A STUDY PLANNING SYSTEM

Morten Milling Jensen
Teddy Kaarløv Nielsen

LYNGBY 2003
EKSAMENSPROJEKT
NR. 70/03

IMM

Trykt af IMM, DTU

Abstract

A study planning system is intended to support automated study planning helping students to elaborate a plan for their course of study. At the time being it is an overwhelming task for the individual student to plan his course of study more than just a couple of terms ahead and a tool for supporting study planning is thus urgently needed.

In this thesis the domain of study planning is thoroughly analyzed from a student's point of view using the Technical University of Denmark as starting point. The analysis is conducted by means of Unified Modelling Language (UML) class diagrams and careful descriptions. A requirements specification is prepared posing requirements with respect to data, functionality and quality. In addition a number of other requirements are stated e.g. legal requirements.

On the basis of the requirements specification a fully-fledged study planning system is implemented. The system is highly configurable, supports various cultures and is based on a 3-tier architecture. The employed development platform is the Microsoft .NET Framework and Microsoft SQL Server is used as relational database system. Also, the Extensible Markup Language (XML) is used for storing data.

Keywords: Automated study planning, study planning, object-oriented programming, requirements specification, 3-tier architecture, database design, .NET, XML.

Resumé

Et studieplanlægningssystem har til formål at understøtte automatiseret studieplanlægning og derved hjælpe studerende med at udarbejde en plan for deres studieforløb. På nuværende tidspunkt er det en nærmest uoverkommelig opgave for den enkelte studerende at planlægge sit studieforløb mere end blot et par semestre frem og et værktøj til at understøtte studieplanlægning er således stærkt påkrævet.

I denne afhandling analyseres domænet for studieplanlægning ud fra en studerendes synsvinkel med udgangspunkt i Danmarks Tekniske Universitet. Analysen udføres ved hjælp af Unified Modelling Language-klassediagrammer (UML-klassediagrammer) og omhyggelige beskrivelser. En kravspecifikation udarbejdes indeholdende krav til data, funktionalitet og kvalitet. I tillæg hertil angives en række andre krav f.eks. lovgivningsmæssige krav.

På grundlag af kravspecifikationen implementeres et fuldt fungerende studieplanlægningssystem. Systemet er i høj grad konfigurerbart, understøtter forskellige kulturer og er baseret på en 3-lags arkitektur. Den benyttede udviklingsplatform er Microsoft .NET Frameworket og Microsoft SQL Server er anvendt som relationelt databasesystem. Ligeledes benyttes Extensible Markup Language (XML) til at lagre data.

Nøgleord: Automatiseret studieplanlægning, studieplanlægning, objekt-orienteret programmering, kravspecifikation, 3-lags arkitektur, databasedesign, .NET, XML.

Preface

The present Master's Thesis has been completed at the Department of Informatics and Mathematical Modelling (IMM) at the Technical University of Denmark (DTU) in the period from July 7, 2003 to December 19, 2003. Our M.Sc. studies are concluded by this project.

The project has been supervised by lektor Jens Thyge Kristensen and assoc. prof. Hans Bruun.

Acknowledgements

We would like to thank our supervisors for helpful guidance and for being obliging on enquiry. Special thanks to Jens Thyge Kristensen for several valuable proofreadings.

Also we would like to thank M.Sc. Torben Gjaldbæk, former co-student, for proofreading and profitable comments.

Kgs. Lyngby, December 19, 2003

Morten Milling Jensen *Teddy Kaarløv Nielsen*

Overview

Contents

List of Figures

Chapter 1

Introduction

A study planning system is intended to support automated study planning helping students to elaborate a plan for their course of study. At the time being it is an overwhelming task for the individual student to plan his course of study more than just a couple of terms ahead and a tool for supporting study planning is thus urgently needed.

The aim of a study planning system is to make students plan in the long view, highlight substance instead of rules and constraints, give rise to a plain leitmotif in the individual course of study and so on.

In this thesis we analyze the domain of study planning and prepare a requirements specification for a study planning system. Moreover, we implement a fully acting system which supports automated study planning.

1.1 Thesis Objectives

The objectives of the current thesis are:

1. To describe and profoundly analyze the domain of study planning.

2. To pose requirements to insist on if a study planning system is to support an expedient, automated study planning.

3. To implement a fully-fledged study planning system that meets the posed requirements.

1.2 Approach

We have used the following approach:

1. **Domain Description** (chapter 2)
   Initially, the different concepts and the coherence between these are described.

2. **Domain Analysis** (chapter 3)
   Then, we analyze the domain and present a logical model of the different entities and relations between the entities.

3. **Requirements Specification** (chapter 4)
   Starting from the domain analysis, we pose requirements to a study planning system.

4. **Implementation** (chapters 5, 6, 7 and 8)
   On the basis of the requirements specification we implement a study planning system.

5. **Testing** (chapter 9)
   Finally, the system is tested.

The domain analysis has been performed by applying an object-oriented approach using Unified Modelling Language[1] (UML) class diagrams as a means of representing the entities of the domain and the coherence between the entities.

The domain description, the domain analysis and the requirements specification (chapters 2, 3 and 4) are organized according to the principles expound in [5] and [18].

1.3 Previous Work

In the period from March 24, 2003 to June 23, 2003 we made a preparatory thesis (confer [15]) in which we analyzed the domain of study planning and determined which requirements to insist on if a study planning system is to support an expedient study planning.

We have based our work on [15] and large parts from this have been included in the current thesis. The sections 2.2-2.4 and 4.2-4.4 have been included from [15] in a by and large unaltered form while chapter 3 and section 4.1 are fundamental revisions of contents from [15].

1.4 Thesis Outline

In chapter 2, we describe the domain in both a concise and complete form. Chapter 3 provides an analysis of the domain. A requirements specification for the study planning system is presented in chapter 4. In chapter 5 the foundations of the study planning system are described. Chapter 6 forms an overview of select, central modules in the implemented system. In chapter 7, we describe how the system has been initialized with data and in chapter 8 the developed user interface is described. In chapter 10, we discuss whether the system fulfills the posed requirements. Chapter 11 presents some suggestions for future work to be done in continuation of this project. In chapter 12, we conclude on the project in general.

Appendix A gives an overview of the `StudyPlanning.DAL` namespace. Accordingly, appendix B provides an overview of the `StudyPlanning.Biz` namespace and appendix C provides an overview of `StudyPlanning.UI` namespace. Appendix D contains a decision diagram for the process of authentication. In appendix E diagrams of the database design are found. Volume

[1] Confer [4].

II contains the source code of the study planning system. Volume III contains code concerning the database.

1.5 Prerequisites

In order to read this thesis the following prerequisites are expected:

- Acquaintance with Unified Modelling Language (UML) class diagrams.
- Knowledge of object-oriented modelling and design.
- Basic knowledge of the Visual C# .NET programming language[2].
- Knowledge of database design and Structured Query Language (SQL).

1.6 Peculiarities

The notation "*&c.*" is used as an abbreviation of the Latin expression "et cetera".

[2] Alternatively, either Visual J# .NET or Java as both of these languages remind sufficiently of Visual C# .NET.

Part I

Domain Analysis & Requirements Specification

Chapter 2

Domain Description

In this chapter the domain of study planning will be described. First a synopsis is presented (section 2.1) in which the scope and span of the domain description are defined. Subsequently, a concise (section 2.2.1) and a more elaborate description (section 2.2.2) of the domain is presented along with the stakeholders of the domain (section 2.4). Finally, a terminology list (section 2.3) provides a quick overview of the concepts used in the domain.

2.1 Synopsis

This section defines the scope and span as they will be considered in this paper.

2.1.1 Scope

The domain covered in this paper is that of study planning.

2.1.2 Span

The domain will be considered from a student's point of view. The process of study planning will be covered extensively as this is an important part of the domain. Other entities in the domain such as the university, departments, lecturers, courses, periods *&c.* will also be covered.

Focus is on the process of study planning and therefore some details which are of no importance in relation hereto have been omitted. For example one might want to register some information about lecturers, e.g. interests, projects, publications *&c.* However, this information has no influence whatsoever on the process of study planning and hence it has been left out in the span of this paper.

It has been attempted to make the descriptions as general as possible. However, as the requirements specification later in this paper presents the requirements for a study planning system which is intended to be implemented at the Technical University of Denmark (DTU), many of the examples and descriptions in the paper relate specifically to the concepts and structures of educations used at DTU.

Many types of students study at DTU. Some examples of student types are given below:

- Bachelor of Science (B.Sc.) students.
- Master of Science (M.Sc.) students.
- Foreign M.Sc. students.
- Ph.D. students.
- Food Science students.

For reasons of demarcation only B.Sc. and M.Sc. students will be included in the domain description.

2.2 The Domain of Study Planning

In this section the domain of study planning is described in a concise and a more complete way.

2.2.1 Concise Description

Study planning is an iterative process of selecting courses from a set of required courses and a set of optional courses and arranging them by timetable module in a timetable for one period at a time. Alongside the courses a number of projects must also be scheduled at certain points in the course of study.

Some courses require knowledge which may be obtained in another course thus introducing a dependency in the creation of the study plan. The prerequisites for a course may be mandatory, technical or desirable. Mandatory prerequisites must inevitably be satisfied, however, students may choose to ignore other prerequisites and sign up for a course anyway.

Some courses are mutually exclusive – also known as point blocking – if they have similar contents and hence only one of these courses should be selected.

Both Bachelor of Science (B.Sc.) and Master of Science (M.Sc.) students must choose a technical package containing a selection of mandatory courses. Moreover, M.Sc. students may aim at obtaining a technical line or a specialization within a given field.

When setting up a study plan the various constraints imposed by the university or legislation must be taken into account as well as the criteria determined by the student.

Examples of constraints and criteria:

1. M.Sc. students must achieve a total amount of 300 points and 15 points hereof must be obtained from courses giving AMS points in order to receive their master's degree. B.Sc. students must achieve a total amount of 210 points to receive their degree.
2. A minimum number of points must be achieved every year in order to maintain student grant.
3. No courses should be scheduled on Tuesday afternoons for the spring period of the year 2003.

Constraint number one is an example of a constraint imposed by the university. The second constraint is imposed by the State Education Fund and the third is a criterion posed by the student.

2.2.2 Complete Description

In this section we shall analyze the descriptions from section 2.2.1 in order to classify, categorize and describe the concepts mentioned herein. In addition, the description is extended with further concepts.

The most significant part of the domain is the *student*. Without students there would be no need for study planning.

A university consists of a number of *departments* which are small independent units with the responsibility of teaching and conducting research within a well-defined field of science, the Humanities, social studies *&c.* – e.g. "Mathematics" or "Manufacturing Engineering and Management".

Students study at a *university*.

Departments offer a number of *courses* and *projects*. A course is the teaching of one or more topics from the field of science of the department for a given period of time. At the end of the period the course is evaluated and the students may either pass or fail the course. A project is the writing of a paper based on an assignment possibly including the solving of a practical exercise. Courses with no timetabled teaching are also regarded as projects as well the trainee job which B.Sc. students must complete at a company.

Passing a course or a project gives the students a number of credit *points* depending on the estimated workload required to pass the course or project.

Courses are taught by *lecturers* who are affiliated with a department at the university. "Lecturers" is a common designation of lecturers, senior lecturers and professors.

Both B.Sc. and M.Sc. students must choose a *technical package* which is a selection of mandatory courses and projects which will give the student a fundamental knowledge of the chosen field of science.

M.Sc. students may choose to obtain a *technical line* which is a specialization within a given field acquired by gathering a certain amount of credit points from passing select courses. Having chosen a technical line, M.Sc. students have the option of following a *specialization* which is a recommended course of study within the chosen technical line.

A course is taught in one or more *timetable modules* in a weekly recurring timetable. A timetable module is the designation of a given period of time on a specific weekday e.g. Mondays from 8:00 to noon is designated the module 1A (figure 2.1 below).

A *timetable* is comprised of all the possible timetable modules.

Time	**Mon**	**Tue**	**Wed**	**Thu**	**Fri**
8.00 – 12.00	1A	3A	5A	2B	4B
12.00 – 13.00	Break	Break	Break	Break	Break
13.00 – 17.00	2A	4A	5B	1B	3B

Figure 2.1: Example of a timetable from DTU showing the timetable modules.

A *Period* is a subdivision of a calendar year specified by a start date and an end date e.g. from February 3rd 2003 to May 13th 2003. There may be different types of periods and periods of varying length.

B.Sc. students must conclude their studies by writing a *Bachelor's Thesis* which is a project with a duration of approximately 5 months. Likewise, M.Sc. students must also conclude their studies by writing a *Master's Thesis* which is a project spanning a period of 5 to 12 months.

Producing a *Study Plan* is the rather involved, iterative process of scheduling courses in timetables for the remaining periods of a student's course of study. Alongside the courses some projects must also be scheduled. The student's interests, technical package, desired technical line, interdependencies between courses, courses and projects which have already been finished as well as the general recommendations from the university as when a given course should ideally be scheduled must be taken into account.

2.3 Terminology

Below we have listed in alphabetical order a brief overview of some important concepts and their meaning from the domain description.

AMS
: is an abbreviation of the Danish words "Arbejdsmiljø, Miljø og Samfundsfag" – in English work environment, environment and civics.

Bachelor of Science (B.Sc.)
: is the title acquired by a person who has studied for 3-3$\frac{1}{2}$ years at a university.

Bachelor's Thesis
: is the project which concludes the studies of a B.Sc. student.

Course
: is the timetabled teaching of select topics within a specific field of science, the Humanities, social studies *&c.*

Department
: is a subdivision of a university with the responsibility of teaching and researching within a specific field of science, the Humanities, social studies *&c.*

ECTS
: is an abbreviation of "European Credit Transfer System" – a unit of measurement used by European universities to denote the amount of credit points which may be obtained by passing a course.

Lecturer
: is a person who teaches one or more courses at a university.

Master of Science (M.Sc.)
: is the title acquired by a person who has studied for at least 5 years at a university.

Master's Thesis
: is the project which concludes the studies of a M.Sc. student.

Module
: is a weekly recurring time interval in which a course is taught.

Point
: is used as a way of expressing the estimated workload required to pass a course.

Project
: is the writing of a paper based on a given assignment, a course with no timetabled teaching or a trainee job in a company.

Specialization
: is a recommended course of study within a given technical line.

Student
: is a person who studies at a university.

Study Plan
: is a collection of courses and projects scheduled for the remaining part of the student's course of study.

Technical Line
: is a specialization within a given field which – if completed – will appear on the student's diploma.

Technical Package
: is a selection of mandatory courses which provide a fundamental knowledge of a given field of science.

University
: is an institution that teaches and performs research within the areas of the Humanities, social studies, natural science *&c.*

2.4 Stakeholders

In this section the various stakeholders in the domain of study planning are briefly described.

According to [5], section 15.3.2, stakeholders are:

> "A closely knit, tightly related group of either people or institutions, pressure groups where the 'fabric' that 'relates' members of the group, 'separates' these from other such stakeholder groups, and from non-stakeholder entities."

The stakeholders of the study planning domain are:

The *students* who study at the university.
The *lecturers* who teach courses at the university.
The *university* as a whole.
The individual *departments* at the university.

Chapter 3

Domain Analysis

In this chapter we present analysis of the domain of study planning. As mentioned in section 1.2, the domain analysis has been performed by applying an object-oriented approach using UML class diagrams as a means of representing a logical model of the domain. Consequently, entities in the domain are referred to as classes and relations between entities are referred to as associations.

For reasons of space and clarity, the logical model is divided up into several figures and hence some classes occur in more than one figure. Each class is named uniquely and in singular. A class with the same name in two different figures thus represents one and the same class.

The individual class and its associations to other classes is meticulously described and explained on the philosophy that however unequivocal a diagram may be it is of small value if the very meaning hereof is not plain.

The headings in this chapter correspond to the names of the various classes by which it is convenient to look up classes in the table of contents.

3.1 Department

Each department of the university is a larger, independent unit which is responsible of teaching and doing research within a number of subjects or areas[1]. A department is represented by the class `Department`.

As it appears from figure 3.1 one or more lecturers may be attached to a department. The attached lecturers are the ones who among other things are responsible of lecturing and coaching in the courses offered by the department.

Also a number of students are possibly attached to the department in the form of an association between the `Department` and `Student` (section 3.5) classes. Far from all students are attached to a department as a student is only attached to a department when he is writing his master or bachelor thesis.

[1] http://www.adm.dtu.dk/institutter/index_d.htm,
http://www.adm.dtu.dk/fakta/rekrut/profiltxt_d.htm,
http://www.adm.dtu.dk/fakta/adm/l_sek/vedtaegt_d.htm (§§18)

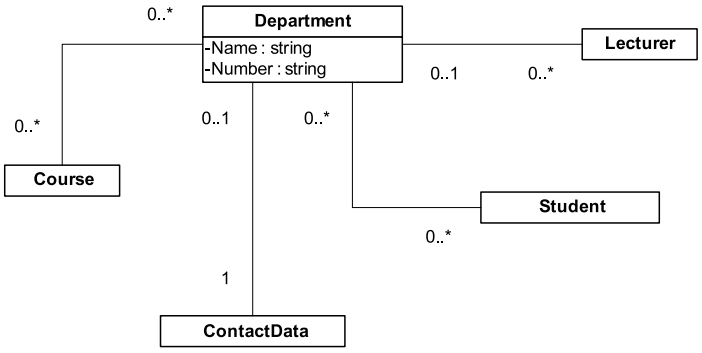


Figure 3.1: Logical model of the `Department` class and its associations to the `Course`, `ContactData`, `Student` and `Lecturer` classes.

The department's address and other contact information are represented by an association to the `ContactData` (section 3.12) class.

A department may offer one or more courses. The contents of the courses offered by the department are directly related to the areas within which the department carries out research.

The `Department` class has the following attributes:

Attribute Name	Description
Name	Name of the department.
Number	The internal number of the department.

3.2 Lecturer

In this context "lecturer" has been selected as a common designation of lecturer, senior lecturer and professor.

The lecturer is a person who is employed by the university with the purpose of teaching courses among other things. A lecturer may also be responsible for conducting research, writing publications and holding seminars. In the domain of study planning, however, only the lecturer's role as a teacher is relevant.

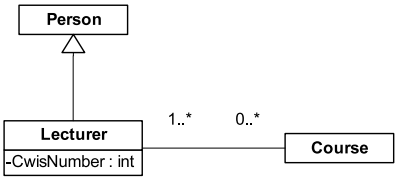


Figure 3.2: Logical model of the `Lecturer` class and its association to the `Course` class as well as the generalization from the `Lecturer` class to the `Person` class.

A lecturer is a person as denoted by the generalization between the `Lecturer` class and the `Person` class (section 3.11).

A lecturer may teach one or more courses and a course may be taught by more than one lecturer.

The `Lecturer` class has the following attribute:

Attribute Name	Description
CwisNumber	The Campus Wide Information System[2] number of the lecturer.

3.3 Course

A course is the timetabled teaching of various topics within a given field of education for a predefined period of time[3]. At the end of the period the attending students are evaluated and may either pass or fail the course. Passing a course gives the student a certain amount of credit points[4].

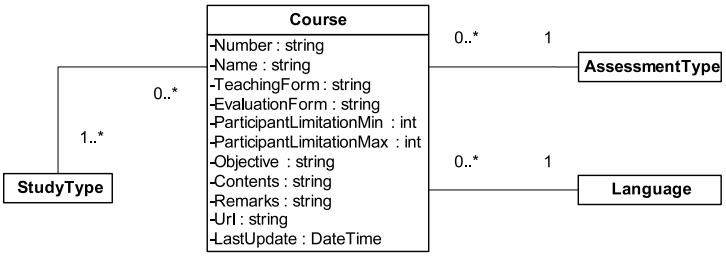


Figure 3.3: Logical model of the `Course` class and its associations to the `StudyType`, `Language` and `AssessmentType` classes.

The `Course` class (figure 3.3 represents a course. Each `Course` object (figure 3.3) is linked to one or more `StudyType` (section 3.8) objects which describe the study types for which the course is intended. The association between the `Course` and `Language` (section 3.19) classes represents the language in which the course is taught. Courses are assessed in exactly one way illustrated by the association between the `Course` and the `AssessmentType` (section 3.17) class.

The `Course` class has the following attributes:

Attribute Name	Description
Number	A unique identification of the course which may contain characters – e.g. “C4222” or “88890”.
Name	The name of the course.
TeachingForm	Text describing how the course is taught e.g. “group discussions”, “lectures” or “laboratory exercises”.
EvaluationForm	Description of how the course is evaluated e.g. “Reports count 60% and the written exam 40%” or “One report with oral defence”.

[2] http://www.adm.dtu.dk/fakta/om_cwis/index_e.htm
[3] http://www.kurser.dtu.dk
[4] http://www.adm.dtu.dk/studier/bekendt/bekend_d.htm
A template for a course description can be found in [11]

Attribute Name	Description
ParticipantLimitationMin	The minimum number of students who must sign up for the course in order for the course to be completed.
ParticipantLimitationMax	The maximum number of students who can attend the course simultaneously.
Objective	A description of the course objective i.e. what the student will achieve by passing the course.
Contents	Text describing the contents of the course i.e. an excerpt of the syllabus for the course.
Remarks	Special remarks regarding the course.
Url	The url for the course home page where further information can be retrieved.
LastUpdate	Date and time for the last update of the course description.

3.3.1 Course Parts

Some courses span several periods and must thus be divided into sequent parts in order to avoid ambiguity and to state exactly which part of the course belongs to which period. Each part of a multi-part course must be signed up for in ascending order and successive parts must be finished in the first possible periods where the course part is taught as shown in figure 3.4. Figure 3.5 shows a scenario which may not be allowed – even though part 2 of the course is taught in periods b and d it is considered another instance of the course part in period d.

Period	*a*	*b*	*c*	*d*	...
Part	1	2			

Figure 3.4: Allowed scenario – course parts must be finished in the first possible period.

Period	*a*	*b*	*c*	*d*	...
Part	1			2	

Figure 3.5: Disallowed scenario – unless part 2 is not taught in periods b and c.

The `CoursePart` class represents a part of a course and a course consists of one or more sequentially numbered course parts depicted by the association between the `Course` and `CoursePart` classes shown in figure 3.6.

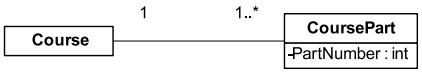


Figure 3.6: Logical model of the associations between the `Course` and `CoursePart` classes.

The `CoursePart` class has the following attribute:

Attribute Name	Description
PartNumber	The number of the course part.

3.3.2 Periods and Credit Points

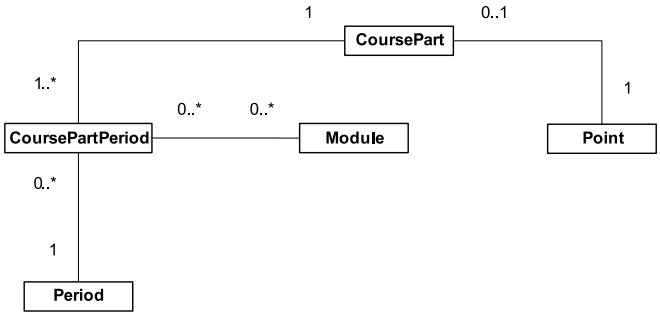


Figure 3.7: Logical model of the associations between the `CoursePart` class and the `CoursePartPeriod`, `Period`, `Module` and `Point` classes.

The total number of credit points which is obtained by passing a course is distributed among the course parts. The credit points for each course part are described by the association between the `CoursePart` and `Point` (section 3.13) classes (figure 3.7). To preserve the rule illustrated in figure 3.4 each course part must be associated with the periods in which it is taught. This association is shown in figure 3.7.

A `CoursePartPeriod` object may be associated with one or more `Module` (section 3.15) objects which correspond to the timetable modules in which the course part is taught.

3.3.3 Modules

The timetable schedule of courses may be specified in a compound form as shown in the next example:

For course `28010` the modules are specified as[5]:

```
E5A and F5A
```

Here *E* means autumn (Danish: efterår) and *F* stands for spring (Danish: forår). *5A* is the module in which the course is taught, Wednesdays from 8:00 to 12:00. So according to this the course consists of two parts, the first part is taught in the autumn and the second part is taught in spring. However, the course description says that the course only spans one teaching period and as the course only gives the student 5 credit points the author probably meant:

```
E5A or F5A
```

Another example is course `28011` which has the following specification of modules[6]:

```
E3A and E5A and January
    or F2B and June
```

From this it appears that the course consists of two parts – the student may take the first part either in the autumn in the modules *3A* and *5A* or in spring in module *2B*. The corresponding second parts of the course are taught in January and June, respectively. The course description confirms this interpretation.

[5] http://www.kurser.dtu.dk/presentation/presentation.asp?coursecode=28010-2
[6] http://www.kurser.dtu.dk/presentation/presentation.asp?coursecode=28011-2

The module specification for course `28010` corresponds to the general specification in figure 3.8 where $\mathcal{M}$ is the set of available modules.

$$\left.\begin{array}{l}(a_1 \vee b_1 \vee \ldots \vee z_1)\\ \wedge\\ (a_2 \vee b_2 \vee \ldots \vee z_2)\\ \wedge\\ \ldots\end{array}\right\} a, b \ldots z \in \mathcal{M}$$

Figure 3.8: Most common module specification.

Whereas the module specification for course `28011` has the form shown in figure 3.9.

$$\left.\begin{array}{l}(a_1 \wedge b_1 \wedge \ldots \wedge z_1)\\ \vee\\ (a_2 \wedge b_2 \wedge \ldots \wedge z_2)\\ \vee\\ \ldots\end{array}\right\} a, b \ldots z \in \mathcal{M}$$

Figure 3.9: Less common module specification.

3.3.4 Course interdependencies

A course may have relations to other courses and the following four types of interdependencies have been identified:

- Mandatory prerequisite courses
- Technical prerequisite courses
- Desirable prerequisite courses
- Point blocking courses

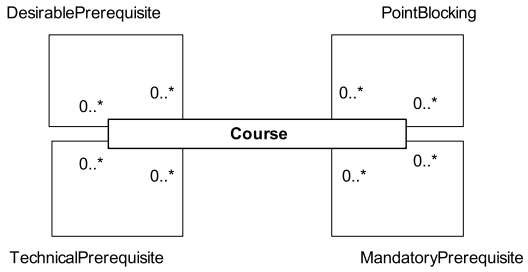


Figure 3.10: Logical model of the four associations from the `Course` class to itself.

The `Course` class has four associations to itself describing the possible interdependencies between courses (figure 3.10). A course may have none or many of the four types of interdependencies.

For course `27031` the technical prerequisites are:

```
27000 / 27001.28120 (if possible at the same time). Mega MAT
```

The / operator should be interpreted as logical *or* and the . (dot) operator means logical *and* in this context[7]. Logically this expression is unambiguous since *and* takes precedence over *or*[8] and bearing in mind that "Mega MAT" has course number `01005` the expression could also be written as:

$$\texttt{27000} \lor (\texttt{27001} \land \texttt{28120} \land \texttt{01005})$$

Judging by the contents of the courses, however, this is most likely not what the author intended as the courses `27000` and `27001` are equivalent and the courses `28120` and `01005` are completely different. What the author actually meant is probably:

$$(\texttt{27000} \lor \texttt{27001}) \land \texttt{28120} \land \texttt{01005}$$

This raises the question of whether to read the expression using the precedence of logical operators in boolean logic or in a left–to–right fashion.

Nonetheless, mandatory prerequisites, desirable prerequisites and point blocking courses are specified in the same way which leads to the general specification of interdependencies between courses shown in figure 3.11 where $\mathcal{C}$ is the set of courses.

$$\left.\begin{array}{l} (a_1 \lor b_1 \lor \ldots \lor z_1) \\ \land \\ (a_2 \lor b_2 \lor \ldots \lor z_2) \\ \land \\ \ldots \end{array}\right\} a, b \ldots z \in \mathcal{C}$$

Figure 3.11: Most commonly used course dependency specification.

The specification in figure 3.11 is used for the vast majority of courses, however, a few courses use a different specification. For instance course `31652` for which the technical prerequisites are described as[9]:

```
(31029.31657)/(01032.41621)
```

This may also be written as:

$$(\texttt{31029} \land \texttt{31657}) \lor (\texttt{01032} \land \texttt{41621})$$

Generally this can be written as the expression shown in figure 3.12.

In order to sign up for a course its mandatory prerequisites must be satisfied – i.e. any mandatory prerequisite courses must have been passed. Mandatory courses are typically

[7] Confer ([11])
[8] Confer ([24])
[9] http://www.kurser.dtu.dk/presentation/presentation.asp?coursecode=31652-2

$$\left.\begin{array}{l}(a_1 \wedge b_1 \wedge \ldots \wedge z_1)\\ \vee\\ (a_2 \wedge b_2 \wedge \ldots \wedge z_2)\\ \vee\\ \ldots\end{array}\right\} a, b \ldots z \in \mathcal{C}$$

Figure 3.12: Sparsely used course dependency specification.

safety related – e.g. prior to working with high voltages or hazardous chemicals a mandatory safety course must be taken.

Technical prerequisite courses are not mandatory, but they do provide some knowledge which the students are assumed to be familiar with prior to signing up for a course. Technical prerequisites may also be specified as qualifications – e.g. for course `02393` the technical prerequisites are described as "Knowledge of basic programming concepts as datatypes, choices and decisions, loops and functions".

Desirable prerequisite courses contain useful knowledge which might be relevant, however, they are neither mandatory nor do they give any competence required to sign up for a course. Desirable prerequisites may too be defined as qualifications – e.g. the desirable prerequisites for course `28252` are "Mathematical and Process technology background".

A course is point blocking in relation to another course if the contents of the two are mostly alike and while a student is allowed to sign up for both courses, only the first one passed will give the student any credit points. Often a course is point blocking in combination with earlier/later versions of the same course so students cannot just easily collect a lot of credit points by taking similar courses. One might also say that a course and its point blocking courses are mutually exclusive with regards to credit points.

$$(A \overset{p}{\bowtie} C) \wedge (B \overset{p}{\bowtie} C) \nRightarrow (A \overset{p}{\bowtie} B)$$

Figure 3.13: Transitivity between point blocking courses may not be assumed.

Figure 3.13 illustrates that if a course A is point blocking in relation to (denoted by the $\overset{p}{\bowtie}$ operator) a course C and another course B is also point blocking in relation to the course C then the courses A and B are often mutually point blocking as well, however, this transitivity does not always hold.

3.3.5 Recommended Placement

As it appears from figure 3.14 courses may have a recommended placement – i.e. at which point in the course of study a course should ideally be taken. The recommended placement may be specified separately for individual study types either specifically like:

> Bachelor of Science: "4th to 5th semester"
> Master of Science: "2nd to 3rd semester"

or as a concept like:

> Bachelor of Science: “At the end of the study”
> Master of Science: “In the middle of the study”

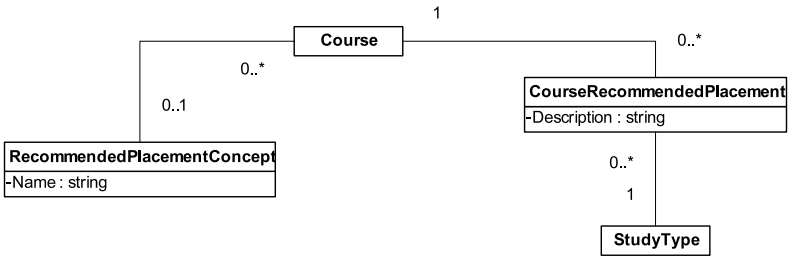


Figure 3.14: Logical model of the association between the `Course`, `RecommendedPlacementConcept`, `CourseRecommendedPlacement` and `StudyType` classes.

The `RecommendedPlacementConcept` class represents the possible concepts of placement recommendations – e.g. “The middle of the study” or “At the end of the study” .

The `RecommendedPlacementConcept` class has the following attribute:

Attribute Name	Description
Name	The name of the concept – e.g. “The middle of the study”.

The `CourseRecommendedPlacement` class has the following attribute:

Attribute Name	Description
Description	Description of the recommended placement of the course in relation to a given study type – e.g. “B.Sc.: 3rd semester” and “M.Sc.: 2nd semester”.

3.3.6 Miscellaneous

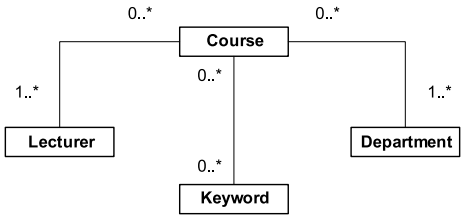


Figure 3.15: Logical model of the association between the `Course` class and the `Lecturer`, `Keyword` and `Department` classes.

Figure 3.15 reflects that each course may have one or more keywords (section 3.10) associated with it. These keywords are selected by the author to describe the contents of the course. A course has one or more contact persons (lecturers) who are usually also the persons who teach the course. Commonly, a course is offered by a specific department, however, some courses are offered through a collaboration between multiple departments.

3.4 Project

A project is a work of more or less extensive character executed within some period having no timetabled teaching[10]. The actual project work is carried out by one or several students in collaboration.

Examples of projects that students have to prepare are given in section 3.5. Notice, some projects are regarded as courses in that timetabled teaching is offered in addition to the project work[11].

A project is represented by the `Project` class (figure 3.16).

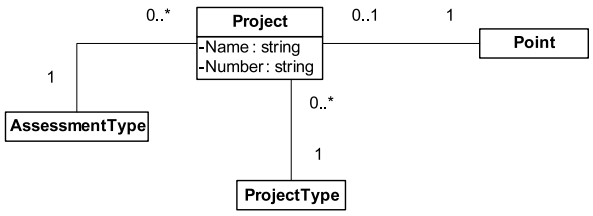


Figure 3.16: Logical model of the `Project` class and its associations to the `AssessmentType`, `ProjectType` and `Point` classes.

A project is of some type and so each `Project` object is linked to a `ProjectType` object (confer subsection below). The workload of a project is measured by means of points and an association thus exists between the `Project` and `Point` (section 3.13) classes.

The outcome of a project may be assessed in different ways e.g. by a grade or pass/fail. This is represented by the association to the `AssessmentType` class (section 3.17).

The `Project` class has the following attributes:

Attribute Name	Description
Name	Name of the project.
Number	A number that identifies the project. The project number may contain characters and is therefore modelled as a string.

According to figure 3.17, a project may be executed by just one student or as a collaboration between several students as expressed by the association between the `Project` class and the `StudentProject` (section 3.5) class.

The fact that a project spans some period is manifested in figure 3.17 by the association between the `Project` class and the `Period` (section 3.14) class. Projects being part of one or more technical packages are not directly associated to a period as the project period is individually specified for each of the relevant technical packages[12].

3.4.1 ProjectType

The `ProjectType` class represents the possible types of projects (figure 3.18).

[10] http://www.adm.dtu.dk/studier/bekendt

[11] As an example technical package projects (projects that must be completed as part of a technical package) are regarded as courses in that teaching in subjects like “collaboration”, “formulation of abstract” and “delimitation” is timetabled during the project work.

[12] Confer section 3.6 for a detailed description of the association between projects and technical packages.

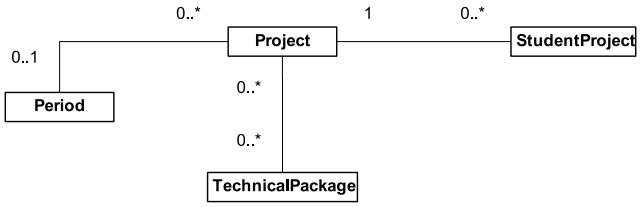


Figure 3.17: Logical model of the associations between the `Project` class and the `StudentProject`, `Period` and `TechnicalPackage` classes.

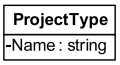


Figure 3.18: Logical model of the `ProjectType` class.

The `ProjectType` class has the following attribute:

Attribute Name	Description
Name	The name of the project type – e.g. "Mid-way Project".

3.5 Student

A student is a person who studies at the university. The individual student may obtain a degree by passing a set of courses and finishing a number of projects.

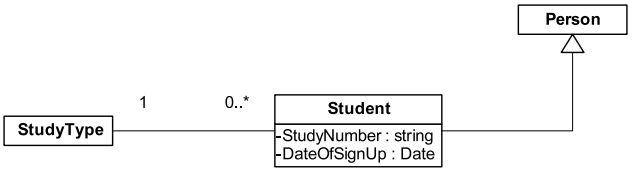


Figure 3.19: Logical model of the `Student` class and its generalization to the `Person` class. Moreover, the association between the `Student` class and the `StudyType` class.

As already mentioned a student is a person. In figure 3.19, that fact is denoted by the generalization between the `Student` class and the `Person` (section 3.11) class.

Every student takes a degree of some type[13] e.g. "Bachelor of Science". In figure 3.19 this is expressed by an association between the `Student` and `StudyType` (section 3.8) classes.

The `Student` class has the following attributes:

[13] http://www.explore.dtu.dk/retning/sr.html

Attribute Name	Description
StudyNumber	A number which uniquely identifies the student. The study number is assigned when the student is signed up at the university. A study number may contain characters like e.g. "s971838" and is therefore a string.
DateOfSignUp	The date at which the student signed up at the university.

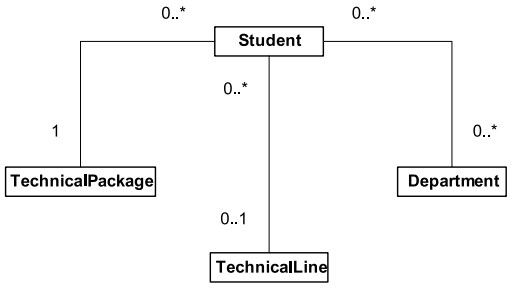


Figure 3.20: Logical model of the association between the `Student` class and the `TechnicalPackage`, `TechnicalLine` and `Department` classes.

A student may be attached to one or more departments (figure 3.20) when he is writing his master's or bachelor's thesis. That is, not all students are attached to a department.

Every student first of all has to complete a technical package[14] (section 3.6). Students may select between various technical packages, however, it is only allowed for a student to complete one technical package.

M.Sc. students have the option to complete a technical line (section 3.7).

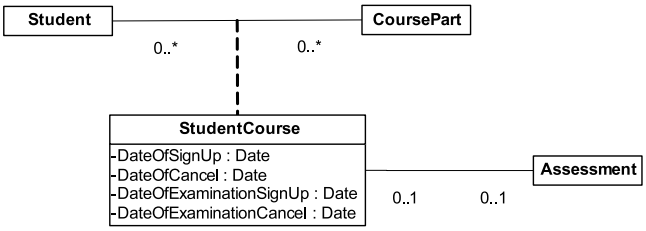


Figure 3.21: Logical model of the association between the `Student` and `Course` classes. As well a logical model of the association class `StudentCourse` and its association to the `Assessment` class.

Every student can sign up for one or more courses (figure 3.21). A sign up may be cancelled by the student later on. A student can finish a course he has signed up for either by getting a "Pass" or a grade of 6 or higher depending on the way whereupon the course is assessed. Hence, a course is not finished (and points therefore not credited) if the student obtains a "Fail" or a grade less than 6. The result obtained by the student in a given course finds expression by the association between the `StudentCourse` class and the `Assessment` (section 3.16) class.

[14] http://www.explore.dtu.dk/studie/civ/uddyb.html

If a student has unsuccessfully attempted to finish a given course he may sign up for that course again, however, rules apply with respect to the permissible number of repeater attempts[15].

The `StudentCourse` class has the following attributes:

Attribute Name	Description
DateOfSignUp	The date at which the student signed up for the course.
DateOfCancellation	The date at which the student has cancelled his previous sign up for the course.
DateOfExaminationSignUp	The date at which the student signed up for examination in the course.
DateOfExaminationCancellation	The date at which the student has cancelled his previous sign up for examination in the course.

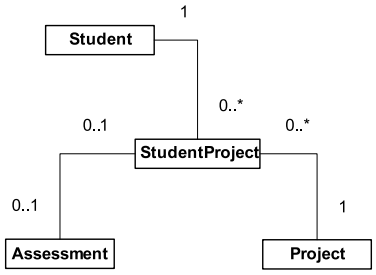


Figure 3.22: Logical model of the association between the `Student` and `StudentProject` classes. As well a logical model of the associations from the `StudentProject` class to the `Project` and `Assessment` classes.

A student has to prepare one or more projects during his study[16] (figure 3.22). As an example Master of Science students have to prepare a so-called "Mid-way Project" in the middle of their study. Similarly, Bachelor of Science students are under the obligation to complete a trainee job at a company. It is worthy of note that a trainee job is considered a project.

Both Bachelor and Master of Science students are supposed to finalize their studies by preparing a bachelor's or a master's thesis, respectively. A thesis is also considered a project.

The assessment of a project is represented by an association between the `StudentProject` and the `Assessment` classes.

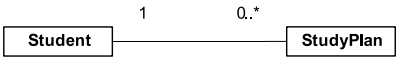


Figure 3.23: Logical model of the association between the `Student` class and the `StudyPlan` class.

Each student can have one or more study plans (figure 3.23) representing different course of study scenarios considered by the student.

[15] [10], page 171, item 5.
[16] [10], pages 202 and 210.

3.6 TechnicalPackage

A technical package[17] is the name of a set of courses and projects which both Master and Bachelor of Science students have to finish in order to fulfil one among other requirements[18].

The `TechnicalPackage` class represents a technical package.

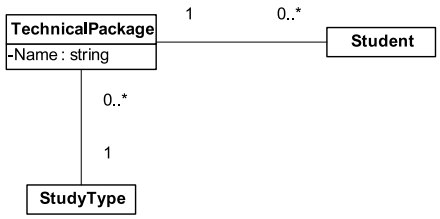


Figure 3.24: Logical model of the `TechnicalPackage` class and its association to the `Student` class and the `StudyType` class.

A technical package may be completed by one or more students and is directed to one particular study type (figure 3.24).

The `TechnicalPackage` class has the following attribute:

Attribute Name	Description
Name	The name of the technical package.

According to figure 3.25 every `TechnicalPackage` object is linked to one or more objects of type `TechnicalPackagePeriod` (confer subsection below). In other words a technical package consists of one or more periods and these periods are furthermore composed of a number of courses.

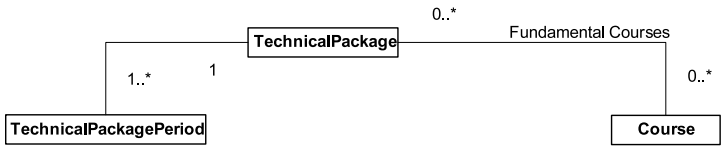


Figure 3.25: Logical model of the associations between the `TechnicalPackage` class and the `TechnicalPackagePeriod` and `Course` classes.

Moreover, a technical package may consist of one or more so-called fundamental courses which are basically just courses. The reason why those courses are named "fundamental" is merely due to the very contents hereof.

The fundamental courses are for some technical packages specified in the form illustrated by figure 3.11 on page 18 i.e. in the same way as course interdependencies are most commonly specified.

[17] Notice, in this context "technical package" has been chosen as a common designation of the Danish terms "fagpakke" and "retning". Master of Science students have to complete a "fagpakke" and Bachelor of Science students have to complete a "retning", however, it is obvious to operate only with one concept as the two afore-mentioned terms cover the same – namely a specification of courses and projects.

[18] http://www.adm.dtu.dk/studieinformation/studinfo/fagpakkeskema/index_d.htm, http://www.diploming.dtu.dk

3.6.1 TechnicalPackagePeriod

As previously mentioned a technical package consists of one or more periods of type `TechnicalPackagePeriod`. Each period of a technical package is further made up of different components.

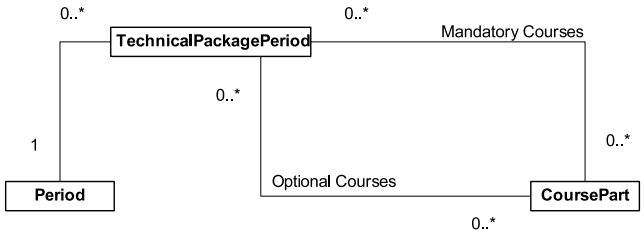


Figure 3.26: Logical model of the `TechnicalPackagePeriod` class and its associations to the `Period`, `CoursePart` classes.

First and foremost it is worth to notice that according to figure 3.26 each `TechnicalPackagePeriod` object is linked to exactly one `Period` object (section 3.25).

Each period of a technical package has a number of mandatory courses specified (the *Mandatory Courses* association) which students taking the current technical package have to follow in the concerned period. Mandatory courses are – as was the case with fundamental courses expound previously in this section – specified in the form illustrated by figure 3.11 on page 18.

The association labelled *Optional Courses* in figure 3.26 indicates a number of courses which it may be appropriate for students taking the current technical package to follow in the concerned period.

3.6.2 TechnicalPackageProject

A technical package may in addition to the mandatory courses specify a number of projects that students have to complete as part of the technical package.

A project of a technical package is represented by the `TechnicalPackageProject` class.

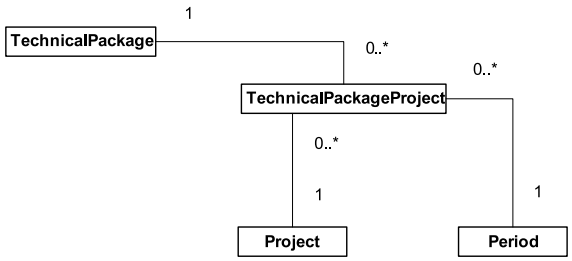


Figure 3.27: Logical model of the association between the `TechnicalPackageProject` class. Moreover, a logical model of the associations between the `TechnicalPackageProject` class and the `Project` and `Period` classes.

Each project of the technical package must be executed in a specific period as reproduced by the association between the `TechnicalPackageProject` class and the `Period` class (figure 3.27). A project can be part of several technical package specifications.

3.7 TechnicalLine

A technical line is a recommended course of study within a given field which makes it possible for Master of Science students to obtain a comprehensive competence within this field[19].

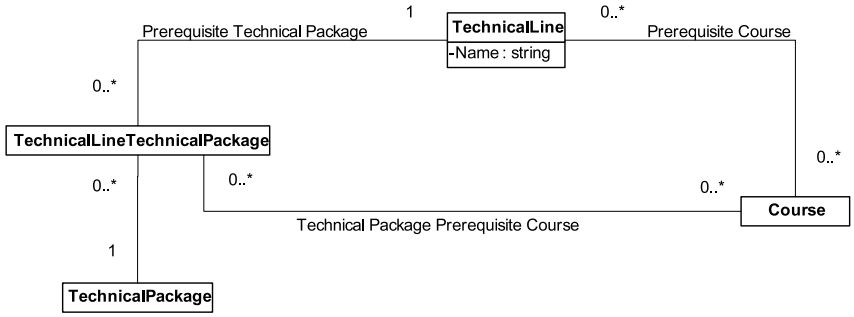


Figure 3.28: Logical model of the `TechnicalLine` class and its associations to the `TechnicalPackage` and `Course` classes as well as the `TechnicalLine-TechnicalPackage` class.

In order to commence a technical line, a specific prerequisite technical package may be required – shown as the *Prerequisite Technical Package* association in figure 3.28. In addition to a technical package a number of entrance requirement courses may be specified for each technical package – depicted as the *Technical Package Prerequisite Course* association in figure 3.28. For example the technical line "Informatics" has the following two possible entrance requirements[20]:

- The "Informatics" technical package and the courses `02130` and `02140`.
- The "Electrical Engineering" technical package and the courses `02130`, `02140`, `02100`, `02105` and `02110`.

Furthermore a number of entrance courses which apply to all students regardless of technical package may be specified. This condition is represented by the *Prerequisite Course* association in figure 3.28.

The `TechnicalLine` class has the following attribute:

Attribute Name	Description
Name	The name of the technical line – e.g. "Energy" or "Construction".

Each technical line (figure 3.29) is divided into one or more subdivisions known as technical fields (section 3.7.1). Furthermore, a technical line is divided into one or more specializations (section 3.7.2) which are analogous and sometimes identical to the technical fields.

To complete a technical line a certain number of credit points must be collected from the technical fields or specializations of the technical line. The association between the `TechnicalLine` and `Point` classes constitutes this requirement.

[19] http://www.adm.dtu.dk/studier/bekendt/civord01_d.html#4.%20Retningsbetegnelse
[20] http://www.adm.dtu.dk/studier/retninger/informatik_d.htm

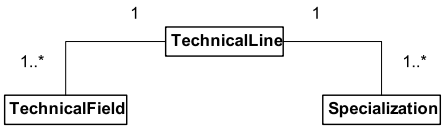


Figure 3.29: Logical model of the association between the `TechnicalLine` and `TechnicalField` classes as well as the association between the `TechnicalLine` and `Specialization` classes.

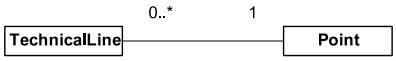


Figure 3.30: Logical model of the association between the `TechnicalLine` and `Point` classes.

3.7.1 TechnicalField

As earlier mentioned a technical field is a subdivision of a technical line – a grouping of related courses. For example within the technical line "Energy" there is a technical field called "Renewable Energy Sources"[21]. The technical field consists of a list of courses from which a certain amount of credit points must be collected. A technical field is represented by the `TechnicalField` class (figure 3.31). A course may be part of one or more technical fields.

Figure 3.31: Logical model of the `TechnicalField` class and its association to the `Course` class.

The `TechnicalField` class has the following attribute:

Attribute Name	Description
Name	The name of the technical field.

3.7.2 Specialization

A specialization is an area within a technical line in which a student may choose to specialize – a recommended course of study. The specialization consists of a list of courses which may be either main or supplementary, point giving or non-point giving (figure 3.32). Main courses are the primary courses of the specialization and thus the student should first of all select courses which are categorized as main courses. Besides the main courses the student may also select a number of supplementary courses which are secondary to the main courses. A course may be point giving which means that the credit points obtained by passing the course are included in the point requirement for the technical line. Analogously, the points obtained from non-point giving courses are not included.

The required amount of credit points for the technical line must be obtained from either main or supplementary courses which are point giving.

The specializations of the technical line "Applied Chemistry and Chemical Engineering" differ from all other specializations by being further divided into subcategories called "fundamental

[21] http://www.adm.dtu.dk/studier/retninger/energi_d.htm

courses", "affiliated courses" and "other courses" each with a point requirement. The descriptions are somewhat equivocal – e.g. the subcategory "other courses" for the specialization "Chemical Product Development"[22] is described as "At least two courses distributed over the technical fields 2 and 4." However, this is rather unclear as the technical fields are not numbered.[23]

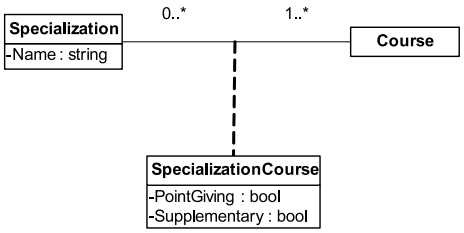


Figure 3.32: Logical model of the `Specialization` class and its association to the `Course` class. As well a logical model of the association class `SpecializationCourse`.

The `Specialization` class has the following attribute:

Attribute Name	Description
Name	The name of the specialization – e.g. "Chemical Product Development" or "Water Resources".

The association class `SpecializationCourse` specifies whether the related courses are point giving and supplementary.

The `SpecializationCourse` association class has the following attributes:

Attribute Name	Description
PointGiving	Describes whether or not the associated course is point giving for the technical line.
Supplementary	Describes whether the associated course is main or supplementary in relation to the specialization.

3.8 StudyType

The `StudyType` class represents the possible study types like "Master of Science" or "Bachelor of Science" (figure 3.33). A student has only one study type and a study type may be associated with one or more students. A `StudyType` may be linked to one or more `TechnicalPackage` objects denoting the technical packages which apply to the study type.

The `StudyType` class has the following attribute:

Attribute Name	Description
Name	The name of the study type – e.g. "Master of Science" or "Bachelor of Science".

[22] http://www.adm.dtu.dk/studier/retninger/specialiseringer/kemitek/specialisering2_d.htm
[23] Requirements for how to handle this specialization are stated in section 4.1.4.

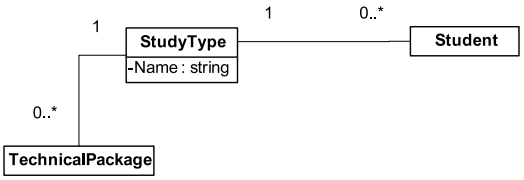


Figure 3.33: Logical model of the `StudyType` class and its associations to the `TechnicalPackage` and `Student` classes.

In order to finish a study of a given study type a certain amount of credit points must be achieved – e.g. Master of Science students must collect 300 ECTS points. These credit point requirements are represented by the association named *Total* in figure 3.34.

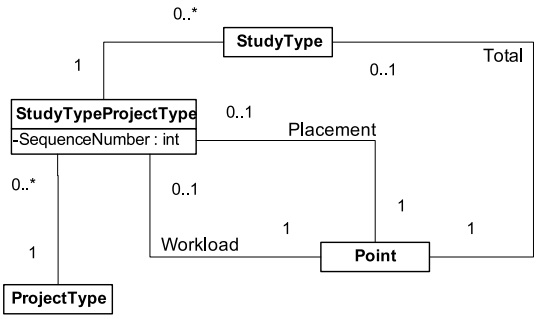


Figure 3.34: Logical model of the association between the `StudyType` and `Point` classes. As well the logical model of `StudyTypeProjectType` class and its associations to the `Point` and `ProjectType` classes.

Furthermore students must also complete a number of projects of specified types during the course of study – e.g. Master of Science students must complete a mid-way project [10]. In figure 3.34 this condition is represented by the `StudyTypeProjectType` and `ProjectType` classes.

The *Placement* association to the `Point` class denotes the interval of credit points which should have been collected prior to scheduling the project and the *Workload* association to the `Point` class denotes the workload of the project.

The `StudyTypeProjectType` class has the following attribute:

Attribute Name	Description
SequenceNumber	If multiple projects are to be finished the sequence number specifies the order in which the projects should be finished.

3.9 StudyPlan

A study plan is a course of study for the periods from when the study plan is made until the student has finished his education. The student tailors his study plan to suit his interests and specific demands of courses, technical package and technical line.

A study plan is represented by the `StudyPlan` class.

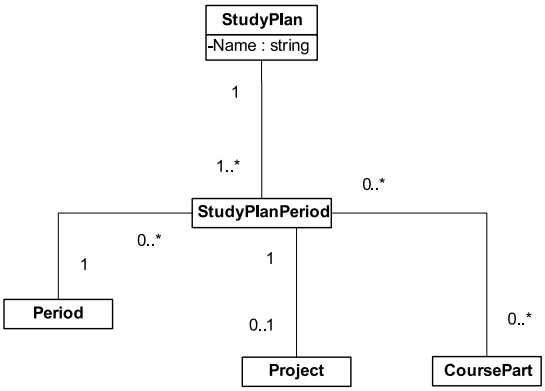


Figure 3.35: Logical model of the `StudyPlan` class and its association to the `StudyPlanPeriod` class. Moreover, a logical model of the associations between the `StudyPlanPeriod` class and the `Period`, `Project` and `CoursePart` classes.

The study plan consists of one or more study plan periods. The duration of a study plan period is determined by the `Period` (section 3.14) object to which it is linked (figure 3.35). Each period of the study plan contains the courses (course parts) or the project which the student has scheduled in that period. At most one project may scheduled in each period since a project has its own individual period which may overlap the course periods.

The `StudyPlan` class has the following attribute:

Attribute Name	Description
Name	The name of the study plan chosen by the student.

3.10 Keyword

To describe the contents of a course the lecturer may use a set of keywords. Students can then find courses which match their interests by searching for keywords associated with the courses.

A keyword is represented by an instance of the `Keyword` class (figure 3.36).

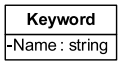


Figure 3.36: Logical model of the `Keyword` class.

The `Keyword` class has the following attribute:

Attribute Name	Description
Name	The name of the keyword – e.g. “Programming”, “Mechanics” or “Molecular geometry”.

3.11 Person

A person is an individual human being and is represented by the `Person` class. Examples of persons are students and lecturers.

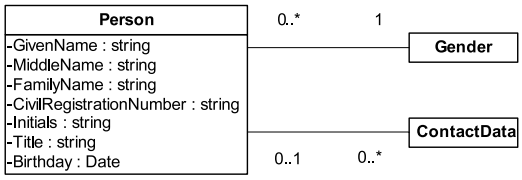


Figure 3.37: Logical model of the `Person` class and its associations to the `ContactData` and `Gender` classes.

A person has a gender (confer subsection below) and each `Person` object is therefore linked to a `Gender` object (figure 3.37). To this may be added one or more sets of contact data (section 3.12) e.g. private and corporate contact data.

The `Person` class has the following attributes:

Attribute Name	Description
GivenName	The given name of the person (also known as first name).
MiddleName	The middle name of the person.
FamilyName	The family name of the person (also known as surname).
CivilRegistrationNumber	The civil registration number of the person. The number uniquely identifies a person in the country of which the person has his primary citizenship. In Denmark the civil registration number is a 10 digit number which all Danish citizens are assigned by the "Centrale Personregister". The first six digits of the number is the citizen's date of birth, the next three digits comprise a serial number and the last digit is a check number.
Initials	The initials of the person.
Title	The title of the person.
Birthday	The person's date of birth.

3.11.1 Gender

Human beings are divided into two groups of gender: the masculine and the feminine gender. A gender is represented by the `Gender` class (figure 3.38).

The `Gender` class has the following attribute:

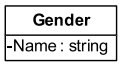


Figure 3.38: Logical model of the `Gender` class.

Attribute Name	Description
Name	The name of the gender e.g. “Female”.

3.12 ContactData

The `ContactData` class represents the contact information – i.e. address, phone numbers *&c.* – of a person or a department (figure 3.39). A person may have several contact data of different types – e.g. “Private” or “Corporate”.

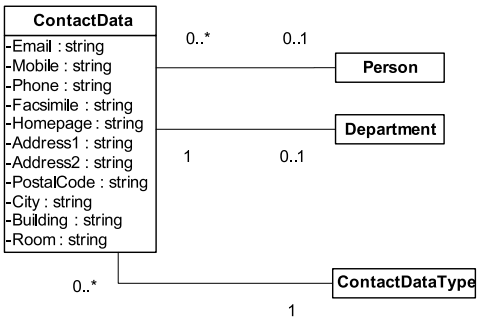


Figure 3.39: Logical model of the `ContactData` class and its associations to the `ContactDataType` and `Person` classes.

The `ContactData` class has the following attributes:

Attribute Name	Description
Email	The email address of the contact.
Mobile	The mobile phone number of the contact.
Phone	The phone number of the contact.
Facsimile	The facsimile number of the contact.
Homepage	The url for the home page of the contact.
Address1	First line of the contact address.
Address2	Second line of the contact address.
PostalCode	The postal (zip) code of the contact.
City	The name of the city of the contact address.
Building	The building number where the contact is located.
Room	The room number of the contact’s office.

3.12.1 ContactDataType

The `ContactDataType` class represents the possible types of contact data (figure 3.40).

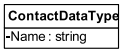


Figure 3.40: Logical model of the `ContactDataType` class.

The `ContactDataType` class has the following attribute:

Attribute Name	Description
Name	The name of the contact type – e.g. "Private" or "Corporate".

3.13 Point

The `Point` class defines a fixed number of credit points for instance 5 points as well as an interval of credit points like for example $[0; 180]$.

When used to represent the number of credit points obtained by passing a course or a project the fixed number is used (figure 3.41). The generalization between the `Point` and `PointFixed` classes represents this property.

The `Point` class is also used to specify a lower and an upper bound of desired workload in a period e.g. $[10; 25]$ or to define a recommended placement (section 3.8) by specifying an interval of credit points which should be collected. This property is represented by the generalization between the `Point` and `PointInterval` classes.

Finally, the `Point` class may also be used to represent a lower bound only – used when the upper bound is unspecified. For example to start his thesis a Master of Science student must have collected at least 235 credit points [10].

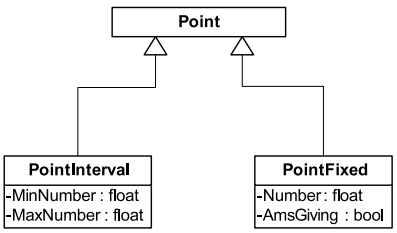


Figure 3.41: Logical model the `PointInterval` and `PointFixed` classes and their generalizations to the `Point` class.

The `PointInterval` class has the following attributes:

Attribute Name	Description
MinNumber	The lower bound of a credit point interval.
MaxNumber	The upper bound of a credit point interval.

The `PointFixed` class has the following attributes:

Attribute Name	Description
Number	Used when the instance is representing the workload of a course or project or required amount of points for a given study type.
AmsGiving	Specifies whether the instance also assigns AMS[24] points to the student.

3.14 Period

A period is a time span of variable length. A given period starts at some point in time and ends at some other point later in time.

The `Period` class represents a period.

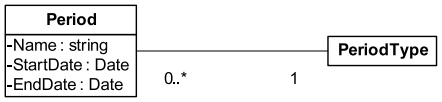


Figure 3.42: Logical model of the `Period` class and its association to the `PeriodType` class.

Each period is of some type as it appears in figure 3.42. The type of a period is represented by an association between the `Period` and the `PeriodType` class.

The `Period` class has the following attributes:

Attribute Name	Description
Name	The name of the period.
StartDate	The start date of the period.
EndDate	The end date of the period.

3.14.1 PeriodType

Each period is of some type and a period type is represented by the `PeriodType` class (figure 3.43). A `PeriodType` object may for instance be "13-week period" which is a common teaching period[25] at DTU – or "Individual period" for periods of individual character.

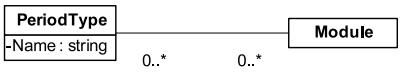


Figure 3.43: Logical model of the `PeriodType` class and its association to the `Module` class.

A period type may be split up into modules – two or more if it is to make sense. Confer section 3.15 for a description of the `Module` class.

The `PeriodType` class has the following attribute:

[24] http://www.adm.dtu.dk/studier/ams_d.htm
[25] [10], page 12.

Attribute Name	Description
Name	The name of the period type.

3.15 Module

A module is the name of a certain period of time on a given weekday. There may be none or several modules on the same weekday (section 3.20), but a module only occurs once a week.

A module is represented by the `Module` class (figure 3.44).

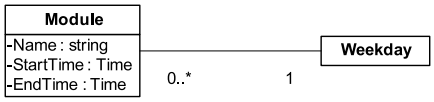


Figure 3.44: Logical model of the `Module` class and its association to the `Weekday` class.

The timetable used at DTU is shown in figure 3.45.

Time	Mon	Tue	Wed	Thu	Fri
8:00 – 12:00	1A	3A	5A	2B	4B
12:00 – 13:00	Break	Break	Break	Break	Break
13:00 – 17:00	2A	4A	5B	1B	3B

Figure 3.45: The original timetable

At DTU there are no modules after 17:00, however, some courses are taught from 17:00 to 21:00 – e.g. courses `02115`, `02393` and `42966` – which makes it convenient to introduce five new modules Monday through Friday from 17:00 to 21:00. With introduction of the new modules the resulting weekly timetable looks like this:

Time	Mon	Tue	Wed	Thu	Fri
8:00 – 12:00	1A	3A	5A	2B	4B
12:00 – 13:00	Break	Break	Break	Break	Break
13:00 – 17:00	2A	4A	5B	1B	3B
17:00 – 21:00	6	7	8	9	10

Figure 3.46: The modified timetable

The `Module` class has the following attributes:

Attribute Name	Description
Name	The name of the module – e.g. “2A” or “3B”.
StartTime	The start time for courses in this module – e.g. 08:00 or 13:00.
EndTime	The end time for courses in this module – e.g. 12:00 or 17:00.

3.16 Assessment

An assessment is the evaluation of an oral or written examination, a paper, a project work *&c.* The `Assessment` class represents an assessment (figure 3.47).

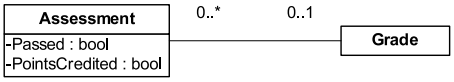


Figure 3.47: Logical model of the `Assessment` class and its association to the `Grade` class.

The `Assessment` class has the following attributes:

Attribute Name	Description
Passed	Indication of whether the student has passed or failed the course or project (specification is only necessary if the course or project is assessed as passed/failed).
PointsCredited	Indication of whether points should be credited. In some cases points are credited even though a "Fail" or a grade less than 6 is given. Special rules apply both for Bachelor of Science students (see [10], page 19) and Master of Science students (see [10], page 71).

3.17 AssessmentType

Courses and projects can be assessed in different ways. The `AssessmentType` class represents a way whereupon a course or a project can be assessed (figure 3.48). An `AssessmentType` object may for instance be "Pass/Fail" or "13-scale".

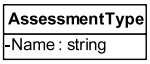


Figure 3.48: Logical model of the `AssessmentType` class.

The `AssessmentType` class has the following attribute:

Attribute Name	Description
Name	The name of the assessment type.

3.18 Grade

The `Grade` class represents the possible grades in the 13-scale grade system[26] (figure 3.49).

The `Grade` class has the following attributes:

[26] http://www.retsinfo.dk/_GETDOCM_/ACCN/B19950051305-REGL

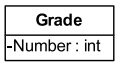


Figure 3.49: Logical model of the `Grade` class.

Attribute Name	Description
Number	Numeric representation of the grade number – i.e. 0, 3, 5, 6, 7, 8, 9, 10, 11 or 13.

3.19 Language

A language is represented by the `Language` class (figure 3.50). A language object may for instance be "English" or "Danish".

Figure 3.50: Logical model of the `Language` class.

The `Language` class has the following attribute:

Attribute Name	Description
Name	The name of the language.

3.20 Weekday

The `Weekday` class represents the days of the week (figure 3.51) as used in the Gregorian Calendar.

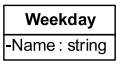


Figure 3.51: Logical model of the `Weekday` class.

The `Weekday` class has the following attribute:

Attribute Name	Description
Name	The name of the weekday e.g. "Monday".

Chapter 4

Requirements Specification

In this chapter we describe the requirements posed to a study planning system. The system has to meet every requirement specified in this chapter.

We more specifically describe the requirements posed with respect to data (section 4.1), functionality (section 4.2) and quality (section 4.3). Furthermore, a number of other requirements are stated (section 4.4).

In the requirements specification we use "supplier" as a designation of the company (or the like) who is supposed to implement the specified system. Accordingly, "customer" is used as a designation of the institution (or the like) who needs to have the specified system implemented.

It is impossible to specify every requirement down to the last detail – and unreasonable if the requirements specification is to be a useful document in the design and implementation of the system. For this reason there may very likely be several implied requirements, however, we expect the supplier to enter into a current dialogue with the customer in order to find solutions to any unclarified issues.

4.1 Data Requirements

In this section we specify the requirements made with respect to data. According to [18] data requirements deal with specifying "the data to be stored in the system, no matter whether data is stored as a database or in some other way".

The system should be capable of storing data corresponding to the logical models and descriptions presented in chapter 3 taking into account the limitations and additional data requirements introduced in this section. No demands are made on the physical data model.

Subsection headings in this section correspond to the names of the various entities by which it is convenient to look up entities in the table of contents.

4.1.1 Course

Courses change over time – for example the number of credit points may be increased/reduced or the assessment type may change. In order to retain a full history of courses it must be possible to store different versions of a course.

As mentioned in the "Modules" section on page 17 the vast majority of courses use the specification shown in figure 3.8 to describe the modules in which the course is taught. So for the purpose of simplicity only the above-mentioned specification of modules should be implemented.

Regarding course interdependencies only three courses use the specification in figure 3.12 to describe dependencies to other courses so only the specification shown in figure 3.11 should be implemented.

4.1.2 Student

A student alters over time – e.g. the student may change to another study type or technical package. In order to retain a full history of the student it must be possible to store different versions of a student.

The system should be capable of storing every sign up and examination attempt of the individual course in that rules apply with respect to the permissible number of repeater attempts.

4.1.3 TechnicalPackage

A technical package changes over time – e.g. at DTU new versions of each technical package are introduced every year. In order to retain a full history of the different technical packages it must be possible to store different versions of a technical package.

As stated in section 3.6.2, a project being part of a technical package must be executed in a specific period. However, a student may fail courses or e.g. be on leave for half a year and thus be delayed in his study. In such cases the student is not able to complete the project work in the specified period. In order to perform an expedient study planning it must therefore be possible to store an interval in the course of study (independent of specific periods) during which the project should be scheduled. This is done by linking each `TechnicalPackageProject` object to a `Point` object (figure 4.1) specifying an interval of points.

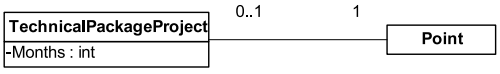


Figure 4.1: Logical model of the association between the `TechnicalPackageProject` class and the `Point` class.

Ideally, the project should be scheduled in the period specified for the project. Owing to some unforeseen circumstances (e.g. pregnancy and maternity leave) this may not be possible and the project should alternatively be scheduled in a period in which the student has at minimum obtained the specified number of points and has not exceeded the specified maximum number of points.

The `TechnicalPackageProject` class has the following attribute:

Attribute Name	Description
Months	The number of months the project work is supposed to last.

4.1.4 TechnicalLine

Technical lines may change over time – e.g. new courses are created and existing courses are discontinued. In order to retain a full history of technical lines it must be possible to store different versions hereof.

As mentioned in section 3.7.2 on page 29 the four specializations of the technical line "Applied Chemistry and Chemical Engineering" differ from the specializations of all other technical lines by being divided into an additional number of subcategories. The specification of specializations must follow the form used by all other technical lines and hence the quirkiness of the above-mentioned technical line shall not be supported in its current form. However, once it has been rewritten this technical line must be supported by the system.

4.1.5 StudyType

Study types may change over time – e.g. the required total number of credit points may change or the study type may apply to different technical packages. In order to retain a full history of study types it must be possible to store different versions hereof.

4.1.6 Grade

Grades must have both a numeric and a string representation. The numeric representation is used when computing grade averages. When presenting the grade to a user the string representation is used. To avoid possible misunderstandings and fraud attempts – e.g. accidentally writing a "1" in front of the grades – "0" and "3" are traditionally represented as "00" and "03" in the string representation which also eliminates the need for conversion between the two representations.

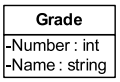


Figure 4.2: Logical model of the `Grade` class.

The extended `Grade` class has the following attributes:

Attribute Name	Description
Number	Numeric representation of the grade number – i.e. 0, 3, 5, 6, 7, 8, 9, 10, 11 and 13.
Name	String representation of the grade number – i.e. "00", "03", "5", "6", "7", "8", "9", "10", "11" and "13".

4.1.7 StudyPlanCriterion

A study plan criterion is an assembly of specifications on the basis of which the system should elaborate a study plan. The `StudyPlanCriterion` class is a representation of a study plan criterion.

It is the individual student who is assumed to state the specifications of a study plan criterion previous to the system's elaboration of a study plan.

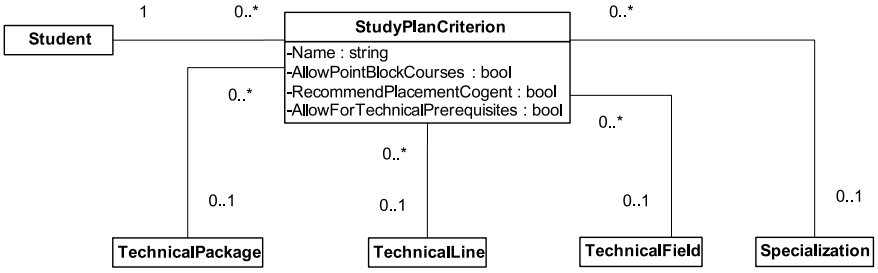


Figure 4.3: Logical model of the `StudyPlanCriterion` class and its associations to the `Student`, `TechnicalPackage`, `TechnicalLine`, `TechnicalField` and `Specialization` classes.

According to figure 4.3 a student may have one or more study plan criteria.

The student may specify that he wants to complete a specific technical package. It is possible to determine which technical package a given student is currently signed up for, however, if the student ponders switching to another technical package it may be very relevant to specify the technical package considered.

Likewise, a Master of Science student may specify which technical line he wants to obtain. For a given student is possible to ascertain which technical line the student is presently signed up for, however, if the student ponders switching to another technical line it may be crucial to specify the technical line considered.

The student may furthermore – if a technical line is specified – choose to follow a recommended course of study by specifying a technical field or a specialization.

The `StudyPlanCriterion` class has the following attributes:

Attribute Name	**Description**
Name	The name of the study plan criterion chosen by the student.
AllowPointBlockCourses	Specifies whether or not point blocking courses may be scheduled.
RecommendedPlacementCogent	Specifies whether the recommended placement of courses should be respected or ignored.
AllowForTechnicalPrerequisites	Specifies whether technical prerequisites for courses should be respected or ignored.

The student may specify a desired workload by means of an amount of credit points which should be obtained within a period (figure 4.4). This may be done either generally per period type or for specific periods.

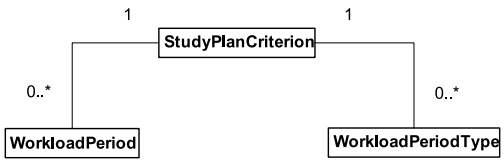


Figure 4.4: Logical model of the associations between the `StudyPlanCriterion` class and the `WorkloadPeriod` and `WorkloadPeriodType` classes.

A `StudyPlanCriterion` object may be linked to one or more `WorkloadPeriod` objects which represent the student's desired workload in specific periods. Moreover, a `StudyPlanCriterion` object may be linked to one or more `WorkloadPeriodType` objects which represent the student's default desired workload in different period types. Confer the subsections "WorkloadPeriod" and "WorkloadPeriodType" below for an in-depth description of these classes.

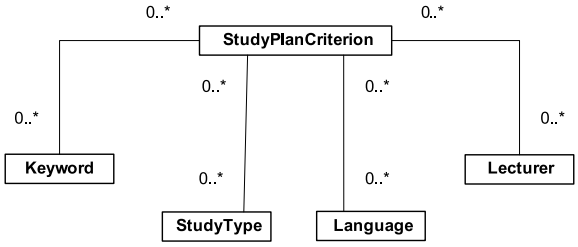


Figure 4.5: Logical model of the associations between the `StudyPlanCriterion` class and the `Keyword`, `StudyType`, `Language` and `Lecturer` classes.

Fields of interest may be specified in the form of keywords (figure 4.5).

The student may choose also to consider courses offered to study types distinct from the student's own study type. One could for instance imagine that a Bachelor of Science student would choose also to consider courses offered to Master of Science students.

Selection of one or more languages is possible. A Danish student may e.g. select both "Danish" and "English" as languages.

In the same way one or more lecturers may be deselected (all lecturers are considered selected by default). A student may e.g. choose to deselect a lecturer if he does not like the lecturer's way of teaching or the like.

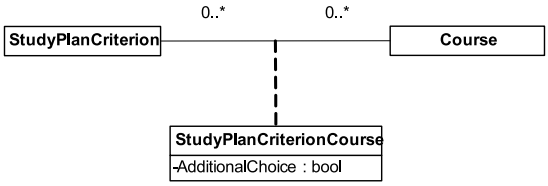


Figure 4.6: Logical model of the association between the `StudyPlanCriterion` and `Course` classes. Moreover, a logical model of the association class `StudyPlanCriterionCourse`.

The student can explicitly choose one or more courses which should be scheduled at some time during the course of study (figure 4.6). These courses may be scheduled at any time during the course of study. Likewise, the student can also deselect any number of courses which means that these courses will never be scheduled (unless this clashes with other criteria). This

property is represented by the `StudyPlanCriterionCourse` association class introduced in figure 4.6.

The `StudyPlanCriterionCourse` association class has the following attribute:

Attribute Name	Description
AdditionalChoice	Specifies whether the course should be scheduled at some time during the course of study or never.

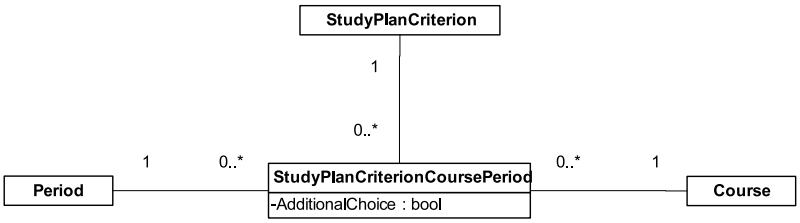


Figure 4.7: Logical model of the association between the `StudyPlanCriterion` and `StudyPlanCriterionCoursePeriod` classes. Moreover, a logical model of the associations between the `StudyPlanCriterionCoursePeriod` class and the `Period` and `Course` classes.

A student may choose to manually schedule one or more select courses in a specific period which corresponds to the traditional, manual study planning (figure 4.7). Likewise, the student can manually deselect courses in a specific period meaning that these courses will not be scheduled in that period.

The `StudyPlanCriterionCoursePeriod` association class has the following attribute:

Attribute Name	Description
AdditionalChoice	Specifies whether the course should be scheduled in the specific period or not.

As described in sections 3.4 and 3.5, students must prepare one or more projects during their studies. In order to execute a more rational study planning the student may choose to specify the workload of each project in different time intervals (figure 4.8). For example the student might want to specify the workload of a 12.5 point project as follows:

Interval	Points
1 – 20%	5.0
21 – 80%	5.0
81 – 100%	2.5

The `ProjectWorkload` class has the following attributes:

Attribute Name	Description
FromPercent	Specifies the lower bound of the workload interval e.g. 20%.
ToPercent	The upper bound of the workload interval e.g. 80%.

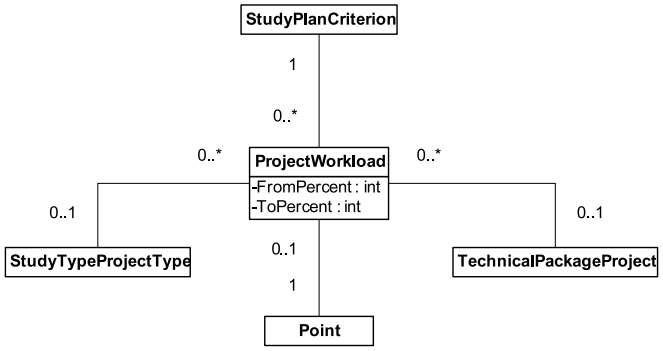


Figure 4.8: Logical model of the association between the `StudyPlanCriterion` and `ProjectWorkload` classes. Moreover, a logical model of the associations between the `ProjectWorkload` class and the `TechnicalPackageProject`, `StudyTypeProjectType` and `Point` classes.

WorkloadPeriod

A workload period describes the workload – i.e. the minimum and maximum number of credit points – desired by the student in a specific period (figure 4.9). For example a student may specify a desired workload of 15 to 25 points for the 13-week period in the spring 2003.

Furthermore, a student may deselect specific modules in a period. This means that no courses will be scheduled in these modules during the period. If a student wants to play tennis every Tuesday afternoon he or she can deselect the module(s) on Tuesday afternoons of the week schedule. In DTU terms that would be the module 4A – confer section 2.2.2 for a description of the week schedule used at DTU.

A workload desired for a specific period takes precedence of the workload desired generally in periods of the same type (confer subsection below).

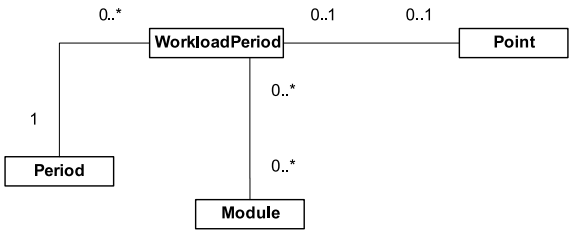


Figure 4.9: Logical model of the associations between the `WorkloadPeriod` class and the `Period`, `Module` and `Point` classes.

WorkloadPeriodType

A workload period type describes the workload – i.e. the minimum and maximum number of credit points – desired by the student in specific period types (figure 4.10). This is the general setting which is used for all periods of the selected period type unless it is overruled by a `WorkloadPeriod` object introduced in figure 4.9.

For example a student may specify a default desired workload of 25 points for all 13-week periods and a workload of 5 points for all 3-week periods. However, suppose the student

wants to go on summer holiday in June 2003 and thus he will specify a desired workload of 0 points in the 3-week period in June 2003. This will then overrule the general setting and no courses will be scheduled for this period.

A student may deselect chosen modules in a specific period type. This means that no courses will be scheduled in these modules for all periods of this particular period type.

Now, consider again the above-mentioned example with the student who wants to play tennis every Tuesday afternoon. Suppose the tennis lessons are rescheduled from Tuesday afternoons to Thursday afternoons for the second half of the year 2003. The student will then specifically reselect the Tuesday afternoon module(s) and deselect the Thursday afternoon module(s) for the specific periods during the second half of the year 2003. No courses will then be scheduled for Thursday afternoons until 2004 where the previous setting is restored.

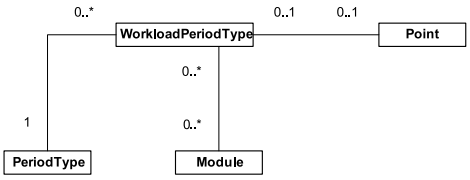


Figure 4.10: Logical model of the associations between the `WorkloadPeriodType` class and the `PeriodType`, `Module` and `Point` classes.

4.2 Functional Requirements

In this section the requirements made with respect to functionality are specified. With reference to [18] functional requirements specify "what data is to be used for, how it is recorded, computed, transformed, updated. . . ".

Below is among other functional requirements described how the student should be able to manage his study plan criteria and elaborate a study plan from a study plan criterion.

4.2.1 Principal Requirement

The system must be web-based and users should therefore be able to operate the system by a web-browser.

4.2.2 Restricted Admission

Admission to the system must be restricted. When a person tries to access the system he should therefore be directed to a login page from which he can be authenticated and thereby get admission to the system.

Login Page

On the login page the user should be able to state his credentials – username and password – and request for admission. Upon request the system must validate the stated username and password.

If either the username or password is wrong the system must refuse the user's request for admission and inform so to the user. The user may try to login again, however, the user only has got a limited number of attempts, confer the subsection "Prevention of Brute Force Attacks" below.

The user should be authenticated if both the username and password are correct. An authenticated user may use the system freely, however, the user may only obtain access to those parts of the system which he has been authorized access to. Confer [12], § 12, and section 4.4.1.

Prevention of Brute Force Attacks

The system must prevent brute force attacks. In other words it must not be possible for unauthorized persons to carry out an exhaustive test of usernames and passwords in combination.

Every refused request for admission must be registered. If a request for admission has been refused three times in succession within the last five minutes for a given username the system should obstruct further requests for that username in a time interval of 20 minutes.

Password Policy

Users should be forced to change their password every 90 days. If the user has not changed his password within this time frame at login he should be forced to change his password before he is given admission to the system.

It is required that passwords meet the following complexity requirements:

- A password must at a minimum be eight characters long.
- A password must contain characters from at least three of the following categories:
  - English Lower Case Letters (a, b, c,... z)
  - English Upper Case Letters (A, B, C,... Z)
  - Westernized Arabic Numerals (0, 1, 2,... 9)
  - Non-alphanumeric, "special characters" (!@#$%&()_*-+={}[],.:/|)

Finally, a password must not be the same as any of the user's previous four passwords. The system must therefore maintain a log of the user's previous passwords.

4.2.3 Main Page

When a student has logged in, a main page shall be presented by the system from where it must be possible to get an overview of previously saved study plan criteria and study plans – where available.

From the main page it must be possible to enter the study plan criterion management (section 4.2.4) or the study plan management (section 4.2.5).

The system must allow deletion of study plan criteria or study plans either from the main page or from the respective management functions. The supplier shall provide a suggestion as to which of the two is the more appropriate solution.

4.2.4 Study Plan Criterion Management

Students must be able to manage a number of study plan criteria in the system. Partly, a student should be able to create a new study plan criterion and partly modify and delete existing study plan criteria. Moreover, it should be possible to elaborate a study plan from a study plan criterion.

Creation

The creation of a study plan criterion consists in the student specifying his interests, specific demands *&c.*

The student should be able to specify which *technical package* he wants to complete – if different from the one he is already signed up for. M.Sc. students should be able to select a technical line as well as a specialization. Only technical package and technical line relevant for the student's study type should be selectable.

As regards the student's *interests* it should be possible for the student to specify those as keywords.

Desired *workload* by means of an amount of credit points should be specifiable both generally per period type and for specific periods.

A study plan should reflect the student's language skills and the student should therefore be able to select one or more *languages* in which he wants to be taught. One or more *lecturers* should be deselectable as all lecturers are selected by default and the student may not wish to be taught by certain lecturers.

The student should be able to choose one or more *study types* in addition to the student's own study type. The different study types should be selectable from a list.

It should be possible for the student to choose one or more *specific courses* which the student would like to have scheduled at some time during the course of study. In the same way the student should be able to choose one or more courses which should never be scheduled. Likewise, for specific periods the student should be able to select and deselect one or more courses – such a specification corresponds to traditional, manual study planning.

The student should be able to define whether the *technical prerequisites* of the courses should be taken into account or not.

As far as *projects* are concerned the student should be able to specify the amount of workload associated with a project in different parts of the project. The student should be able to make such a specification for each of the projects the student is going to prepare.

The student should be able to state a *name* for the study plan criterion such that the student can identify it later on.

Finally, it should be possible to *save* a study plan criterion.

Modification and Deletion

A student should be able to modify and delete existing study plan criteria. It should be possible to modify the same criteria which were possible to specify at the creation of a study plan criterion. If a student chooses to delete a study plan criterion he should confirm this choice before the actual deletion is executed.

Study Plan Elaboration

From a given study plan criterion the student should be able to request an elaboration of a study plan.

A study plan should be elaborated according to the guidelines described in section 4.2.6.

4.2.5 Study Plans

The system must allow the user to save any number of study plans and to assign a name to each individual study plan in order to facilitate identification of the study plan.

Presentation of a study plan should be as a list of courses (course number and course name) as well as projects grouped by study plan periods. The study plan period to which the courses or project belong must be clearly identified. Moreover, all courses shown must be linked to its associated description such that clicking a course will bring up the detailed description for the course.

The system should allow deletion of the study plans created by the user who is currently logged in. It should not be possible to edit study plans manually as this may violate the assumptions made for other study plan periods and thus introduce inconsistency in the study plan.

4.2.6 Study Plan Routine

The study plan routine should elaborate a study plan on the basis of a given study plan criterion. In the following is outlined the chief mode of operation of the study plan routine.

The routine should begin by generating a set of courses which matches the student's interests i.e. the specified keywords. To this set must be added those courses explicitly desired by the student.

If any of the courses in the set are taught in a language or by a lecturer deselected by the student these courses should be removed from the set. Courses comprised by the student's technical package (or the one selected) should be added to the set and if the student has desired to complete either a technical line or a specialization then the courses comprised hereby should be added to the set.

Those courses explicitly deselected by the student should be removed from the set. The same applies for those courses already finished by the student.

The set of courses should be distributed among the different periods in consideration of desired workload and possible parallel projects. The succession in which courses are arranged should be expedient by considering mandatory and possibly technical prerequisites.

If a course consists of multiple parts, the individual parts must be scheduled sequentially in ascending order in the first possible periods in which they are taught as mentioned in section 3.3.1.

The study plan should be elaborated such that it is primarily composed of courses from the technical package and from a possible specified technical line or specialization. The courses

explicitly selected are of secondary importance. Courses matching the student's interests by means of keywords have of all lowest priority[1].

The routine should – given a student's identity – be able to compute how many credit points the student has achieved and has to achieve in order to complete his study. Having this information the routine has knowledge of how many credit points in the form of courses for which a study plan should be made.

The further particulars with regard to how the study plan routine should function is to be mapped out in close and current collaboration between the customer and the supplier.

4.2.7 Culture Versioning

The system shall support Danish and British cultures including the correct formatting of culture dependent fields such as dates, currencies, numbers, sort orders for strings *&c.*

For future use the system shall be able to cope with other European cultures.

4.2.8 Persistent Functionality

The system shall allow the user to perform the following functions anywhere in the system i.e. the functions must always be accessible:

- Switch to the home page.
- Log off the system.

4.3 Quality Requirements

In this section we specify the requirements made with respect to quality – also known as non-functional requirements. According to [18] quality requirements specify "how well the system must perform its functions".

Below we more specifically describe capacity, performance, usability, security and extensibility requirements.

4.3.1 Capacity

It is expected that on average 2% of the users (students) – corresponding to approximately 120 users[2] – will use the system simultaneously. However, the system must be able to handle peak loads of up to five times as many simultaneous users which are likely to occur the last few days before the course sign up deadlines for each period.

Below some initial capacity requirements are introduced along with the expected yearly growth rate.

[1] Immediately, it may sound peculiar that the student's interests have a low priority, however, the mandatory courses must inevitably be passed in order to start the technical line and specialization (which the student also has selected by interest). Furthermore the student's interest are represented by keywords and since the keywords associated with the courses are not weighted it is impossible to determine how large a part of the course actually matches the various keywords.

[2] The estimate is based on the number of students at DTU in 2003.

Object type	Initial capacity	Expected yearly growth
Student	6.000	1.000
Course	1.000	100
Department	20	0
Lecturer	1.200	120
Study Plan	12.000	4.000

Initially, there are no study plans in the system, however, it is anticipated that every student will create an average of two study plans shortly after implementation of the system.

Initial capacity

The initial capacity of the system must be sufficient to store all students, all courses, all departments, all lecturers *&c.* Furthermore, the system must initially have a capacity that can handle at least two study plans per student.

Scalability

The system must be scalable since it is anticipated that most students will use the option to create multiple study plans. Moreover, new courses and projects are created on a regular basis and existing courses and projects are altered and thus resulting in a new version of the course or project.

4.3.2 Performance

Response time for elaborating a study plan shall be at most 20 seconds in 95% of the cases. In 50% of the cases the response time shall be less than ten seconds.

The specified response time is on the server side alone disregarding any internet delay as the supplier cannot be held responsible for possible internet lags. The system must run on existing hardware which the above-mentioned response times allow for. Specifications for the hardware are available upon request.

4.3.3 Usability

Use of the system should to a great extent be self-explanatory and users should be able to use the basic features of the system without prior instruction. Notice, the primary target group of the system is students which is a group of young users likely of being experienced with the Web and using web applications.

The system should efficiently support the user's tasks. As an example a user should not have to split a given task up into unnecessarily many sub-tasks and execute those in different parts of the system in order to complete the overall task.

The system must be capable of taking over tasks on a detailed level such that users need not enter information which the system is able to derive from existing information.

4.3.4 Security

As regards security the system must take precautions against a number of issues on that subject.

When transmitting information over the internet there is a risk that the information is read by third party. In some cases information of sensitive character (e.g. grades) have to be transmitted and such information are encompassed by §§ 7-8 in [12], confer section 4.4.1. In order to ensure confidentiality the transmission between client and server must be done via the https protocol in conformity with recommendation of "The Danish Data Protection Agency" ([25]).

The system must not be vulnerable to injections into the data base (SQL, XPath or the like injections) and buffer overruns. As the described system is to become a web application the system should also make countermeasures in respect of URL tampering, content suction and script injections[3].

4.3.5 Extensibility

The system must be designed and implemented in a way that allows for future extensions which may reasonably be expected. Extensions must be implementable without requiring a fundamental redesign of the system.

4.4 Other Requirements

In this section the remaining requirements are specified. These requirements do not fall into the categories of the requirements already specified.

4.4.1 Legal Requirements

The system must abide by the regulations of [12] on the protection of individuals with regard to the processing of personal data.

In this requirements specification a number requirements have been presented which ensure observance of the regulations in [12]. However, on the basis of the afore-mentioned act the supplier must determine whether further initiative is necessary in order to satisfy the requirements specified herein. If this is the case the supplier must see to it that the system supports these measures.

4.4.2 User Interface Requirements

The supplier should draw up prototypes of the user interface (also known as virtual windows) before developing the actual user interface. These prototypes should be debated with the customer and altered accordingly. The actual user interface should of course then be developed on the basis of the final prototypes.

[3] At discretion, examples of the different types of attacks are given in [2], chapter 2.

At all times the user should have a general view of the current part of the system being used and where that part is placed in the navigation hierarchy.

The graphical design should be professional and simple in order to prevent unnecessary absent-mindedness from the user's tasks.

The system should be usable in the screen resolutions as follows: 800×600, 1024×768 and 1280×1024. No matter which screen resolution the user has the whole screen area should be exploited.

4.4.3 Technical Requirements

On the server side the system must be able to run on the existing hardware. Specifications for the existing hardware are available upon request.

On the client side the system will be used from a number of different platforms and therefore the client must comply to the following standards in the version specified or higher:

- JavaScript version 1.2
- Cascading Style Sheets (CSS) level 2
- Document Object Model (DOM) version 1
- HyperText Markup Language (HTML) version 4.01
- Secure Sockets Layer (SSL) version 3.0

4.4.4 System Design and Programming

Only a few requirements are made on system design and programming.

It is required that the system architecture is divided into at least three tiers: one tier for handling presentation, one tier for managing business logic and one tier for handling storage of data. The system architecture may be divided into further tiers e.g. by dividing the business tier into a user-oriented and a data-oriented tier, however, the principal requirement is that the system has a clear-cut separation between presentation logic, business logic and data.

The system is required to be developed based on object-oriented principles.

4.4.5 Documentation

Documentation is required to be supplied with the system.

The documentation should include:

- Description of data structures. The entities of the physical data model are to be described and for each entity it should appear:
  - what purpose the entity serves.
  - how the entity is related to other entities.
- Description of file structure. For each file it should appear:
  - what purpose the file serves.
  - how the file is possibly related to other files.
- Description of components. For each component it should appear:

  - what purpose the component serves.
  - how the component is possibly related to other components and entities in the database.
- Description of other parts such that adequate documentation is available of every part of the system.

The system documentation is expected to be supplied in electronic form as a number of coherent web pages and images.

In addition, there should be comments in the source code itself. The comments are required to have an extent such that outsiders with necessary technical competence within reasonable time are able to get a clear comprehension of what the concerned source code does.

System documentation and source code comments should be in English.

Part II

System Design & Implementation

Chapter 5

Foundations

Based on the prepared requirements specification, we have implemented a system that supports automated study planning.

In this chapter we describe the foundations of the implemented study planning system. The following subjects will be covered:

- The chosen technology (section 5.1)
- Select aspects of the database design (section 5.2)
- The main architecture of the system (section 5.3)
- The distinct so-called tiers of the system (sections 5.4, 5.5 and 5.6)
- How errors are handled (section 5.7)

Documentation of the implemented system can be found at:

http://www.studyplanning.dk/documentation

5.1 Choice of Technology

As an initial step we will explain our choice of technology and considerations made in this connection.

5.1.1 Development Platform

Several development platforms qualify to support the development of the study planning system. We have, however, chosen to implement the system using Microsoft's .NET Framework.

The main reason for choosing .NET is that the system is intended to be employed at DTU. At the present time DTU already offers a number of services which are implemented by use of Microsoft technologies and it will – other things being equal – be easier to integrate the study planning system with these other systems if the systems are implemented in technologies from the same vendor.

In .NET the choice of programming language is a question of personal preference as several programming languages are supported[1], including:

[1] http://msdn.microsoft.com/vstudio/productinfo/whitepapers

- Visual Basic .NET
- Visual C# .NET
- Visual C++ .NET
- Visual J# .NET (Java in .NET)

We have chosen C# as the language to be. The choice of C# seemed direct as both of the authors have previously developed systems using Java and in our opinion it holds to argue that C# *is* indeed Java improved in a handful of fields. However, most often such a choice is a question of personal preference.

5.1.2 Storage of Data

As it appeared from section 4.1 the system needs to store a fairly heterogeneous quantity of data. The data has to be stored in some organized form and we first of all decided on using the following technologies for storing data:

- **A relational database system**
  A mature, prevalent, theoretically well-founded and efficient way of storing structured data. In addition, data is queried by means of the standardized SQL[2] language.

- **Extensible Markup Language – XML**
  A flexible, standardized way of storing semi-structured data. Moreover, data is queried by means of the standardized XPath and XQuery languages[3].

Which specific relational database system to choose? Again, several systems qualify. We have chosen to use Microsoft SQL Server as relational database system.

A number of reasons apply for this choice:

- SQL Server is well-integrated with .NET both as regards classes in the framework and efficiency.
- Storage and querying of XML is supported.
- The concept of stored procedures is supported which we prefer for architectural and security reasons.
- Both authors had prior experience with the concerned database system.

5.2 Database Design

In this section we elaborate on some select aspects of the database design. The following subjects will be covered:

- Reflections concerning primary keys (section 5.2.1)
- Globally unique identifiers (section 5.2.2)
- How versioning is handled (section 5.2.3)
- Tracking of changes (section 5.2.4)
- How culture versioning is handled (section 5.2.5)
- Considerations regarding referential integrity (section 5.2.6)

[2]SQL is an abbreviation for Structured Query Language – the language is described in e.g. [27], chapters 5-7.

[3]Descriptions of the XPath and XQuery languages can be found at http://www.w3c.org.

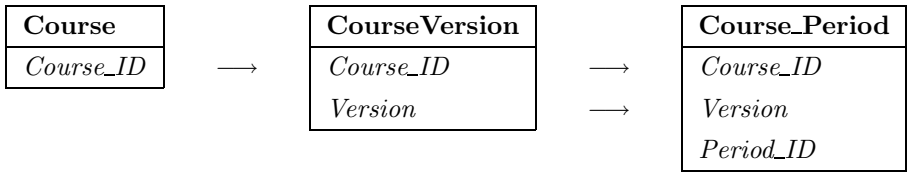
5.2.1 Primary Keys

When designing relational database tables an important part of the design is the definition of primary keys. Basically, the choice is between simple keys and compound keys or a combination of these.

To illustrate the differences between the two approaches an example using the two main entities for courses is shown in the following.

Using compound keys

The base table `Course` has the primary key `Course_ID`, the `CourseVersion` table has the compound primary key consisting of `Course_ID` and `Version` and the `Course_Period` table has a compound primary key consisting of `Course_ID`, `Version` and `Period_ID` as shown below where keys are in *italic*:

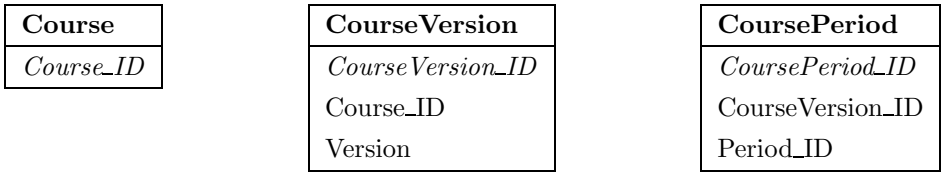


Using compound keys means that any table referencing the key from another table must include the primary key from that table as part of its own compound key. This is illustrated by the `CourseVersion` table which includes the primary key from the `Course` table and by the `Course_Period` table which includes the primary key from the `CourseVersion` table.

Extending the primary key of the `Course` table in the above-mentioned example with another attribute into a compound key would also imply changing the `CourseVersion` and `CoursePeriod` tables since their keys have been derived from the primary key of the `Course` table.

Using simple keys

The base table `Course` has the simple primary key `Course_ID`, the `CourseVersion` table has the simple primary key `Course_Version_ID` and the `CoursePeriod` table the simple primary key `Course_Period_ID` as shown below where keys are in *italic*:



When using simple keys each table has its own primary key consisting of only one attribute which is independent of other keys.

Compared to compound keys, simple keys are easier to reference since only one attribute must be referenced and not a compound key consisting of several attributes – e.g. when passing the key from one function to another. This also applies when programming the business and presentation tiers since a single value uniquely identifies the desired data record.

Since the primary key itself represents no data it is easy to extend the table with new attributes as the key will not have to be changed.

5.2.2 Globally Unique Identifiers

When defining keys for database tables one may use either logical keys or surrogate keys. A logical key consists of the actual data, which identify the record – e.g. for a table containing contacts the key might be the family name of the contacts. A surrogate key consists of unique, arbitrary values which abstractly represent the records in the table[4] – e.g. by assigning a number to each contact in the above-mentioned example.

Using logical keys is disadvantageous if the actual data in the key is subject to changes since foreign keys of all referencing tables would have to be updated each time the data changes.

Surrogate keys are typically represented as integers and many database systems can even increment the key automatically when inserting a new row into the table thus preventing duplicate keys. However, often the (automatically generated) key from one table must be inserted as a foreign key into another table which again depends on the foreign key of a third table. This means that the insertions must be carefully planned and executed in a specific order for this to work.

Moving data between multiple instances of the database – e.g. from a production system to a test system – may also be difficult when using this approach. The surrogate keys generated in the production system may already exist in the test system and thus the data cannot be inserted into the tables of the test system unless the offending records are deleted from the test system or the values of the keys are changed which may be an overwhelming task if several hundred tables are involved.

Another approach is to use a surrogate key which is guaranteed to be (almost) globally unique. Microsoft SQL Server and the .NET Framework have a built-in data type called "GUID" which is an abbreviation of "Globally Unique Identifier". The GUID is a 128 bit integer which is guaranteed to be (practically) globally unique. The generation of a GUID uses the encoding of the MAC address of the computer's network card as well as other machine specific information including some randomness (confer [13]). A GUID is usually written in hexadecimal form like the example below[5]:

```
936DA01F-9ABD-4D9D-80C7-02AF85C822A8
```

Using GUIDs solves the above-mentioned problems with integer keys, however GUIDs also have some disadvantages.

- They are long and difficult to type and as such unfit for presentation and printing on invoices *&c.*
- It is not possible to see in which sequence two GUIDs have been generated.
- Being 128 bit integers they do take up more space than regular 32 bit integers and scanning a database index containing GUIDs may be slightly slower than if the index contained 32 bit integers.

Considering the pros and cons of using integers or GUIDs we have come to the conclusion that the advantage of GUIDs being globally unique makes up for the space and speed disadvantages.

To make the best of approaches the study planning system uses both integer keys and GUID keys. Manually incremented integer keys are used on entities which are static or only rarely updated whereas GUIDs are used on entities which are updated regularly.

[4] http://www.miswebdesign.com/resources/articles/wrox-beginning-php-4-chapter-3-3.html
[5] http://msdn.microsoft.com/library/en-us/acdata/ac_8_con_03_2uox.asp

5.2.3 Versioning

Versioning is used whenever an instance is likely to change a number of times – e.g. courses, students or technical lines. For example the duration of a technical package is at least two and maximum four years. Technical packages are usually changed (slightly) every year which means that up to four different versions of a technical package must coexist.

If the amount of credit points for a course is changed from say 7.5 to 5 without creating a new version of the course, all students who have signed up for the course prior to the change are suddenly only credited 5 points for the course and they have no way of proving that the course was supposed to give 7.5 credit points because the previous information has been overwritten. If instead a new version of the course had been created the historical information would have been retained and could have been used as documentation.

The versioning is handled uniformly in the system by means of *base* entities and *version* entities. The base entity contains nothing but the identification of the instance whereas the version entity contains all information which is subject to changes. In the following example `Course` is the base entity and `CourseVersion` is the version entity:

Course
Course_ID
...

CourseVersion
CourseVersion_ID
Course_ID
Version
Number
Name
...

When to update the existing instance or create a new version is entirely up to the administrators of the system. However, for the versioning to make sense it is recommended that an instance is only updated when correcting a spelling or punctuation mistake and anything beyond these simple corrections should result in a new version of the instance. Changes of contents (semantic changes) such as a course description or the modules in which a course is taught should certainly result in a new version.

5.2.4 Change Tracking

Keeping track of changes to the database is important when trying to locate errors, for statistical purposes and as documentation of occurrences in the system.

To enable an expedient tracking of changes the following four attributes occur on every single table in the system:

Attribute	**DataType**	**Description**
Created	DateTime	Date and time of when the record was created
CreatedBy	GUID	User-id of the user who created the record
Updated	DateTime	Date and time of when the record was last updated
UpdatedBy	GUID	User-id of the user who last updated the record

Since it is not possible to determine in which sequence GUIDs[6] have been generated the `Created` attribute may be used to sort the records in sequence or to extract all records which have been created on a given date. If an erroneous change has been made to some data it is easy to locate the user who made the change by looking at the contents of the `UpdatedBy` attribute and then ask the user about the change.

5.2.5 Culture Versioning

The study planning system is going to be used by many students – foreign as well as Danish. Therefore the system shall at least support the Danish and English languages. However, restricting the system to a fixed number of languages may be a problem if the university some day wants to support German or French as well.

First and foremost culture versioning includes all texts in the study planning system which must be representable in multiple languages. Secondly culture versioning also affects the way dates and currencies are displayed as well as the sort order of strings, however, only the representation of texts in multiple languages will be covered in this section.

One way of introducing text in multiple languages is to store all texts in a separate table as in the example shown below where keys are in *italic*:

CourseVersion
CourseVersion_ID
Version
~~Name~~
~~Objective~~
...

TextTable
Key
TextGroup
TextNumber
Culture
Text

All text attributes are removed from the entities and instead represented as rows in the text table. In the example shown above the attributes `Name` and `Objective` will be removed from the `CourseVersion` table and transformed into rows in the `TextTable` like for example:

Key	**TextGroup**	**TextNumber**	**Culture**	**Text**
ABC	123	456	da-DK	Systemanalyse
ABC	123	456	en-GB	System Analysis
ABC	234	567	da-DK	A. At give en grundlæggende ...
ABC	234	567	en-GB	A. The students shall basically ...
...	...	...	...	...

This design is quite flexible when it comes to introducing new text attributes which can be added without the need for altering the table, however, it does have some disadvantages. Since all text attributes are gathered in one single table extracting data for statistical purposes requires many join operations to retrieve the correct texts for each record in a table. It may also be difficult to maintain a general view of the database when text attributes are gathered in one table while all other attributes are scattered around other tables. Furthermore, the `Key` field on the `TextTable` must be able to hold all types of keys like integers, strings, dates

[6] See section 5.2.2.

&c. Compound keys need special treatment as they must be concatenated in order to fit into the single `Key` field.

Another possibility is to use a combination of relational and semi-structured data like XML. In doing so the structure of the database remains intact – i.e. no attributes are removed from the entities. Instead multiple translations of texts are stored as XML in a single attribute of the entity. Using this approach on the example introduced above the attributes `Name` and `Objective` will contain the following:

Name	**Objective**
<cultures> <culture> <cultureID>en−GB</cultureID> <value>System Analysis</value> </culture> <culture> <cultureID>da−DK</cultureID> <value>Systemanalyse</value> </culture> </cultures>	<cultures> <culture> <cultureID>en−GB</cultureID> <value>A. The students shall basically ...</value> </culture> <culture> <cultureID>da−DK</cultureID> <value>A. At give en grundlæggende ...</value> </culture> </cultures>

Storing multiple values in a single attribute violates the rule of 1st normal form in database design which states that any attribute must contain only one single value ([27], page 153) however, this is a deliberate choice which has been made.

The advantage of combining semi-structured and relational data is that the structure of the database persists while providing the possibility of storing an arbitrary number of translations of a text in a single attribute. All records need not have the same number of cultures and introducing a new culture is a simple task.

One disadvantage is that a lot of space is wasted on meta-data. In the example above 174 characters are used to represent the contents of the `Name` attribute in two languages. Of this 79 % of the space is occupied by meta-data and the remaining 21 % contain the actual data of two languages. The amount of meta-data is constant and thus the meta-data to data ratio will be less striking when greater amounts of data is stored.

Moreover, searching for a specific text in the XML string may provide some challenges in a non-XML enabled database system. Suppose a user searches for the word “culture” in the `Objective` attribute shown above. The search would match every single record in the table.

Given the advantages and disadvantages of the various ways of performing culture versioning the combination of relational and semi-structured data has been selected. Regardless of which approach has been chosen it has no consequences whatsoever for the business tier as all data access is handled transparently by the data tier.

5.2.6 Referential Integrity

In theory using Referential Integrity (RI) may seem like a good idea when designing a database because it enforces the relations between tables either by restricting deletes and inserts or cascading deletes. RI may prevent programmers and others who are tampering directly with the database from introducing an inconsistency into the database.

However, properly written applications will make sure that the referential integrity constraints are not violated and thus additional checks on the database level will only slow down performance. In large systems it may be possible to make such a complex topology of relations

that it becomes very difficult and in some cases impossible to back up or restore individual tables[7].

In the study planning system a prominent quantity of initializing data containing all the rules, types, periods *&c.* which are used in the system must be entered manually. Using RI would introduce a lot of constraints on the sequence in which the data is entered.

We find that business logic should not be built into the database by means of RI since it may restrict the possibilities of changing the business logic which is likely to change more often than the database itself. Therefore RI is not used in the database for the study planning system.

5.3 Main Architecture

In this section we present the main architecture of the study planning system. We have chosen to divide the system into the following three tiers[8]:

Tier Name	**Description**
Data Tier	Encompasses all interactions with the database i.e. creation, retrieval, alteration and deletion of data.
Business Tier	Encompasses rules, processes, workflow *&c.*
Presentation Tier	Encompasses all logic associated with the user interface.

This way of dividing the application is prevalent and is generally referred to as a 3-tier architecture. The implemented architecture complies with the requirements posed in section 4.4.4.

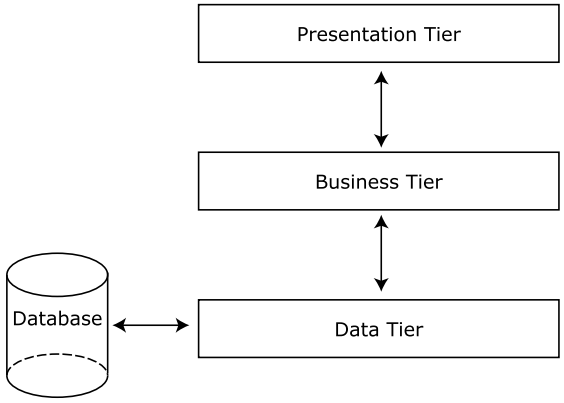


Figure 5.1: Illustration of the three tiers in the architecture.

There are numerous advantages of dividing the system up into tiers, the main points being[9]:

[7] http://www.mysql.com/doc/en/ANSI_diff_Foreign_Keys.html

[8] We have been inspired by the architectures presented in [3] and [22] and by the discussions made in chapter 10 of [16].

[9] The listed advantages (especially the "Reuse" and "Consistency" points) are closely related to componentizing the application which can said to be a direct part of dividing the application up into tiers.

- **Encapsulation** – plain interfaces hide irrelevant details from users of the interfaces. A case in point is that programmers working with the business tier do not have to possess any knowledge of how to interact with the database – they just use the interfaces provided by the data tier.

- **Adaptability** – the application is very adaptable to modified needs and changes in its environment. The presentation tier and business tier are for instance entirely unaffected by changes in the data tier, say, if the application changes to use another database system.

- **Reuse** – different parts of an application have similar functional needs. Two different parts of the presentation tier may have the need to present the fairly same set of data and may therefore reuse a common functionality in the business tier.

- **Consistency** – a specific functionality is handled in exactly one part of the application and is therefore handled consistently (a consequence of the "Reuse" point). Handling a certain business rule in several parts of an application involves the risk that the rule will be handled inconsistently at some point if the rule is not altered systematically everywhere in the application when it has to be handled differently.

From the above it appears that we have chosen not to divide the business tier further up into a user-oriented and a data-oriented tier as was suggested in section 4.4.4 as an option of further application division. The choice is deliberate and based on the fact that the system in its present version is only addressed to a single type of user agent, namely desktop client browsers, and the benefit of a further division is thus not sufficient. Provided that the system in the future is to be addressed to other types of user agents (like e.g. rich mobile clients and handheld PCs) it would most likely be worth to spend some resources splitting up the business tier further.

The division into tiers has the implication from a design point of view that the code, which manages the tasks in a certain tier, is organized within the same namespace. The three tiers in the study planning system have been organized in the following namespaces:

Tier Name	**Namespace**	**Described in**
Data Tier	`StudyPlanning.DAL`	Section 5.4
Business Tier	`StudyPlanning.Biz`	Section 5.5
Presentation Tier	`StudyPlanning.UI`	Section 5.6

As an example all the code that performs operations on the database is sited directly in the `StudyPlanning.DAL` namespace or in a namespace being subordinate hereto.

The three tiers are described in more detail in the sections stated in the above table. Please note, error handling in the three tiers is described in a common section, namely section 5.7.

5.4 Data Tier

The purpose of the data tier is to manage all interaction with the database. Any form of creation, retrieval, alteration and deletion of data in the database is thus handled by the data tier.

By encapsulating all database interactions within a separate tier one obtains among other things that the other parts of the application are not troubled with the quibblings and issues usually involved with interacting with a database. The other application parts just use the interfaces provided by the data tier.

The data tier first and foremost contains classes that represent the individual tables of the database. That is, a one-to-one mapping exists between the tables in the database and classes in the data tier as illustrated by figure 5.2.

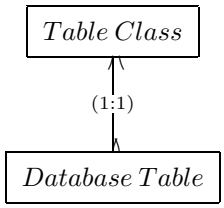


Figure 5.2: Illustration of the one-to-one mapping between a table class and a database table.

In addition to the table classes the data tier contains a number of common classes that support the table classes. These common classes are described in the subsections below.

The data tier is organized within the following namespace:

```
StudyPlanning.DAL
```

This namespace is divided up into a number of sub-namespaces. Confer appendix A for an overview of how the data tier is organized and brief descriptions of the individual classes in the tier.

The source code of the classes in the data tier can be found in chapter 1 in volume II.

5.4.1 Base Class

All the classes in the data tier that represent a table in the database inherit from a base class named `DbObject`. It is beneficial to use a base class seeing that it becomes very easy to add or change an aspect common to all of the table classes.

The `DbObject` class is fairly plain in that it simply creates a connection to the database. Hereby all subclasses have a database connection once instantiated.

5.4.2 Interaction with Stored Procedures

The table classes do not contain any inline SQL statements. We have chosen an approach where the table classes only operates on the database through stored procedures[10] as illustrated in figure 5.3.

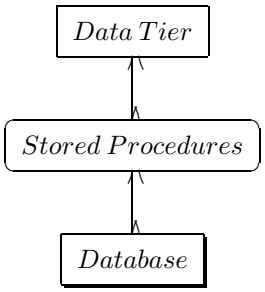


Figure 5.3: Illustration of the communication between data tier, stored procedures and the database.

One can speak of inserting a separate tier between the data tier and the database, however, we conceive the stored procedures as part of the database.

The reasons for the chosen approach are several[11]:

1. **Security:**
   (a) **Parameterized Queries**: Use of parameterized queries is enforced by the use of stored procedures. If one wants to invoke a stored procedure with some input it has to be done by means of input parameters. Similarly, as to output. By exclusively using parameterized queries the system is not vulnerable to security flaws like SQL and XPath injections. Use of parameterized queries thus complies with the security requirements posed in section 4.3.4.

   (b) **Fine-grained Authorization**: The majority of database systems, including SQL Server[12], provide security mechanisms that make it possible on a fine-grained level to restrict the use of a stored procedure. Use of stored procedures thus freely makes an extra level of security available.

2. **Reuse**: Two or more methods in a table class may have the need for the same query. Placing the query in a stored procedure allows for reuse of queries.

3. **Convenience**: T-SQL is an independent language and it seems convenient to have T-SQL statements placed in a self-contained media – the alternative being a cumbersome build-up of strings (holding the queries) in the table classes.

The drawback of the chosen approach is that not all database systems support stored procedures for which reason it may be more difficult to switch over to some database systems than others.

[10] A stored procedure is a set of SQL statements stored within the database in compiled form.
[11] Additional advantages of stored procedures is discussed at pages 255-257 and 352-355 in [16].
[12] http://msdn.microsoft.com/library/en-us/adminsql/ad_security_3whf.asp

In our implementation each individual stored procedure only operates on a single database table. The individual table class may interact with several stored procedures and several methods of a given table class may interact with the same stored procedure.

The T-SQL statements of the stored procedures can be found in chapter 2 in volume III.

5.4.3 Attribute Types

Every table class has public properties corresponding to the attributes of the table which the class represents.

These attributes are declared as one of the types listed in the following table:

Class Name	Description
`DalBool`	Models an attribute (of a database table) representing a boolean.
`DalDateTime`	Models an attribute (of a database table) representing date and time of day.
`DalFloat`	Models an attribute (of a database table) representing floating point data.
`DalGuid`	Models an attribute (of a database table) representing a globally unique identifier.
`DalInt`	Models an attribute (of a database table) representing an integer.
`DalString`	Models an attribute (of a database table) representing a string.
`DalStringLocalizable`	Models an attribute (of a database table) representing a localizable string. This class is described in more detail in section 5.4.4.

The types in the above table are referred to as "attribute types". We have been obliged to introduce the attribute types in the data tier due to the special `NULL` value that exists in databases. As is well known it is possible to assign a `NULL` value to an attribute of a database table in addition to assigning other values. A `NULL` value is assigned if the value is e.g. unknown or inapplicable[13]. The built-in types in the .NET Framework (e.g. `System.Int32` or `System.String`) can justly not have assigned `NULL` values to them and for this reason we have introduced the attributes types.

The attribute types are basically minor extensions of the built-in types – one of the extensions being the possibility of representing a `NULL` value. All the attribute types inherit from a base class named `DalType` as shown in figure 5.4. The `DalType` class contains a few properties and validation methods common to the attribute types.

5.4.4 The `DalStringLocalizable` Class

In section 5.2.5 we described how we have chosen to store semi-structured data in those attributes that need to be culture versioned i.e. localized.

[13] Interpretations of `NULL` values can be found at pages 239-240 in [27].

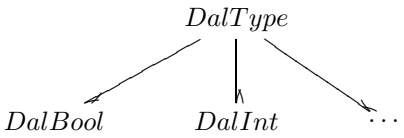


Figure 5.4: Illustration of how the various attribute types inherit from the `DalType` class.

The `DalStringLocalizable` class represents a text attribute that can be localized. The class is modelled as a hashtable – also known as a collection of key-and-value pairs. The key is the ID of a certain culture and the value is the text in that culture.

A localizable attribute may for instance be the name of a course:

```
DalStringLocalizable CourseName = new DalStringLocalizable(false);
```

, the `false` parameter indicating that the attribute may not be `NULL`. The `Add` method is used in order to localize the course name into a new culture:

```
Name.Add("da-DK", "Afløbssystemer");
Name.Add("en-GB", "Urban drainage systems");
```

In order to get the English name of a course one uses the following syntax:

```
string message;
message = "You have just signed up for course ";
message += Name["en-GB"];
```

Similarly, for other cultures.

5.4.5 The `DataLog` Class

As described in section 5.2.4 the attributes `Created`, `CreatedBy`, `Updated` and `UpdatedBy` occur on every single database table in the system on account of change tracking.

The `DataLog` class has been introduced as an obvious representation of the four attributes. Accordingly, each table class has a public `Log` property which is declared as a `DataLog` object.

The class has public properties corresponding to the afore-mentioned four attributes besides which a number of methods are offered in order to make it convenient and efficient for developers of table classes to work with the change tracking attributes.

5.4.6 Basic Methods

The various table classes most often implement one or more of the basic methods described in the table below.

Method Name	Description
`public void Create()`	Creates a new row in the current table based on the values of the properties of the current object.
`public void Retrieve()`	Retrieves a row from the current table using the value of property that represents the primary key of the table.
`public bool Update(...)`	Updates a row in the current table using the values of properties of the current object.
`public bool Delete()`	Deletes a row in the current table using the value of the property that represents the primary key of the table.

The listed methods correspond to the four basic statements in SQL as follows: `INSERT`, `SELECT`, `UPDATE` and `DELETE`. Invocation of one of the methods essentially leads to the execution of the corresponding SQL statement.

A great many of the table classes implement all four methods. Table classes representing a type table in most cases merely implement the `Retrieve` method. Naturally, the table classes implement other methods than the basic ones – e.g. several table classes implement methods that retrieve lists based on different criteria.

5.5 Business Tier

The purpose of the business tier is to manage how the system should function. The business tier interacts massively with the data tier and can be thought of as a servant to the presentation tier.

The most essential classes in the business tier are briefly described in the table below:

Class Name	Description
`Course`	Represents a course.
`Project`	Represents a project.
`Student`	Represents a student.
`StudyPlan`	Represents a study plan.
`StudyPlanCriterion`	Represents a study plan criterion i.e. an assembly of specifications on the basis of which the system should elaborate a study plan.
`StudyPlanElaborator`	Elaborates a study plan on the basis of a specified study plan criterion.
`CourseGrabber`	Grabs the courses in the "DTU Course Catalogue" and stores the courses in the database.

So perhaps most interestingly the business tier contains the algorithm that elaborates a study plan. Furthermore, the tier contains classes that:

- represent technical packages, technical lines, technical fields and specializations.

- handle security.
- provide localized text groups to the user interface.

The various parts of the business tier are described in chapter 6. The business tier is organized within the following namespace:

```
StudyPlanning.Biz
```

This namespace is again divided up into a number of sub-namespaces. Confer appendix B for an overview of how the business tier is organized and brief descriptions of the individual classes in the tier.

The source code of the classes in the business tier can be found in chapter 3 in volume II.

5.5.1 Base Class

Every class in the business tier inherits from a base class named `BizObject`. In the current version of the system the `BizObject` actually does not contain any properties or methods and does as such not add anything to its subclasses. Nevertheless, the `BizObject` has been introduced in order to easily support prospective common needs in the business tier.

5.6 Presentation Tier

The purpose of the presentation tier is to provide a user interface such that users can make use of the system. The presentation tier acts upon the user's actions and passes on user input to the business tier.

The presentation tier is implemented using ASP.NET i.e. the part of the .NET Framework for building web applications. Basically, the presentation tier is composed of a number of dynamic web pages – in this context so-called ASPX pages. The presentation code may be organized in many ways, however, ASP.NET supports a concept referred to as "code separation" and we have chosen that approach. Using the "code separation" approach, imperative code (e.g. C#) is conveniently separated from declarative code (e.g. HTML and XML) – in other words presentation code is separated from layout and design.

The individual ASPX page contains the HTML code, server controls and the like. In addition, the ASPX page inherits from a class in the so-called "code-behind" file associated to the ASPX page as illustrated in figure 5.5. The class in the "code-behind" file handles the events raised by the user on the ASPX page and manages the interaction with the business tier.

Using the "code separation" approach one can speak of inserting a separate tier between the presentation tier and the business tier, however, we conceive the "code-behind" classes as part of the presentation tier.

Several benefits apply to using code separation[14], the two most striking in our opinion being:

- **Layout Easier to Modify** – the layout and design can to a great extent be changed without the need of recompiling the presentation logic.
- **Division of Labour** – web designers can focus on the look and feel and developers can concentrate on the presentation logic. A favourable side effect moreover being that the presentation tier can be implemented in shorter time.

[14] Benefits of code separation are discussed e.g. at pages 305-306 in [16].

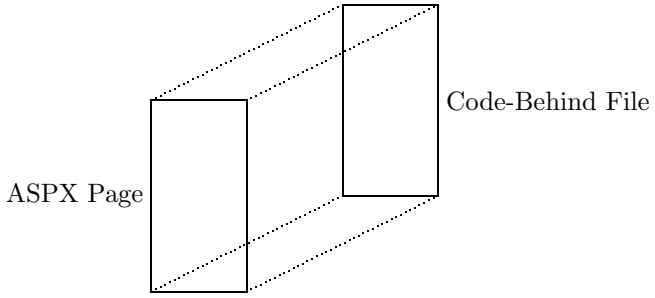


Figure 5.5: The individual ASPX page inherits from a class in its "code-behind" file.

The presentation tier is organized within the following namespace:

```
StudyPlanning.UI
```

Confer appendix C for an overview of the how the namespace is organized.

The source code of the presentation tier can be found in chapter 5 in volume II. The implemented user interface is described in chapter 8.

5.7 Error Handling

An important issue is how to handle errors. There is always the risk that errors may arise even though an application has been thoroughly tested and a profound handling of errors is therefore crucial.

Errors may arise in any of the application tiers. Chiefly, two types of errors can arise in the data tier:

Class Name	Description
`SqlException`	Exception that is thrown when the SQL Server returns a warning or an error.
`XmlException`	Exception that is thrown when an error occurs parsing an XML document.

We have introduced the `DalException` class as a common container for the exceptions that may arise in the data tier. Hereby, the business tier only has to deal with this single exception. In order to make tests on the different error types more convenient we operate with the following error codes in the `DalException` class:

Error Code	Description
4	**Validation error on a property**. For instance the value of an attribute, that is not allowed to be `NULL`, has not been set.
8	**Row does not exist**. For instance non-existing row has been attempted to be retrieved, updated or deleted.
12	**Row already exists**. For instance a row with an already existing primary key has been attempted to be created and the primary key constraint has thereby been violated.
16	**A severe error has occurred**. For instance the SQL server may be unreachable or an error has arisen parsing an XML document.

The error codes are arbitrarily chosen. The `DalException` class may be instantiated specifying either an `SqlException` or an `XmlException`.Recipients of a `DalException` may (in addition to the returned error code) get the exact error message by looking at the `Message` property of the thrown `DalException` object.

The recipients of thrown exceptions in the data tier are the classes in the business tier. The business tier may be able to cope with caught errors, however, some errors can not be handled and in such cases an exception must be thrown to the presentation tier. We have introduced the `BizException` class as a common container for unmanageable errors arisen in the business tier. A `BizException` is supposed to be thrown only if an error is severe and unmanageable. Finally, the presentation tier is to catch exceptions thrown by the business tier, provide a useful message to the user and possibly guide the user in what to do.

As a sum up we have introduced the following exceptions classes in the study planning application:

Exception Name	Description
`DalException`	**Represents an exception in the data tier**. When an error occurs in the data tier a `DalException` is thrown to the business tier. The business tier may be able to handle the error and proceed, however, in case of a severe error arisen in the data tier, the business tier is not able to continue and an error must be given to the user.
`BizException`	**Represents an exception in the business tier**. When an error occurs in the business tier a `BizException` is thrown to the presentation tier. A `BizException` is supposed to be thrown only if an error is severe and unmanageable and the current part of business tier is in no way able to continue. The presentation tier is based on a caught `BizException` responsible of presenting a useful message to the user.

Chapter 6

System Modules

In sections 5.3-5.6 we took a horizontal look by describing the data tier, business tier and presentation tier, respectively, in a general manner. In this chapter we will take a vertical look at the system.

We do so by describing select parts of the system referred to as system modules. As illustrated by figure 6.1, a system module is to be conceived as some entity or functionality transverse to the business and data tier possibly including a number of database tables.

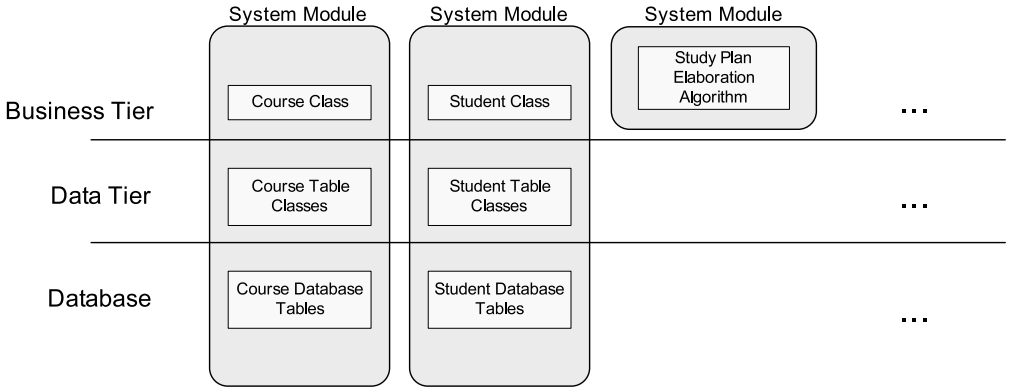


Figure 6.1: Illustration of how a system module is to be conceived.

In the description of the different system modules we will describe select parts of the database design and select parts of the business tier. We will refrain from describing the data tier and the stored procedures for each of the system modules for the following reasons:

1) The data tier and the stored procedures work more or less alike for all the different system modules.
2) Section 5.4 should have given adequate insight in the data tier and how it interacts with the database through stored procedures.

We will cover the following system modules (section number in brackets): “Courses” (6.1), “Projects” (6.2), “Students” (6.3), “Study Plans” (6.4), “Study Plan Criteria” (6.5), “Study Plan Elaboration” (6.6) and “Users and Security” (6.7).

6.1 Courses

The greater part of a student's study consists of taking courses – attend lectures, hand in assignments, take examination *&c.* For that reason courses form a weighty part of a study plan and are thus important with regard to study planning.

Courses are composed of much information and are cumbersome to deal with:

- courses are only taught in certain periods.
- teaching in courses is timetabled in certain modules.
- courses may be taught in several parts.
- courses may have complex interdependencies with other courses.
- *&c.*

A course is represented by the `Course` class in the business tier. The chief purpose of the methods in this class is to support the study plan elaboration algorithm.

Overview of this section:

- In subsection 6.1.1 select parts of the database design for courses are described.
- In subsection 6.1.2 a selection of methods provided by the `Course` class are described.

6.1.1 Database Design

In the table below some select database tables used to store course data are presented. Confer appendix E.3 for diagrams of all the database tables used for storing course information.

Table Name	Description
`CourseVersion`	This is the main table for storing course information, including e.g. course number and objective description.
`Course_Lecturer`	Stores the lecturers who are responsible for and teach the courses. The table maps the primary keys of the `CourseVerion` and the `Lecturer` tables.
`Course_Keyword`	Stores the keywords associated with courses by mapping the primary keys of the `CourseVersion` and `Keyword` tables.
`Course_Period`	Contains the periods in which course parts are taught.
`Course_Point`	Stores the credit points of course parts.
`Course_RecommendedPlacement`	This table contain the recommended placement for all courses for distinct study types.

Course Interdepencies

Recall from subsection 4.1.1 that course interdependencies must be specifiable in the form shown in figure 3.11.

The interdependencies between courses are specified by using the two tables `Course_RelationCourse` and `Course_RelationCourseItem` shown in the table below:

Table Name	Description
`Course_RelationCourse`	Each record in this table represents a disjunction list.
`Course_RelationCourseItem`	Entries in this table represent the items in the disjunction lists.

A course may have multiple entries in the `Course_RelationCourse` table meaning that it has a conjunction list of disjunctions lists since there is implicit conjunction between entries in the `Course_RelationCourse` table as illustrated in figure 6.2.

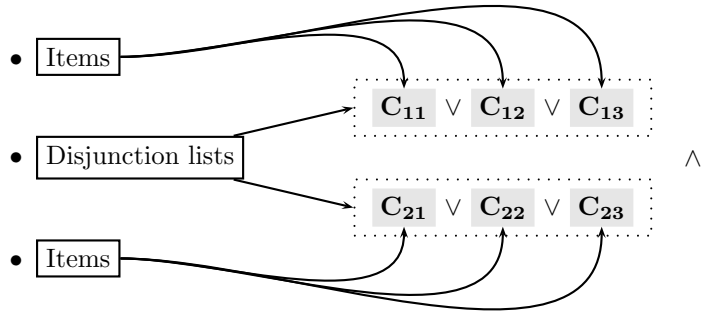


Figure 6.2: Illustration of the representation of course interdependencies.

6.1.2 Select Methods

In this section a selection of methods provided by the `Course` class are presented.

```
public static Guid[][] GetPointBlockingCourses(Guid courseVersion_ID)
```

Retrieves a list of point blocking courses represented by Course_IDs for a given CourseVersion_ID.

```
public static Guid[][] GetPrerequisites(
    Guid courseVersion_ID,
    PrerequisiteCourseType type)
```

Gets a list of prerequisite courses of the specified prerequisite type for the specified course version.

```
public static float GetWorkload(
    Guid courseVersion_ID,
    int part)
```

Retrieves the workload for a course part.

```
public static float[] GetRecommendedPlacement(
    Guid courseVersion_ID,
    int studyType_ID)
```

Retrieves the recommended placement for a course and studytype, expressed as a lower and upper limit of points. Confer sections 6.1.3 and 6.1.4.

```
public static Guid[][] GetKeywordMatchingCourses(
    string[] keywords,
    string culture_ID)
```

Retrieves a list of the course versions which match the specified keywords in the specified culture. The result is sorted such that courses matching the most keywords appear first in the resulting array.

In the following a selection of the methods generating chains of course prerequisites are presented:

```
public static Guid[][] GetPrerequisiteCourseCombinations(
    Guid courseVersion_ID,
    PrerequisiteCourseType type,
    bool chain)
```

Generates a list of prerequisite courses for a given course. The type of prerequisite is specified by the `type` parameter. The `chain` parameter specifies the depth of the prerequisite chains generated. If `TRUE` the complete chains will be generated such that the prerequisites of a prerequisite course are recursively included. If `FALSE` only the first level of prerequisites is generated.

```
public static Guid[][] SortCombinationsByComplexity(
    Hashtable passedGoingPlannedCourses,
    Guid[][] prerequisiteCombinations)
```

This method sorts the course chains by ascending complexity such that the shortest chains appear first in the result. The `passedGoingPlannedCourses` parameter is a hashtable containing all the courses which the student has passed at this point either for real or as estimated by the study plan elaborator. The method then removes the passed courses from the dependency chains and sorts the result.

6.1.3 Measurable Recommended Placements

To perform an automated study planning which takes the recommended placement of courses into consideration the recommended placements must be translated into a measurable quantity – such as credit points – for each study type. This can be achieved by associating the recommended placement with a point interval for each study type as shown in the example below:

Study Type	**Recommended Placement**	**Point Min**	**Point Max**
Bachelor of Science	4th to 5th semester	90	150
Master of Science	3rd to 4th semester	60	120

The example shows that before taking the course Bachelor of Science students should ideally have collected between 90 and 150 credit points whereas Master of Science students need to gather between 60 and 120 credit points to follow the placement recommendation of the course. Using credit points instead of semester numbers also has the advantage that the recommendation works even if a student has been delayed in his study which is not the case when using semester numbers.

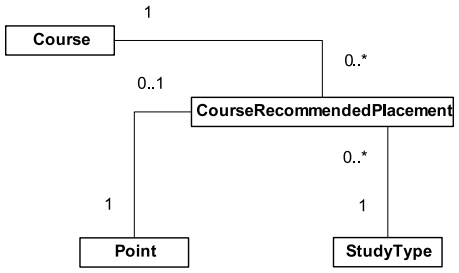


Figure 6.3: Logical model of the `Course` class and its associations to the `CourseRecommendedPlacement`, `StudyType` and `Point` classes.

Figure 6.3 shows a part of the logical model from figure 3.14 which has been extended to include the translation into point intervals. A course may have one or more recommended placements – one for each study type. Each `CourseRecommendPlacement` object is linked to a `Point` object which represents the recommended point interval for the given `StudyType` and `Course` objects.

The measurable recommended placement of courses is stored in the following database table:

Table Name	Description
`Course_RecommendedPlacement`	Contains the placement recommendation for all courses for each study type to which the course applies. The placement is specified by a `Point_ID` which refers to a point interval in the `Point` table. Alternatively, the placement is specified as a `RecommendedPlacementConcept_ID` (confer section 6.1.4).

6.1.4 RecommendedPlacementConcept

As mentioned in section 3.3.5 on page 20 the placement recommendation for a course may also be specified as a concept. The concept must be translated into a measurable quantity in order to perform an automated study planning.

The `RecommendedPlacementConceptStudyType` class is used to translate the recommended placement concepts into credit points (figure 6.4).

The translation depends on the study type as shown in the example below:

Study Type	Recommended Placement Concept	Point Min	Point Max
Bachelor of Science	At the end of the study	140	210
Master of Science	At the end of the study	200	300

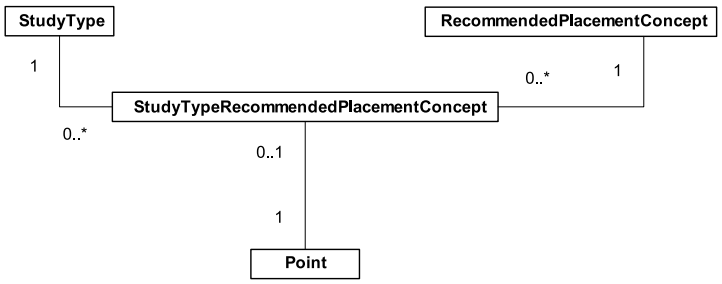


Figure 6.4: Logical model of the association between the `StudyType`, `RecommendedPlacementConcept`, `StudyTypeRecommendedPlacementConcept` and `Point` classes.

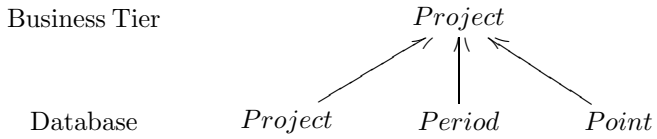
The recommended placement concepts associated with courses are stored in the following database tables:

Table Name	Description
`Course_RecommendedPlacement`	Contains the placement recommendation for all courses for each study type to which the course applies. If the placement is specified as a `Point_ID` the placement has been specifically specified (confer section 6.1.3). The placement may also be expressed as a `RecommendedPlacementConcept_ID` which refers to an entry in the `RecommendedPlacementConcept` table.
`RecommendedPlacementConcept`	Maps the recommended placement concepts into `Point_IDs` which refer to the `Point` table containing a point interval describing the placement recommendation.

6.2 Projects

As mentioned in section 3.4 a project may be e.g. a course with no timetabled teaching, the writing of a paper based on an assignment or a trainee job at a company.

A project is represented by the `Project` class in the business tier. In the database information concerning a project is stored in three different tables as illustrated in the following figure:



The `Project` class contains attributes which are distributed among the three entities in the database – `Project`, `Period` and `Point` as shown above. The `Project` class has the following attributes:

- Name and number of the project

- Type of the project
- Assessment type of the project
- Start date and end date of the project period
- Type of the project period
- Name of the project period
- Points assigned for the project

6.2.1 Database Design

Projects are stored in the following two database tables. The complete database diagrams for projects may be found in appendix E.14.

Table Name	Description
`Project`	The primary table for storing information concerning projects e.g. name of the project and project number.
`Project_Type`	Contains the possible project types in the system.

6.2.2 Methods

The `Project` class provides the following methods which may be performed on an instance of the class. The methods must primarily support the study plan elaborator:

public Guid Create()

Creates a new project by inserting data into the `Project`, `Period` and `Point` tables using the values of the current instance and returns the identification of the newly inserted row in the `Project` table.

public void Retrieve()

Retrieves information from the `Project`, `Period` and `Point` tables using the value of the `Project_ID` of the current instance. Data is also retrieved from the `ProjectType`, `PeriodType` and `AssessmentType` tables to get the names of all the types.

public float GetWorkload(Guid period_ID)

Gets the workload of the project represented by the current instance in a specific common period such as a 13-week period.

The `GetWorkload` method which retrieves the estimated workload of a project in a given period calculates the workload based on the workload intervals defined in the `StudyPlanCriterion_ProjectWorkload` table. The workload is calculated according to the following formula:

$$\sum_{i=1}^{n} \frac{o_i}{t_i} \cdot w_i$$

The variables are defined as follows:

n is the total number of workload intervals specified.

o is the number of overlapping days in the interval.

t is the total number of days in the interval.

w is the total workload specified for the interval.

Project				
w_1		w_2		w_3
a	b	c	d	e

In the example shown above the workload of a project has been specified by a student in three intervals of w_1, w_2 and w_3 points (confer also figure 4.8). The project is overlapped by a common period in sections b and c. Calculating the workload of the project in the common period is the performed as follows:

$$workload = \frac{|b|}{|a| + |b|} \cdot w_1 + \frac{|c|}{|c| + |d|} \cdot w_2 + \frac{0}{|e|} \cdot w_3$$

6.3 Students

A student is represented by the `Student` class in the business tier.

6.3.1 Database Design

In the table below some select database tables for storing information related to students are shown. The complete database diagrams containing all the database tables related to students may be found in appendix E.18.

Table Name	**Description**
`StudentVersion`	The main table for storing information about students such as study number and study type.
`Student_Course`	Contains the course parts which the student has signed up for.
`Student_Project`	Stores the projects for which the student has signed up.
`Student_StudyPlan`	This table contains the student's study plans.
`Student_StudyPlanCriterion`	Stores the student's study plans criteria.

6.3.2 Select Methods

The `Student` class provides the following methods which are used by the study plan elaboration function to determine which courses to schedule for the student:

```
public static float GetTotalPoints(Guid studentVersion_ID)
```

Retrieves the total sum of credit points which the student has obtained so far.

```
public static bool HasPassedCourse(
    Guid studentVersion_ID,
    Guid course_ID)
```

Determines whether a given student has passed a given course.

```
public static Guid[] GetPassedCourses(Guid studentVersion_ID)
```

Gets a list of the courses which the student has passed – i.e. courses where all course parts have been signed up for and passed.

```
public static CoursePart[] GetCourseParts(Guid studentVersion_ID)
```

For a given student this function retrieves the course parts for which the student has signed up – regardless of whether the student has passed the courses yet. This function provides the basis for the elaboration of a study plan.

6.4 Study Plans

As mentioned in section 3.9 a study plan is a list of periods, each containing the courses and projects which the student should sign up for. A study plan is represented by the `StudyPlan` and `StudyPlanPeriod` classes in the business tier.

6.4.1 Database Design

Elaboration of a study plan criterion results in a study plan which is stored in the database tables shown below. The complete database diagram for study plans may be found in appendix E.19.

Table Name	Description
`StudyPlan`	This is the main table for storing study plans.
`StudyPlan_Period`	Contains the periods in a study plan. A study plan may have arbitrarily many study plan periods. Since a project has its own individual period a study plan period may contain either a project or a number of course parts stored in the `StudyPlan_PeriodCourse` table.
`StudyPlan_PeriodCourse`	Stores the course parts which have been scheduled in a study plan period in a study plan.

6.4.2 Select Methods

The `StudyPlan` class provides the following methods:

public static DataTable GetStudyPlans(Guid student_ID)

Retrieves a list of a student's study plans and returns them in a DataTable object.

public static bool Delete(Guid studyPlan_ID)

Deletes the study plan identified by the supplied studyPlan_ID.

The `StudyPlanPeriod` class provides the following methods which all operate on the current instance. These methods are used by the study plan elaborator:

public void AddCoursePart(CoursePart cp)

Adds the specified course part to the current instance. The remaining workload is decremented and the available modules of study plan period are then updated to reflect the addition of the course part.

public void RemoveCoursePart(CoursePart cp)

Removes the specified course part from the current instance. Afterwards the remaining workload is adjusted and the modules occupied by the course part are added to the list of available modules to reflect the removal of this course part.

public bool ContainsCoursePart(CoursePart cp)

Determines whether the current instance contains the specified course part.

public CoursePart GetCoursePart(int index)

Retrieves the course part with the specified index from the list of course parts of the current instance. If no course part with the specified index exists a null value is returned.

6.5 Study Plan Criteria

A study plan criterion describes some requirements as to how a study plan should be elaborated as mentioned in section 4.1.7. In the business tier the study plan criterion is represented by the `StudyPlanCriterion` class.

6.5.1 Database Design

A study plan criterion consists of a set of database tables as shown in appendix E.20 which hold the student's preferences as to how a study plan should be elaborated. Some select tables are shown below:

Table Name	Description
`StudyPlanCriterion`	This is the main table for storing study plan criteria.
`StudyPlanCriterion_Keyword`	Contains the keywords selected by the student to match his interests.
`StudyPlanCriterion_Language`	Stores the languages which in which the student wants to be taught.
`StudyPlanCriterion_Lecturer`	This table contains the lecturers which the student does not want to be taught by.
`StudyPlanCriterion_Course`	Contains the courses which the student has either specifically selected or deselected.
`StudyPlanCriterion_ProjectWorkload`	Stores the workload of projects in user specified intervals. By default the workload is distributed evenly over the whole project period.

6.5.2 Select Methods

The data stored in the tables is retrieved through the data tier and manipulated by means of the following methods used exclusively by the study plan elaboration function:

```
public static int[] GetLanguages(Guid studyPlanCriterion_ID)
```

Retrieves a list of the languages which have been selected in a study plan criterion.

```
public static int[] GetLecturers(Guid studyPlanCriterion_ID)
```

Fetches a list of the lecturers which have been deselected in a study plan criterion.

```
public static int[] GetStudyTypes(Guid studyPlanCriterion_ID)
```

Gets the list of study types which the student has selected in a study plan criterion.

```
public static Guid[] GetKeywords(Guid studyPlanCriterion_ID)
```

Given a study plan criterion this method retrieves a list of the keywords which have been selected by the student.

```
public static float[] GetWorkload(
    Guid studyPlanCriterion_ID,
    Guid period_ID)
```

For a given study plan criterion and a period this method returns the desired workload expressed as an upper and a lower bound of credit points. If no specific workload demands have been set up for the period in question the general setting derived from the type of the specified period is used.

```
public static int[] GetModules(
    Guid studyPlanCriterion_ID,
    Guid period_ID)
```

Retrieves the available timetable modules for a given study plan criterion and period. If no timetable modules have been deselected in the specified period the universal setting deduced from the period type is used.

```
public static Guid[] GetSelectedCourses(
    Guid studyPlanCriterion_ID,
    [Guid period_ID])
```

For a given study plan criterion this function gets the courses which have been specifically selected. If a period is specified as well, only courses which have been selected in that particular period are returned.

```
public static Guid[] GetDeSelectedCourses(
    Guid studyPlanCriterion_ID,
    [Guid period_ID])
```

Retrieves the courses which have been deselected in a given study plan criterion. If a period is also specified, only courses which have been deselected in that specific period are returned.

```
public static Project[] GetProjects(
    Guid studyPlanCriterion_ID)
```

Retrieves a list of the projects which must be scheduled for this study plan criterion.

The methods `GetWorkload` and `GetModules` are a little special in that they will return either a specific or a general selection of workload or modules. For instance when calling the `GetWorkload` function supplying a study plan criterion and a period the function will first try to fetch a specific workload selection for the supplied period from the `StudyPlanCriterion_WorkloadPeriod` table (appendix E.20.3). If no such selection exists the period type is derived from the period by looking it up in the `Period` table (appendix E.11) and the general setting, which is created by default in a study plan criterion, for the period type is retrieved from the `StudyPlanCriterion_WorkloadPeriodType` table (appendix E.20.4).

The `GetProjects` function fetches the required projects from the `StudyPlanCriterion_ProjectWorkload` table (figure E.20.9). If the project originates from the technical package, the `TechnicalPackage_Project` table (figure E.24.4) is read in order to get details about placement of the project and its duration. If the project is determined by the study type the `StudyType_ProjectType` table (figure E.21.3) is read to get the details concerning the project. Using this information a `Project` object (section 6.2) is created for each project.

6.6 Study Plan Elaboration

The elaboration of a study plan is based on a Study Plan Criterion (SPC) (section 4.1.7) which describes the criteria for which to create a study plan. The elaboration of a study plan starts by retrieving the SPC and information about the student for which to elaborate a study plan.

In the SPC the student has the possibility of specifying a technical package (section 3.6) different from the one he is currently signed up for. If no technical package has been selected in the SPC, the student's current technical package is used. Additionally the student may select a technical line (section 3.7) in the SPC otherwise the technical line for which the student is signed up – if any – is used.

In the SPC some criteria are enforced generally for the whole study plan and some are specific for each period in the study plan. The general criteria are:

- Specifically selected or deselected courses.
- Selected language – at least one language must be selected.
- Deselected lecturers.
- Selected study types – at least one study type must be selected.
- Selected keywords.

Courses and projects which the student has already finished or signed up for previously are retrieved. According to the selected technical package and study type (section 3.8) the required projects are retrieved as well as the fundamental courses for the technical package.

Next step is to find the first possible period (section 3.14) in which to commence the scheduling process. The first possible period is the next period which has a start date greater than the current date.

Now the scheduling can begin. The following items are retrieved for each period:

- Mandatory courses for the technical package for this period.
- Mandatory courses for the technical package for previous periods which have not yet been passed.
- Optional courses for the technical package for this period.
- Courses which have been specifically selected/deselected in the given period.
- The desired workload and timetable modules for the period.

The courses for the period are scheduled by traversing the list of courses according to the priority rules in table 6.1 where 1 is the highest priority.

Priority	**Description**
1	Courses which have been specifically selected for the given period.
2	Mandatory courses for the technical package in the given period.
3	Mandatory courses for the technical package for previous periods.
4	Fundamental courses for the technical package.
5	Courses which have been specifically selected.
6	Prerequisite courses for a technical line where applicable.
7	Courses for a selected technical field where applicable.
8	Point giving courses for a specialization where applicable.
9	Supplementary courses for a specialization where applicable.
10	Optional courses for the technical package in the given period.
11	Keyword matching courses.

Table 6.1: Priorities according to which courses are selected.

Steps 6 through 9 of table 6.1 are bypassed until the technical package has been finished.

The priorities in table 6.1 have been chosen such that all mandatory courses are scheduled first as they must inevitably be passed and because they teach some fundamental skills which may be required by other courses. Next, courses which have been specifically selected by the student are scheduled. If a technical line has been selected the prerequisite courses for the technical line must have higher priority than the technical line itself. For specializations the point giving courses have higher priority than the supplementary courses. Courses matching one or more keywords selected by the student have the lowest priority in this context because the matching keyword(s) may describe only a small part of the course.

For each course the user specified constraints below are checked:

1. Is the course taught in one of the selected languages?
2. Is the course taught by a deselected lecturer?
3. Has the course been generally deselected?
4. Has the course been deselected in this period?
5. Is the recommended placement of the course satisfied?
6. Does the study type of the course match one of the selected study types?

However, none of the above-mentioned constraints except #4 are checked for mandatory courses in the technical package as well as fundamental courses since these courses must be scheduled no matter what.

If the constraints are satisfied the algorithm proceeds to course part scheduling where the individual parts of a course are scheduled. If part one of the course is taught in the period currently being scheduled and the timetable modules in which the course part is taught are available as well as the workload of the course part being less than the remaining workload of the period, the course part is scheduled. When a course part has been scheduled the available timetable modules and the remaining workload of the period must be updated.

If a course consists of multiple parts the first part may be placed in any period where the part is taught. However, as soon as part one has been placed the following parts must be scheduled in sequential order in the first possible periods hereafter as mentioned in section 3.3.1 on page 16. If for some reason a part cannot be scheduled according to this rule all parts of the course must be unscheduled and whole course must be moved to a new period if allowed.

6.6.1 Rescheduling

Whenever a course part has been unscheduled from a period it leaves an open space in the schedule and thus the period is not exploited optimally. To prevent this the period is rescheduled meaning that all possible course parts for the period are traversed again according to their priority until a suitable course part – if any – which will fill out the space has been found.

Prior to each rescheduling the state of the period is saved – i.e. the currently scheduled course parts for the period. This is done to avoid cyclic rescheduling where a course part is unscheduled (U in figure 6.5) from a period and hence a rescheduling (R in figure 6.5) is performed which schedules (S in figure 6.5) the same course part again which in turn is unscheduled again and so on. So before performing the actual rescheduling the current state is compared to all the previously stored states of the period. If a match is found the rescheduling is cancelled due to a cycle, otherwise the state is saved and the rescheduling is performed.

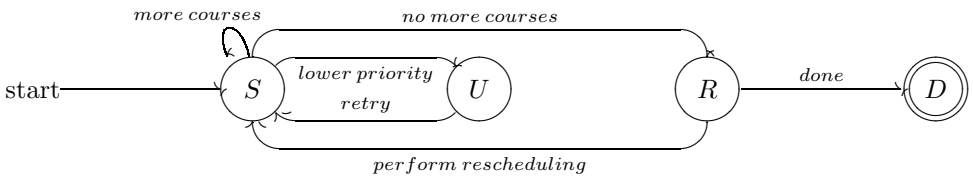


Figure 6.5: Illustration of a possible cyclic rescheduling.

6.6.2 Scheduling Course Chains

Some courses have dependencies to other courses (section 3.3.4, page 18) which in turn must be scheduled first if the student has specified so in the SPC. However, the prerequisite courses may also have dependencies to other courses thus introducing a chain of dependencies. When a course is to be scheduled its dependency chains are generated using a recursive function. Courses which have already been passed or signed up for in an earlier period are removed from the chains and the result is sorted by ascending complexity such that course chain containing the fewest courses appears at the beginning. The study plan elaborator then tries to schedule the courses in the dependency chain. If it fails it will retry using one of the other chains. If scheduling of all chains fail the course will not be scheduled if the chains contain mandatory prerequisites.

Scheduling Projects

Alongside the courses some projects must also be scheduled (sections 3.4 and 3.8). The required projects are stored in the SPC. Whenever a study plan period has been completed the elaborator algorithm determines whether a project should be scheduled at this point by comparing the accumulated number of credit points with the placement recommendation(s) of the project(s). If the project must be scheduled it is placed in its own individual study plan period which may overlap other study plan periods. The workload of the project in the overlapped periods is calculated from the specifications in the SPC (section 4.1.7 page 47).

6.6.3 Description of Flow

The diagram on page 97 shows how one period is scheduled as well as courses spanning multiple periods. Scheduling of projects or chains of prerequisite courses is not shown in the diagram. In the following the circled numbers like (1) refer to the correspondingly numbered transitions in the diagram.

The flow starts at (1) by fetching a course according to the priority rules in table 6.1. If the course has been generally deselected (2) in the SPC the flow returns to (A).

Next the language in which the course is taught is retrieved. If the course language is not among the languages selected in the SPC (4) the flow returns to (A) otherwise the lecturers teaching the course are fetched. If the course is taught by at least one lecturer who has been deselected in the SPC (6) the flow returns to (A) otherwise the recommended placement of the course if retrieved along with the estimated number of credit points for the period being

scheduled. If the recommended placement of the course is satisfied (9) the study types of the course are fetched else the algorithm returns to (A).

Provided that the course has one of the study types selected in the SPC (11) the course is checked for deselection in the given period. If the course has been deselected in the current period (13) it checked whether to allow moving the course to another period. If the course has not been deselected in the current period (12) part 1 of the course is retrieved.

If part 1 is taught in the current period the possible modules for the course part are fetched (15) otherwise the algorithm checks if it is permissible to move the course (14). If moving is allowed (17) the next period is retrieved and a cycle is performed using the transitions (12), (13) and (14) until a period in which the course is taught and is not deselected has been found (15). If no such period is found (19) the course part cannot be scheduled and the flow returns to (A).

If the modules in which the course part is taught are available (27) the workload of the course part is fetched. However, if the modules are not available (26) the algorithm checks whether any course parts with lower priority (confer table 6.1) have been scheduled in the given period. If this is the case (25) these course parts are unscheduled and the module check is repeated.

If no parts with lower priority have been scheduled or the unscheduling did not free the desired timetable modules the course part cannot be scheduled and the flow moves on (24) to checking whether the course part is greater than one. Since all course parts greater than one must be scheduled in the first periods in which they are taught all previous parts must be unscheduled (23) if the part is greater than one.

Unscheduling is done by updating the remaining workload for each period where a previous course part has been scheduled. The modules of the course part are added to the set of available modules and a rescheduling of the period is ordered since the removed course part now leaves behind a gap in the schedule. If moving the course is allowed (20) a new period is fetched and the algorithm then tries scheduling the course again in the new period starting from part one.

If the modules were available (27) the workload of the course part is retrieved along with the remaining workload of the current period. If the workload of the course part exceeds the remaining workload the course part cannot be scheduled (28) and the flow proceeds as if the modules were unavailable. If the remaining workload of the period allows the addition of the course part (29) the course part is scheduled, its modules are removed from the set of available modules and the remaining workload of the period is updated.

If there are more course parts (32) these must be scheduled as soon as possible. The next course part is fetched and and loop is performed (30) (31) until the first period has been found in which the course part is taught. Should there be no more course parts (33) the algorithm determines if the the desired workload for the period has been reached. If this is the case (34) scheduling the period has been finished otherwise (35) it is determined whether more courses are available in the period. If no more courses are available the algorithm returns (36) otherwise (37) the next course is fetched and the flow starts all over again.

6.7 Users and Security

The system is primarily intended to be used by students, however, the system has been designed in a way such that persons of all types (e.g. also lecturers) can be set up as a user in the system. A user is represented by the `User` class in the business tier.

A number of requirements were posed to the security of the system in the requirements specification:

- Admission to the system must be restricted.
- A user may only obtain access to those parts of system which the user has been authorized access to.
- The system must not be vulnerable to brute force attacks.
- A strict password policy should be managed by the system.

The posed security requirements are flexible in that no requirements are made with respect to data and no preferred approach of complying with the requirements has been stated.

We have chosen to implement a role-based security system making it possible to manage system policy in a general manner[1]. The security system operates with the following notions and relations:

1) A user has one or more roles.
2) A role has one or more permissions.

---

3) A user has one or more permissions.

Item 3) is a consequence of 1) and 2). The security system does not support assignment of a permission to the individual user. Both a role and a permission may give admission to various parts of the system. Usually, a permission allows for carrying out a specific action while a role gives a more general admission.

In the current version of the system a "Student" role and a convenient number of permissions have been created. However, the system could easily be configured also to have e.g. a "Lecturer" role and other permissions.

In the subsections below different parts of the security system are described, including authentication of users (section 6.7.2) and authorization (section 6.7.3).

6.7.1 Database Design

In the below table we provide a general view of the most important database tables used to store data by the user and security system. Confer appendix E.16 and E.25 for diagrams of the described tables.

[1] We have been inspired by [6], chapter 5 in [3] and not least [2].

Table Name	Description
`SecurityPermission`	Contains the permissions of the system.
`SecurityRole`	Contains the roles of the system.
`SecurityRole_SecurityPermission`	Stores a permission's membership of a role. The table maps the primary keys of the `SecurityPermission` and `SecurityRole` tables.
`User`	The main table for storing user information, including username and password.
`User_SecurityRole`	Stores a user's membership of a role. The table maps the primary keys of the `User` and `SecurityRole` tables.
`User_Login`	Stores rejected and approved authentication of a user. The data is used to prevent brute force attacks.
`User_PasswordHistory`	Stores a user's previous passwords. The data is used to enforce password policy.

6.7.2 Authentication

Authentication of users is a rather complex matter. The authentication module has to prevent brute force attacks, enforce users to renew password if it has expired *&c.* In order to gain insight into the many security checks made as part of the authentication process we refer to the authentication decision diagram in appendix D.

When a user attempts to log into the system by specifying a username and password, the presentation tier invokes the `Authenticate` method in the `User` class of the business tier:

```
public static AuthenticationResult Authenticate(
    string username,
    string password)
```

Authenticates the user having the specified username using the specified password. The method relies on the `UserAuthenticator` class (described below) which in fact performs the process of authentication.

The following classes support the authentication of users:

Class Name	Description
`AuthenticationResult`	Represents the result of a user authentication. The result specifies whether authentication of the user should be rejected or approved. In addition the result holds among other things a return code specifying the security case that lead to the rejection or approval.
`UserAuthenticator`	Supports authentication of a user. The `Authenticate` method of the class is the one that performs the security checks specified in the authentication decision diagram in appendix D – the method returns an `AuthenticationResult` object.

In the presentation tier we have made use of a module in ASP.NET called "Forms Authentication" which assists authentication of users and does a lot of troublesome work with cookies, encryption and the like behind the scenes[2]. Moreover, the module sees to it that any unidentified users are redirected to the login page.

6.7.3 Authorization

Authorization is the process of granting a user access to a particular part of the system or allowing the user to carry out some specific action. The presentation tier is responsible of performing the necessary authorization.

The `User` class in the business tier makes the following methods available to the presentation tier:

public static int[] GetRoles(Guid user_ID)

Retrieves a list of roles assigned to the specified user.

public static bool HasPermission(Guid user_ID, int permission_ID)

Checks whether the specified user has the specified permission. The method is based on the `GetRoles` method and the `HasPermission` method of the `Role` class.

public static bool HasRole(Guid user_ID, int role_ID)

Checks whether the specified user has the specified role. The method is based on the `GetRoles` method.

The methods in the `User` class are supported by the following methods in the `Role` class in the business tier:

public static int[] GetPermissions(int role_ID)

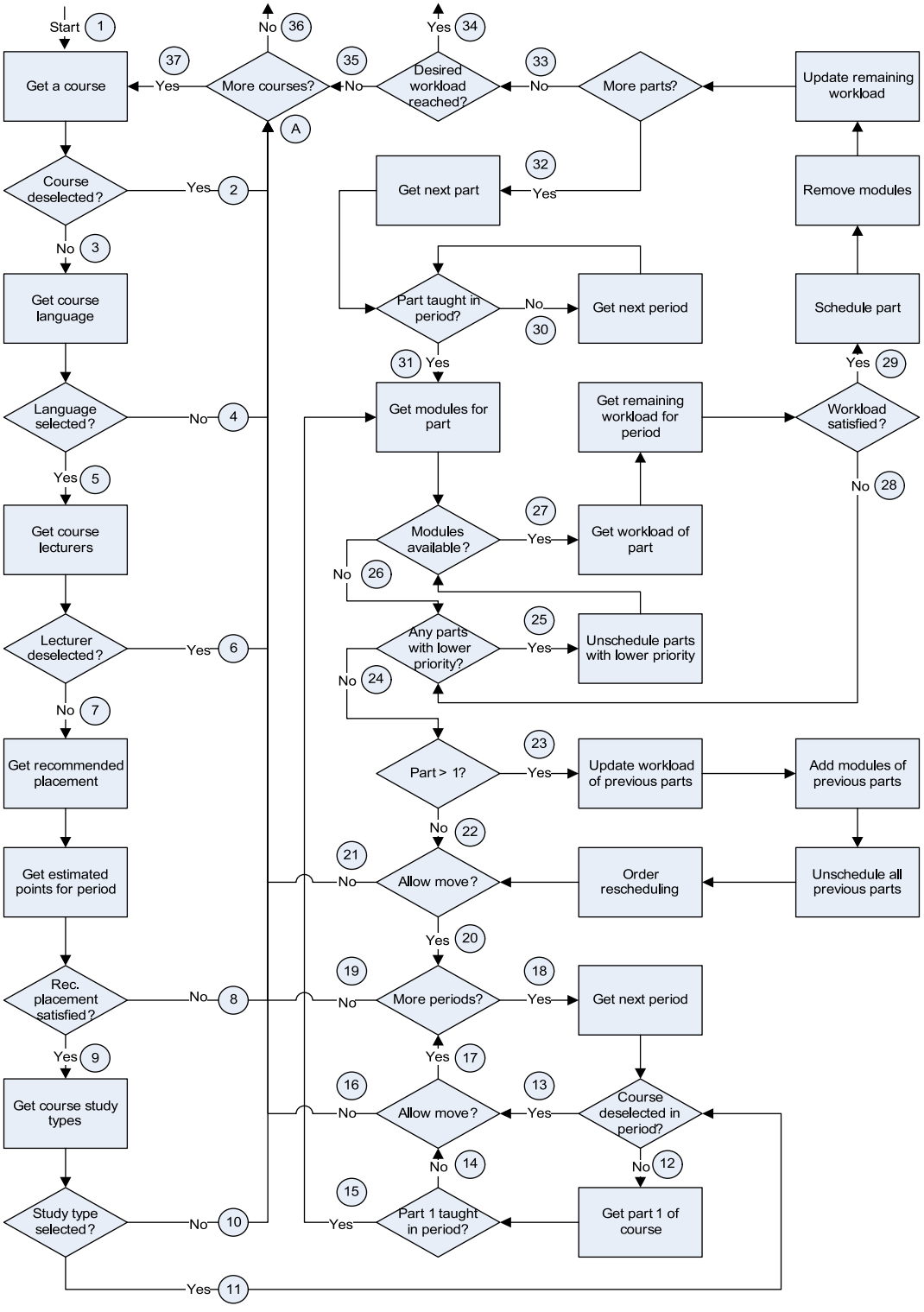
Retrieves a list of permissions assigned to the specified role.

public static bool HasPermission(int role_ID, int permission_ID)

Checks whether the specified role has the specified permission. The method is based on the `GetPermissions` method.

[2]The "Forms Authentication" module is described in chapter 9 in [2].

In a representative security check scenario, the presentation tier invokes the `HasPermission` or `HasRole` methods in the `User` class with a hard-coded identity of a role or a permission.


Start
1
Get a course
37
Yes
More courses?
No
36
A
35
No
Desired workload reached?
Yes
34
33
No
More parts?
Update remaining workload
Course deselected?
Yes
2
Get next part
32
Yes
Remove modules
No
3
Get course language
Part taught in period?
No
30
Get next period
Schedule part
31
Yes
Yes
29
Language selected?
No
4
Get modules for part
Get remaining workload for period
Workload satisfied?
Yes
5
No
28
Get course lecturers
Modules available?
27
Yes
Get workload of part
No
26
Lecturer deselected?
Yes
6
Any parts with lower priority?
25
Yes
Unschedule parts with lower priority
No
24
No
7
Get recommended placement
Part > 1?
23
Yes
Update workload of previous parts
Add modules of previous parts
No
22
Get estimated points for period
21
No
Allow move?
Order rescheduling
Unschedule all previous parts
Yes
20
Rec. placement satisfied?
No
8
19
No
More periods?
18
Yes
Get next period
Yes
9
Yes
17
Get course study types
16
No
Allow move?
13
Yes
Course deselected in period?
No
14
No
12
Study type selected?
No
10
15
Yes
Part 1 taught in period?
Get part 1 of course
Yes
11

Chapter 7

Initial Data

A vast set of initial data must be present in the system before the elaboration of study plans can take place. In this chapter we describe how the system has been initialized with data.

Only course data can be entered automatically into the system as described in section 7.4. The remaining data is either not available in a sufficiently structured form or not available at all (subsections 7.1-7.3) and we have therefore had to create these data manually by means of SQL commands. The SQL commands for creating the initial data can be found in volume III, chapter 3.

7.1 Common Periods

All the common periods such as "3-week-periods" and "13-week-periods" have been created manually starting from the year 1998 to 2012. For passed and current periods the actual start and end dates have been used. For future periods the start and end dates have been estimated by using a calendar. The data for creating periods is shown in volume III, section 3.11.

7.2 Technical Packages

Technical packages are not available in a structured form like the courses. This implies that all technical packages have been entered into the database by hand. For M.Sc. students the technical packages have been created from 1998 to 2003 using the historical information[1]. From 2004 to 2011 the technical packages have been created as copies of the 2003 versions. However, the technical packages from 1998 to 2001 do not work due to some discontinued courses. For each year a new version of the technical package has been created – even if the technical package is unchanged compared to the previous year.

For B.Sc. students the history of technical packages is not available so the technical packages from 2004 to 2011 have been created as copies of the 2003 version. Furthermore most B.Sc.

[1] http://www.adm.dtu.dk/studieinformation/studinfo/fagpakkeskema/index_d.htm

technical packages start both in February and September so for these, two different technical packages have been created.

The "Mechanical Engineering" technical package for B.Sc. students differs from the other technical packages. The first part is common for all students but from the 3rd semester the technical package branches off as shown in figure 7.1.

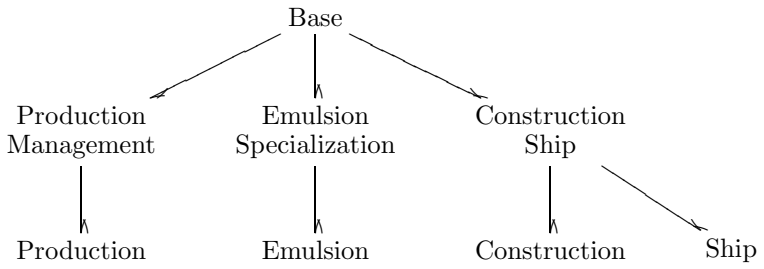


Figure 7.1: Illustration of how the "Mechanical Engineering" technical package branches off for various periods.

The technical packages have been created by the SQL listed in volume III section 3.21.

7.3 Miscellaneous Data

As was the case with technical packages, also technical lines, technical fields and specializations have been entered into the database by hand.

In the study planning system numerous type tables exist. These tables represent the type of another entity. For example the type of a `Period` is represented by the `PeriodType` table. Typically the type tables only contain a few rows which also have been entered by hand.

7.4 Course Data

The courses at DTU can be found in a structured format in the "DTU Course Catalogue"[2]. In order to initialize the study planning system with the current courses at DTU, a comprehensive component for grabbing the courses from the course catalogue has been implemented. The course grabbing component is represented by the `CourseGrabber` class in the business tier.

The HTML structure of the course catalogue web pages has been systematically analyzed and from the analysis a greater number of regular expressions have been prepared. The `CourseGrabber` component establishes a HTTP connection to the course catalogue and iterates through all the courses. For each course the various course data are grabbed by use of the regular expressions and finally the data are stored in the database.

Due to the fact that the specification of prerequisites and timetable modules of many courses is unstructured, inconsistent and in some cases actually erroneous this information had to be

[2]The "DTU Course Catalogue" is available at http://www.kurser.dtu.dk.

processed manually after grabbing the courses (confer volume III, chapter 4). The "*course part*" concept does not presently exist in the course catalogue[3] and we thus had to work in this concept in the creation of the afore-mentioned manual data.

[3]Only an indication of whether a course spans multiple terms is present which is not the same as course parts.

Chapter 8

User Interface

In this chapter we present principal aspects of the user interface[1]. We will refrain from describing the individual pages of the user interface as one can see these by own personal inspection of the system.

Section 8.1 presents the login page and section 8.2 describes the persistent navigation in the user interface. The support for different cultures in the user interface is briefly introduced in section 8.3.

8.1 Login Page

When an unidentified user requests for a page in the system, the user will be directed to a login page just as described in section 6.7.2.

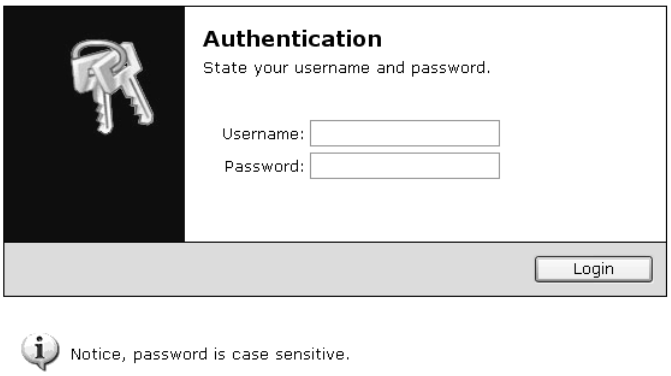


Figure 8.1: Screen shot of the login page.

Figure 8.1 shows a screen shot of the login page. The login page acts according to the requirements posed in section 4.2.2.

[1] We have been greatly inspired by the recommendations and the points presented in [17] in the preparation of the user interface.

8.2 Persistent Navigation

The user interface contains a persistent navigation i.e. a number of navigation elements that appear on every page in the user interface (apart from the login page). Partly, the persistent navigation is composed of primary and secondary navigation elements, partly, a so-called breadcrumb. Moreover, two utility links are also part of the persistent navigation. The constituent parts are described individually below.

8.2.1 Primary and Secondary Navigation

Every page (apart from the login page) contains a number of primary and secondary navigation elements. These elements appear in the same place on every page (namely at the top) and work in a consistent way.

The primary navigation elements consist in a number of main sections. In the current version of the system, the following main sections exist: "Study Planning", "Course Base", "Study Info" and "Control Panel". An illustration of the primary and secondary navigation elements are given in figure 8.2.

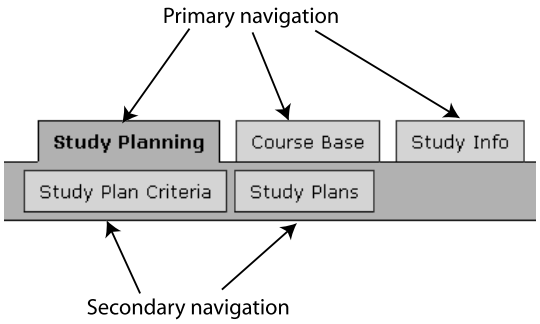


Figure 8.2: Illustration of the primary and secondary navigation elements.

8.2.2 Breadcrumb

As part of the persistent navigation we have included a breadcrumb (alias a "You are here" indicator). This a common aid used in many web applications and we quote from [17], page 74:

"One of the many ways navigation can counteract the Web's inherent 'lost in space' feeling is by showing me where I am in the scheme of things, the same way that a 'You are here' indicator does on the map in a shopping mall..."

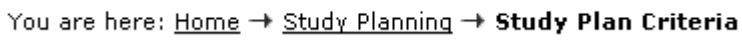


Figure 8.3: Screen shot of the breadcrumb included as part of the persistent navigation in the user interface.

As it appears from figure 8.3, the breadcrumb shows the path from the Home page to the page the user is currently at. That is, the user is at all times ensured a general view hereby meeting one of the requirements posed in section 4.4.2.

8.2.3 Utilities

In the upper right corner of each page a link to the home page is present. Next to the home page link, a link for signing out of the system is present. These links are depicted in figure 8.4.

Figure 8.4: Screen shot of the breadcrumb included as part of the persistent navigation in the user interface.

8.3 Support for Different Cultures

The user interface has been culture versioned such that text strings, date formats *&c.* can be shown in various cultures. Currently, the Danish and British cultures are supported but other European cultures can easily be added.

When a user enters the login page, the culture of the user's session is set according to the user's preference. In any standard browser the user can configure his preferred languages – e.g. as illustrated in figure 8.5.

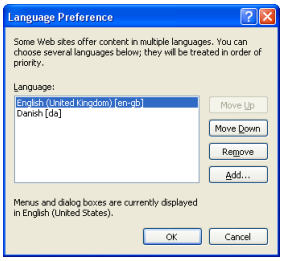


Figure 8.5: Screen shot of the "Language Preference" dialogue in Microsoft Internet Explorer.

If the user has not selected any of the supported languages, the culture is default set to the British culture.

Part III

System Test

Chapter 9

Test

In this chapter the testing of the system is described. First the overall test strategy will be introduced in section 9.1. Next, the testing of the system components will be presented in section 9.2 and finally an integration test may be found in section 9.3.

9.1 Test Strategy

The strategy for testing the system is first to perform a component test where each component is tested separately. In this context a component is considered an atomic module which does not depend on other components. When all components work satisfactorily the modules which rely on several components are tested in an integration test.

9.2 Component Test

The system consists of a number of components which have been tested separately to eliminate fundamental errors and mistakes. As mentioned in section 5.4 each component in the data tier controls the access to a specific database table. In order to conduct a thorough testing a test driver has been created for each component. The test driver consists of a web page which provides an interface to the component as well as its four operations `CREATE`, `READ`, `UPDATE` and `DELETE`. For example the test driver for the `PeriodType` class is shown in figure 9.1.

The creation of an entry may be tested by entering some data into the text fields and pressing the “Create” button. If the operation is successful the identification of the newly created entry is returned in the top field – the `PeriodType_ID` in this case. Retrieving an entry is done by entering the key of the entry in the key field – the `PeriodType_ID` in this case – and pressing the “Retrieve” button. The retrieved data is then displayed in the text fields. The data may then be changed and updated by pressing the “Update” button. Finally an entry may deleted by entering the key of the entry and pressing the “Delete” button. The “Reset” button blanks all the text fields.

If an error occurs it is displayed below the text fields. Figure 9.2 shows the error which occurs if an entry with the specified key was not found in the database.

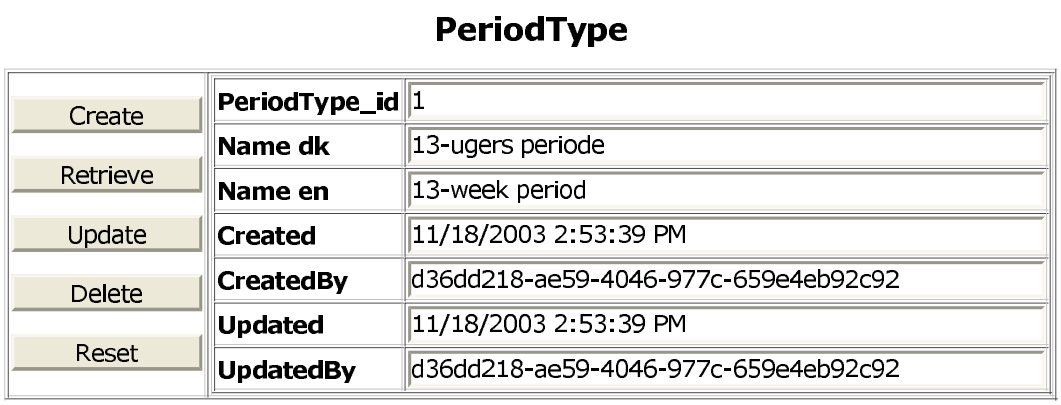


Figure 9.1: Illustration of the test driver for the `PeriodType` class.

```
StudyPlanning.DAL.DalException: Row does not exist. at
StudyPlanning.DAL.PeriodTypes.PeriodType.Retrieve() ...
```

Figure 9.2: Error from Data Tier.

For some components only the `RETRIEVE` operation has been implemented which is also the case for the above-mentioned `PeriodType` class. The test drivers have been prepared for all four operations, however, in this case pressing any other button than "Retrieve" or "Reset" will have no effect.

9.3 Integration Test

Some modules in the business tier like the `CourseGrab` module and the `StudyPlanElaborator` module also have test drivers similar to the one shown for the `PeriodType`. The test driver for the `StudyPlanElaborator` allows the user to enter the identification of a student and a study plan criterion. Pressing the "Elaborate" button starts the elaboration of a study plan and the result is displayed as shown in the following example:

The resulting study plan is validated manually by looking up the scheduled courses in the course catalogue [7] and by checking the various constraints.

Some of the following validations may fail if the student has specified conflicting demands in the study plan criterion. For example by specifically selecting a course which must be scheduled at some point and then deselecting the module(s) in which the course is taught for all periods. Furthermore, some validations may also fail due to the fact that not all constraints are checked for mandatory and fundamental courses in the technical package as mentioned in section 6.6.

A course must not be scheduled more than once

This constraint is easily checked by looking at the courses in the study plan.

A timetable module may only be used once in each period.

The test driver prints out the timetable modules occupied by each course in every period so this rule is checked by assuring that no timetable module occurs more than once in a period.

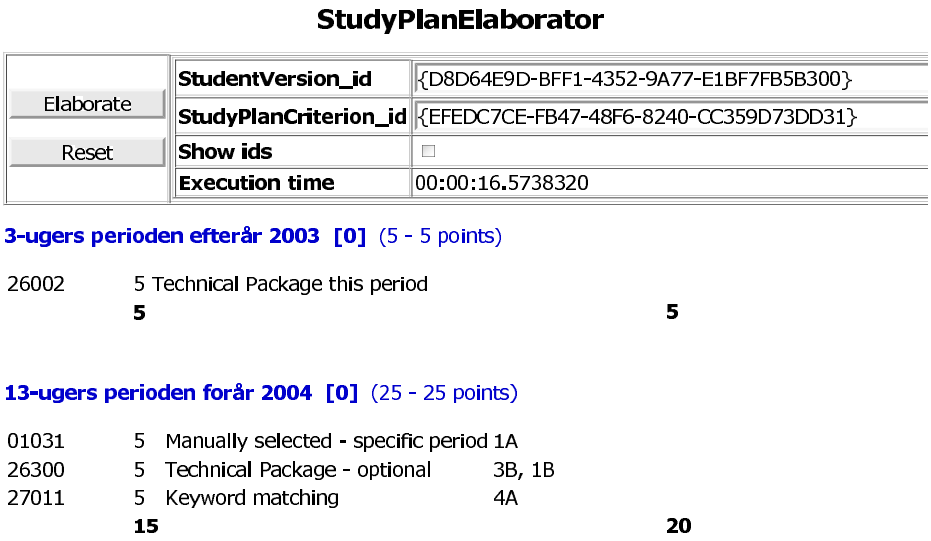


Figure 9.3: Illustration of the test driver for the `StudyPlanElaborator` class.

The desired maximum workload for a period must not be exceeded.

For each period the lower and upper bound of desired workload is printed along with the estimated number of points for the scheduled course parts in the period. The sum of points for scheduled course parts must be less than or equal to the upper bound of desired workload.

The parts of multiple part courses must be scheduled sequentially in ascending order and in the first possible period where the course part is taught.

Multiple part courses are displayed as *(Part#/Parts)* so checking that the course parts are scheduled in ascending order is quite trivial. Checking that course parts are scheduled in the first possible period requires a little more effort. If the next part is not scheduled in the first coming period it must be verified that the course part is not taught in that period as shown below where C_i is the course part i and P_j is period j:

$$\left.\begin{array}{c} scheduled(C_i, P_j) \wedge \neg scheduled(C_{i+1}, P_{j+k}) \Rightarrow \\ \neg taught(C_{i+1}, P_{j+k}) \end{array}\right\} i, j \in \mathbb{N}, k = \{1, 2, \ldots\}$$

Only courses being taught in the selected languages may be scheduled.

Check the language of the scheduled courses by looking them up in the course catalogue.

A course may only be scheduled if it has one of the selected study types.

Verify that the scheduled courses all have one of the selected study types.

Courses being taught by deselected lecturers may not be scheduled.

Consult the course catalogue for all scheduled courses and verify that they are not taught by one or more deselected lecturers.

Deselected courses may not be scheduled at all.

Verify that no deselected courses occur in the schedule.

Courses which have been specifically selected must be scheduled at some point.

Check that the specifically selected courses occur in the study plan.

Specifically deselected courses in a given period may not be scheduled in that period.

Verify that the deselected courses have not been scheduled in the specified period.

Courses which have been specifically selected in a given period must be scheduled in that period.

Check that the specifically selected courses have been scheduled in specified period.

Deselected modules may not be used.

Verify that no courses have been scheduled in deselected modules.

Courses must be scheduled according to their priority shown in table 6.1.

This validation requires that all the courses from the categories of the above-mentioned table are checked such that when multiple choices are available the course with the highest priority is scheduled. For each course in the study plan the test driver for `StudyPlanElaborator` prints the reason why the course has been scheduled here. Furthermore there is a logging function which displays all the decisions made by the `StudyPlanElaborator`

The results of the conducted tests will be further discussed in section 10.3.

Part IV

Summary

Chapter 10

Discussion

In this chapter we discuss whether the objectives of this thesis have been accomplished and whether the used approach have been appropriate.

The chapter is set out as follows:

- In section 10.1, different aspects of the conducted domain analysis and requirements specification are discussed.
- In section 10.2, discussions concerning the implemented system are made – including whether the posed requirements have been fulfilled.
- In section 10.3, a few comments are made on the conducted test.

Suggestions to future work are given in chapter 11.

10.1 Objectives

In the following two subsections we will discuss different aspects of the conducted domain analysis and requirements specification, respectively.

10.1.1 Domain Analysis

Analysis of the domain has been performed by applying an object-oriented approach and using UML class diagrams as a means of describing the entities of the domain. Compared to using a specification language[1] we find that in this context UML diagrams provide a more intuitive and two-dimensional description. UML class diagrams have been chosen over E/R (Entity Relationship) diagrams because class diagrams are object-oriented by nature and the representation of attributes in E/R is inconvenient when having anything but a few attributes.

The span of the domain analysis contains only B.Sc. students and M.Sc. students. It would have been relevant to include foreign M.Sc. students, Ph.D. students and food engineering students into the domain analysis as well, however, they have been omitted for reasons of

[1] For example RAISE

demarcation. We find that this omission is reasonable given the fact that B.Sc. and M.Sc. students represent the largest target group.

Considering the domain merely from the student's point of view has some drawbacks. Omitting lecturers from the domain analysis means that the any interactions between lecturer and student are not thoroughly analyzed. For example some lecturers offer individual projects which are not announced anywhere.

We find that the domain of study planning has been profoundly analyzed and that all relevant aspects of the domain as seen from the student's point of view have been carefully examined and documented.

10.1.2 Requirements Specification

As previously stated the requirements specification in chapter 4 is organized according to the principles expound in [18]. We value the concerned reference to be useful and we have found the prepared requirements specification to be a profitable "document" in the implementation of the study planning system.

Certainly, one could pose more detailed requirements, however, specifying too many details can be fatal to the creativity in the implementation of system and the possibility of handling problems in novel and slick ways is less likely.

We have chosen to describe the functional requirements in text. Another approach could have been to use UML "Use Case" diagrams which is also mentioned as an option in [18].

10.2 Implementation

In the subsections to come we discuss different aspects of the implemented study planning system. At first we verify whether the system complies with the posed requirements (section 10.2.1). Thereupon, the chosen architecture of the system is discussed (section 10.2.2).

Furthermore, we question:

- Is the system configurable? (section 10.2.3)
- Is the system extensible? (section 10.2.4)

The performance of the system is finally discussed in section 10.2.5.

10.2.1 Fulfillment of Requirements

In this subsection we discuss whether the implemented study planning system fulfills the requirements posed in the requirements specification presented in chapter 4.

Data Requirements

A systematic examination of the database design (appendix E) and the implemented data tier (section 5.4, appendix A) will show that the system is in fact capable of storing data as required in section 4.1. Moreover, section 5.1, subsections 5.2.3 and 5.4.4 and chapter 6 substantiate that claim.

Functional Requirements

Below we discuss whether the requirements posed with respect to functionality have been fulfilled.

- **Principal Requirement**
  As required, the system can be used by a web browser (confer section 5.6 and chapter 8). Into the bargain, a user interface to other types of user clients can relatively easy be provided in that all business and data logic have been separated from the web browser interface.
- **Elaboration of Study Plans**
  The system is indeed able to elaborate both correct as well as expedient study plans. This issue is discussed in section 10.3.
- **Various Management Functionality**
  As described in chapter 8, users are able to manage study plan criteria and view elaborated study plans by the user interface.
- **Restricted Admission *&c.***
  As expound in section 6.7, admission to the system is restricted, precautions are taken against brute force attacks and a strict password policy is managed. The requirements posed in section 4.2.2 are thus fulfilled.
- **Culture Versioning**
  Both the Danish and British cultures are supported by the current version of the system (confer subsections 5.2.5 and 5.4.4 and also chapter 8) and other European cultures can easily be added as exemplified in section 10.2.3.

In conclusion, the functional requirements have been fulfilled.

Quality Requirements

Below we discuss whether the requirements posed with respect to quality have been fulfilled.

- **Capacity**
  On account of the time available we have *not* carried out scalability tests and *only* verified that a subset of the initial capacity can be handled.
- **Performance**
  The system does indeed perform as required. The aspect of performance is discussed in section 10.2.5.
- **Usability**
  The posed usability requirements are difficult to measure, however, it is most likely that users can use the system without prior instruction. Moreover, the system only demands information from the user if the information cannot be derived from existing information.
- **Security:**
  - The system is not vulnerable to injections into the database as the data tier only interacts with the database using parameterized queries as explained in section 5.4.2.

  - Hackers cannot attack or make the system break down by exploiting buffer overruns as the system is based on the .NET Framework which offers a managed environment.
  - Nor does the system have a weak spot when it comes to URL tampering, content suction and script injections as countermoves have been made in that respect (confer section 5.6).
  - The transmission between the client and server is *not* done via the https protocol as it depends on a so-called SSL certificate which is not free of charge. However, this security aspect can easily be added later on.

- **Extensibility**
  The implemented system is indeed extensible. The aspect of extensibility is discussed in section 10.2.4.

In sum, except for the missing test of capacity, the implemented system can reasonably be said the meet all the posed quality requirements.

Other Requirements

Below we discuss whether the remaining requirements posed to the system have been fulfilled.

- **Legal Requirements**
  Because of the time available we have *not* taken specific initiative towards making the system fulfill the requirements posed in section 4.4.1.
- **User Interface Requirements**
  Referring to chapter 8 it is evident that the worked out user interface is in accordance with the requirements described in section 4.4.2.
- **Technical Requirements**
  As expound in section 5.6 the system only forwards output to the client (web browser) in a format such that a client, which complies to the standards specified in section 4.4.3, can actually use the system.
- **System Design and Programming**
  The system is based on a 3-tier architecture as expound in section 5.3 and object-oriented principles as it appears from e.g. sections 5.4-5.7. Consequently, the requirements made on system design and programming in section 4.4.4 have been fulfilled.

  Further discussion of the implemented architecture is given in section 10.2.2.
- **Documentation**
  The documentation requirements in section 4.4.5 can reasonably be said to have been fulfilled.

In short, apart from the legal requirements, the requirements posed in section 4.4 have as well been fulfilled.

10.2.2 Architecture

The various advantages of the chosen 3-tier architecture were enumerated in section 5.4. In addition to this we have a few comments based on our experience working with the architecture.

Undoubtedly, we felt it somewhat cumbersome to implement the data tier. However, the efforts proved to be rewarded as the business tier in many cases was very efficient to implement. Likewise, we found it a little incentive that the functionality of the system (e.g. elaborated study plans) were not visualized until late in the development process.

In addition the chosen approach concerning the one-to-one mapping between the table classes in the data tier and tables in the database, one could introduce a number of classes representing a join between two or more tables[2]. The disadvantage of a such insertion would be that several classes depend on the structure of a table and thus more dependencies have to be taken into account. The advantage would be less iterations in the business tier and possibly a better performing system.

10.2.3 Configurability

In the implementation of the system emphasis has been on configurability. The system is highly configurable which means that many changes may be performed without changing a single line of code. For example all types in the system are stored in database tables instead of being hard coded into the system modules. Some of the types used in the system are:

- Study Type
- Project Type
- Period Type
- ...

Adding a new study type to the system is thus merely a matter of inserting a few rows into the `StudyType` and `StudyTypeVersion` tables as well as other associated tables if applicable.

Representing course descriptions in a third language (besides Danish and English) such as German is as simple as inserting a new `<culture>` node in the XML string representing the course description (section 5.2.5).

If all the types had been hard coded into the system, development of the system might have been somewhat easier and quicker and the system would have worked just as well – maybe even faster since there would be less data to retrieve from the database. Using this approach, however, would have made the system quite static. Adding a new type would mean changing the code many places in the system with the possible risk of forgetting something. Introducing a new language might even require extension of some database tables as well as code alterations depending on the implementation.

In a dynamic world the requirements to a system (including a study planning system) are likely to change. A static system which requires the involvement of developers for making a small alteration is not desirable for the customer who has to pay for every change.

The approach used in the study planning system does not eliminate the need for code and database changes in order to cope with new requirements, but being configurable the number

[2] Alternatively, one could introduce a view in the database.

of changes needed is significantly lower when compared to hard coding. Hence the maintenance cost of the system is lower and the time-to-market is improved.

Section 11.1 describes how the system could be made even more configurable especially from the student's point of view.

10.2.4 Extensibility

We claim the system to be extensible and shall briefly exemplify how the system can be said to be so.

The utilized 3-tier architecture certainly conduces to the system being extensible. For instance, one can relatively easy support new types of user agents in that the new user interfaces can reuse the present logic in the business and data tiers. As another instance of flexibility, the system could bring new data sources into play (e.g. a web service) and the business tier would remain unaffected by this change in data strategy.

The fact that the system is implemented using an object-oriented programming language, namely C#, also makes the system extensible. As an example, existing classes can easily be extended with new methods and new classes can – without further ado – inherit from the present base classes.

In addition, the database design is made in a way such that data concerning distinct entities are stored in separate tables. Thereby, one can without exceptional difficulties alter data for an existing entity and introduce new entities.

10.2.5 Performance

Since study plans are to be elaborated online while the user is waiting, performance must be reasonable to an extent such that the user does not lose his patience. With the complexity of generating a study plan in mind most users will show understanding for the fact that the elaboration takes more than a few seconds.

Tests have shown that the elaboration of a study plan takes approximately 17 seconds on average[3]. Compared to the alternative of generating the study plans by an hourly running batch job, we find that a waiting time of 17 seconds is a small price to pay for an online generated result.

If the performance is not acceptable there are several possible ways of optimizing the performance of the study plan elaboration as described in the following.

Introducing Redundancy

Currently, the study plan elaboration algorithm uses some redundancy of data to improve performance. For example the elaborator often needs to know if a given course has already been scheduled when checking prerequisites and to prevent scheduling of the same course multiple times. This may be determined by performing a linear search through the preliminary study plan until the course has been found or the end of the study plan has been reached

[3]On an AMD Athlon XP 2000 with 256 MB RAM using Microsoft Internet Information Server 5.1, Microsoft SQL Server 2000 and version 1.1 of the .NET Framework

– hence the worst case runtime is $O(n)$ where n is the number of courses currently in the study plan. By using an additional hashtable for storing the courses which have already been scheduled the search time may be reduced to approximately $O(1)$ in most cases[4] The improvement in search time is significant even when taking the extra overhead of maintaining the redundant data into consideration.

For further optimization the prerequisite course chains for all courses could be generated and stored in the database for easy retrieval. Furthermore, the contents of the individual course part objects which are generated on the fly could be stored in a `CoursePart` table containing:

- Course_ID
- CourseVersion_ID
- Part number
- Total number of parts
- Workload for this part

Introducing Customized Retrieve Methods

The basic implementation of the 3-tier architecture implies that when retrieving data from a database table, all fields are retrieved regardless of whether only a single field is needed. If it appears that in for example 50% of the cases only two fields are needed, a new `RETRIEVE` operation dedicated to fetching only these two fields could improve performance. Special join modules which retrieve data from a join of multiple tables could also improve performance by eliminating the need for some loops of `RETRIEVE` operations.

10.3 Test

In the following subsections the test of the study planning system is discussed. First the correctness of the study plans generated by the study plan elaborator is discussed in section 10.3.1. Subsequently, it is discussed whether the elaborated study plans are appropriate for the students (section 10.3.2).

10.3.1 Correctness

The implementation of the study plan elaborator ensures that the generated study plans are correct. Being correct means that the resulting study plan does not violate the requirements specified in the corresponding study plan criterion and other constraints imposed by the university *&c.*

The integration test described in section 9.3 performed on a selection of different study plan criteria has shown that the resulting study plans are indeed correct. If a student has specified conflicting requirements in the study plan criterion he might not get the desired result, however, the study plan is still correct because the most restrictive of the conflicting requirements is used.

1) Select course 12345 which is taught in module 3A.
2) Deselect module 3A in all periods.

[4] http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpref/html/frlrfsystemcollectionshashtableclasscontainstopic.asp

If a student specifies the two above-mentioned requirements in a study plan criterion it may clearly be seen that the requirements are conflicting. However, since requirement 2) is the more restrictive the elaborator uses requirement 2) and thus requirement 1) cannot be satisfied in the resulting study plan.

10.3.2 Expediency

Even if a study plan satisfies all the requirements in the corresponding study plan criterion i.e. it is a correct study plan, it is not necessarily an expedient study plan for the student.

The priorities of courses (table 6.1) encourage scheduling of mandatory and fundamental courses at the beginning of the course of study. A study plan may thus contain one or more periods where exclusively fundamental courses have been scheduled. This may be perfectly correct, however, most students will not find such a study plan very appropriate due to the nature of these courses. Fundamental courses are often labour-intensive, they contain difficult material and hence the worst case scenario of scheduling for example five such courses in the same period would be overwhelming to the student.

Tests have shown that in practice, however, the study plan elaborator rarely schedules more than one or two fundamental courses in the same period unless a student has passed the middle of his study and has not yet taken any fundamental courses.

Section 11.1 describes some suggestions of how the system could be extended to allow the students to customize the priority of courses or to specify a maximum number of courses from each category of the priority table (table 6.1) which may be scheduled in one period.

Chapter 11

Future Work

In this chapter we present suggestions to future work to be done in continuation of this project.

11.1 More Individualized Configuration

Presently the system makes a number of predefined choices when elaborating a study plan for a given study plan criterion. Among these are the priority in which courses are selected in each period, when to check constraints and which alternatives to use when multiple choices are available.

Courses are selected according to the priorities listed in table 6.1 which aims at finishing the technical package and the fundamental courses as soon as possible whereas course matching keywords selected by the student have the lowest priority of all. However, some students might want a different priority which could be accomplished by introducing the possibility of customizing the priority rules.

Currently, some constraints such as deselected lecturers and preferred languages are not checked for mandatory courses in the technical package as well as fundamental courses. This setting has been selected because these courses must inevitably be finished if the student wants to obtain his degree. There is a time limit on the technical package which must be finished within four years after enrollment, however, if the student follows the recommended schedule the technical package will be finished after only two years and thus some students may want to exploit the given time limit a little more. Fundamental courses have no time limit except the student will not receive his degree until the fundamental courses have been passed. Being able to enable/disable checking of the various constraints for each group of courses might help the students to produce a more suitable study plan.

For many students fundamental courses are a necessary evil and are generally not regarded as particularly interesting. Some students want to finish the fundamental courses as soon as possible to get it over and done with. Others prefer taking one fundamental course every once in a while along with some other courses which match their interests. As the fundamental courses have a high priority in the elaboration of a study plan (confer table 6.1) chances are that several fundamental courses are scheduled in the same period which may not suit all

students. Therefore, as part of future work an option could be introduced as to how many fundamental courses may be scheduled in one period.

At present the elaboration of a study plan ensures that the mandatory courses in the technical package have been passed before any of the courses from a selected technical line are scheduled. Nonetheless, some students probably want to be able to bypass this restriction, so as a part of some future work this setting could be made user-customizable.

11.2 Representation of Skills

The specification of prerequisites is for some courses expressed as skills instead of or in addition to fixed course numbers. The following prerequisites are examples from the DTU course catalogue:

- Fundamental environmental chemistry.
- Knowledge of relational databases and query languages.
- Acquaintance with the so-called general linear model.
- Elementary knowledge of the SAS software system.

At first a specification of a skill seems be composed of the following constituents:

- **Level of skill**
  Superficial, elementary, extended, expert *&c.*
- **Type of skill**
  Knowledge of, acquaintance with, experience with *&c.*
- **The skill itself**
  A field, a method, a paradigm, a specific product *&c.*

However, further analysis is imperative and an independent project could be to evolve a way to represent skills. The elaboration of study plans would better reflect how study planning is executed in practice if skills could be specified in a well-defined way as part of a course prerequisite.

11.3 More Flexible Prerequisite Specifications

As mentioned in section 3.3.4 for some courses the dependencies are specified in the following form where $\mathcal{C}$ is the set of courses:

$$\left.\begin{array}{l}(a_1 \wedge b_1 \wedge \ldots \wedge z_1)\\ \vee\\ (a_2 \wedge b_2 \wedge \ldots \wedge z_2)\\ \vee\\ \ldots\end{array}\right\} a, b \ldots z \in \mathcal{C}$$

This form is presently not supported by the study planning system. However, for future work the above-mentioned specification could be implemented as part of allowing a more flexible specification of course interdependencies.

For some courses the specification of prerequisites allows that the course and its prerequisite courses are followed simultaneously. The following example where $\mathcal{C}$ is the set of courses illustrates this:

$$(a \wedge b \wedge \ldots \wedge z) \text{ at least simultaneously } \} \, a, b \ldots z \in \mathcal{C}$$

At the moment the elaboration of a study plan always assumes that prerequisites must be finished before scheduling a course, but as part of some future work this would be a possible extension.

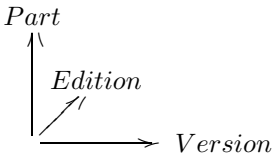
11.4 Editions of Courses

Many courses exist as regular and repeater courses created as two completely separate courses with different course numbers. There is no connection between the two courses apart from the course number which usually ends with a zero for the regular course and a one for the repeater course, however, this numbering convention cannot be taken for granted.

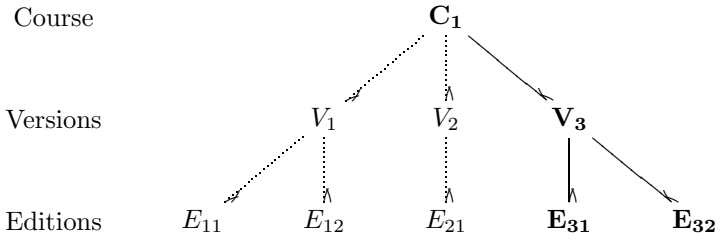
Since the courses are identical they should ideally be point blocking in relation to the same courses, but sometimes both courses are not specified as point blocking to other courses as the repeater course is typically forgotten. So when elaborating a study plan a course may be wrongfully scheduled due to errors in the course catalogue, however, this might also happen when performing the study planning by hand and hence it is not a problem inferred by the automated study planning.

To eliminate the need for regular and repeater courses a new dimension called *editions* could be added to the system. An edition is a variation of a course which inherits from the base course meaning that the connections between multiple instances can clearly be seen since they all inherit from the same base course. Editions differ from versions in that only one version of a course – i.e the newest – may be signed up for at any given time since it is used for retaining the history of a course whereas multiple editions of a course may coexist.

Adding the editions means that there are now three dimensions with respect to courses – parts, versions and editions.



The principles of versions and editions are illustrated in the following diagram:



In the diagram C_1 is a course. There are three versions of the course named V_1, V_2 and V_3. The versions V_1 and V_2 are obsolete as only the newest version of a course (V_3) is active. The editions inherit from the versions and hence the editions E_{11}, E_{12} and E_{21} have become obsolete since the versions from which they inherit are obsolete. So according to the example there is only one active version V_3 and two active editions E_{31} and E_{32}.

11.5 Taxonomy for Contents

Currently contents of courses are stated as keywords. However, this an unsophisticated and inadequate way of representing course contents. If a student e.g. specifies "physics" as one of his interests the search will not match a course having the keyword "quantum theory" even though quantum theory is a field within physics.

An independent project could be to devise how to represent course contents in a rich and intelligent way – a taxonomy for contents. In part prepare how to specify such a taxonomy and in part to map out the taxonomy itself. Having a well-developed taxonomy for course contents at disposal would make it possible to perform a more expedient study planning.

11.6 Student Grant

In order to maintain student grant the individual student may not be more than 12 months delayed in his studies compared with the prescribed time of study. In part for this reason the student has an interest in making continuous progress in his studies.

Receiving student grant for meeting living costs is an important aspect for the greater part of students and by it a factor which influences the study planning. A future version of the system could therefore profitably include student grant in the elaboration of study plans. As an alternative of specifying a desired workload in the different period types one could imagine a general option saying something like "Schedule courses and projects representing a workload which entitle me to maintain my student grant".

In fact the existing student grant rules are very accessible from an automated study planning point of view. To illustrate this we shall briefly describe the existing rules below taking Master of Science students as our starting point.

As mentioned elsewhere Master of Science students altogether have to collect 300 credit points in order obtain their degree. During his study a Master of Science student is entitled to receive at most 70 grants. The individual grant is not paid without further ado in that the student must have collected a fixed number of credit points (at least) in order to receive the grant in question. The precise point requirements for Master of Science students are given in the table below[1].

[1] Point requirements for the different study types are specified at 189-192 in [10].

Grant#	**Point**	**Grant#**	**Point**	**Grant#**	**Point**	**Grant#**	**Point**	**Grant#**	**Point**
1-12	0	24	42.5	36	97.5	48	157.5	60	225.0
13	2.5	25	47.5	37	102.5	**49**	**165.0**	61	230.0
14	5.5	26	52.5	38	107.5	50	170.0	62	237.5
15	7.5	27	57.5	39	112.5	51	175.0	63	242.5
16	10.0	28	62.5	40	115.0	52	180.0	64	250.0
17	15.0	29	67.5	41	120.0	53	185.0	65	255.0
18	17.5	30	70.0	42	125.0	54	190.0	66	262.5
19	20.0	31	75.0	43	130.0	55	195.0	67	270.0
20	25.0	32	80.0	44	135.0	56	202.5	68	275.0
21	30.0	33	85.0	45	142.5	57	207.5	69	280.0
22	35.5	34	90.0	46	147.5	58	212.5	70	287.5
23	40.0	35	95.0	47	152.5	59	217.5		

E.g. a Master of Science student must have collected at least 165 credit points in order to receive grant number 49. Similar points requirements apply to other study types. As is apparent from the example the existing rules is indeed measurable by disposition.

Chapter 12

Conclusion

In this thesis we have made a profound analysis of the domain of study planning from a student's point of view. Based on the analysis of the domain, a requirements specification has been prepared in which the necessary requirements for the implementation of a study planning system have been stated. A system which supports automated elaboration of study plans based on some criteria has been implemented – and can reasonably be said to fulfill the posed requirements.

The implemented system is among other things highly configurable and based on a 3-tier architecture. Furthermore, the system has been initialized with a complete set of the data related to study planning and not just an extract hereof.

Tests of the system have been conducted and discussions of the test results have been made. In addition we have also made discussions concerning different aspects of the implemented system. Finally, suggestions for future work to be done in continuation of this thesis have been presented.

We find that the resulting study planning system is a great step towards making it easier for students to plan their studies more carefully and more than just a couple of terms ahead. Hopefully, in the future some students will benefit from our efforts.

References

[1] Richard Anderson, Brian Francis, Alex Homer, Rob Howard, Dave Sussman, and Karli Watson. *Professional ASP.NET 1.0*. Wrox Press, 2002.

[2] Russ Basiura, Richard Conway, Brady Gaster, Dan Kent, Sitaraman Lakshminarayanan, Enrico Sabbadin, Doug Seven, and Srinivasa Sivakumar. *Professional ASP.NET Security*. Wrox Press, 2002.

[3] Marco Bellinaso and Kevin Hoffman. *ASP.NET Website Programming, Problem - Design - Solution*. Wrox Press, 2002.

[4] Simon Bennett, Steve McRobb, and Ray Farmer. *Object-Oriented Systems Analysis and Design using UML*. McGraw-Hill Publishing Company, 1999.

[5] Dines Bjørner. *Software Engineering – Theory & Pratice*, volume 2. Department of Informatics and Mathematical Modelling, Technical University of Denmark, DK-2800 Kgs. Lyngby, 2000.

[6] Microsoft Corporation. Role-Based Security in the .NET Framework.
http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpguide/html/cpconrole-basedsecurity.asp.

[7] Danmarks Tekniske Universitet. Kursusbasen.
http://www.kurser.dtu.dk.

[8] Danmarks Tekniske Universitet. Studieplaner – diplomingeniør.
http://www.adm.dtu.dk/studieinformation/studinfo/uddannelser/diploming_d.htm.

[9] Danmarks Tekniske Universitet. Studieplaner - civilingeniør.
http://www.adm.dtu.dk/studieinformation/studinfo/uddannelser/civiling_d.htm.

[10] Danmarks Tekniske Universitet. Studiehåndbog 2003/2004, Civilingeniøruddannelsen, Diplomingeniøruddannelsen, Levnedsmiddeluddannelserne, Studieplaner, Regelsamling, 2003.

[11] Danmarks Tekniske Universitet, Studieforvaltningen. Skabelon til oprettelse af nye kurser.
http://www.adm.dtu.dk/intern/adm/studforv/infosite_for_undervisere/skabelon.pdf.

[12] Department of Justice. The Act on Processing of Personal Data.
http://www.datatilsynet.dk/attachments/20001061548/ENGELSK%20LOV.doc, May 2000.

[13] Alex Homer and Dave Sussman. *ASP.NET Distributed Data Applications*. Wrox Press, 2002.

[14] Helge Elbrønd Jensen and Tom Høholdt. *Grundlæggende Matematik for Dataloger*. Matematisk Institut, Danmarks Tekniske Højskole, 1993.

[15] Morten M. Jensen and Teddy K. Nielsen. Study Planning System – Domain Description & Requirements Specification. Department of Informatics and Mathematical Modelling, Technical University of Denmark, June 2003.

[16] John Kauffman, Brian Matsik, Eric N. Mintz, Jan D. Narkiewicz, Kent Tegels, John West, Donald Xie, Jesudas Chinnathampi, Fabio Claudio Ferracchiati, and James Greenwood. *Beginning ASP.NET Databases using C#*. Wrox Press, 2002.

[17] Steve Krug. *Don't Make Me Think – A Common Sense Approach to Web Usability*. New Riders Publishing, 2000.

[18] Soren Lauesen. *Software Requirements - Styles and Techniques*. Addison Wesley, 2002.

[19] Håkon Wium Lie and Bert Boss. *Cascading Style Sheets – Designing for the web*. Addison Wesley, 2nd edition, 1999.

[20] Scott Mitchell. Using MD5 to Encrypt Passwords in a Database.
http://aspnet.4GuysFromRolla.com//articles/103002-1.aspx.

[21] Jakob Nielsen and Marie Tahir. *Homepage Usability – 50 Websites Deconstructed*. New Riders Publishing, 2001.

[22] Wayne Plourde. 15 Seconds: Creating a Data Access Layer in .NET - Part 1.
http://www.15seconds.com/issue/030317.htm.

[23] Wayne Plourde. 15 Seconds: Creating a Data Access Layer in .NET - Part 2.
http://www.15seconds.com/issue/030401.htm.

[24] Dr. Seeler. Boolean Algebra and Two State Logic.
ww2.lafayette.edu/~seelerk/me478/ps/Boolean.pdf.
Department of Mechanical Engineering, Lafayette College.

[25] The Danish Data Protection Agency. Sikkerhedsvejledning (Vejl. nr. 37 af 2. april 2001).
http://www.datatilsynet.dk/attachments/20014171359/ny.sikkerh.vejl.WORD70.doc, April 2001.

[26] Chris Ullman, Chris Goode, Juan T. Llibre, Ollie Cornes, Rob Birdwell, Ajoy Krishnamoorthy, Christopher L. Milling, Neil Raybould, and David Sussman. *Beginning ASP.NET using C#*. Wrox Press, 2001.

[27] Jeffrey D. Ullman and Jennifer Widom. *A First Course In Database Systems*. Prentice-Hall International, Inc., 1997.

Part V

Appendices

Appendix A

Overview of the `StudyPlanning.DAL` Namespace

In this appendix an overview of the `StudyPlanning.DAL` namespace is provided. The namespace constitutes the data tier.

The appendix is organized as follows:

- Section A.1 describes the classes that are organized in the root of the `StudyPlanning.DAL` namespace.
- Section A.2 describes the classes that are organized within a subnamespace to the `StudyPlanning.DAL` namespace.

A.1 Root Classes

A number of classes are organized directly in the root of `StudyPlanning.DAL` namespace:

This section is organized as follows:

- Subsection A.1.1 describes the classes that represent attributes of database tables.
- Subsection A.1.2 describes the classes that represent database tables.

A.1.1 Attribute Classes

The classes listed in the below table represent attributes of database tables. All of the listed classes are organized in the root of the `StudyPlanning.DAL` namespace.

Class Name	Description
DalType	Base class for the classes that model a single attribute of a database table.
DalBool	Models an attribute (of a database table) representing a boolean.
DalDateTime	Models an attribute (of a database table) representing date and time of day.
DalFloat	Models an attribute (of a database table) representing floating point data.
DalGuid	Models an attribute (of a database table) representing a globally unique identifier.
DalInt	Models an attribute (of a database table) representing an integer.
DalString	Models an attribute (of a database table) representing a string.
DalStringLocalizable	Models an attribute (of a database table) representing a localizable string.
DataLog	Represents the attributes Created, CreatedBy, Updated and UpdatedBy which occur on every single table in the database.

A.1.2 Table Classes

The classes listed in the below table represent tables in the database. All of the listed classes are organized in the root of the StudyPlanning.DAL namespace.

Class Name	Description
`Assessment`	Represents the “Assessment” table in the database.
`AssessmentType`	Represents the “AssessmentType” table in the database.
`ContactData`	Represents the “ContactData” table in the database.
`ContactDataType`	Represents the “ContactDataType” table in the database.
`Department`	Represents the “Department” table in the database.
`Gender`	Represents the “Gender” table in the database
`Grade`	Represents the “Grade” table in the database.
`Keyword`	Represents the “Keyword” table in the database.
`Language`	Represents the “Language” table in the database.
`Lecturer`	Represents the “Lecturer” table in the database.
`Module`	Represents the “Assessment” table in the database.
`Period`	Represents the “Period” table in the database.
`Project`	Represents the “Project” table in the database.
`ProjectType`	Represents the “ProjectType” table in the database.
`Text`	Represents the “Text” table in the database.
`Weekday`	Represents the “Weekday” table in the database.

A.2 Subnamespaces

The `StudyPlanning.DAL` namespace has been divided into the subnamespaces stated in the below table.

Each namespace contains classes that operate on database tables storing data concerning a specific part of the system e.g. the `StudyPlanning.DAL.Courses` namespace contains classes that operate on database tables storing course data.

Namespace	Described in
StudyPlanning.DAL.Courses	Section A.2.1
StudyPlanning.DAL.PeriodTypes	Section A.2.2
StudyPlanning.DAL.RecommendedPlacementConcepts	Section A.2.3
StudyPlanning.DAL.SecurityRoles	Section A.2.4
StudyPlanning.DAL.Specializations	Section A.2.5
StudyPlanning.DAL.Students	Section A.2.6
StudyPlanning.DAL.StudyPlanCriteria	Section A.2.7
StudyPlanning.DAL.StudyPlans	Section A.2.8
StudyPlanning.DAL.StudyTypes	Section A.2.9
StudyPlanning.DAL.TechnicalFields	Section A.2.10
StudyPlanning.DAL.TechnicalLines	Section A.2.11
StudyPlanning.DAL.TechnicalPackages	Section A.2.12
StudyPlanning.DAL.Users	Section A.2.13

A.2.1 StudyPlanning.DAL.Courses

The StudyPlanning.DAL.Courses namespace contains classes that operate on database tables storing data concerning **courses**.

The below table describes the classes contained in the namespace.

Class Name	Description
`Course`	Represents the "Course" table in the database.
`CourseGrab`	Represents the "CourseGrab" table in the database.
`CourseVersion`	Represents the "CourseVersion" table in the database.
`Department`	Represents the "Course_Department" table in the database.
`Keyword`	Represents the "Course_Keyword" table in the database.
`Lecturer`	Represents the "Course_Lecturer" table in the database.
`Period`	Represents the "Course_Period" table in the database.
`PeriodModule`	Represents the "Course_PeriodModule" table in the database.
`PeriodModuleItem`	Represents the "Course_PeriodModuleItem" table in the database.
`Point`	Represents the "Course_Point" table in the database.
`RecommendedPlacement`	Represents the "Course_RecommendedPlacement" table in the database.
`RelationCourse`	Represents the "Course_RelationCourse" table in the database.
`RelationCourseItem`	Represents the "Course_RelationCourseItem" table in the database.
`StudyType`	Represents the "Course_StudyType" table in the database.
`StudyTypeCategory`	Represents the "Course_StudyTypeCategory" table in the database.

A.2.2 `StudyPlanning.DAL.PeriodTypes`

The `StudyPlanning.DAL.PeriodTypes` namespace contains classes that operate on database tables storing data concerning **period types**.

The below table describes the classes contained in the namespace.

Class Name	Description
`PeriodType`	Represents the "PeriodType" table in the database.
`Module`	Represents the "PeriodType_Module" table in the database.

A.2.3 `StudyPlanning.DAL.RecommendedPlacementConcepts`

The `StudyPlanning.DAL.RecommendedPlacementConcepts` namespace contains classes that operate on database tables storing data concerning **recommended placement concepts**.

Class Name	Description
`RecommendedPlacementConcept`	Represents the "RecommendedPlacementConcept" table in the database.
`StudyType`	Represents the "RecommendedPlacementConcept_StudyType" table in the database.

A.2.4 StudyPlanning.DAL.SecurityRoles

The StudyPlanning.DAL.SecurityRoles namespace contains classes that operate on database tables storing data concerning **security roles** (user roles).

The below table describes the classes contained in the namespace.

Class Name	Description
SecurityPermission	Represents the "SecurityRole_SecurityPermission" table in the database.

A.2.5 StudyPlanning.DAL.Specializations

The StudyPlanning.DAL.Specializations namespace contains classes that operate on database tables storing data concerning **specializations**.

The below table describes the classes contained in the namespace.

Class Name	Description
Specialization	Represents the "Specialization" table in the database.
Course	Represents the "Specialization_Course" table in the database.
SpecializationVersion	Represents the "SpecializationVersion" table in the database.

A.2.6 StudyPlanning.DAL.Students

The StudyPlanning.DAL.Students namespace contains classes that operate on database tables storing data concerning **students**.

The below table describes the classes contained in the namespace.

Class Name	Description
Student	Represents the "Student" table in the database.
Course	Represents the "Student_Course" table in the database.
Department	Represents the "Student_Department" table in the database.
Project	Represents the "Student_Project" table in the database.
StudentVersion	Represents the "StudentVersion" table in the database.
StudyPlan	Represents the "Student_StudyPlan" table in the database.
StudyPlanCriterion	Represents the "Student_StudyPlanCriterion" table in the database.

A.2.7 StudyPlanning.DAL.StudyPlanCriteria

The StudyPlanning.DAL.StudyPlanCriteria namespace contains classes that operate on database tables storing data concerning **study plan criteria**.

The below table describes the classes contained in the namespace.

Class Name	Description
StudyPlanCriterion	Represents the "StudyPlanCriterion" table in the database.
Course	Represents the "StudyPlanCriterion_Course" table in the database.
CoursePeriod	Represents the "StudyPlanCriterion_CoursePeriod" table in the database.
Keyword	Represents the "StudyPlanCriterion_Keyword" table in the database.
Language	Represents the "StudyPlanCriterion_Language" table in the database.
Lecturer	Represents the "StudyPlanCriterion_Lecturer" table in the database.
ProjectPeriodPoint	Represents the "StudyPlanCriterion_ProjectPeriodPoint" table in the database.
StudyType	Represents the "StudyPlanCriterion_StudyType" table in the database.

The namespace is further divided up into the following two subnamespaces:

- StudyPlanning.DAL.StudyPlanCriteria.WorkloadPeriods (described below)
- StudyPlanning.DAL.StudyPlanCriteria.WorkloadPeriodTypes (described below)

StudyPlanning.DAL.StudyPlanCriteria.WorkloadPeriods

The below table describes the classes contained in the namespace.

Class Name	Description
WorkloadPeriod	Represents the "StudyPlanCriterion_WorkloadPeriod" table in the database.
Module	Represents the "StudyPlanCriterion_WorkloadPeriod_Module" table in the database.

StudyPlanning.DAL.StudyPlanCriteria.WorkloadPeriodTypes

The below table describes the classes contained in the namespace.

Class Name	Description
WorkloadPeriodType	Represents the "StudyPlanCriterion_WorkloadPeriodType" table in the database.
Module	Represents the "StudyPlanCriterion_WorkloadPeriod-Type_Module" table in the database.

A.2.8 StudyPlanning.DAL.StudyPlans

The StudyPlanning.DAL.StudyPlans namespace contains classes that operate on database tables storing data concerning **study plans**.

The below table describes the classes contained in the namespace.

Class Name	Description
StudyPlan	Represents the "StudyPlan" table in the database.
Period	Represents the "StudyPlan_Period" table in the database.
PeriodCourse	Represents the "StudyPlan_PeriodCourse" table in the database.

A.2.9 StudyPlanning.DAL.StudyTypes

The StudyPlanning.DAL.StudyTypes namespace contains classes that operate on database tables storing data concerning **study types**.

The below table describes the classes contained in the namespace.

Class Name	Description
StudyType	Represents the "StudyType" table in the database.
ProjectType	Represents the "StudyType_ProjectType" table in the database.
StudyTypeVersion	Represents the "StudyTypeVersion" table in the database.
TechnicalLine	Represents the "StudyType_TechnicalLine" table in the database.

A.2.10 StudyPlanning.DAL.TechnicalFields

The StudyPlanning.DAL.TechnicalFields namespace contains classes that operate on database tables storing data concerning **technical fields**.

The below table describes the classes contained in the namespace.

Class Name	Description
TechnicalField	Represents the "TechnicalField" table in the database.
Course	Represents the "TechnicalField_Course" table in the database.
TechnicalFieldVersion	Represents the "TechnicalFieldVersion" table in the database.

A.2.11 StudyPlanning.DAL.TechnicalLines

The `StudyPlanning.DAL.TechnicalLines` namespace contains classes that operate on database tables storing data concerning **technical lines**.

The below table describes the classes contained in the namespace.

Class Name	Description
`TechnicalLine`	Represents the "TechnicalLine" table in the database.
`PrerequisiteCourse`	Represents the "TechnicalLine_PrerequisiteCourse" table in the database.
`PrerequisiteTechnicalPackage`	Represents the "TechnicalLine_PrerequisiteTechnicalPackage" table in the database.
`PrerequisiteTechnicalPackageCourse`	Represents the "TechnicalLine_PrerequisiteTechnicalPackageCourse" table in the database.
`Specialization`	Represents the "TechnicalLine_Specialization" table in the database.
`TechnicalField`	Represents the "TechnicalLine_TechnicalField" table in the database.
`TechnicalLineVersion`	Represents the "TechnicalLineVersion" table in the database.

A.2.12 StudyPlanning.DAL.TechnicalPackages

The `StudyPlanning.DAL.TechnicalPackages` namespace contains classes that operate on database tables storing data concerning **technical packages**.

The below table describes the classes contained in the namespace.

Class Name	Description
TechnicalPackage	Represents the "TechnicalPackage" table in the database.
FundamentalCourse	Represents the "TechnicalPackage_FundamentalCourse" table in the database.
FundamentalCourseItem	Represents the "TechnicalPackage_FundamentalCourseItem" table in the database.
Period	Represents the "TechnicalPackage_Period" table in the database.
PeriodCourse	Represents the "TechnicalPackage_PeriodCourse" table in the database.
PeriodCourseItem	Represents the "TechnicalPackage_PeriodCourseItem" table in the database.
PeriodOptionalCourse	Represents the "TechnicalPackage_PeriodOptionalCourse" table in the database.
Project	Represents the "TechnicalPackage_Project" table in the database.
TechnicalPackageVersion	Represents the "TechnicalPackageVersion" table in the database.

A.2.13 StudyPlanning.DAL.Users

The StudyPlanning.DAL.Users namespace contains classes that operate on database tables storing data concerning **users**.

The below table describes the classes contained in the namespace.

Class Name	Description
User	Represents the "User" table in the database.
Login	Represents the "User_Login" table in the database.
PasswordHistory	Represents the "User_PasswordHistory" table in the database.
SecurityRole	Represents the "User_SecurityRole" table in the database.

Appendix B

Overview of the `StudyPlanning.Biz` Namespace

In this appendix an overview of the `StudyPlanning.Biz` namespace is provided. The namespace constitutes the business tier.

The appendix is organized as follows:

- Section B.1 describes the classes that are organized in the root of the `StudyPlanning.´Biz` namespace.
- Section B.2 describes the classes that are organized within a subnamespace to the `StudyPlanning.Biz` namespace.

B.1 Root Classes

The classes described in the below table are organized directly in the root of `StudyPlanning.Biz` namespace.

Class Name	Description
Course	Represents a course.
CoursePart	Represents a course part.
Lecturer	Represents a lecturer.
Specialization	Represents a specialization.
Student	Represents a student.
StudyPlan	Represents a study plan.
StudyPlanCriterion	Represents a study plan criterion.
StudyPlanElaborator	Supports elaboration of a study plan.
TechnicalField	Represents a technical field.
TechnicalLine	Represents a technical line.
TechnicalPackage	Represents a technical package.
TextItem	Represents a text string.
User	Represents a user.
UserAuthenticator	Supports authentication of a user.
AuthenticationResult	Represents the result of a user authentication.
BizObject	Base class in the business tier. Other classes in the business tier inherit from the BizObject class (some, however, only indirectly).
BizException	Represents an error in the business tier.
BizStringLocalizable	Represents a localizable string.
BizType	Base class for types specific to the business tier. Presently only the BizStringLocalizable class inherits from the BizType class.

B.2 Subnamespaces

The StudyPlanning.Biz namespace has been divided into the subnamespaces stated in the below table.

Namespace	Described in
StudyPlanning.Biz.Course.Grab	Section B.2.1
StudyPlanning.Biz.Security	Section B.2.2

B.2.1 StudyPlanning.Biz.Course.Grab

The StudyPlanning.Biz.Course.Grab only contains the CourseGrabber class as described below.

Class Name	Description
`CourseGrabber`	Grabs courses from the DTU course catalogue at http://www.kurser.dtu.dk/.

B.2.2 `StudyPlanning.Biz.Security`

The `StudyPlanning.DAL.PeriodTypes` namespace contains classes related to security.

The below table describes the classes contained in the namespace.

Class Name	Description
`Password`	Represents a password of a user.
`Role`	Represents a user role.

Appendix C

Overview of the `StudyPlanning.UI` Namespace

In this appendix an overview of the `StudyPlanning.UI` namespace is provided. The classes within the namespace along the various ASPX pages constitute the presentation tier.

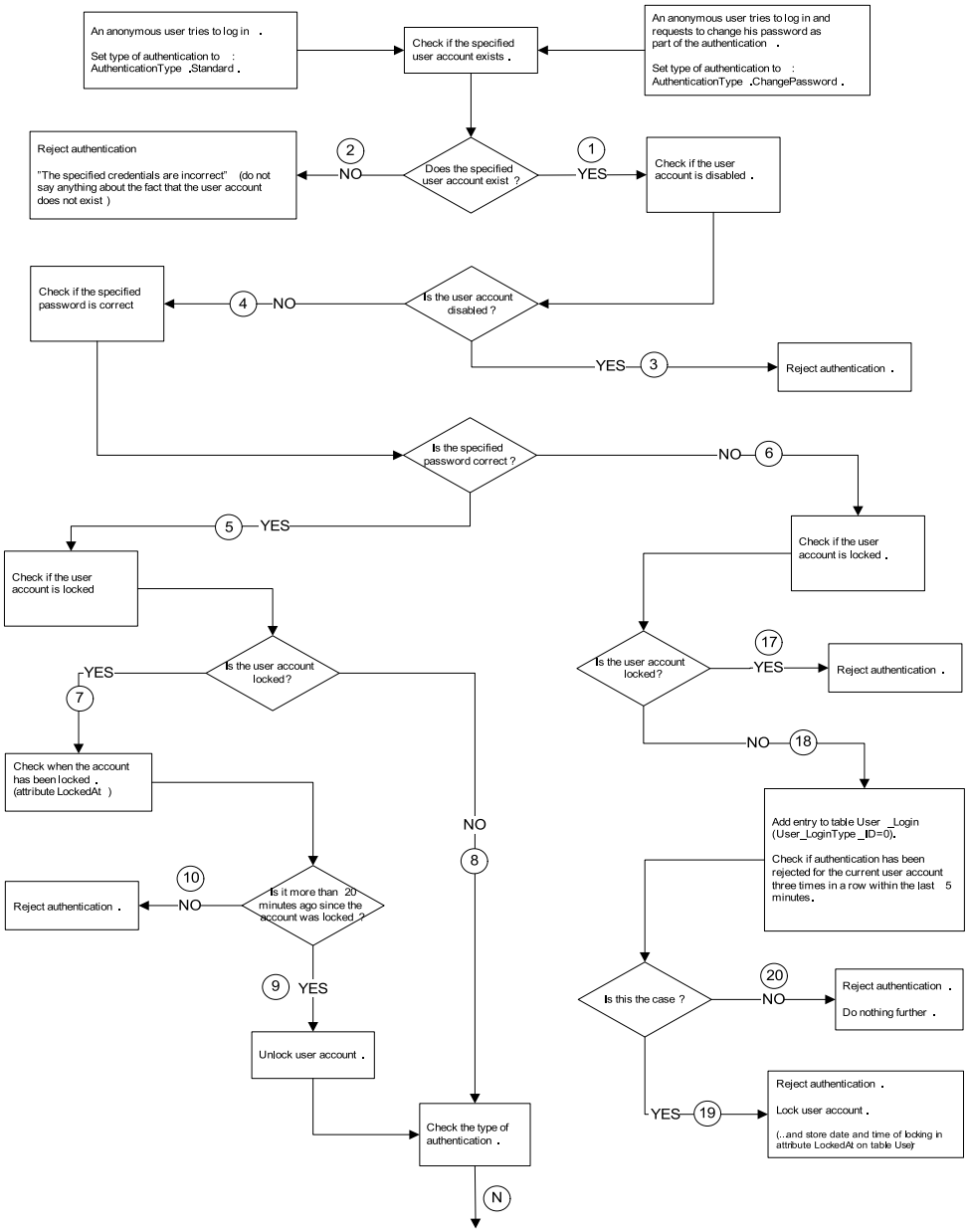
The namespace is divided into the subnamespaces described in the below table and are introduced according to the different primary navigation sections described in section 8.

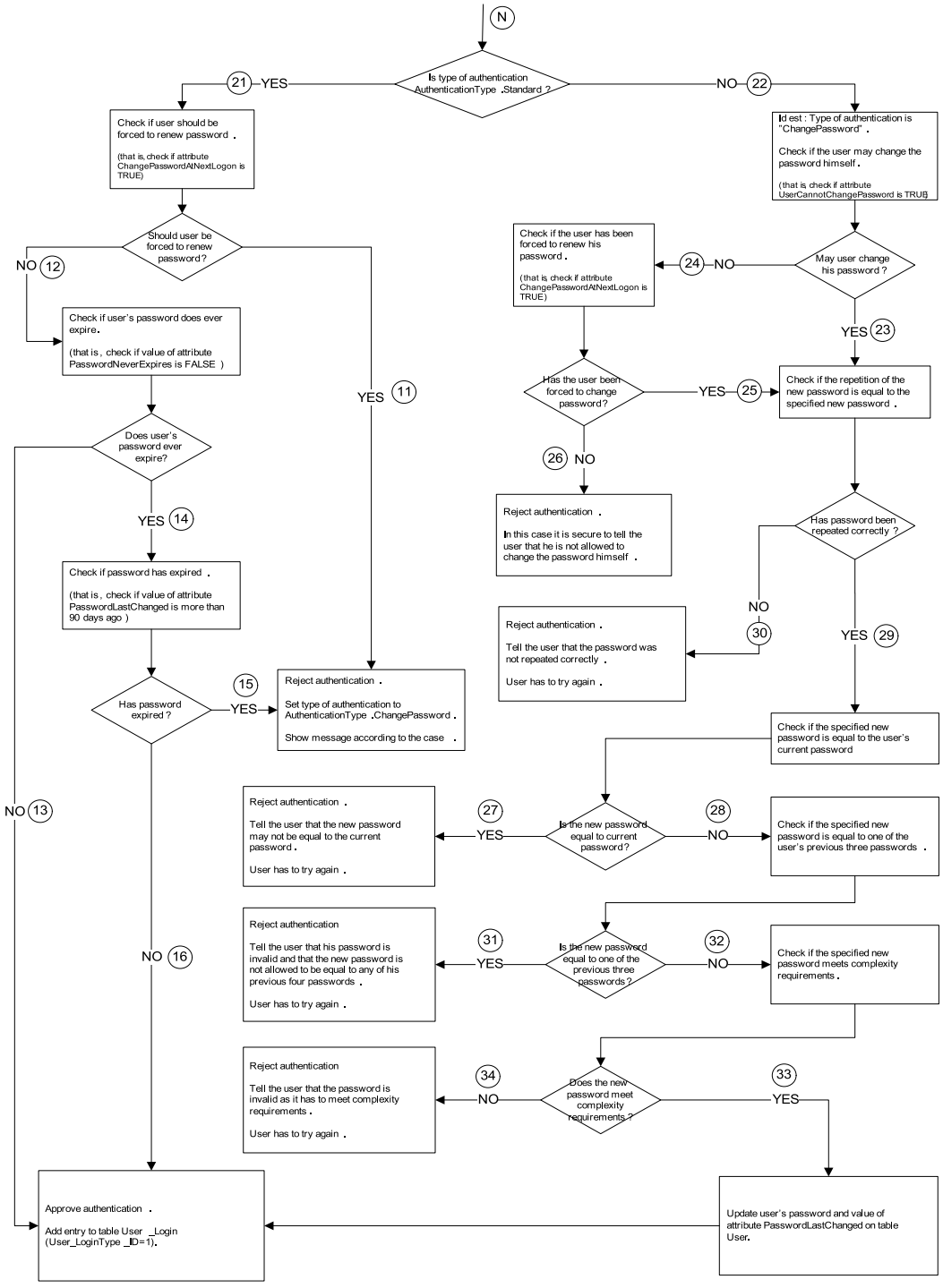
Namespace	**Described in**
`StudyPlanning.UI.studyplanning`	Contains classes that support the ASPX pages responsible of presenting study plans, enabling students to manage their study plan criteria *&c.*
`StudyPlanning.DAL.coursebase`	Contains classes that support the ASPX pages responsible of presenting information related to courses. The namespace is introduced for illustrative purposes.
`StudyPlanning.DAL.studyinfo`	Contains classes that support the ASPX pages responsible presenting study info of various kinds. The namespace is introduced for illustrative purposes.
`StudyPlanning.DAL.controlpanel`	Contains classes that support the ASPX pages that enables the user to manage different settings in the system, including changing his password. The namespace is introduced for illustrative purposes.
`StudyPlanning.DAL.administration`	Contains classes that support the ASPX pages enabling administrative staff to configure global settings of the system. The namespace is introduced for illustrative purposes.

Appendix D

Authentication Decision Diagram

For reasons of space the decision diagram has been split up into two parts and each part has been placed on its own independent page beginning at next page.

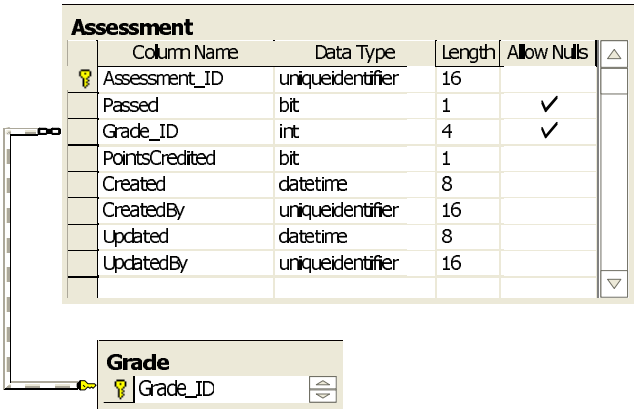

An anonymous user tries to log in .
Set type of authentication to : AuthenticationType .Standard .
Check if the specified user account exists .
An anonymous user tries to log in and requests to change his password as part of the authentication .
Set type of authentication to : AuthenticationType .ChangePassword .
Reject authentication
"The specified credentials are incorrect" (do not say anything about the fact that the user account does not exist)
2
NO
Does the specified user account exist ?
1
YES
Check if the user account is disabled .
Check if the specified password is correct
4
NO
Is the user account disabled ?
YES
3
Reject authentication .
Is the specified password correct ?
NO
6
5
YES
Check if the user account is locked
Check if the user account is locked .
Is the user account locked?
YES
7
17
YES
Reject authentication .
NO
18
Check when the account has been locked . (attribute LockedAt)
Add entry to table User _Login (User_LoginType _ID=0).
Check if authentication has been rejected for the current user account three times in a row within the last 5 minutes.
NO
8
10
Reject authentication .
NO
Is it more than 20 minutes ago since the account was locked ?
9
YES
Is this the case ?
20
NO
Reject authentication .
Do nothing further .
Unlock user account .
YES
19
Reject authentication .
Lock user account .
(..and store date and time of locking in attribute LockedAt on table User)
Check the type of authentication .
N

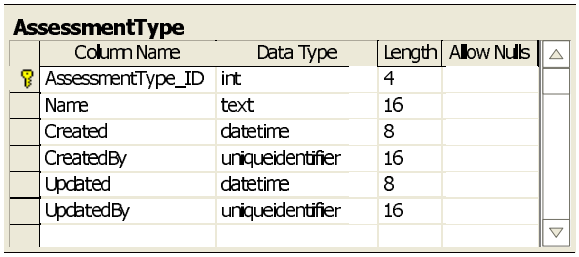

N
Is type of authentication AuthenticationType .Standard ?
21
YES
NO
22
Check if user should be forced to renew password .
(that is, check if attribute ChangePasswordAtNextLogon is TRUE)
Id est : Type of authentication is "ChangePassword" .
Check if the user may change the password himself .
(that is, check if attribute UserCannotChangePassword is TRUE)
Should user be forced to renew password?
NO
12
Check if the user has been forced to renew his password .
(that is, check if attribute ChangePasswordAtNextLogon is TRUE)
24
NO
May user change his password ?
YES
23
Check if user's password does ever expire.
(that is, check if value of attribute PasswordNeverExpires is FALSE)
YES
11
Has the user been forced to change password?
YES
25
Check if the repetition of the new password is equal to the specified new password .
Does user's password ever expire?
26
NO
Reject authentication .
In this case it is secure to tell the user that he is not allowed to change the password himself .
YES
14
Has password been repeated correctly ?
Check if password has expired .
(that is, check if value of attribute PasswordLastChanged is more than 90 days ago)
NO
30
Reject authentication .
Tell the user that the password was not repeated correctly .
User has to try again .
YES
29
Has password expired ?
15
YES
Reject authentication .
Set type of authentication to AuthenticationType .ChangePassword .
Show message according to the case .
Check if the specified new password is equal to the user's current password
NO
13
Reject authentication .
Tell the user that the new password may not be equal to the current password .
User has to try again .
27
YES
Is the new password equal to current password?
28
NO
Check if the specified new password is equal to one of the user's previous three passwords .
Reject authentication
Tell the user that his password is invalid and that the new password is not allowed to be equal to any of his previous four passwords .
User has to try again .
31
YES
Is the new password equal to one of the previous three passwords?
32
NO
Check if the specified new password meets complexity requirements .
NO
16
Reject authentication
Tell the user that the password is invalid as it has to meet complexity requirements .
User has to try again .
34
NO
Does the new password meet complexity requirements ?
33
YES
Approve authentication .
Add entry to table User_Login (User_LoginType _ID=1).
Update user's password and value of attribute PasswordLastChanged on table User.

Appendix E

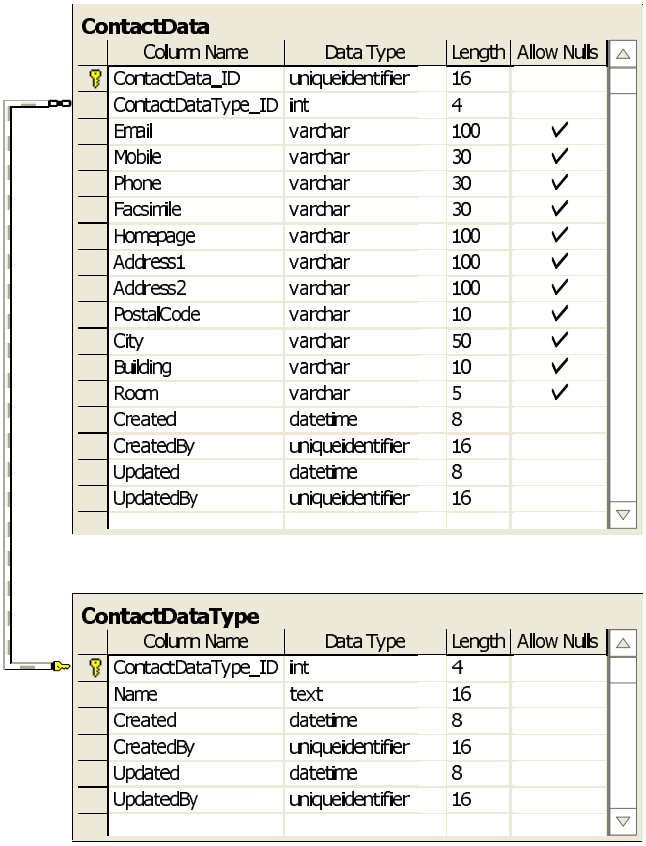
Database Diagrams

E.1 Assessment





E.2 ContactData



ContactData

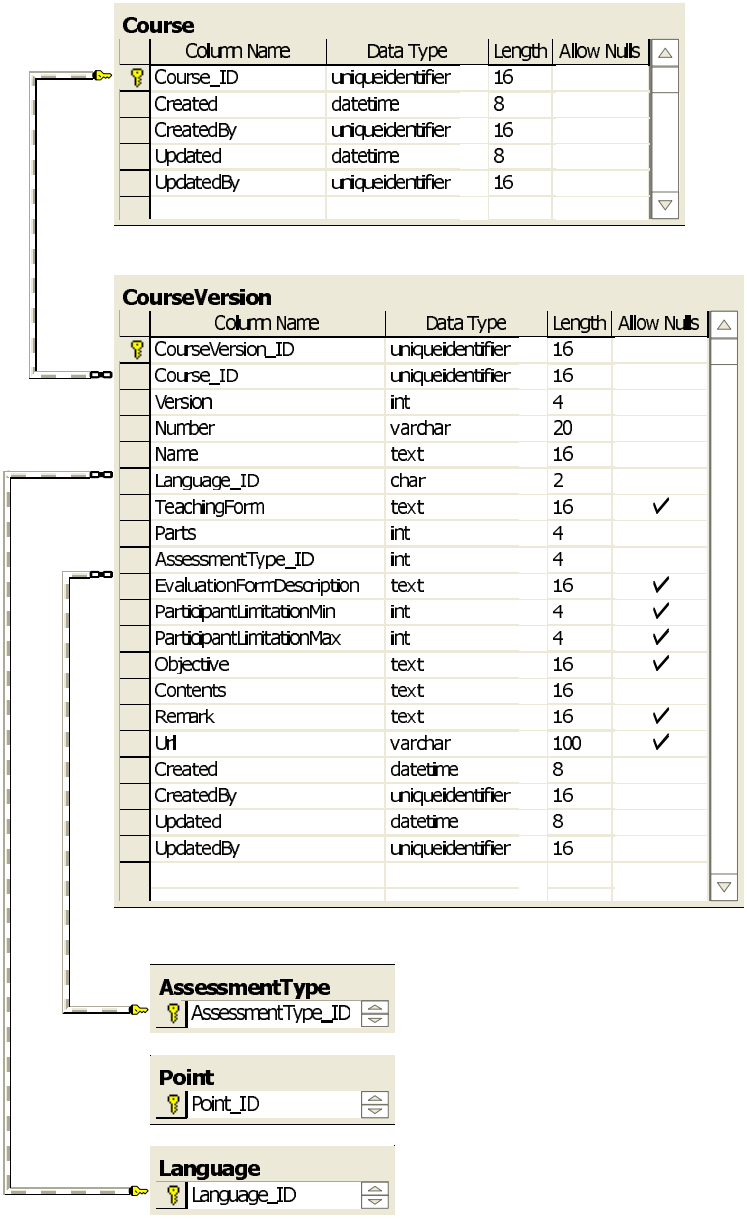
Column Name	Data Type	Length	Allow Nulls
ContactData_ID	uniqueidentifier	16	
ContactDataType_ID	int	4	
Email	varchar	100	✓
Mobile	varchar	30	✓
Phone	varchar	30	✓
Facsimile	varchar	30	✓
Homepage	varchar	100	✓
Address1	varchar	100	✓
Address2	varchar	100	✓
PostalCode	varchar	10	✓
City	varchar	50	✓
Building	varchar	10	✓
Room	varchar	5	✓
Created	datetime	8	
CreatedBy	uniqueidentifier	16	
Updated	datetime	8	
UpdatedBy	uniqueidentifier	16	

ContactDataType

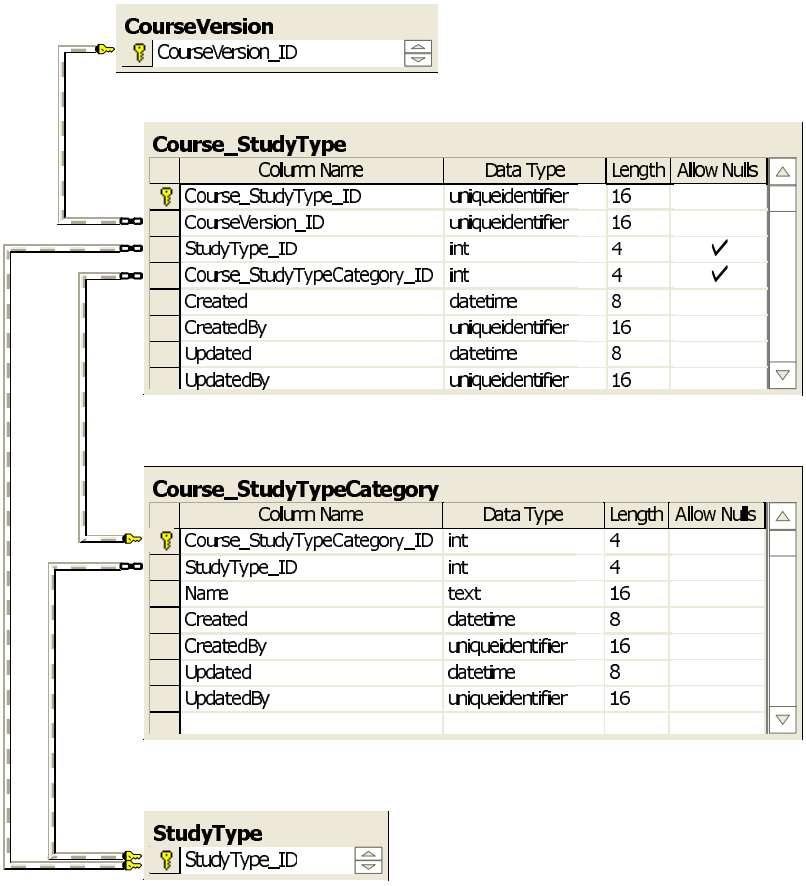
Column Name	Data Type	Length	Allow Nulls
ContactDataType_ID	int	4	
Name	text	16	
Created	datetime	8	
CreatedBy	uniqueidentifier	16	
Updated	datetime	8	
UpdatedBy	uniqueidentifier	16	

E.3 Course

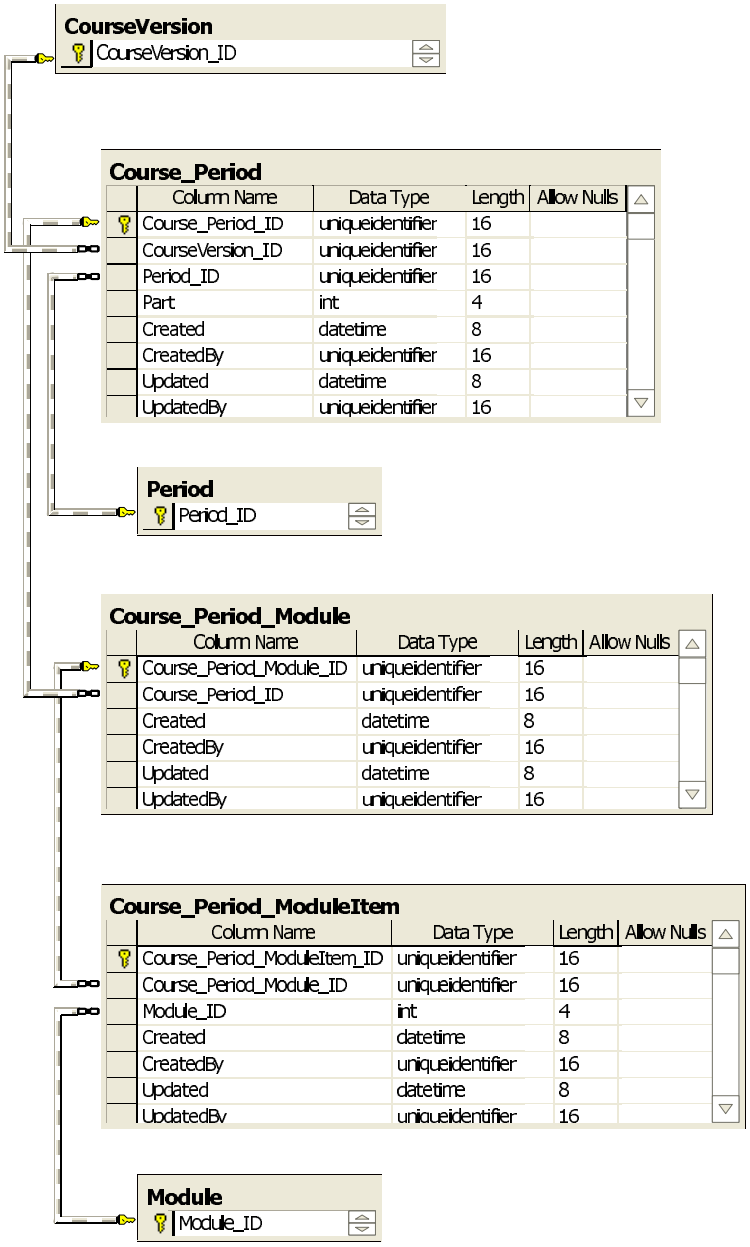
E.3.1 Course Diagram 1



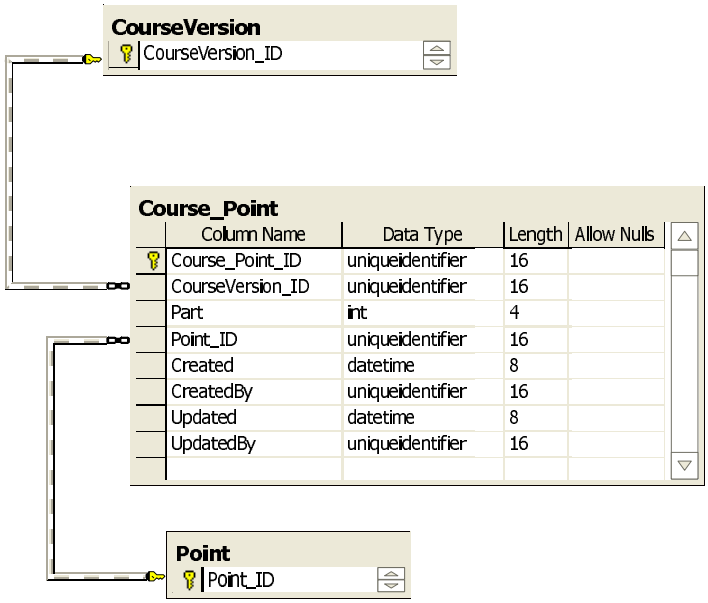
E.3.2 Course Diagram 2



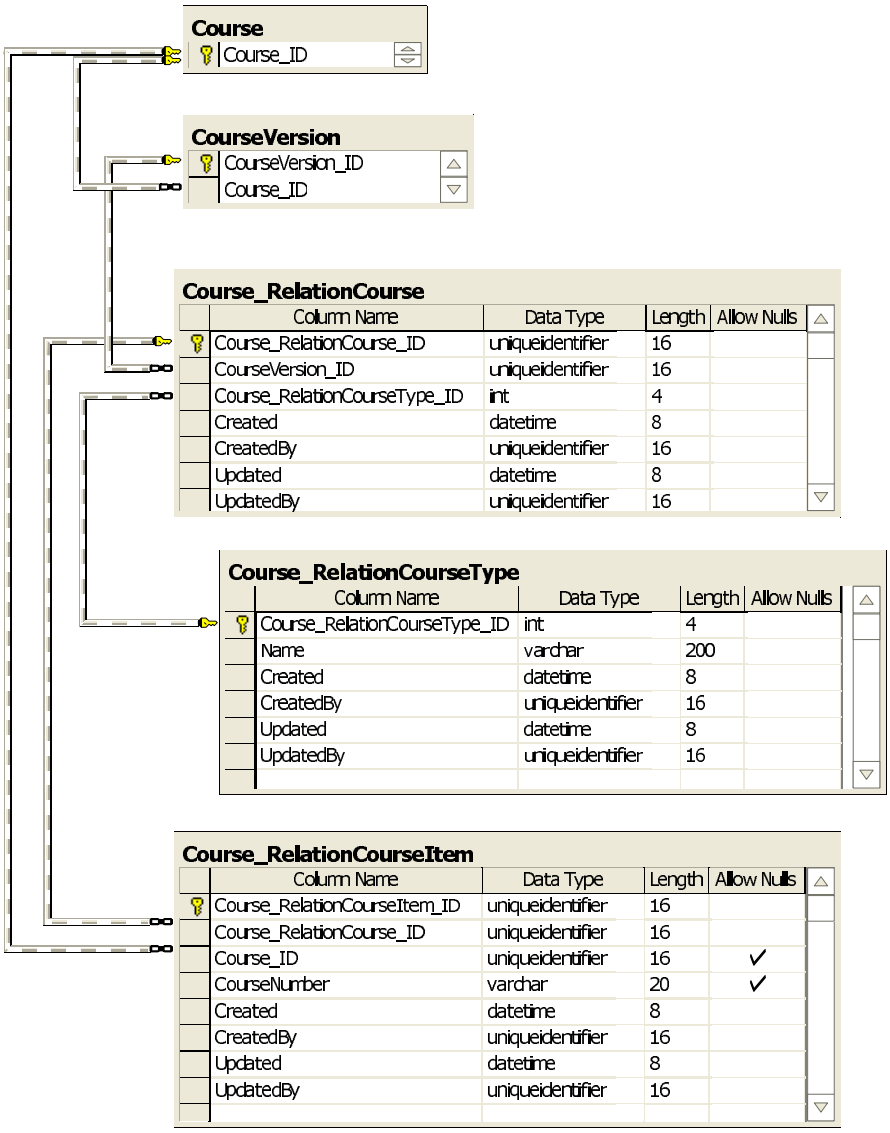
E.3.3 Course Diagram 3



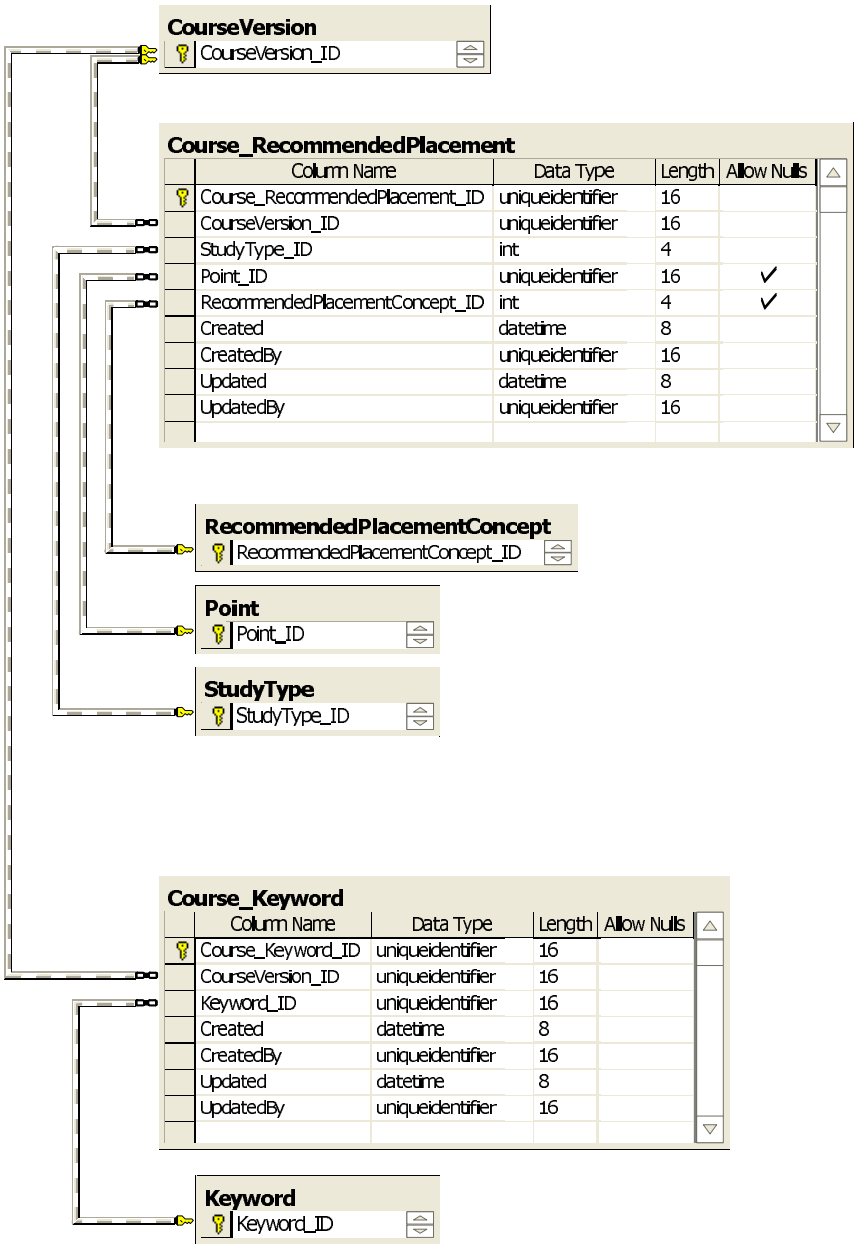
E.3.4 Course Diagram 4



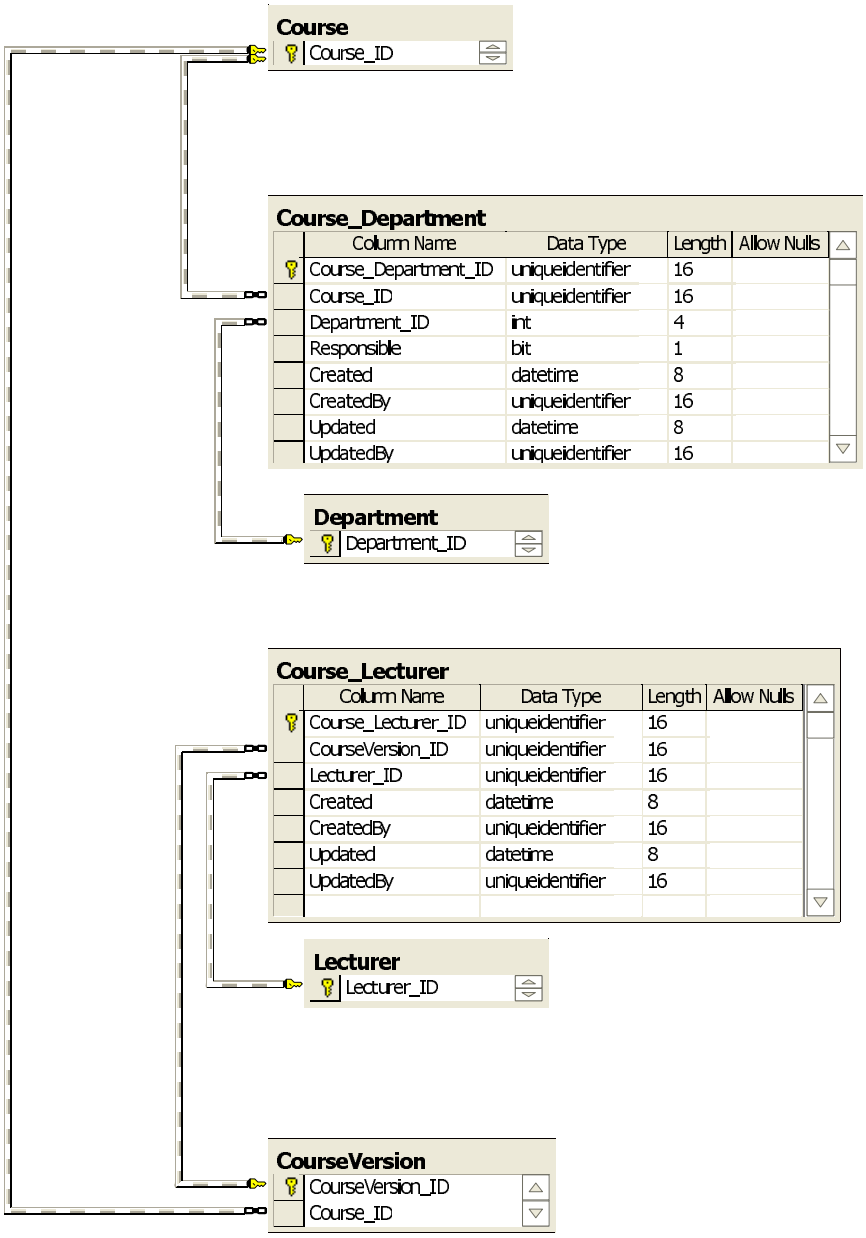
E.3.5 Course Diagram 5



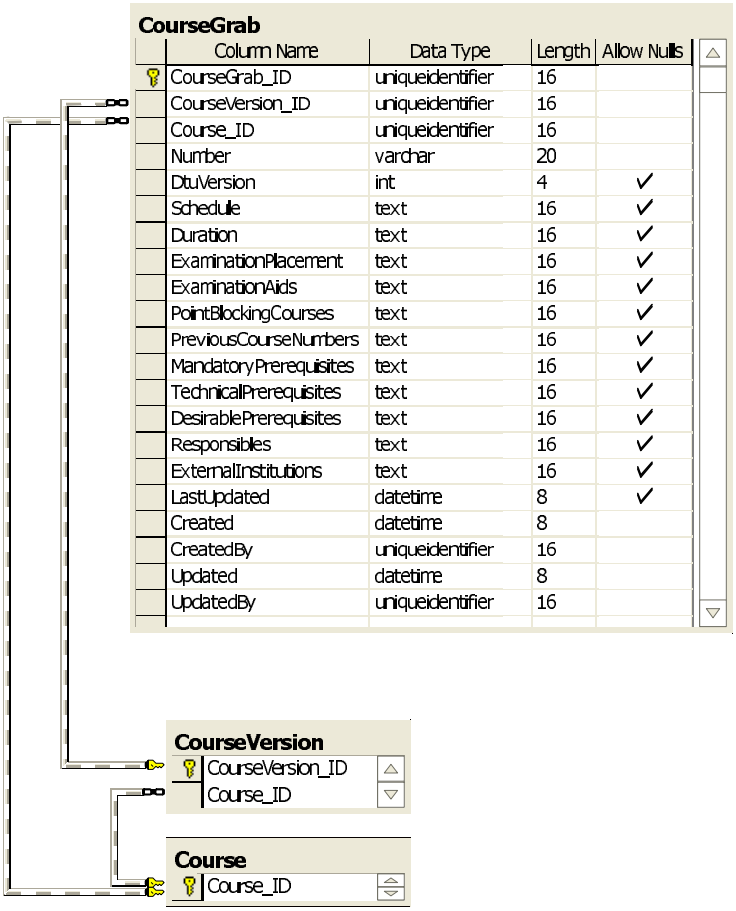
E.3.6 Course Diagram 6



E.3.7 Course Diagram 7



E.4 CourseGrab

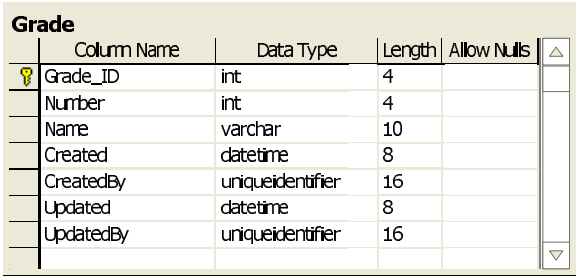


E.5 Department

Department

	Column Name	Data Type	Length	Allow Nulls
🔑	Department_ID	int	4	
	ContactData_ID	uniqueidentifier	16	✓
	Name	text	16	
	Number	int	4	
	Created	datetime	8	
	CreatedBy	uniqueidentifier	16	
	Updated	datetime	8	
	UpdatedBy	uniqueidentifier	16	

E.6 Grade

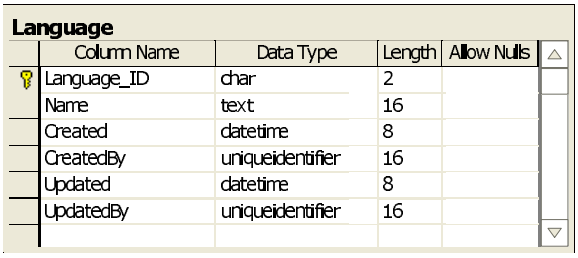

Grade
Column Name
Data Type
Length
Allow Nulls
Grade_ID
int
4
Number
int
4
Name
varchar
10
Created
datetime
8
CreatedBy
uniqueidentifier
16
Updated
datetime
8
UpdatedBy
uniqueidentifier
16

E.7 Keyword

Keyword

Column Name	Data Type	Length	Allow Nulls
Keyword_ID	uniqueidentifier	16	
Name	varchar	100	
Culture_ID	varchar	10	
Created	datetime	8	
CreatedBy	uniqueidentifier	16	
Updated	datetime	8	
UpdatedBy	uniqueidentifier	16	

E.8 Language



Language

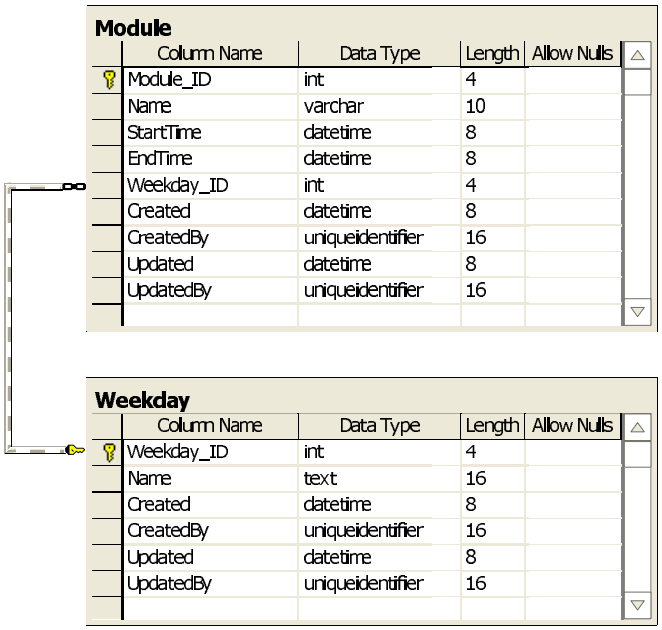
Column Name	Data Type	Length	Allow Nulls
Language_ID	char	2	
Name	text	16	
Created	datetime	8	
CreatedBy	uniqueidentifier	16	
Updated	datetime	8	
UpdatedBy	uniqueidentifier	16	

E.9 Lecturer

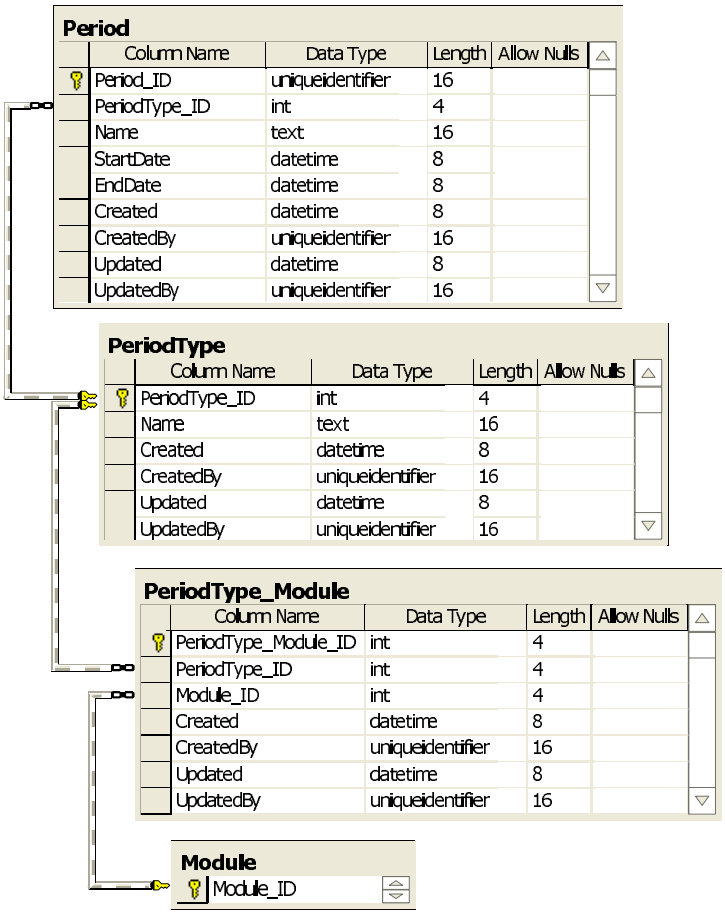
Lecturer

Column Name	Data Type	Length	Allow Nulls
Lecturer_ID	uniqueidentifier	16	
CwisNumber	int	4	✓
Person_ID	uniqueidentifier	16	
Created	datetime	8	
CreatedBy	uniqueidentifier	16	
Updated	datetime	8	
UpdatedBy	uniqueidentifier	16	

E.10 Module

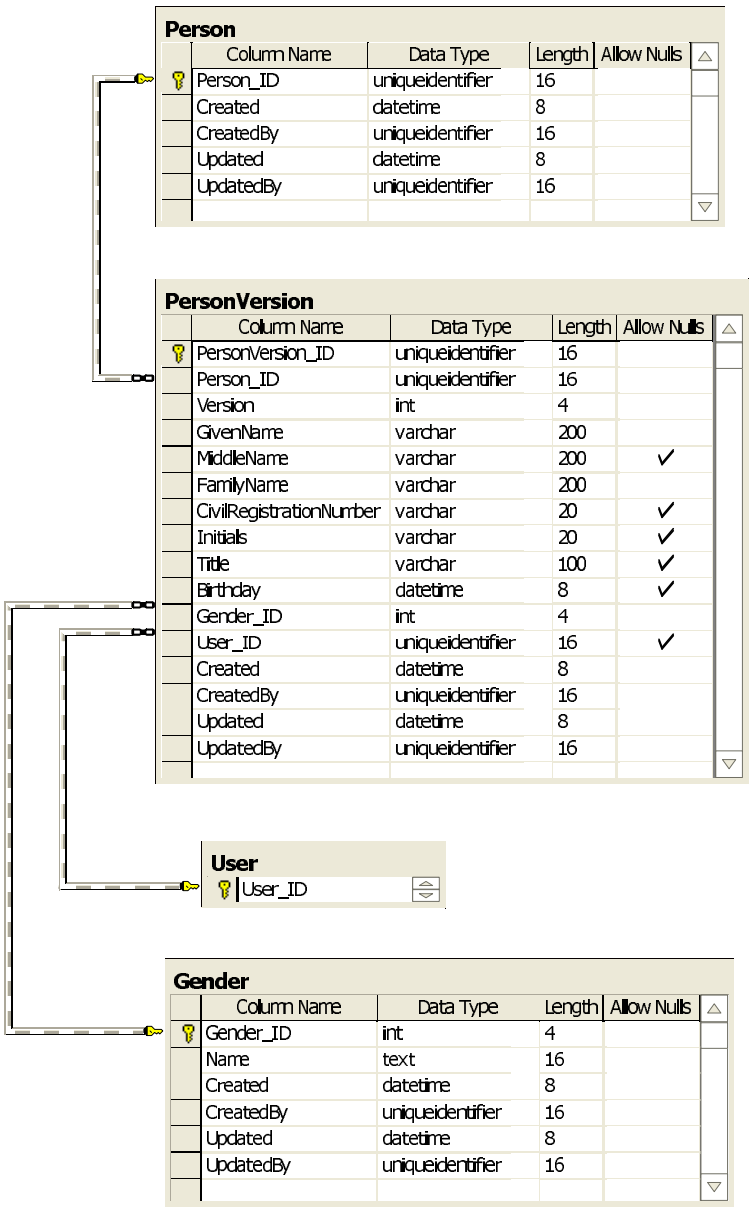

Module
Column Name
Data Type
Length
Allow Nulls
Module_ID
int
4
Name
varchar
10
StartTime
datetime
8
EndTime
datetime
8
Weekday_ID
int
4
Created
datetime
8
CreatedBy
uniqueidentifier
16
Updated
datetime
8
UpdatedBy
uniqueidentifier
16
Weekday
Column Name
Data Type
Length
Allow Nulls
Weekday_ID
int
4
Name
text
16
Created
datetime
8
CreatedBy
uniqueidentifier
16
Updated
datetime
8
UpdatedBy
uniqueidentifier
16

E.11 Period

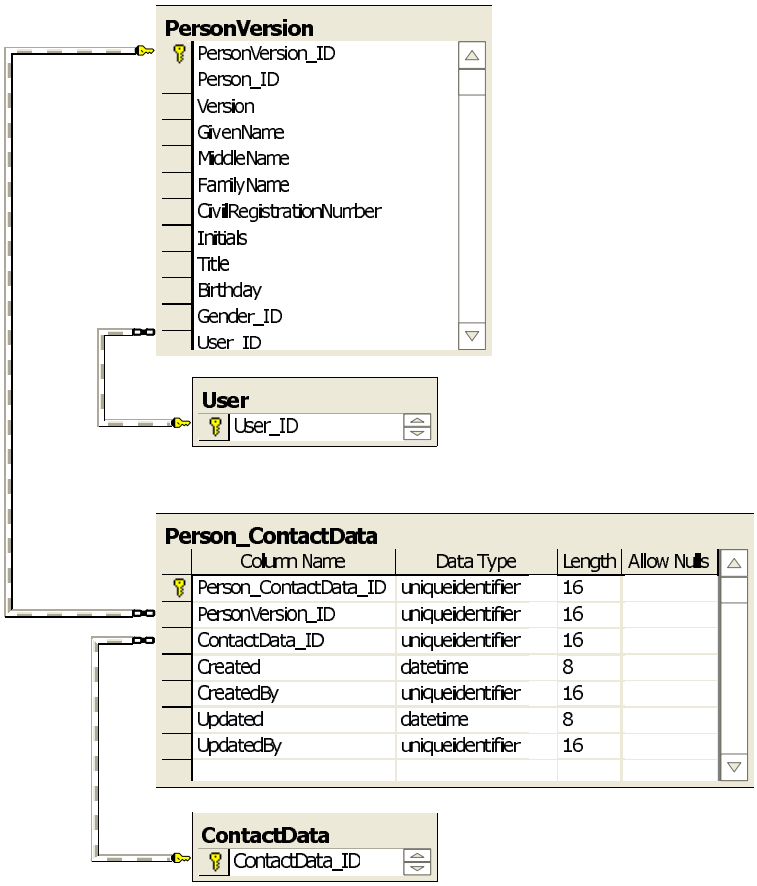

Period
Column Name
Data Type
Length
Allow Nulls
Period_ID
uniqueidentifier
16
PeriodType_ID
int
4
Name
text
16
StartDate
datetime
8
EndDate
datetime
8
Created
datetime
8
CreatedBy
uniqueidentifier
16
Updated
datetime
8
UpdatedBy
uniqueidentifier
16
PeriodType
Column Name
Data Type
Length
Allow Nulls
PeriodType_ID
int
4
Name
text
16
Created
datetime
8
CreatedBy
uniqueidentifier
16
Updated
datetime
8
UpdatedBy
uniqueidentifier
16
PeriodType_Module
Column Name
Data Type
Length
Allow Nulls
PeriodType_Module_ID
int
4
PeriodType_ID
int
4
Module_ID
int
4
Created
datetime
8
CreatedBy
uniqueidentifier
16
Updated
datetime
8
UpdatedBy
uniqueidentifier
16
Module
Module_ID

E.12 Person

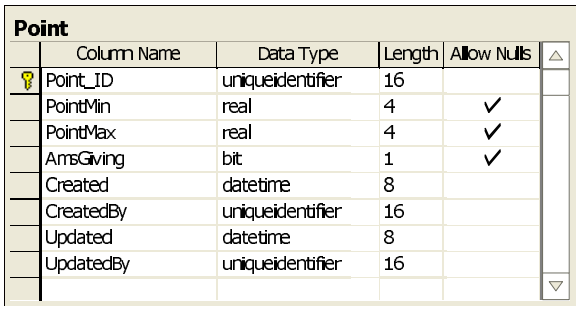
E.12.1 Person Diagram 1



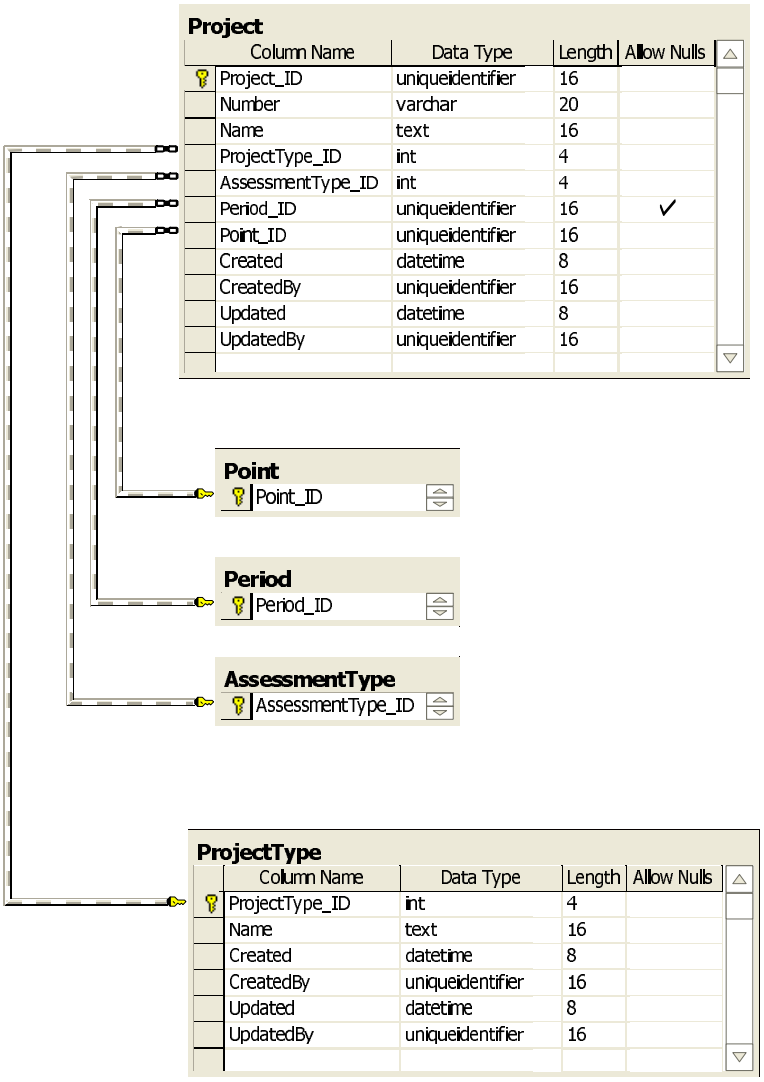
E.12.2 Person Diagram 2



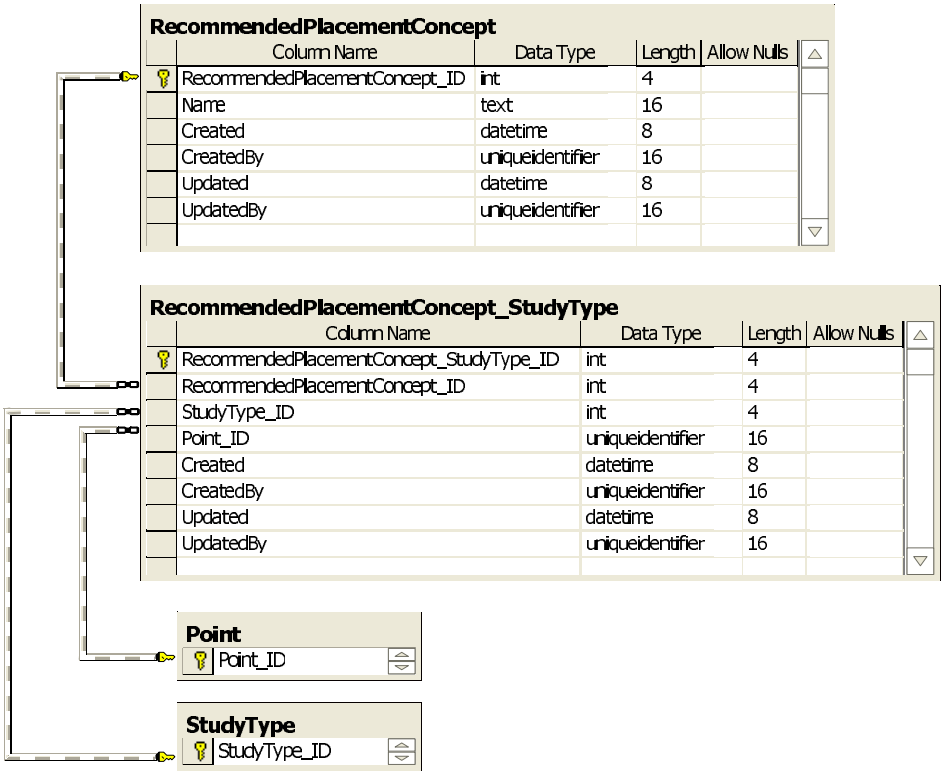
E.13 Point


Point
Column Name
Data Type
Length
Allow Nulls
Point_ID
uniqueidentifier
16
PointMin
real
4
✓
PointMax
real
4
✓
AmsGiving
bit
1
✓
Created
datetime
8
CreatedBy
uniqueidentifier
16
Updated
datetime
8
UpdatedBy
uniqueidentifier
16

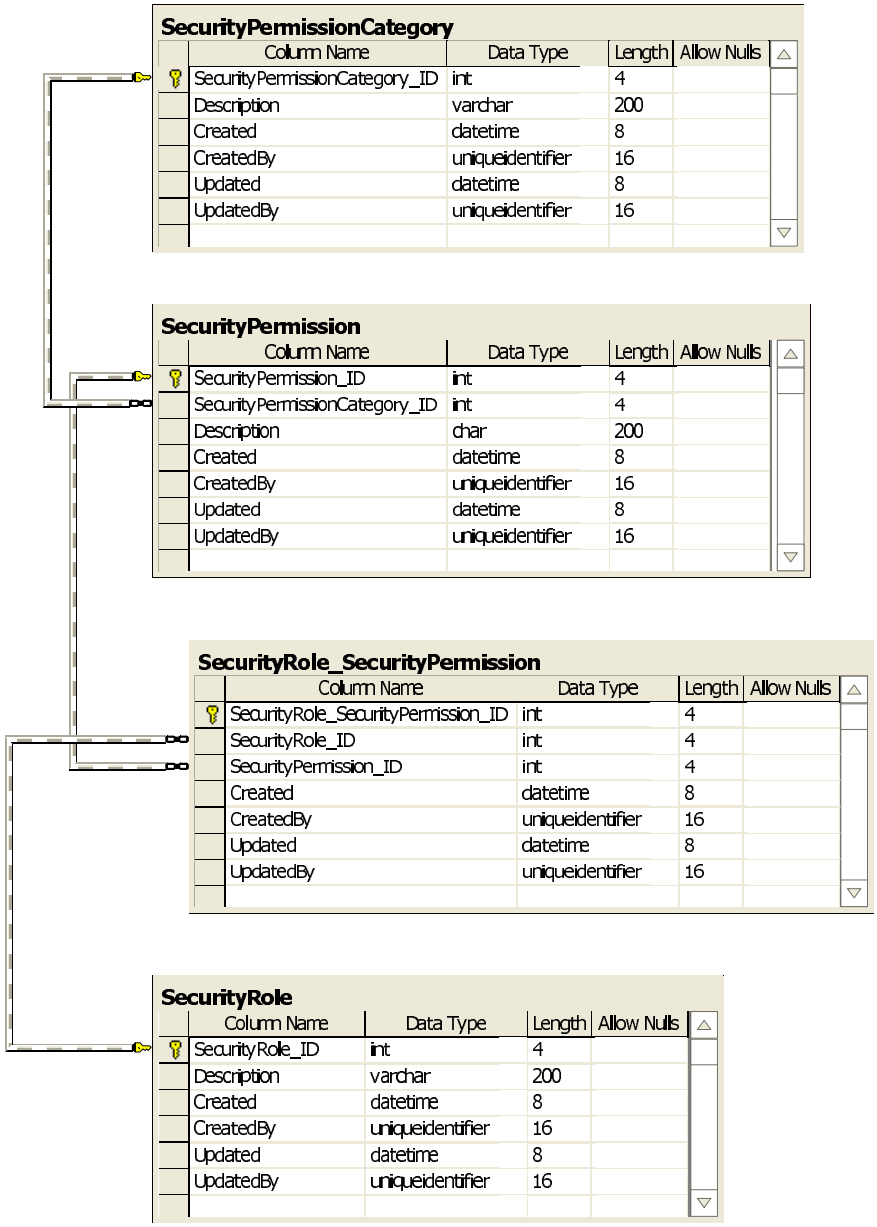
E.14 Project


Project
Column Name
Data Type
Length
Allow Nulls
Project_ID
uniqueidentifier
16
Number
varchar
20
Name
text
16
ProjectType_ID
int
4
AssessmentType_ID
int
4
Period_ID
uniqueidentifier
16
✓
Point_ID
uniqueidentifier
16
Created
datetime
8
CreatedBy
uniqueidentifier
16
Updated
datetime
8
UpdatedBy
uniqueidentifier
16
Point
Point_ID
Period
Period_ID
AssessmentType
AssessmentType_ID
ProjectType
Column Name
Data Type
Length
Allow Nulls
ProjectType_ID
int
4
Name
text
16
Created
datetime
8
CreatedBy
uniqueidentifier
16
Updated
datetime
8
UpdatedBy
uniqueidentifier
16

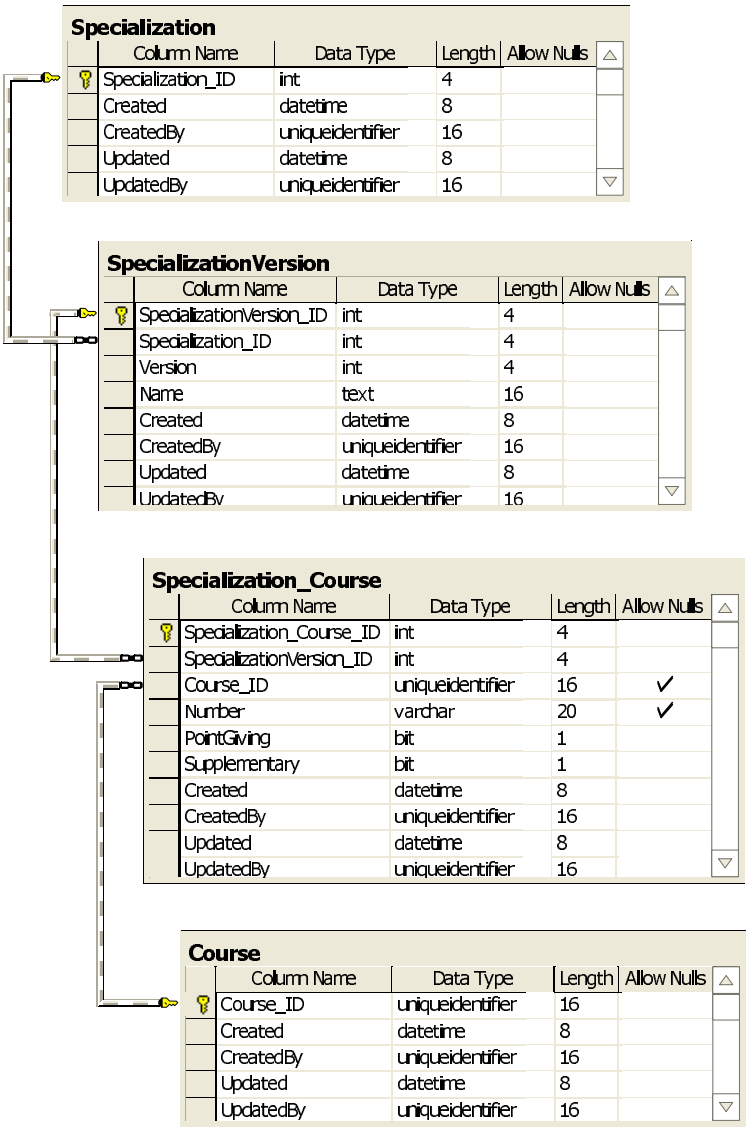
E.15 RecommendedPlacementConcept



E.16 Security

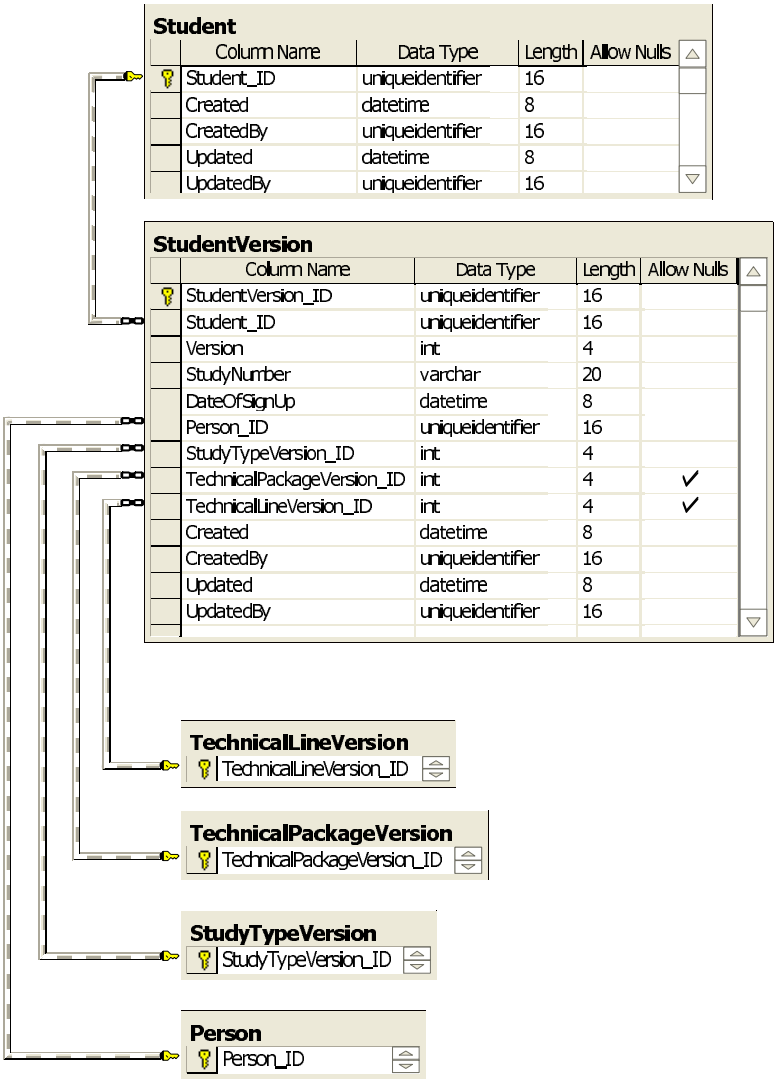


E.17 Specialization

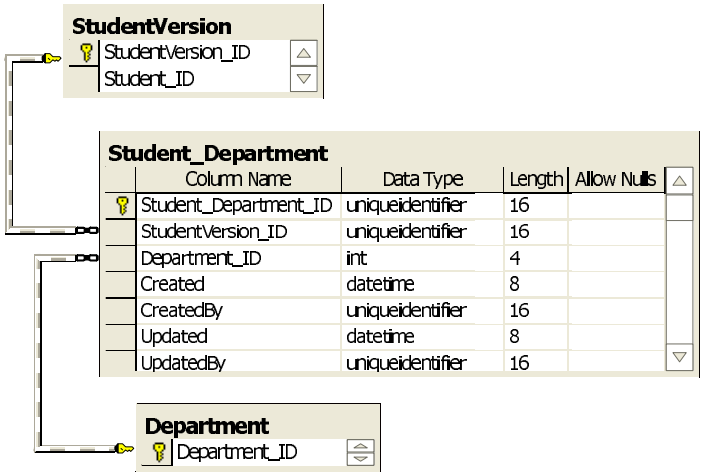


E.18 Student

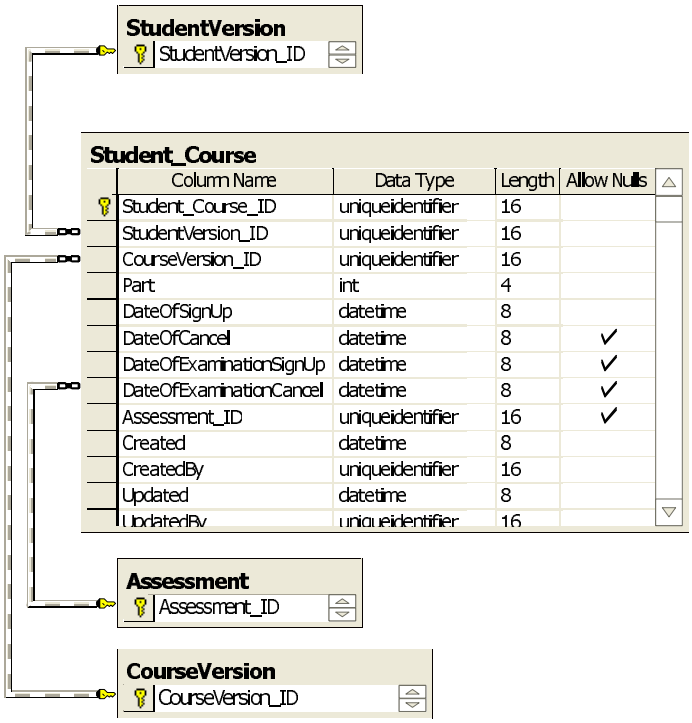
E.18.1 Student Diagram 1



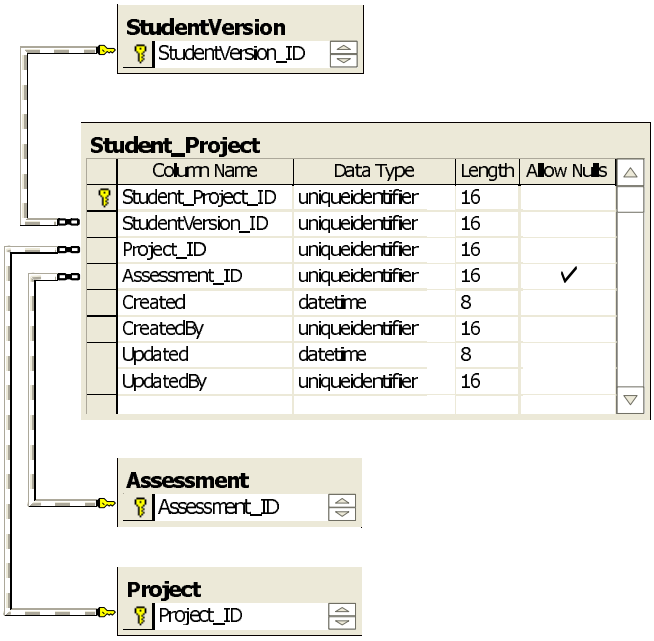
E.18.2 Student Diagram 2



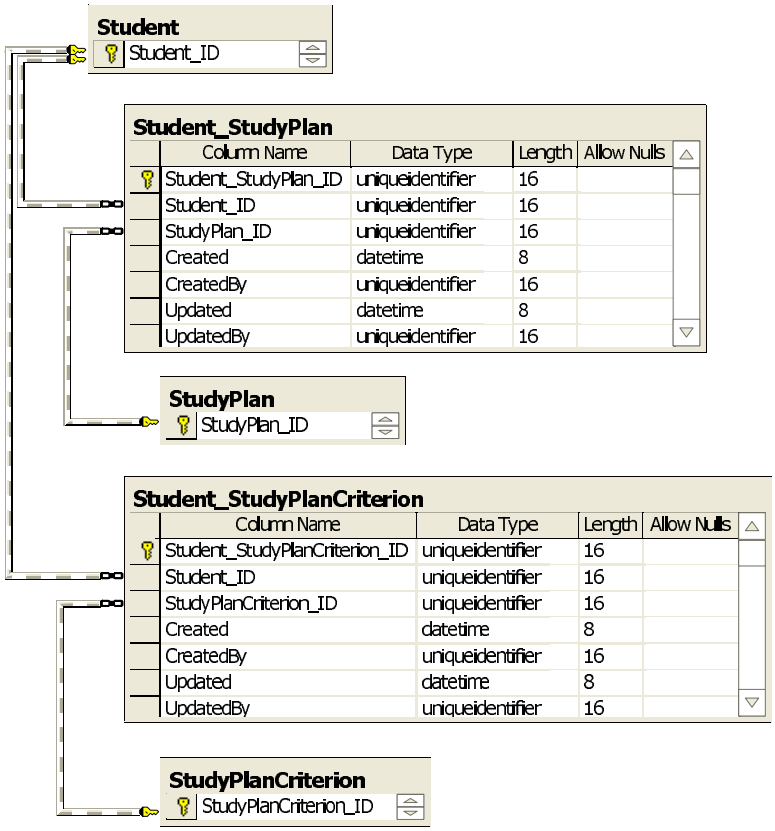
E.18.3 Student Diagram 3



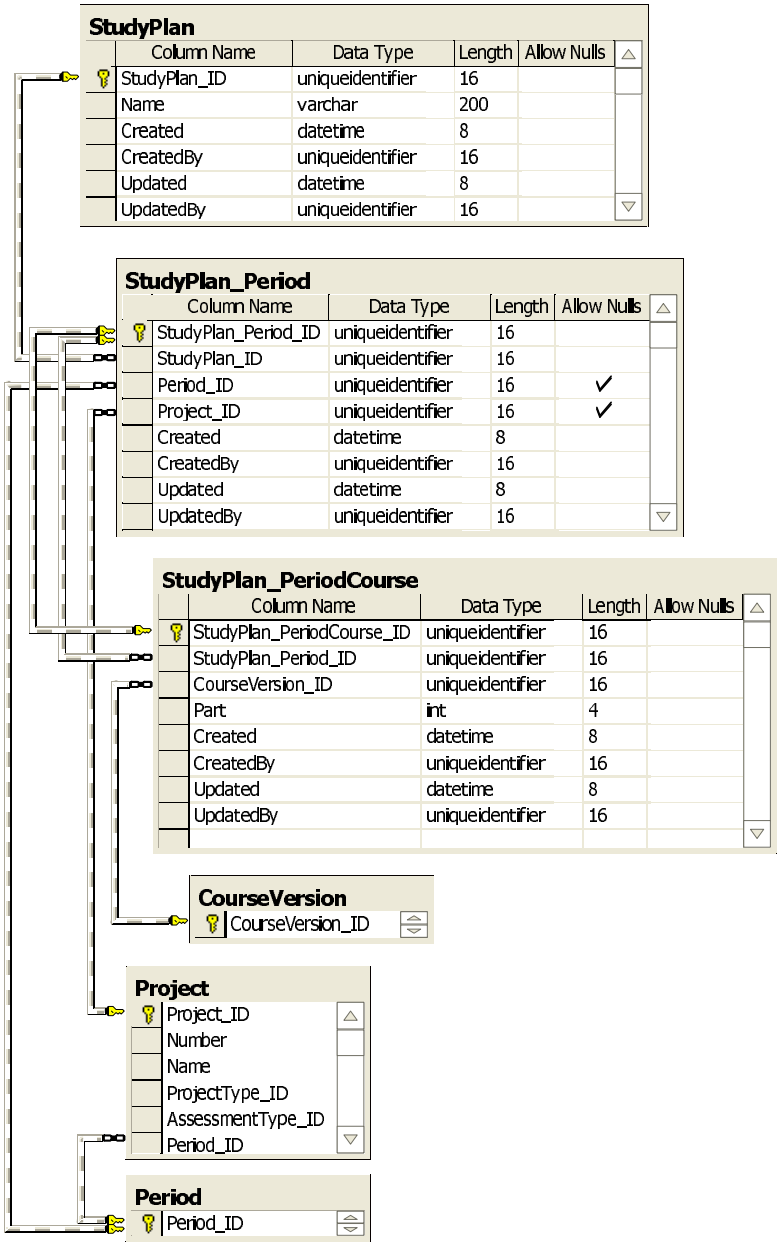
E.18.4 Student Diagram 4



E.18.5 Student Diagram 5

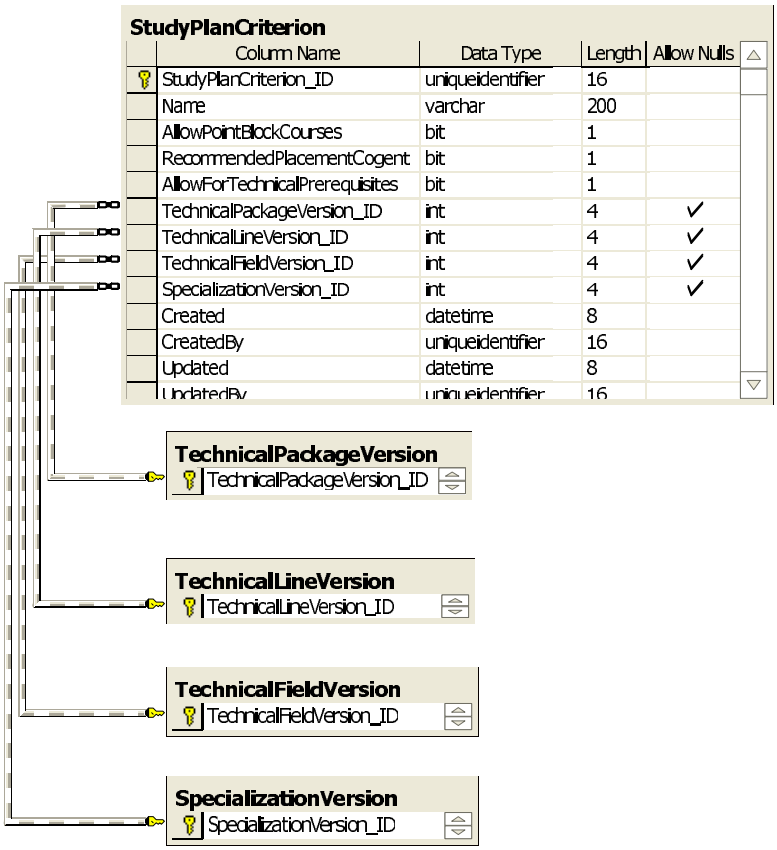


E.19 StudyPlan

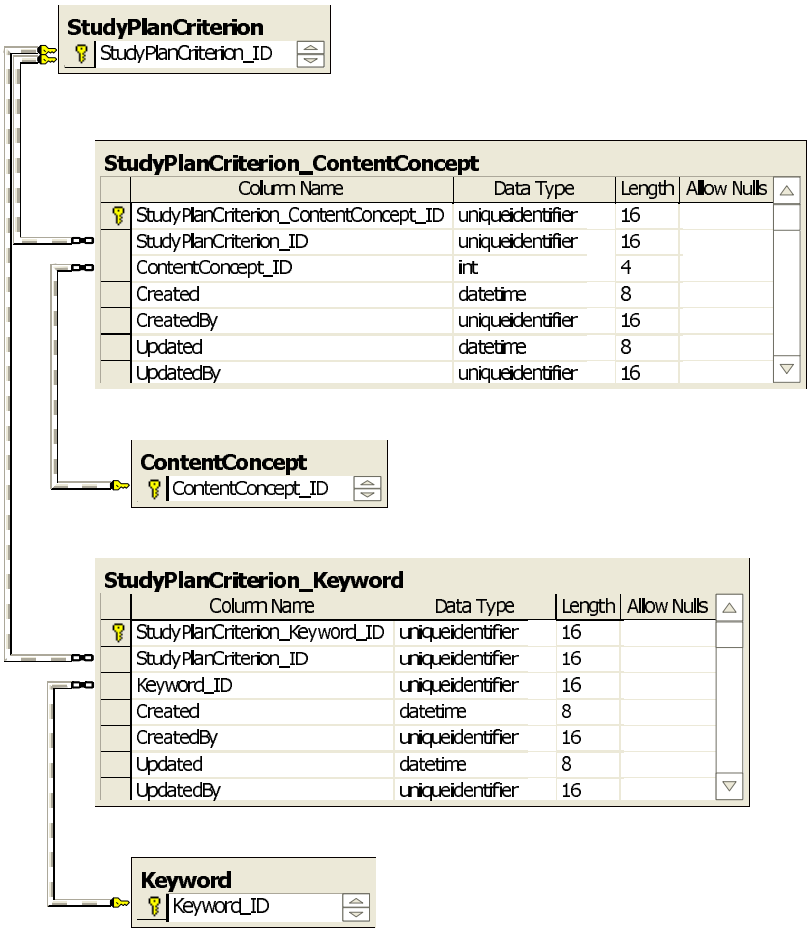


E.20 StudyPlanCriterion

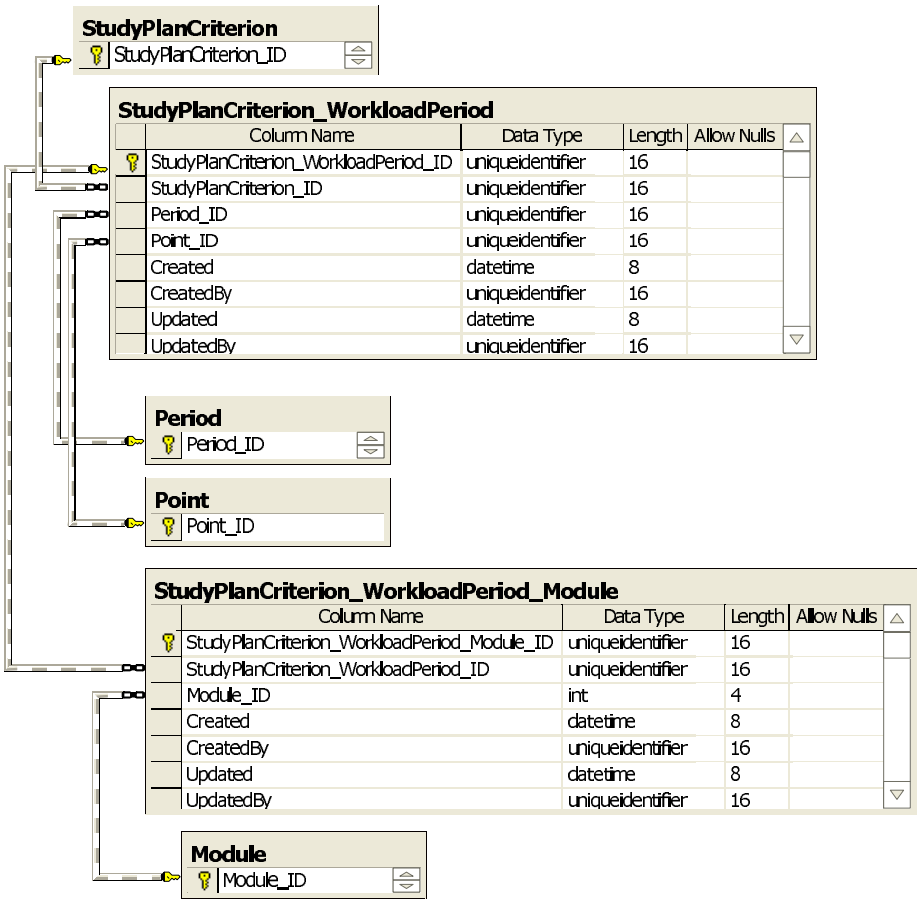
E.20.1 StudyPlanCriterion Diagram 1



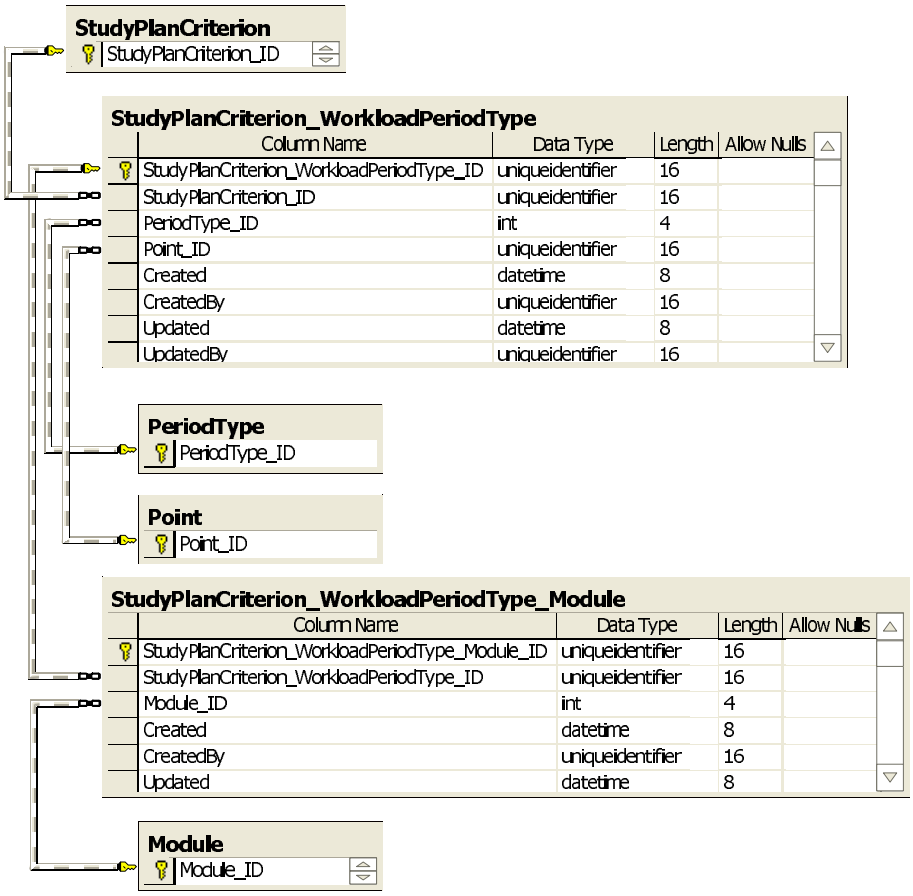
E.20.2 StudyPlanCriterion Diagram 2



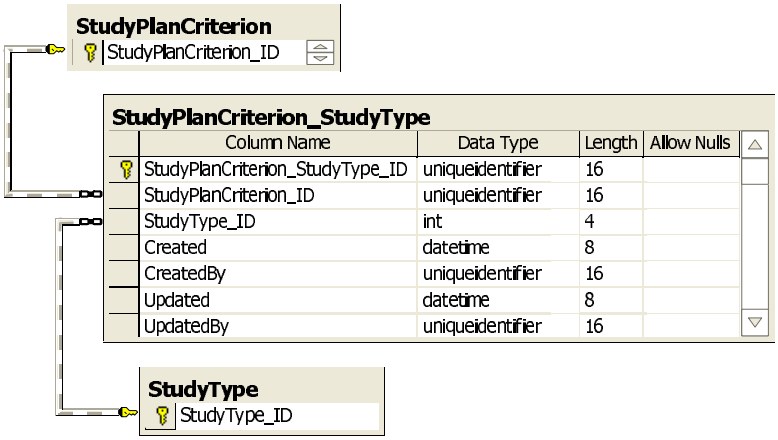
E.20.3 StudyPlanCriterion Diagram 3



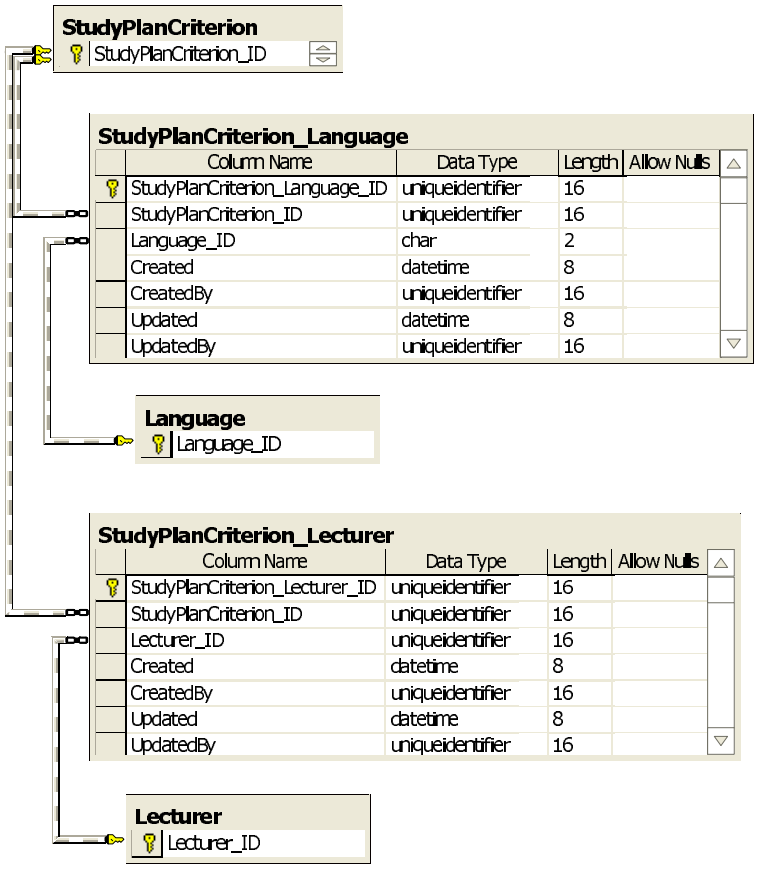
E.20.4 StudyPlanCriterion Diagram 4



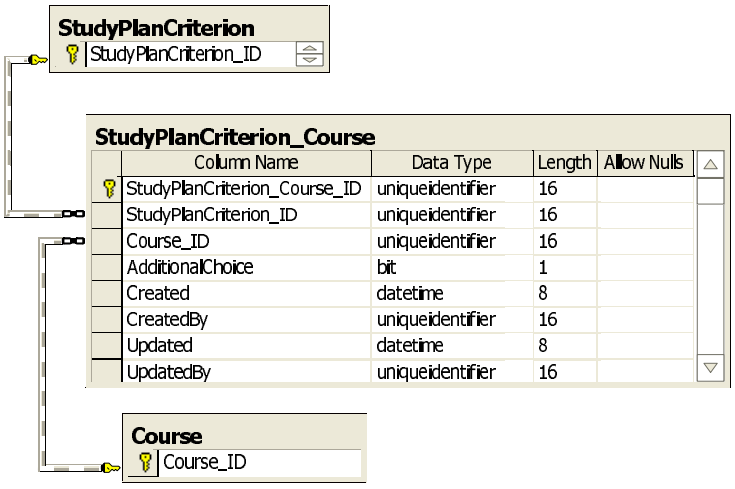
E.20.5 StudyPlanCriterion Diagram 5



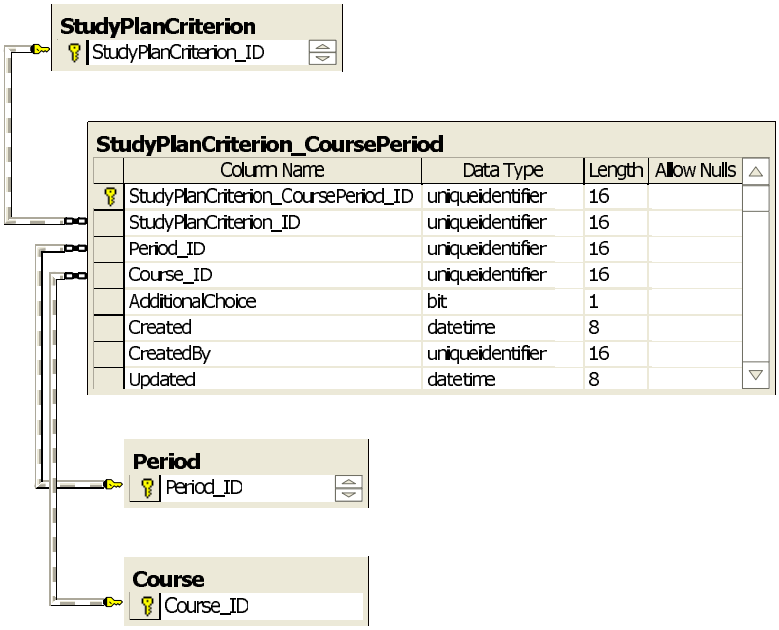
E.20.6 StudyPlanCriterion Diagram 6



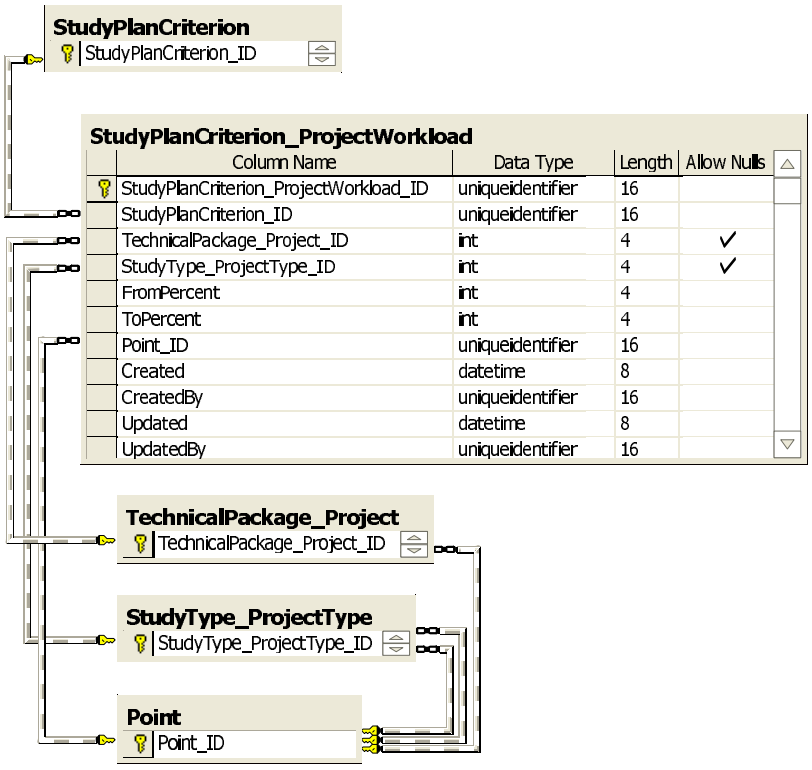
E.20.7 StudyPlanCriterion Diagram 7



E.20.8 StudyPlanCriterion Diagram 8

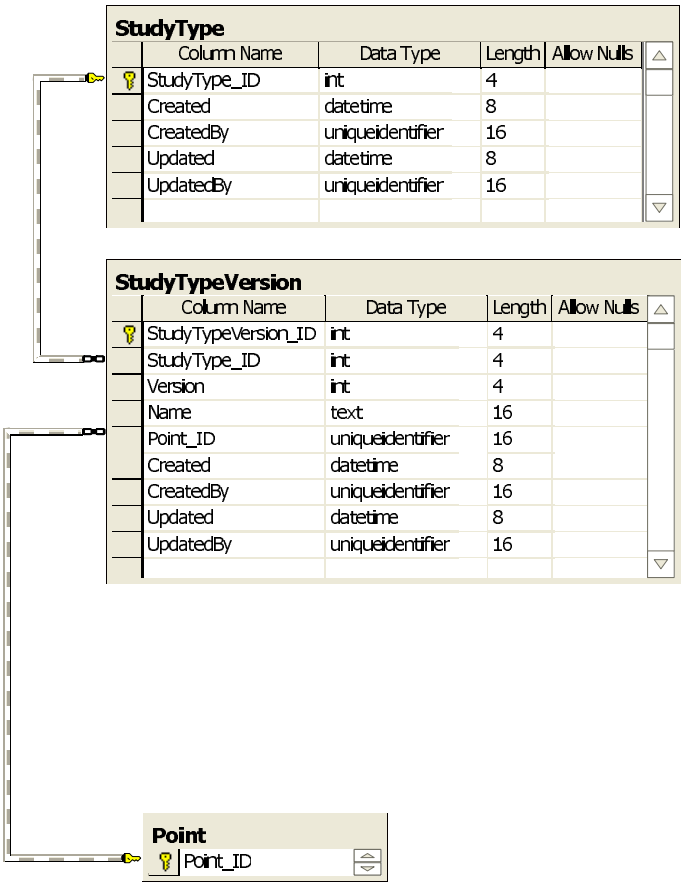


E.20.9 StudyPlanCriterion Diagram 9

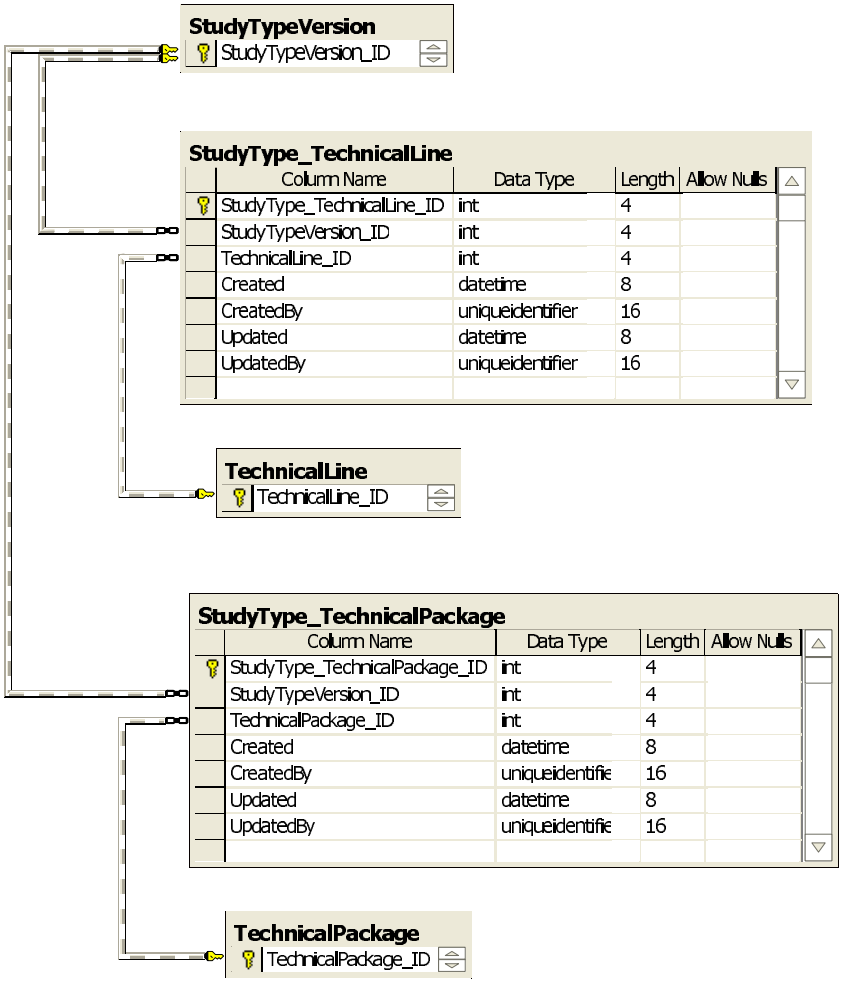


E.21 StudyType

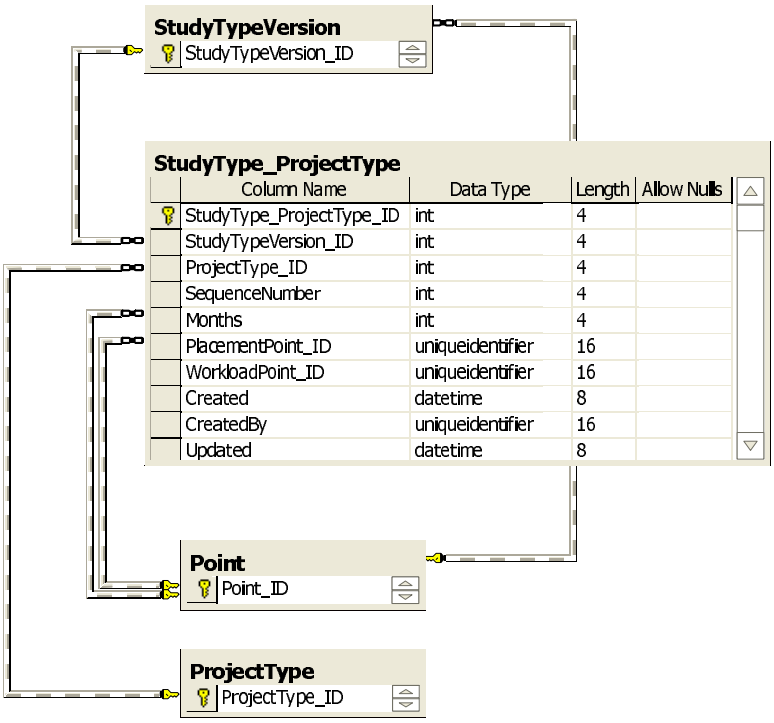
E.21.1 StudyType Diagram 1



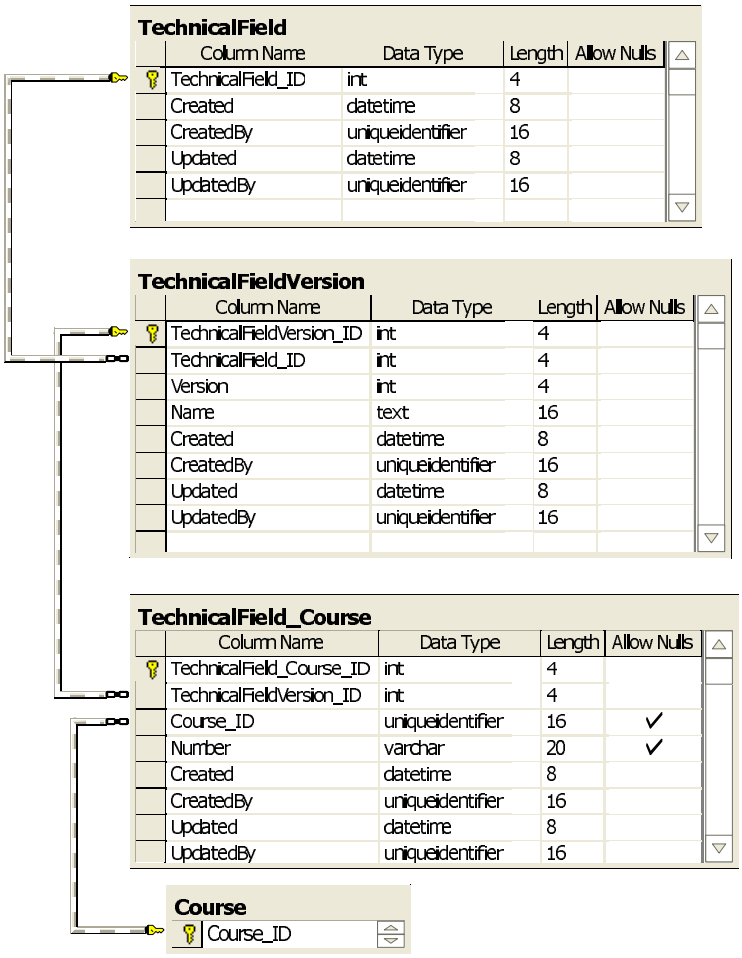
E.21.2 StudyType Diagram 2



E.21.3 StudyType Diagram 3

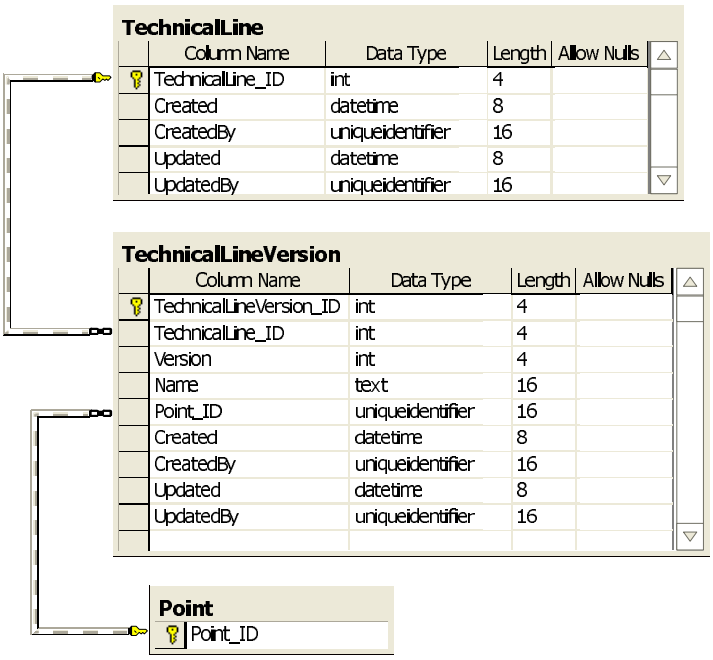


E.22 TechnicalField

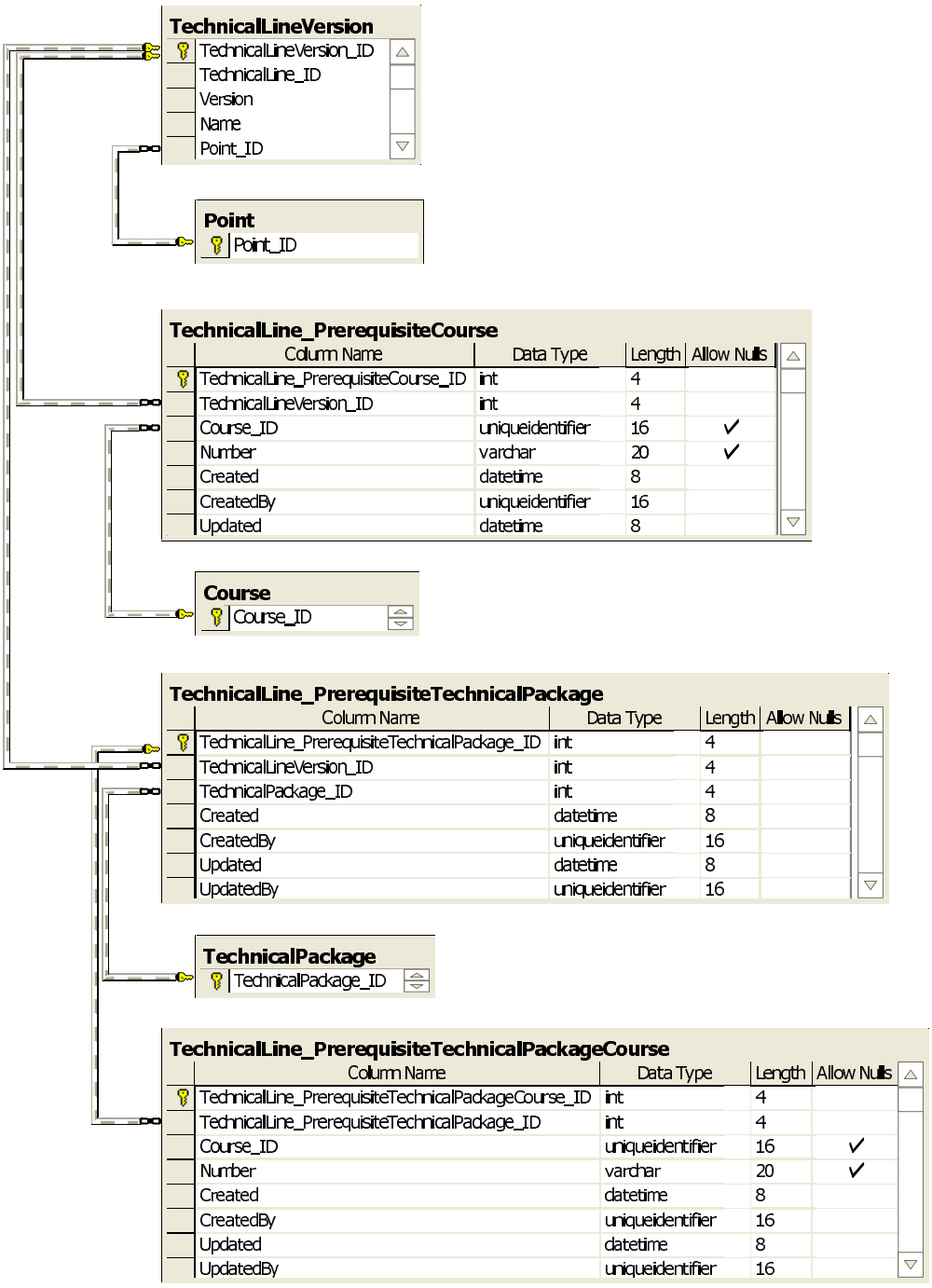


E.23 TechnicalLine

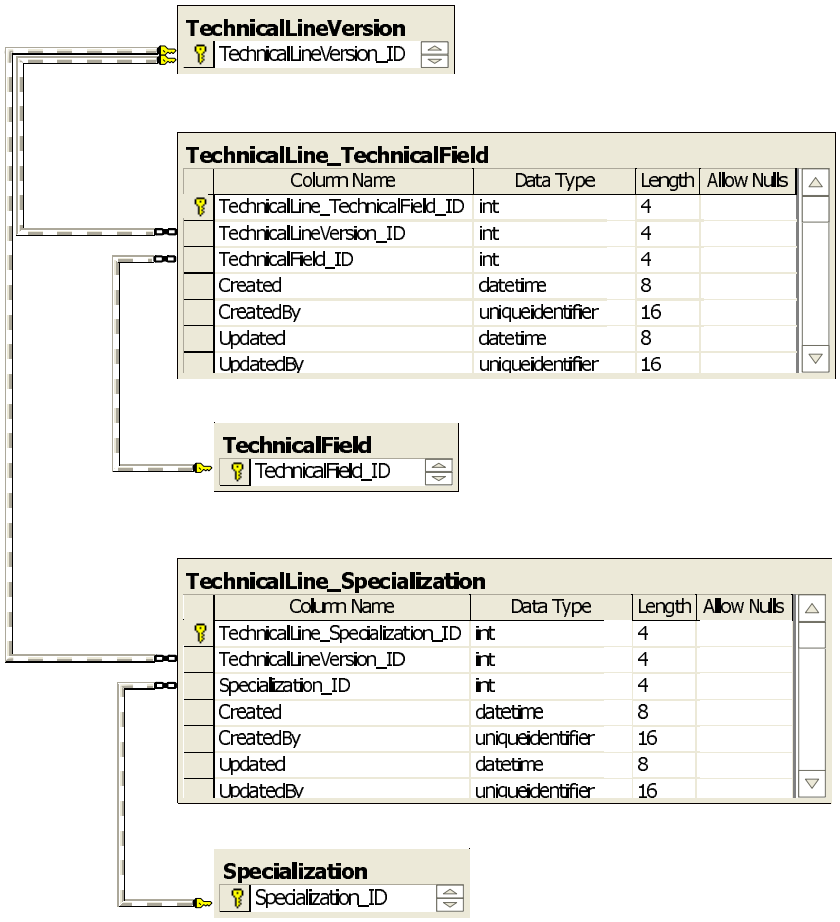
E.23.1 TechnicalLine Diagram 1



E.23.2 TechnicalLine Diagram 2

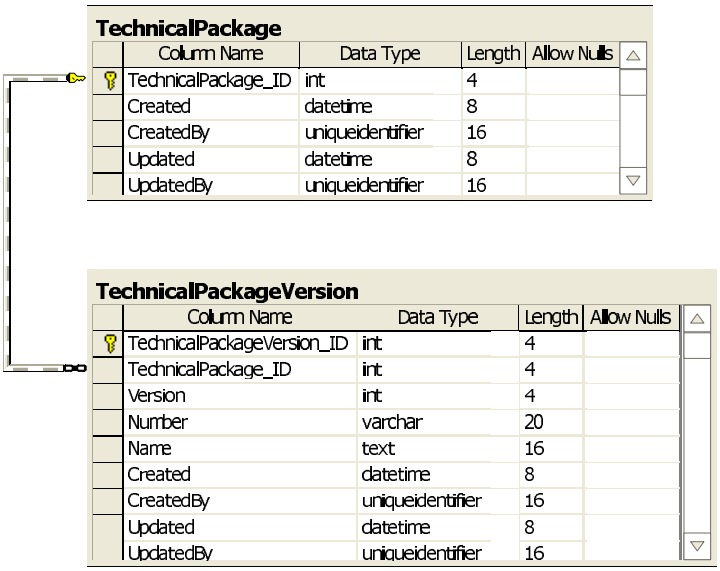


E.23.3 TechnicalLine Diagram 3

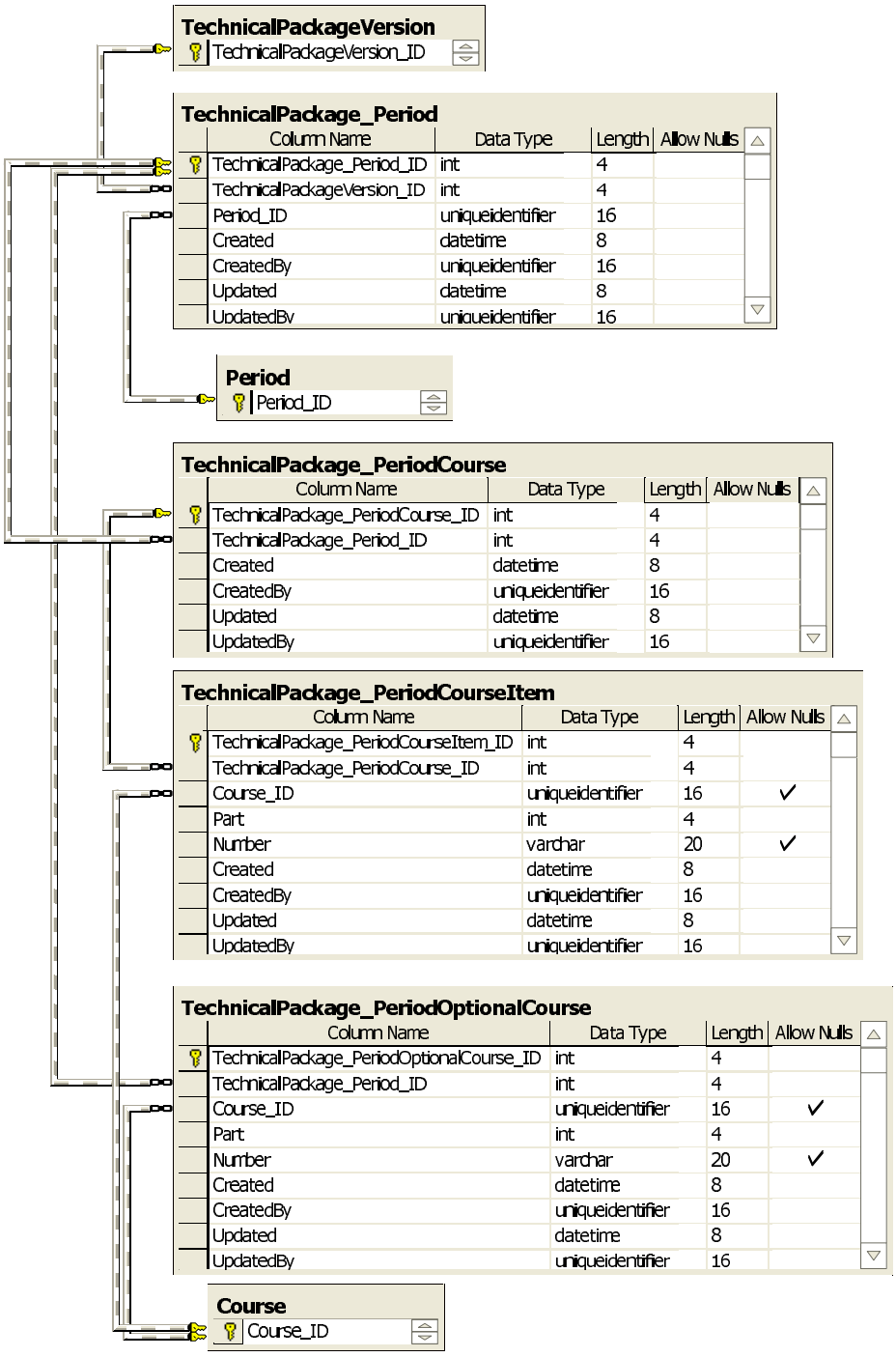


E.24 TechnicalPackage

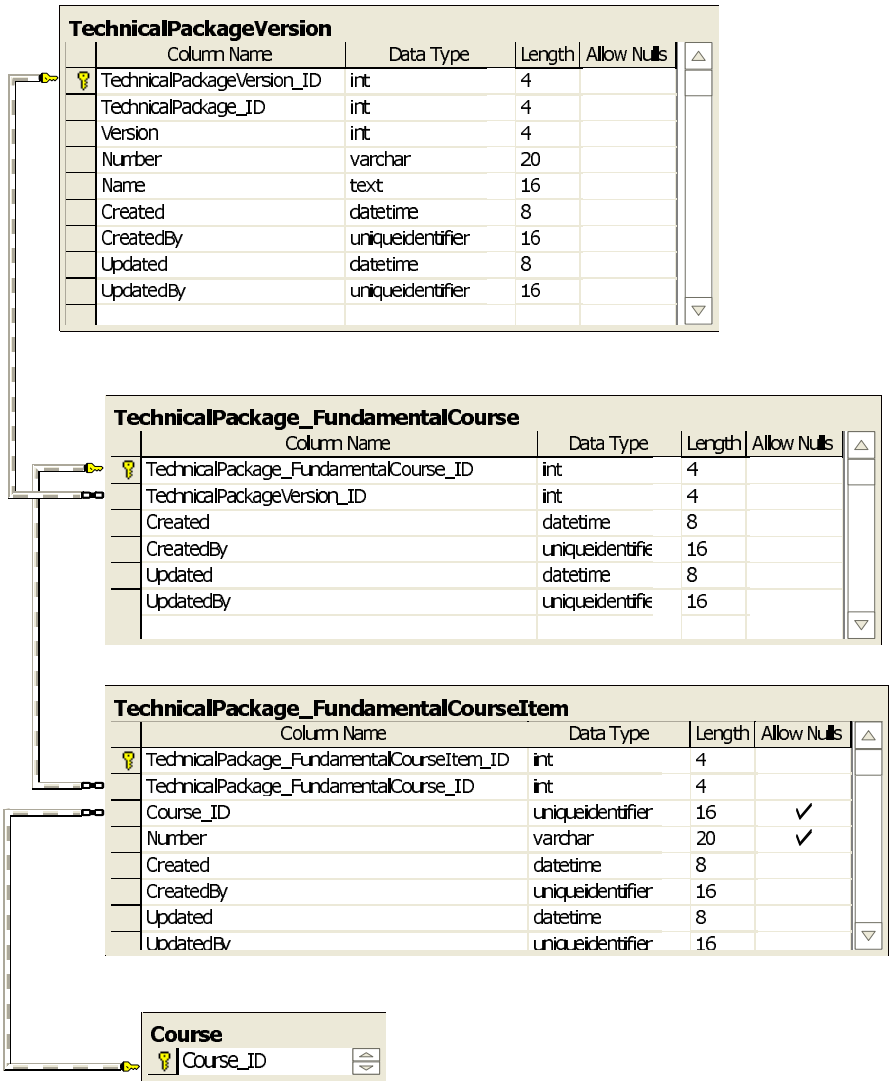
E.24.1 TechnicalPackage Diagram 1



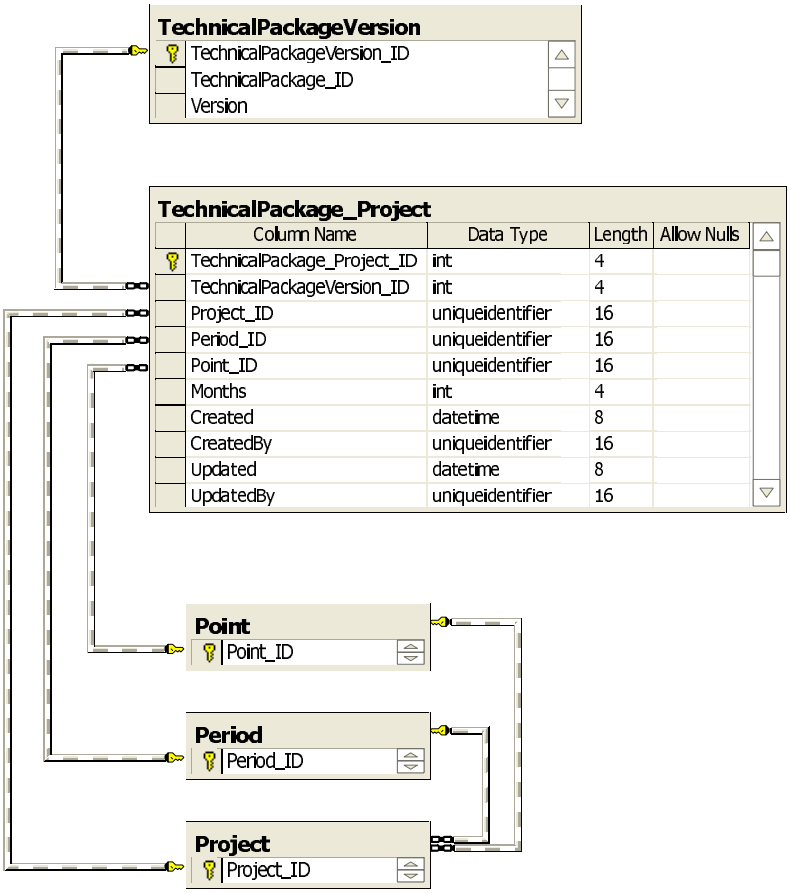
E.24.2 TechnicalPackage Diagram 2



E.24.3 TechnicalPackage Diagram 3

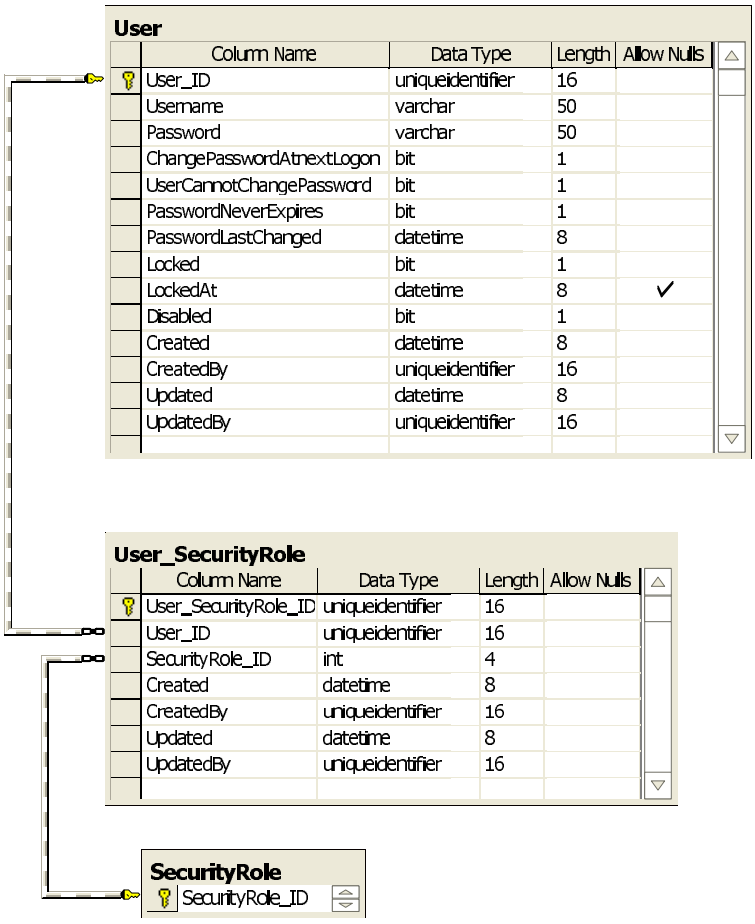


E.24.4 TechnicalPackage Diagram 4

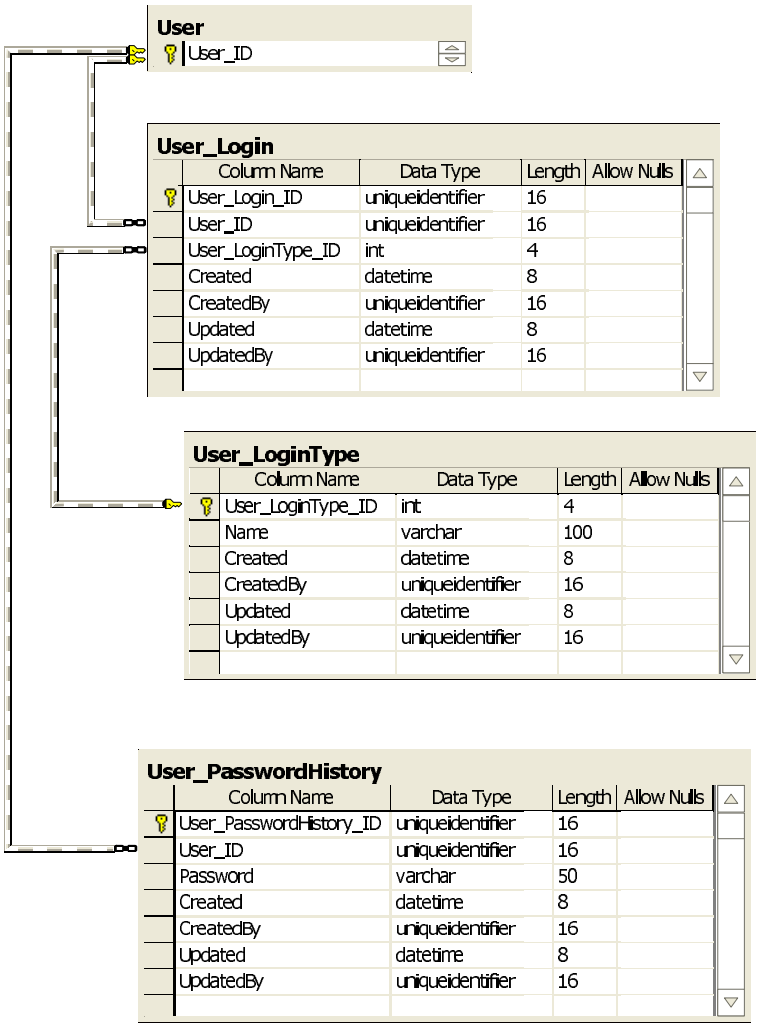


E.25 User

E.25.1 User Diagram 1



E.25.2 User Diagram 2



Source Code

Database

A STUDY PLANNING SYSTEM

VOLUME II

SOURCE CODE

Morten Milling Jensen
Teddy Kaarløv Nielsen

LYNGBY 2003
EKSAMENSPROJEKT
NR. 70/03

IMM

Trykt af IMM, DTU

Contents

Chapter 1

Data Tier

1.1 DalTypes

1.1.1 DalBool

```
using System;

namespace StudyPlanning.DAL
{
  /// <summary>
  /// Represents a Boolean value in the data access layer.
  /// </summary>
  public sealed class DalBool : DalType
  {
    private bool _Value;

    /// <summary>
    /// Initializes a new instance of the <see cref="StudyPlanning.DAL.DalBool">DalBool</see> class
    /// with the specified value of the Boolean.
    /// </summary>
    /// <param name="blnValue">The value of the Boolean.</param>
    /// <param name="allowNull">Indicates whether the value is allowed to be null.<br/>
    /// <strong>true</strong>, if the value is allowed to be null; otherwise, <strong>false</strong>.
    /// </param>
    public DalBool(bool blnValue, bool allowNull)
    {
      this.Value = blnValue;
      this._allowNull = allowNull;
    }

    /// <summary>
    /// Initializes a new instance of the <see cref="StudyPlanning.DAL.DalBool">DalBool</see> class.
    /// </summary>
    /// <param name="allowNull">Indicates whether the value is allowed to be null.<br/>
    /// <strong>true</strong>, if the value is allowed to be null; otherwise, <strong>false</strong>.
    /// </param>
    public DalBool(bool allowNull)
    {
      this._allowNull = allowNull;
    }

    /// <summary>
    /// Gets or sets the value of the integer.
    /// </summary>
    public bool Value
    {
      get { return this._Value; }
      set
      {
        this._Value = value;
        this._isNull = false;
      }
    }
  }
}
```

1.1.2 DalDateTime

```
using System;

namespace StudyPlanning.DAL
{
  /// <summary>
  /// Represents an instant in time in the data access layer,
  /// typically expressed as a date and time of day.
  /// </summary>
  public sealed class DalDateTime : DalType
  {
    private DateTime _Value;

    /// <summary>
    /// Initializes a new instance of the <see cref="StudyPlanning.DAL.DalDateTime">DalDateTime</see>
    /// class with the specified <see cref="System.DateTime">DateTime</see> object.
    /// </summary>
    /// <param name="dateTime">The <see cref="System.DateTime">DateTime</see> object.</param>
    /// <param name="allowNull">Indicates whether the value is allowed to be null.<br/>
    /// <strong>true</strong>, if the value is allowed to be null; otherwise, <strong>false</strong>.
    /// </param>
    public DalDateTime(DateTime dateTime, bool allowNull)
    {
      this.Value = dateTime;
      this._allowNull = allowNull;
    }

    /// <summary>
    /// Initializes a new instance of the <see cref="StudyPlanning.DAL.DalDateTime">DalDateTime</see>
    /// class.
    /// </summary>
    /// <param name="allowNull">Indicates whether the value is allowed to be null.<br/>
    /// <strong>true</strong>, if the value is allowed to be null; otherwise, <strong>false</strong>.
    /// </param>
    public DalDateTime(bool allowNull)
    {
      this._allowNull = allowNull;
    }

    /// <summary>
    /// Gets or sets the value of the instant in time.
    /// </summary>
    public DateTime Value
    {
      get { return this._Value; }
      set
      {
        this._Value = value;
        this._isNull = false;
      }
    }
  }
}
```

1.1.3 DalException

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Xml;

namespace StudyPlanning.DAL
{
  /// <summary>
  /// Summary description for DalException.
  /// </summary>
  public class DalException : System.SystemException
  {
    #region Private Properties
    private int _Number;
    private string _Message;
    private string _PropertyName;
    private SqlException _Sql;
    private XmlException _Xml;
    #endregion //Private Properties

    #region Public Properties

    /// <summary>
    /// Gets or sets a number that identifies the type of error.
    /// </summary>
    public int Number
    {
      get { return _Number; }
      set { _Number = value; }
    }

    /// <summary>
    /// Gets the text describing the error.
    /// </summary>
    /// <remarks>Invoke the <strong>RetrieveMessage</strong> method in order
    /// to set the <strong>Message</strong> property.</remarks>
    public override string Message
    {
      get { return _Message; }
    }
```

```
/// <summary>
/// Gets or sets a string containing the name of the property that caused
/// the exception to be thrown.
/// </summary>
public string PropertyName
{
  get { return _PropertyName; }
  set { _PropertyName = value; }
}

/// <summary>
/// Gets or sets the <see cref="System.Data.SqlClient.SqlException">
/// SqlException</see> that caused the exception to be thrown.
/// </summary>
public SqlException Sql
{
  get { return _Sql; }
  set { _Sql = value; }
}

#endregion

#region Constructors

/// <summary>
/// Initializes a new instance of the <see cref="StudyPlanning.DAL.DalException">DalException</see>
/// class.
/// </summary>
public DalException() {}

/// <summary>
/// Initializes a new instance of the <see cref="StudyPlanning.DAL.DalException">DalException</see>
/// class with the specified error number.
/// </summary>
/// <param name="errorNumber">The number that identifies the type of error.</param>
public DalException(int errorNumber)
{
  _Number = errorNumber;
  RetrieveMessage();
  _Sql = null; //new SqlException();
}

/// <summary>
/// Initializes a new instance of the <see cref="StudyPlanning.DAL.DalException">DalException</see>
/// class with the specified error number and attribute name.
/// </summary>
/// <param name="errorNumber">The number that identifies the type of error.</param>
/// <param name="propertyName">The name of the property that causes the error.</param>
public DalException(int errorNumber, string propertyName)
{
  _Number = errorNumber;
  _PropertyName = propertyName;
  RetrieveMessage();
  _Sql = null; //new SqlException();
}

/// <summary>
/// Initializes a new instance of the <see cref="StudyPlanning.DAL.DalException" />
/// class with the specified <see cref="System.Data.SqlClient.SqlException" /> object.
/// </summary>
/// <param name="objException">The <see cref="System.Data.SqlClient.SqlException" />
/// object which has caused the <see cref="StudyPlanning.DAL.DalException" /> to
/// be thrown.
/// </param>
public DalException(SqlException objException)
{
  if (objException.Number == 50000)
  {
    int errNum = Convert.ToInt32(objException.Message);
    switch (errNum)
    {
      case 50001:
      {
        _Number = 8;
        break;
      }
      case 50101:
      {
        _Number = 8;
        break;
      }
      case 50201:
      {
        _Number = 8;
        break;
      }
      default:
      {
        _Number = 8;
        break;
      }
    }
  }
  else if (objException.Number == 2627)
  {
    _Number = 12;
  }
  else
```

```
      {
        _Sql = objException;
        _Number = 16;
      }

      RetrieveMessage();
    }

    /// <summary>
    /// Initializes a new instance of the <see cref="StudyPlanning.DAL.DalException" />
    /// class with the specified <see cref="System.Xml.XmlException" /> object.
    /// </summary>
    /// <param name="objException">The <see cref="System.Xml.XmlException" />
    /// object which has caused the <see cref="StudyPlanning.DAL.DalException" /> to
    /// be thrown.
    /// </param>
    public DalException(XmlException objException)
    {
      _Xml = objException;
      _Number = 16;

      RetrieveMessage();
    }

    #endregion //Constructors

    #region Methods

    /// <summary>
    /// Retrieves the message corresponding to the error number of
    /// the Error property of the current instance.
    /// </summary>
    private void RetrieveMessage()
    {
      const int TextGroup_ID = 523;
      const string Culture_ID = "en-GB";

      TextItem objText = TextItem.Retrieve(TextGroup_ID, Convert.ToString(_Number), Culture_ID);
      _Message = objText.Text.Value;

      if (_Number == 4)
      {
        string strText = objText.Text.Value;
        strText = strText.Replace("@@V1", _PropertyName);
        _Message = strText;
      }
    }

    #endregion //Methods
  }
}
```

1.1.4 DalFloat

```
using System;

namespace StudyPlanning.DAL
{
  /// <summary>
  /// Represents a single-precision 32-bit floating point number in the data access layer.
  /// </summary>
  public sealed class DalFloat : DalType
  {
    private float _Value;

    /// <summary>
    /// Initializes a new instance of the <see cref="StudyPlanning.DAL.DalType"/> class
    /// with the specified value of the integer.
    /// </summary>
    /// <param name="floatValue">The value of the float point number.</param>
    /// <param name="allowNull">Indicates whether the value is allowed to be null.<br/>
    /// <strong>true</strong>, if the value is allowed to be null; otherwise, <strong>false</strong>.
    /// </param>
    public DalFloat(float 'oatValue, bool allowNull)
    {
      this.Value = 'oatValue;
      this._allowNull = allowNull;
    }

    /// <summary>
    /// Initializes a new instance of the <see cref="StudyPlanning.DAL.DalFloat"/> class.
    /// </summary>
    /// <param name="allowNull">Indicates whether the value is allowed to be null.<br/>
    /// <strong>true</strong>, if the value is allowed to be null; otherwise, <strong>false</strong>.
    /// </param>
    public DalFloat(bool allowNull)
    {
      this._allowNull = allowNull;
    }

    /// <summary>
    /// Gets or sets the value of the float point number.
    /// </summary>
    public float Value
    {
      get { return this._Value; }
```

```
      set
      {
        this._Value = value;
        this._isNull = false;
      }
    }
  }
}
```

1.1.5 DalGuid

```
using System;

namespace StudyPlanning.DAL
{
  /// <summary>
  /// Represents a globally unique identifier (GUID) in the data access layer.
  /// </summary>
  public sealed class DalGuid : DalType
  {
    private Guid _Value;

    /// <summary>
    /// Initializes a new instance of the <see cref="StudyPlanning.DAL.DalGuid">DalGuid</see> class
    /// with the specified <see cref="System.Guid">Guid</see> object.
    /// </summary>
    /// <param name="guid">The <see cref="System.Guid">Guid</see> object.</param>
    /// <param name="allowNull">Indicates whether the value is allowed to be null.<br/>
    /// <strong>true</strong>, if the value is allowed to be null; otherwise, <strong>false</strong>.
    /// </param>
    public DalGuid(Guid guid, bool allowNull)
    {
      this.Value = guid;
      this._allowNull = allowNull;
    }

    /// <summary>
    /// Initializes a new instance of the <see cref="StudyPlanning.DAL.DalGuid">DalGuid</see> class.
    /// </summary>
    /// <param name="allowNull">Indicates whether the value is allowed to be null.<br/>
    /// <strong>true</strong>, if the value is allowed to be null; otherwise, <strong>false</strong>.
    /// </param>
    public DalGuid(bool allowNull)
    {
      this._allowNull = allowNull;
    }

    /// <summary>
    /// Gets or sets the value of the globally unique identifier.
    /// </summary>
    public Guid Value
    {
      get { return this._Value; }
      set
      {
        this._Value = value;
        this._isNull = false;
      }
    }
  }
}
```

1.1.6 DalInt

```
using System;

namespace StudyPlanning.DAL
{
  /// <summary>
  /// Represents a 32−bit signed integer in the data access layer.
  /// </summary>
  public sealed class DalInt : DalType
  {
    private int _Value;

    /// <summary>
    /// Initializes a new instance of the <see cref="StudyPlanning.DAL.DalInt">DalInt</see> class
    /// with the specified value of the integer.
    /// </summary>
    /// <param name="intValue">The value of the integer.</param>
    /// <param name="allowNull">Indicates whether the value is allowed to be null.<br/>
    /// <strong>true</strong>, if the value is allowed to be null; otherwise, <strong>false</strong>.
    /// </param>
    public DalInt(int intValue, bool allowNull)
    {
      this.Value = intValue;
      this._allowNull = allowNull;
    }

    /// <summary>
    /// Initializes a new instance of the <see cref="StudyPlanning.DAL.DalInt">DalInt</see> class.
    /// </summary>
```

```
    /// <param name="allowNull">Indicates whether the value is allowed to be null.<br/>
    /// <strong>true</strong>, if the value is allowed to be null; otherwise, <strong>false</strong>.
    /// </param>
    public DalInt(bool allowNull)
    {
      this._allowNull = allowNull;
    }

    /// <summary>
    /// Gets or sets the value of the integer.
    /// </summary>
    public int Value
    {
      get { return this._Value; }
      set
      {
        this._Value = value;
        this._isNull = false;
      }
    }
  }
}
```

1.1.7 DalString

```
using System;

namespace StudyPlanning.DAL
{
  /// <summary>
  /// Represents a text (series of Unicode characters) in the data access layer.
  /// </summary>
  public sealed class DalString : StudyPlanning.DAL.DalType
  {
    private string _Value;

    /// <summary>
    /// Initializes a new instance of the <see cref="StudyPlanning.DAL.DalString">DalString</see>
    /// class with the specified value of the string.
    /// </summary>
    /// <param name="strValue">The value of the string</param>
    /// <param name="allowNull">Indicates whether the value is allowed to be null.<br/>
    /// <strong>true</strong>, if the value is allowed to be null; otherwise, <strong>false</strong>.
    /// </param>
    public DalString(string strValue, bool allowNull)
    {
      Value = strValue;
      this._allowNull = allowNull;
    }

    /// <summary>
    /// Initializes a new instance of the <see cref="StudyPlanning.DAL.DalString">DalString</see>
    /// class.
    /// </summary>
    /// <param name="allowNull">Indicates whether the value is allowed to be null.<br/>
    /// <strong>true</strong>, if the value is allowed to be null; otherwise, <strong>false</strong>.
    /// </param>
    public DalString(bool allowNull)
    {
      this._allowNull = allowNull;
    }

    /// <summary>
    /// Gets or sets the text of the string.
    /// </summary>
    public string Value
    {
      get { return _Value; }
      set
      {
        _Value = value;
        this._isNull = false;
      }
    }
  }
}
```

1.1.8 DalStringLocalizable

```
using System;
using System.Collections;
using System.Xml;

namespace StudyPlanning.DAL
{
  /// <summary>
  /// Represents a localizable text (series of Unicode characters)
  /// in the data access layer.
  /// </summary>
  public class DalStringLocalizable : StudyPlanning.DAL.DalType
  {
    private Hashtable _Values = new Hashtable();
```

```
/// <summary>
/// Initializes a new instance of the <see cref="StudyPlanning.DAL.DalStringLocalizable">
/// DalStringLocalizable</see> class.
/// </summary>
/// <param name="allowNull">Indicates whether the value is allowed to be null.<br/>
/// <strong>true</strong>, if the value is allowed to be null; otherwise, <strong>false</strong>.
/// </param>
public DalStringLocalizable(bool allowNull)
{
  this._allowNull = allowNull;
}

/// <summary>
/// Gets or sets the text for the specified culture.
/// </summary>
public string this[string culture_ID]
{
  get
  {
    string strValue;
    strValue = Convert.ToString(_Values[culture_ID]);
    return strValue;
  }

  set
  {
    _Values[culture_ID] = value;
  }
}

/// <summary>
/// Gets a text containing an xml representation of the localized string.
/// </summary>
public string Xml
{
  get
  {
    string strXml = "";
    IDictionaryEnumerator enumValues = _Values.GetEnumerator();

    //XmlDocument xmlDoc = new XmlDocument();
    //xmlDoc.CreateElement("cultures");
    strXml = "<cultures>";

    while (enumValues.MoveNext())
    {
      strXml += "<culture>";
      strXml += "<cultureID>" + enumValues.Key + "</cultureID>";
      strXml += "<value>" + enumValues.Value + "</value>";
      strXml += "</culture>";

      //XmlElement elemCulure = xmlDoc.CreateElement("culture");
      //(System.Xml.XmlElement)elemCulture.
    }
    strXml += "</cultures>";

    return strXml;
  }
}

/// <summary>
/// Gets a text containing an xml representation of the localized string.
/// </summary>
public string Value
{
  get { return this.Xml; }
}

/// <summary>
/// Gets the number of cultures into which the string has been localized.
/// </summary>
public int Count
{
  get { return _Values.Count; }
}

/// <summary>
/// Localizes the string into the specified culture with the specified text.
/// </summary>
/// <param name="culture_ID">The culture for which the string should be
/// localized into.</param>
/// <param name="text">The text of the string for the specified culture.</param>
public void Add(string culture_ID, string text)
{
  _Values.Add(culture_ID, text);
  _isNull = false;
}

/// <summary>
/// Removes the localization of the string for the specified culture.
/// </summary>
/// <param name="culture_ID">The culture for which the string should no
/// longer be localized into.</param>
public void Remove(string culture_ID)
{
  _Values.Remove(culture_ID);
  if (_Values.Count == 0)
  {
```

```
        _isNull = true;
      }
    }

    /// <summary>
    /// Determines whether the string is localized into a specific culture.
    /// </summary>
    /// <param name="culture_ID">The culture for which it should be determined if the
    /// string has been localized into.</param>
    /// <returns><strong>true</strong>, if the string is localized into the specified culture;
    /// otherwise, <strong>false</strong>.</returns>
    public bool Contains(string culture_ID)
    {
      if (_Values.Contains(culture_ID))
        return true;
      else
        return false;
    }

    /// <summary>
    /// Loads an xml document from the specified string.
    /// </summary>
    /// <param name="xml">String containing the XML document to load.</param>
    public void LoadXml(string xml)
    {
      XmlDocument xDoc = new XmlDocument();
      try
      {
        xDoc.LoadXml(xml);
      }
      catch (XmlException objEx)
      {
        throw new DalException(objEx);
      }
      XmlNode xmlRoot = xDoc.FirstChild;

      foreach(XmlNode xn in xmlRoot.ChildNodes)
      {
        if (_Values.Contains(xn.ChildNodes[0].InnerText))
        {
          /*
           * The string has already been localized into the current
           * culture and the text for the current culture is just
           * updated.
           */
          _Values[xn.ChildNodes[0].InnerText] = xn.ChildNodes[1].InnerXml;
        }
        else
        {
          _Values.Add(xn.ChildNodes[0].InnerText, xn.ChildNodes[1].InnerXml);

        }
      }

      if (_Values.Count == 0)
      {
        _isNull = true;
      }
      else
      {
        _isNull = false;
      }
    }

  }
}
```

1.1.9 DalType

```
using System;

namespace StudyPlanning.DAL
{
  /// <summary>
  /// Represents a type in the data access layer.
  /// </summary>
  public abstract class DalType
  {
    /// <summary>
    /// Indicates whether the value of the type is null.
    /// </summary>
    protected bool _isNull = true;

    /// <summary>
    /// Indicates whether the value of the type is allowed to be null.
    /// </summary>
    protected bool _allowNull = false;

    /// <summary>
    /// Validates the type. If the property AllowNull is
    /// <strong>false</strong> and the property IsNull is
    /// <strong>true</strong> a
    /// <see cref="StudyPlanning.DAL.DalException">DalException</see>
    /// is thrown.
    /// </summary>
```

```
    /// <exception cref="StudyPlanning.DAL.DalException">Throws a
    /// <see cref="StudyPlanning.DAL.DalException" /> if the property
    /// AllowNull is false and the property IsNull is true.
    /// </exception>
    public void Validate()
    {
      if (!_allowNull && _isNull)
      {
        DalException objEx = new DalException(4);
        throw objEx;
      }
    }

    /// <summary>
    /// Validates the type. If the property AllowNull is
    /// <strong>false</strong> and the property IsNull is
    /// <strong>true</strong> a
    /// <see cref="StudyPlanning.DAL.DalException">DalException</see>
    /// is thrown.
    /// </summary>
    /// <exception cref="StudyPlanning.DAL.DalException">Throws a
    /// <see cref="StudyPlanning.DAL.DalException" /> if the property
    /// AllowNull is false and the property IsNull is true.
    /// </exception>
    public void Validate(string propertyName)
    {
      if (!_allowNull && _isNull)
      {
        DalException objEx = new DalException(4, propertyName);
        throw objEx;
      }
    }

    /// <summary>
    /// Gets or sets a value indicating whether the value of
    /// the type is null.
    /// </summary>
    /// <remarks>The <strong>IsNull</strong> property may only
    /// be set to <strong>true</strong> - if set to <strong>
    /// false</strong> a <see cref="StudyPlanning.DAL.DalException">
    /// DalException</see> is thrown.
    /// </remarks>
    public bool IsNull
    {
      get { return _isNull; }
      set
      {
        if (Convert.ToBoolean(value))
        {
          _isNull = true;

          /*
          string strTest;
          int intTest;
          if (typeof(_Value).GetType == strTest.GetType())
          {
            _Value = "";
          }
          else if (typeof(_Value).GetType == intTest.GetType())
          {
            _Value = 0;
          }
          */
        }
        else
        {
          DalException objEx = new DalException(4);
          throw objEx;
        }
      }
    }

    /// <summary>
    /// Gets a value indicating whether the value of the type
    /// is allowed to be null.
    /// </summary>
    public bool AllowNull
    {
      get { return _allowNull; }
    }

  }
}
```

1.1.10 DataLog

```
using System;
using System.Data;
using System.Data.SqlClient;

namespace StudyPlanning.DAL
{
  /// <summary>
  /// Represents the properties common to all data classes.
  /// </summary>
```

```
/// <remarks>Every row in the database holds the following information:
/// <list type="">
///  <item>Creation date and time</item>
///  <item>ID of user who created the row</item>
///  <item>Date and time of last update</item>
///  <item>ID of user who last updated the row</item>
/// </list>
/// </remarks>
public class DataLog
{
  #region Private Properties
  private DalDateTime _Created = new DalDateTime(false);
  private DalGuid _CreatedBy = new DalGuid(false);
  private DalDateTime _Updated = new DalDateTime(false);
  private DalGuid _UpdatedBy = new DalGuid(false);
  #endregion //Private Properties

  #region Public Properties

  /// <summary>
  /// Gets the date and time at which the row in the database
  /// was created.
  /// </summary>
  public DalDateTime Created
  {
    get { return _Created; }
  }

  /// <summary>
  /// Gets the ID of the user who created the row in the
  /// database.
  /// </summary>
  public DalGuid CreatedBy
  {
    get { return _CreatedBy; }
  }

  /// <summary>
  /// Gets the date and time at which the row in the database
  /// was last updated.
  /// </summary>
  public DalDateTime Updated
  {
    get { return _Updated; }
  }

  /// <summary>
  /// Gets the ID of the user who last updated the row in the
  /// database.
  /// </summary>
  public DalGuid UpdatedBy
  {
    get { return _UpdatedBy; }
  }

  #endregion //Public Properties

  #region Constructors
  /// <summary>
  /// Initializes a new instance of the
  /// <see cref="StudyPlanning.DAL.DataLog">DataLog</see> class.
  /// </summary>
  public DataLog() {}

  /// <summary>
  /// Initializes a new instance of the
  /// <see cref="StudyPlanning.DAL.DataLog"/> class with
  /// the date and time of creation, the ID of the user who created the
  /// data, the date and time at which the data was last updated, and
  /// the ID of the user who last updated the data.
  /// </summary>
  /// <param name="created">The creation date and time of the data.</param>
  /// <param name="createdBy">The ID of the user who has created the data.</param>
  /// <param name="updated">The date and time at which the data was last updated.</param>
  /// <param name="updatedBy">The ID of the user who last updated the data.</param>
  public DataLog(
    DateTime created,
    Guid createdBy,
    DateTime updated,
    Guid updatedBy)
  {
    _Created.Value = created;
    _CreatedBy.Value = createdBy;
    _Updated.Value = updated;
    _UpdatedBy.Value = updatedBy;
  }

  #endregion //Constructors

  #region Methods

  /// <summary>
  /// Adds the data log properties as parameter to the specified
  /// <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see> object.
  /// </summary>
  /// <param name="objCommand">The <see cref="System.Data.SqlClient.SqlCommand">SqlCommand
  /// </see> object to which data log parameters should be added.</param>
  /// <param name="isOriginal"><strong>true</strong> if the current data
  /// log object is associated to an original data object (for handling concurrent programming
```

```
    /// issues); <strong>false</strong>, otherwise.</param>
    /// <returns>The <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see>
    /// object with the data log parameters added.</returns>
    public SqlCommand AddParameters(SqlCommand objCommand, bool isOriginal)
    {
      SqlParameter objParam;

      if (!isOriginal)
        objParam = objCommand.Parameters.Add("@Created", SqlDbType.DateTime, 8);
      else
        objParam = objCommand.Parameters.Add("@Original_Created", SqlDbType.DateTime, 8);

      objParam.Value = _Created.Value;

      if (!isOriginal)
        objParam = objCommand.Parameters.Add("@CreatedBy", SqlDbType.UniqueIdenti↓er, 16);
      else
        objParam = objCommand.Parameters.Add("@Original_CreatedBy", SqlDbType.UniqueIdenti↓er, 16);

      objParam.Value = _CreatedBy.Value;

      if (!isOriginal)
        objParam = objCommand.Parameters.Add("@Updated", SqlDbType.DateTime, 8);
      else
        objParam = objCommand.Parameters.Add("@Original_Updated", SqlDbType.DateTime, 8);

      objParam.Value = _Updated.Value;

      if (!isOriginal)
        objParam = objCommand.Parameters.Add("@UpdatedBy", SqlDbType.UniqueIdenti↓er, 16);
      else
        objParam = objCommand.Parameters.Add("@Original_UpdatedBy", SqlDbType.UniqueIdenti↓er, 16);

      objParam.Value = _CreatedBy.Value;

      return objCommand;
    }

    /// <summary>
    /// Validates the <see cref="StudyPlanning.DAL.DataLog"/> object. If the
    /// value of the Created, CreatedBy, Updated or UpdatedBy property is
    /// null a <see cref="StudyPlanning.DAL.DalException"/> is thrown.
    /// </summary>
    public void Validate()
    {
      try
      {
        _Created.Validate("Created");
        _CreatedBy.Validate("CreatedBy");
        _Updated.Validate("Updated");
        _UpdatedBy.Validate("UpdatedBy");
      }
      catch (DalException excp)
      {
        throw excp;
      }
    }

    /// <summary>
    /// Sets the values of the properties of the current object taking
    /// the values from the specified <see cref="System.Data.DataRow"/>
    /// object.
    /// </summary>
    /// <param name="dataRow">The <see cref="System.Data.DataRow"/> object
    /// from which the values should be taken.</param>
    public void SetValues(DataRow dataRow)
    {
      _Created.Value = (DateTime)dataRow["Created"];
      _CreatedBy.Value = (Guid)dataRow["CreatedBy"];
      _Updated.Value = (DateTime)dataRow["Updated"];
      _UpdatedBy.Value = (Guid)dataRow["UpdatedBy"];
    }

    #endregion //Methods

  }
}
```

1.1.11 DbObject

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Con↓guration;

namespace StudyPlanning.DAL
{
  /// <summary>
  /// Base class for the data access layer (also known as data services tier).
  /// The class is declared as abstract which means that this class can act as
  /// a basis for other classes, but cannot be instantiated.
  /// </summary>
  public abstract class DbObject
  {
    /// <summary>
```

```
    /// Gets the connection to the SQL Server database.
    /// </summary>
    public SqlConnection objConnection;
    private string _ConnectionString;

    /// <summary>
    /// Creates a new instance of the DbObject class.
    /// </summary>
    public DbObject()
    {
      _ConnectionString =
        System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
      objConnection = new SqlConnection(_ConnectionString);
    }

    /// <summary>
    /// Gets the database connection string.
    /// </summary>
    protected string ConnectionString
    {
      get { return _ConnectionString; }
    }

  }
}
```

1.2 Assessment

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL
{
  /// <summary>
  /// Represents an assessment of a course or project.
  /// </summary>
  public class Assessment : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalGuid _Assessment_ID = new DalGuid(false);
    private DalBool _Passed = new DalBool(true);
    private DalInt _Grade_ID = new DalInt(true);
    private DalBool _PointsCredited = new DalBool(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.Assessment"/>
    /// class.
    /// </summary>
    public Assessment() {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the Assessment_ID.
    /// </summary>
    /// <value></value>
    public DalGuid Assessment_ID
    {
      get { return _Assessment_ID; }
    }

    /// <summary>
    /// Gets the Passed property of the assessment.
    /// </summary>
    public DalBool Passed
    {
      get { return _Passed; }
    }

    /// <summary>
    /// Gets the Grade_ID of the assessment.
    /// </summary>
    public DalInt Grade_ID
    {
      get { return _Grade_ID; }
    }

    /// <summary>
    /// Gets the PointsCredited property of the assessment.
    /// </summary>
    public DalBool PointsCredited
    {
```

```
    get { return _PointsCredited; }
  }

  /// <summary>
  /// Gets the data log of the course.
  /// </summary>
  public DataLog Log
  {
    get { return _Log; }
  }

  #endregion //Public Properties

  #region Public Methods

  #region Create Methods

  /// <summary>
  /// Creates an Assessment object in the database using the values of the properties
  /// of the current instance.
  /// </summary>
  public void Create()
  {
    try
    {
      this.Validate();
    }
    catch (DalException excp)
    {
      throw excp;
    }

    SqlCommand objCommand = new SqlCommand("Assessment_Insert", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, false);

    try
    {
      objConnection.Open();
      objCommand.ExecuteNonQuery();
      objConnection.Close();
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
  }

  #endregion

  #region Retrieve Methods

  /// <summary>
  /// Retrieves the Assessment from the database using the value of the
  /// Assessment_ID property of the current instance
  /// </summary>
  public void Retrieve()
  {
    try
    {
      this._Assessment_ID.Validate("Assessment_ID");
    }
    catch (DalException e)
    {
      throw e;
    }

    try
    {
      RetrieveExecute(_Assessment_ID.Value, this, objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }
  }

  /// <summary>
  /// Retrieves the Assessment with the specified identifier from the database.
  /// </summary>
  /// <param name="assessment_ID"> Identification of the assessment.</param>
  public static Assessment Retrieve(Guid assessment_ID)
  {
    string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
    SqlConnection objConnection = new SqlConnection(connString);

    Assessment objAssessment = new Assessment();

    try
    {
      RetrieveExecute(assessment_ID, objAssessment, objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }
    return objAssessment;
  }
```

```
private static void RetrieveExecute(
  Guid assessment_ID,
  Assessment assessment,
  SqlConnection sqlConnection)
{
  DataRow objDataRow;

  SqlCommand objCommand = new SqlCommand("Assessment_Select", sqlConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  SqlParameter objParam;

  objParam = objCommand.Parameters.Add("@Assessment_ID", SqlDbType.UniqueIdenti↓er);
  objParam.Value = assessment_ID;

  DataSet objDataSet = new DataSet("Assessment");
  SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

  try
  {
    sqlConnection.Open();
    objAdap.Fill(objDataSet, "Assessment");
    sqlConnection.Close();
    objDataRow = objDataSet.Tables["Assessment"].Rows[0];
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
  assessment._Assessment_ID.Value = (Guid)objDataRow["Assessment_ID"];

  if (objDataRow["Passed"].Equals(DBNull.Value))
    assessment._Passed.IsNull = true;
  else
    assessment._Passed.Value = (bool)objDataRow["Passed"];

  if (objDataRow["Grade_ID"].Equals(DBNull.Value))
    assessment._Grade_ID.IsNull = true;
  else
    assessment._Grade_ID.Value = (int)objDataRow["Grade_ID"];

  assessment._PointsCredited.Value = (bool)objDataRow["PointsCredited"];
  assessment._Log.SetValues(objDataRow);
}

#endregion //Retrieve Methods

#region Update Methods

/// <summary>
/// Updates the current Assessment in the database using the original
/// Assessment to resolve possible concurrency issues.
/// </summary>
/// <param name="originalAssessment">The <see cref="StudyPlanning.DAL.Assessment"/> object.</param>
/// <returns><strong>true</strong> if the update was succesfully executed; otherwise, <strong>false</strong>.</returns>
public bool Update(Assessment originalAssessment)
{
  bool blnResult = false;

  try
  {
    this.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }

  SqlCommand objCommand = new SqlCommand("Assessment_Update", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, false);
  AddParameters(originalAssessment, objCommand, true);

  try
  {
    objConnection.Open();
    int rowsA↑ected = objCommand.ExecuteNonQuery();
    objConnection.Close();

    if (rowsA↑ected > 0)
      blnResult = true;
  }
  catch(SqlException excp)
  {
    throw new DalException(excp);
  }
  return blnResult;
}

#endregion //Update Methods

#region Delete Methods

/// <summary>
/// Deletes the current Assessment from the database.
/// </summary>
/// <returns><strong>true</strong> if the deletion is succesfully executed; otherwise, <strong>false</strong>.</returns>
```

```
public bool Delete()
{
  bool blnResult = false;

  try
  {
    _Assessment_ID.Validate("Assessment_ID");
  }
  catch (DalException excp)
  {
    throw excp;
  }

  SqlCommand objCommand = new SqlCommand("Assessment_Delete", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, true);

  try
  {
    objConnection.Open();
    int rowsA↑ected = objCommand.ExecuteNonQuery();
    objConnection.Close();

    if (rowsA↑ected > 0)
      blnResult = true;
  }
  catch(SqlException e)
  {
    throw new DalException(e);
  }
  return blnResult;
}

#endregion //Delete Methods

#endregion //Public Methods

#region Private Methods

/// <summary>
/// Adds the relevant parameters to the specified <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see> object
/// setting the values to ones of the specified <see cref="StudyPlanning.DAL.Assessment"/> object.
/// </summary>
/// <param name="assessment">
/// The <see cref="StudyPlanning.DAL.Assessment"/> object containing the values that the parameters have to be
/// initialized to.
/// </param>
/// <param name="objCommand">
/// The <see cref="System.Data.SqlClient.SqlCommand"/> object to which parameters should be added.
/// </param>
/// <param name="isOriginal">
/// <strong>true</strong> if the parameters to be added represent original
/// data (for handling concurrent programming issues); otherwise, <strong>false</strong>.</param>
/// <returns>The <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see>
/// object with the relevant parameters added.</returns>
private static void AddParameters(Assessment assessment, SqlCommand objCommand, bool isOriginal)
{
  string paramName = "";
  SqlParameter objParam;

  paramName = "Assessment_ID";
  objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er);
  objParam.Value = assessment._Assessment_ID.Value;
  if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

  paramName = "Passed";
  objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.Bit);
  if (assessment._Passed.IsNull)
    objParam.Value = DBNull.Value;
  else
    objParam.Value = assessment._Passed.Value;
  if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

  paramName = "Grade_ID";
  objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.Int);
  if (assessment._Grade_ID.IsNull)
    objParam.Value = DBNull.Value;
  else
    objParam.Value = assessment._Grade_ID.Value;
  if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

  paramName = "PointsCredited";
  objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.Bit);
  objParam.Value = assessment._PointsCredited.Value;
  if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

  assessment._Log.AddParameters(objCommand, isOriginal);
}

/// <summary>
/// Validates the current <see cref="StudyPlanning.DAL.Assessment"/> object. If the
/// value of either of the private properties is null a
/// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
/// </summary>
```

```
    private void Validate()
    {
      try
      {
        _Assessment_ID.Validate("Assessment_ID");
        _Passed.Validate("Passed");
        _Grade_ID.Validate("Grade_ID");
        _PointsCredited.Validate("PointsCredited");
        _Log.Validate();
      }
      catch (DalException excp)
      {
        throw excp;
      }
    }
    #endregion //Private Methods
  }
}
```

1.3 AssessmentType

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL
{
  /// <summary>
  /// Represents a type whereupon a course or a project can be assessed.
  /// </summary>
  public class AssessmentType : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalInt _AssessmentType_ID = new DalInt(false);
    private DalStringLocalizable _Name = new DalStringLocalizable(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.AssessmentType" />
    /// class.
    /// </summary>
    public AssessmentType()
    {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the ID of the assessment.
    /// </summary>
    public DalInt AssessmentType_ID
    {
      get { return _AssessmentType_ID; }
    }

    /// <summary>
    /// Gets the name of the assessment.
    /// </summary>
    public DalStringLocalizable Name
    {
      get { return _Name; }
    }

    /// <summary>
    /// Gets the data log of the assessment.
    /// </summary>
    public DataLog Log
    {
      get { return _Log; }
    }

    #endregion //Public Properties

    #region Methods

    #region Retrieve Methods

    /// <summary>
    /// Retrieves the assessment from the database using the value of the
    /// AssessmentType_ID property of the current instance.
    /// </summary>
    public void Retrieve()
    {
      try
      {
```

```
        this._AssessmentType_ID.Validate("AssessmentType_ID");
      }
      catch (DalException excp)
      {
        throw excp;
      }

      try
      {
        RetrieveExecute(_AssessmentType_ID.Value, this, objConnection);
      }
      catch (DalException objExc)
      {
        throw objExc;
      }
    }

    /// <summary>
    /// Retrieves the assessment with the specified identifier from the database.
    /// </summary>
    /// <param name="assessment_ID">The globally unique identifier of the assessment.</param>
    public static AssessmentType Retrieve(int assessment_ID)
    {
      string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
      SqlConnection objConnection = new SqlConnection(connString);

      AssessmentType objAssessmentType = new AssessmentType();

      try
      {
        RetrieveExecute(assessment_ID, objAssessmentType, objConnection);
      }
      catch (DalException objExc)
      {
        throw objExc;
      }

      return objAssessmentType;
    }

    private static void RetrieveExecute(
      int assessment_ID,
      AssessmentType objAssessmentType,
      SqlConnection sqlConnection)
    {
      SqlCommand objCommand = new SqlCommand("AssessmentType_Select", sqlConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      SqlParameter objParam;

      objParam = objCommand.Parameters.Add("@AssessmentType_ID", SqlDbType.Int, 4);
      objParam.Value = assessment_ID;

      DataSet objDataSet = new DataSet("AssessmentType");
      SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

      try
      {
        sqlConnection.Open();
        objAdap.Fill(objDataSet, "AssessmentType");
        sqlConnection.Close();
        DataRow objDataRow = objDataSet.Tables["AssessmentType"].Rows[0];

        objAssessmentType.AssessmentType_ID.Value = (int)objDataRow["AssessmentType_ID"];
        objAssessmentType.Name.LoadXml(objDataRow["Name"].ToString());
        objAssessmentType.Log.SetValues(objDataRow);
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }
    }

    #endregion //Retrive Methods

    #endregion //Methods
  }
}
```

1.4 ContactData

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL
{
  /// <summary>
  /// Represents contact information of e.g. a person, department etc.
  /// </summary>
  public class ContactData : StudyPlanning.DAL.DbObject
  {
    #region Private Properties
```

```
private DalGuid _ContactData_ID = new DalGuid(false);
private DalInt _ContactDataType_ID = new DalInt(false);
private DalString _Email = new DalString(true);
private DalString _Mobile = new DalString(true);
private DalString _Phone = new DalString(true);
private DalString _Facsimile = new DalString(true);
private DalString _Homepage = new DalString(true);
private DalString _Address1 = new DalString(true);
private DalString _Address2 = new DalString(true);
private DalString _PostalCode = new DalString(true);
private DalString _City = new DalString(true);
private DataLog _Log;

#endregion //Private Properties

#region Constructors

/// <summary>
/// Initializes a new instance of the <see cref="StudyPlanning.DAL.ContactData"/>
/// class.
/// </summary>
public ContactData()
{
  _Log = new DataLog();
}

#endregion

#region Public Properties

/// <summary>
/// Gets the ID of the contact data.
/// </summary>
public DalGuid ContactData_ID
{
  get { return _ContactData_ID; }
}

/// <summary>
/// Gets the ID of the contact data type.
/// </summary>
public DalInt ContactDataType_ID
{
  get { return _ContactDataType_ID; }
}

/// <summary>
/// Gets the email address of the contact.
/// </summary>
public DalString Email
{
  get { return _Email; }
}

/// <summary>
/// Gets the mobile phone number of the contact.
/// </summary>
public DalString Mobile
{
  get { return _Mobile; }
}

/// <summary>
/// Gets the phone number of the contact.
/// </summary>
public DalString Phone
{
  get { return _Phone; }
}

/// <summary>
/// Gets the facsimile number of the contact.
/// </summary>
public DalString Facsimile
{
  get { return _Facsimile; }
}

/// <summary>
/// Gets url of the homepage of the contact.
/// </summary>
public DalString Homepage
{
  get { return _Homepage; }
}

/// <summary>
/// Gets the first line of the contact address.
/// </summary>
public DalString Address1
{
  get { return _Address1; }
}

/// <summary>
/// Gets the second line of the contact address.
/// </summary>
public DalString Address2
```

```
{
  get { return _Address2; }
}

/// <summary>
/// Gets the postal code of the contact (also known as zip code)
/// </summary>
public DalString PostalCode
{
  get { return _PostalCode; }
}

/// <summary>
/// Gets the name of the city of the contact address.
/// </summary>
public DalString City
{
  get { return _City; }
}

/// <summary>
/// Gets the data log of the contact data.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Methods

/// <summary>
/// Creates a new contact in the database using the values of the properties
/// of the current instance.
/// </summary>
public void Create()
{
  try
  {
    this.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }

  SqlCommand objCommand = new SqlCommand("ContactData_Insert", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, false);

  try
  {
    objConnection.Open();
    objCommand.ExecuteNonQuery();
    objConnection.Close();
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
}

#region Retrieve Methods

/// <summary>
/// Retrieves the contact data from the database using the value of the
/// ContactData_ID property of the current instance.
/// </summary>
public void Retrieve()
{
  try
  {
    this._ContactData_ID.Validate("ContactData_ID");
  }
  catch (DalException excp)
  {
    throw excp;
  }

  try
  {
      RetrieveExecute(_ContactData_ID.Value, this, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
}

/// <summary>
/// Retrieves the contact data with the specified identifier from the database.
/// </summary>
/// <param name="contactData_ID">The globally unique identifier of the contact data.</param>
public static ContactData Retrieve(Guid contactData_ID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);
```

```
    ContactData objCd = new ContactData();

    try
    {
      RetrieveExecute(contactData_ID, objCd, objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }

    return objCd;
  }

  private static void RetrieveExecute(
    Guid contactData_ID,
    ContactData objContactData,
    SqlConnection sqlConnection)
  {
    SqlCommand objCommand = new SqlCommand("ContactData_Select", sqlConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam;

    objParam = objCommand.Parameters.Add("@ContactData_ID", SqlDbType.UniqueIdenti↓er, 16);
    objParam.Value = contactData_ID;

    DataSet objDataSet = new DataSet("ContactData");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
      sqlConnection.Open();
      objAdap.Fill(objDataSet, "ContactData");
      sqlConnection.Close();
      DataRow objDataRow = objDataSet.Tables["ContactData"].Rows[0];

      objContactData.ContactData_ID.Value = (Guid)objDataRow["ContactData_ID"];
      objContactData.ContactDataType_ID.Value = (int)objDataRow["ContactDataType_ID"];

      if (objDataRow["Email"].Equals(System.DBNull.Value))
        objContactData.Email.IsNull = true;
      else
        objContactData.Email.Value = (string)objDataRow["Email"];

      if (objDataRow["Mobile"].Equals(System.DBNull.Value))
        objContactData.Mobile.IsNull = true;
      else
        objContactData.Mobile.Value = (string)objDataRow["Mobile"];

      if (objDataRow["Phone"].Equals(System.DBNull.Value))
        objContactData.Phone.IsNull = true;
      else
        objContactData.Phone.Value = (string)objDataRow["Phone"];

      if (objDataRow["Facsimile"].Equals(System.DBNull.Value))
        objContactData.Facsimile.IsNull = true;
      else
        objContactData.Facsimile.Value = (string)objDataRow["Facsimile"];

      if (objDataRow["Homepage"].Equals(System.DBNull.Value))
        objContactData.Homepage.IsNull = true;
      else
        objContactData.Homepage.Value = (string)objDataRow["Homepage"];

      if (objDataRow["Address1"].Equals(System.DBNull.Value))
        objContactData.Address1.IsNull = true;
      else
        objContactData.Address1.Value = (string)objDataRow["Address1"];

      if (objDataRow["Address2"].Equals(System.DBNull.Value))
        objContactData.Address2.IsNull = true;
      else
        objContactData.Address2.Value = (string)objDataRow["Address2"];

      if (objDataRow["PostalCode"].Equals(System.DBNull.Value))
        objContactData.PostalCode.IsNull = true;
      else
        objContactData.PostalCode.Value = (string)objDataRow["PostalCode"];

      if (objDataRow["City"].Equals(System.DBNull.Value))
        objContactData.City.IsNull = true;
      else
        objContactData.City.Value = (string)objDataRow["City"];

      objContactData.Log.SetValues(objDataRow);
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
  }

  #endregion //Retrive Methods

  /// <summary>
  /// Updates the current contact data in the database using the original
  /// contact data to resolve possible concurrency issues.
```

```
/// </summary>
/// <param name="originalContactData">The original <see cref="StudyPlanning.DAL.ContactData"/> object.</param>
/// <returns><strong>true</strong> if the update was succesfully executed; otherwise, <strong>false</strong>.</returns>
public bool Update(ContactData originalContactData)
{
  try
  {
    this.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }

  bool blnResult = false;

  SqlCommand objCommand = new SqlCommand("ContactData_Update", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, false);
  AddParameters(originalContactData, objCommand, true);

  int rowsA↑ected;

  try
  {
    objConnection.Open();
    rowsA↑ected = objCommand.ExecuteNonQuery();
    objConnection.Close();

    if (rowsA↑ected > 0)
      blnResult = true;
  }
  catch(SqlException excp)
  {
    throw new DalException(excp);
  }

  return blnResult;
}

/// <summary>
/// Deletes the current contact data from the database.
/// </summary>
/// <returns><strong>true</strong> if the deletion is succesfully executed; otherwise, <strong>false</strong>.</returns>
public bool Delete()
{
  bool blnResult = false;

  try
  {
    this.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }

  SqlCommand objCommand = new SqlCommand("ContactData_Delete", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, true);

  try
  {
    objConnection.Open();
    int rowsA↑ected = objCommand.ExecuteNonQuery();
    objConnection.Close();

    if (rowsA↑ected > 0)
      blnResult = true;
  }
  catch(SqlException objEx)
  {
    throw new DalException(objEx);
  }

  return blnResult;
}

/// <summary>
/// Adds the relevant parameters to the specified <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see> object
/// setting the values to ones of the specified <see cref="StudyPlanning.DAL.ContactData"/> object.
/// </summary>
/// <param name="contactData">
/// The <see cref="StudyPlanning.DAL.ContactData"/> object containing the values that the parameters have to be
/// initialized to.
/// </param>
/// <param name="objCommand">
/// The <see cref="System.Data.SqlClient.SqlCommand"/> object to which parameters should be added.
/// </param>
/// <param name="isOriginal">
/// <strong>true</strong> if the parameters to be added represent original
/// data (for handling concurrent programming issues); otherwise, <strong>false</strong>.</param>
/// <returns>The <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see>
/// object with the relevant parameters added.</returns>
private static void AddParameters(ContactData contactData, SqlCommand objCommand, bool isOriginal)
{
  string paramName = "";
```

```
SqlParameter objParam;

//ContactData_ID
paramName = "ContactData_ID";
objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

if (isOriginal)
  objParam.ParameterName = "@Original_" + paramName;

objParam.Value = contactData.ContactData_ID.Value;

//ContactDataType_ID
paramName = "ContactDataType_ID";
objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.Int, 4);

if (isOriginal)
  objParam.ParameterName = "@Original_" + paramName;

objParam.Value = contactData.ContactDataType_ID.Value;

//Email
paramName = "Email";
objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.VarChar, 100);

if (isOriginal)
  objParam.ParameterName = "@Original_" + paramName;

if (contactData.Email.IsNull)
  objParam.Value = System.DBNull.Value;
else
  objParam.Value = contactData.Email.Value;

//Mobile
paramName = "Mobile";
objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.VarChar, 30);

if (isOriginal)
  objParam.ParameterName = "@Original_" + paramName;

if (contactData.Mobile.IsNull)
  objParam.Value = System.DBNull.Value;
else
  objParam.Value = contactData.Mobile.Value;

//Phone
paramName = "Phone";
objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.VarChar, 30);

if (isOriginal)
  objParam.ParameterName = "@Original_" + paramName;

if (contactData.Phone.IsNull)
  objParam.Value = System.DBNull.Value;
else
  objParam.Value = contactData.Phone.Value;

//Facsimile
paramName = "Facsimile";
objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.VarChar, 30);

if (isOriginal)
  objParam.ParameterName = "@Original_" + paramName;

if (contactData.Facsimile.IsNull)
  objParam.Value = System.DBNull.Value;
else
  objParam.Value = contactData.Facsimile.Value;

//Homepage
paramName = "Homepage";
objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.VarChar, 100);

if (isOriginal)
  objParam.ParameterName = "@Original_" + paramName;

if (contactData.Homepage.IsNull)
  objParam.Value = System.DBNull.Value;
else
  objParam.Value = contactData.Homepage.Value;

//Address1
paramName = "Address1";
objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.VarChar, 100);

if (isOriginal)
  objParam.ParameterName = "@Original_" + paramName;

if (contactData.Address1.IsNull)
  objParam.Value = System.DBNull.Value;
else
  objParam.Value = contactData.Address1.Value;

//Address2
paramName = "Address2";
objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.VarChar, 100);

if (isOriginal)
  objParam.ParameterName = "@Original_" + paramName;
```

```
      if (contactData.Address2.IsNull)
        objParam.Value = System.DBNull.Value;
      else
        objParam.Value = contactData.Address2.Value;

      //PostalCode
      paramName = "PostalCode";
      objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.VarChar, 10);

      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      if (contactData.PostalCode.IsNull)
        objParam.Value = System.DBNull.Value;
      else
        objParam.Value = contactData.PostalCode.Value;

      //City
      paramName = "City";
      objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.VarChar, 50);

      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      if (contactData.City.IsNull)
        objParam.Value = System.DBNull.Value;
      else
        objParam.Value = contactData.City.Value;

      //Log
      objCommand = contactData.Log.AddParameters(objCommand, isOriginal);
    }

    /// <summary>
    /// Validates the current <see cref="StudyPlanning.DAL.ContactData"/> object. If the
    /// value of either of the private properties is null a
    /// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
    /// </summary>
    private void Validate()
    {
      try
      {
        _ContactData_ID.Validate("ContactData_ID");
        _ContactDataType_ID.Validate("ContactDataType_ID");
        _Email.Validate("Email");
        _Mobile.Validate("Mobile");
        _Phone.Validate("Phone");
        _Facsimile.Validate("Facsimile");
        _Homepage.Validate("Homepage");
        _Address1.Validate("Address1");
        _Address2.Validate("Address2");
        _PostalCode.Validate("PostalCode");
        _City.Validate("City");
        _Log.Validate();
      }
      catch (DalException excp)
      {
        throw excp;
      }
    }

    #endregion //Methods
  }
}
```

1.5 ContactDataType

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL
{
  /// <summary>
  /// Represents contact information type.
  /// </summary>
  public class ContactDataType : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalInt _ContactDataType_ID = new DalInt(false);
    private DalStringLocalizable _Name = new DalStringLocalizable(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.ContactDataType" />
    /// class.
    /// </summary>
    public ContactDataType()
```

```
{
  _Log = new DataLog();
}

#endregion

#region Public Properties

/// <summary>
/// Gets the ID of the contact data type.
/// </summary>
public DalInt ContactDataType_ID
{
  get { return _ContactDataType_ID; }
}

/// <summary>
/// Gets the name of the contact data type.
/// </summary>
public DalStringLocalizable Name
{
  get { return _Name; }
}

/// <summary>
/// Gets the data log of the contact data type.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Methods

#region Retrieve Methods

/// <summary>
/// Retrieves the contact data type from the database using the value of the
/// ContactDataType_ID property of the current instance.
/// </summary>
public void Retrieve()
{
  try
  {
    this._ContactDataType_ID.Validate("ContactDataType_ID");
  }
  catch (DalException excp)
  {
    throw excp;
  }

  try
  {
    RetrieveExecute(_ContactDataType_ID.Value, this, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
}

/// <summary>
/// Retrieves the contact data type with the specified identifier from the database.
/// </summary>
/// <param name="contactDataType_ID">The globally unique identifier of the contact data type.</param>
public static ContactDataType Retrieve(int contactDataType_ID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  ContactDataType objContactDataType = new ContactDataType();

  try
  {
    RetrieveExecute(contactDataType_ID, objContactDataType, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }

  return objContactDataType;
}

private static void RetrieveExecute(
  int contactDataType_ID,
  ContactDataType objContactDataType,
  SqlConnection sqlConnection)
{
  SqlCommand objCommand = new SqlCommand("ContactDataType_Select", sqlConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  SqlParameter objParam;

  objParam = objCommand.Parameters.Add("@ContactDataType_ID", SqlDbType.Int, 4);
  objParam.Value = contactDataType_ID;
```

```
      DataSet objDataSet = new DataSet("ContactDataType");
      SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

      try
      {
        sqlConnection.Open();
        objAdap.Fill(objDataSet, "ContactDataType");
        sqlConnection.Close();
        DataRow objDataRow = objDataSet.Tables["ContactDataType"].Rows[0];

        objContactDataType.ContactDataType_ID.Value = (int)objDataRow["ContactDataType_ID"];
        objContactDataType.Name.LoadXml(objDataRow["Name"].ToString());
        objContactDataType.Log.SetValues(objDataRow);
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }
    }

    #endregion //Retrive Methods

    #endregion //Methods
  }
}
```

1.6 Courses

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.Courses
{
  /// <summary>
  /// Represents a course.
  /// </summary>
  public class Course : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalGuid _Course_ID = new DalGuid(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.Courses.Course" />
    /// class.
    /// </summary>
    public Course()
    {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets or sets the ID of the course.
    /// </summary>
    public DalGuid Course_ID
    {
      get { return _Course_ID; }
    }

    /// <summary>
    /// Gets or sets the data log of the course.
    /// </summary>
    public DataLog Log
    {
      get { return _Log; }
    }

    #endregion //Public Properties

    #region Methods

    #region Create Methods

    /// <summary>
    /// Creates a course in the database using the values of the properties
    /// of the current instance.
    /// </summary>
    public void Create()
    {
      try
      {
        _Course_ID.Validate();
        _Log.Validate();
```

```
    }
    catch (DalException excp)
    {
      throw excp;
    }

    SqlCommand objCommand = new SqlCommand("Course_Insert", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam;

    objParam = objCommand.Parameters.Add("@Course_ID", SqlDbType.UniqueIdenti↓er, 16);
    objParam.Value = _Course_ID.Value;

    objCommand = _Log.AddParameters(objCommand, false);

    DataSet objDataSet = new DataSet("Course");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    DataRow objDataRow;

    try
    {
      objConnection.Open();
      objAdap.Fill(objDataSet, "Course");
      objConnection.Close();
      objDataRow = objDataSet.Tables["Course"].Rows[0];
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }

    _Course_ID.Value = (Guid)objDataRow["Course_ID"];
    _Log.Created.Value = (DateTime)objDataRow["Created"];
    _Log.CreatedBy.Value = (Guid)objDataRow["CreatedBy"];
    _Log.Updated.Value = (DateTime)objDataRow["Updated"];
    _Log.UpdatedBy.Value = (Guid)objDataRow["UpdatedBy"];
  }

  #endregion

  #region Retrieve Methods

  /// <summary>
  /// Retrieves the course from the database using the value of the
  /// Course_ID property of the current instance.
  /// </summary>
  public void Retrieve()
  {
    try
    {
      _Course_ID.Validate();
    }
    catch (DalException excp)
    {
      throw excp;
    }

    try
    {
      DataRow objDataRow =
        RetrieveExecute(_Course_ID.Value, objConnection);

      this._Course_ID.Value = (Guid)objDataRow["Course_ID"];
      this._Log.SetValues(objDataRow);
      /*
      this._Log.Created.Value = (DateTime)objDataRow["Created"];
      this._Log.CreatedBy.Value = (Guid)objDataRow["CreatedBy"];
      this._Log.Updated.Value = (DateTime)objDataRow["Updated"];
      this._Log.UpdatedBy.Value = (Guid)objDataRow["UpdatedBy"];
      */
    }
    catch (DalException objExc)
    {
      throw objExc;
    }
  }

  /// <summary>
  /// Retrieves the course with the specified identifier from the database.
  /// </summary>
  /// <param name="course_ID">The globally unique identifier of the course.</param>
  /// <returns>A <see cref="StudyPlanning.DAL.Courses.Course"/> object representing the course with the specified ID.</returns>
  public static Course Retrieve(Guid course_ID)
  {
    string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
    SqlConnection objConnection = new SqlConnection(connString);

    Course objCourse = new Course();

    try
    {
      DataRow objDataRow =
        RetrieveExecute(course_ID, objConnection);

      objCourse.Course_ID.Value = (Guid)objDataRow["Course_ID"];
      objCourse.Log.SetValues(objDataRow);
```

```
      /*
      objCourse.Log.Created = (DateTime)objDataRow["Created"];
      objCourse.Log.CreatedBy = (Guid)objDataRow["CreatedBy"];
      objCourse.Log.Updated = (DateTime)objDataRow["Updated"];
      objCourse.Log.UpdatedBy = (Guid)objDataRow["UpdatedBy"];
      */
    }
    catch (DalException objExc)
    {
      throw objExc;
    }

    return objCourse;
  }

  private static DataRow RetrieveExecute(
    Guid course_ID,
    SqlConnection sqlConnection)
  {
    SqlCommand objCommand = new SqlCommand("Course_Select", sqlConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam;

    objParam = objCommand.Parameters.Add("@Course_ID", SqlDbType.UniqueIdenti↓er, 16);
    objParam.Value = course_ID;

    DataSet objDataSet = new DataSet("Course");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
      sqlConnection.Open();
      objAdap.Fill(objDataSet, "Course");
      sqlConnection.Close();
      DataRow objDataRow = objDataSet.Tables["Course"].Rows[0];

      return objDataRow;
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
  }

  #endregion //Retrive Methods

  #region Update Method

  /// <summary>
  /// Updates the current course in the database using the original
  /// course to resolve possible concurrency issues.
  /// </summary>
  /// <param name="originalCourse">The original <see cref="StudyPlanning.DAL.Courses.Course"/> object.</param>
  /// <returns><strong>true</strong> if the update was succesfully executed; otherwise, <strong>false</strong>.</returns>
  public bool Update(Course originalCourse)
  {
    try
    {
      _Course_ID.Validate();
      _Log.Validate();
    }
    catch (DalException excp)
    {
      throw excp;
    }

    bool blnResult = false;

    SqlCommand objCommand = new SqlCommand("Course_Update", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam;

    objParam = objCommand.Parameters.Add("@Course_ID", SqlDbType.UniqueIdenti↓er, 16);
    objParam.Value = this._Course_ID.Value;

    objCommand = this._Log.AddParameters(objCommand, false);

    objParam = objCommand.Parameters.Add("@Original_Course_ID", SqlDbType.UniqueIdenti↓er, 16);
    objParam.Value = originalCourse._Course_ID.Value;

    objCommand = originalCourse.Log.AddParameters(objCommand, true);

    int rowsA↑ected;

    try
    {
      objConnection.Open();
      rowsA↑ected = objCommand.ExecuteNonQuery();
      objConnection.Close();

      if (rowsA↑ected > 0)
        blnResult = true;
    }
    catch(SqlException excp)
    {
      throw new DalException(excp);
    }
```

```
      return blnResult;
    }

    #endregion

    #region Delete Method

    /// <summary>
    /// Deletes the current course from the database.
    /// </summary>
    /// <returns><strong>true</strong> if the deletion is succesfully executed; otherwise, <strong>false</strong>.</returns>
    public bool Delete()
    {
      bool blnResult = false;

      if (_Log == null)
      {
        throw new DalException(4,"Log");
      }

      try
      {
        _Course_ID.Validate();
        _Log.Validate();
      }
      catch (DalException excp)
      {
        throw excp;
      }

      SqlCommand objCommand = new SqlCommand("Course_Delete", objConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      SqlParameter objParam;

      objParam = objCommand.Parameters.Add("@Original_Course_ID", SqlDbType.UniqueIdenti↓er, 16);
      objParam.Value = this._Course_ID.Value;

      objCommand = this._Log.AddParameters(objCommand, true);

      int rowsA↑ected;
      try
      {
        objConnection.Open();
        rowsA↑ected = objCommand.ExecuteNonQuery();
        objConnection.Close();

        if (rowsA↑ected > 0)
          blnResult = true;
      }
      catch(SqlException excp)
      {
        throw new DalException(excp);
      }

      return blnResult;
    }

    #endregion

    #endregion //Methods
  }
}


using System;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.Courses
{
  /// <summary>
  /// Represents a course part in the data accessc tier.
  /// A course part consists of a Course_ID and a Part number.
  /// </summary>
  public class CoursePart : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private Guid _Course_ID;
    private int _Part;

    #endregion //Private Properties

    #region Public Properties

    /// <summary>
    /// Gets or sets the value of the Course_ID property.
    /// </summary>
    public Guid Course_ID
    {
      get { return _Course_ID; }
      set { _Course_ID = value; }
    }

    /// <summary>
    /// Gets or sets the value of the Part property.
    /// </summary>
    public int Part
```

```
    {
      get { return _Part; }
      set { _Part = value; }
    }

    #endregion //Public Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.Courses.CoursePart"/>
    /// class.
    /// </summary>
    public CoursePart() {}

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.Courses.CoursePart"/>
    /// class using the supplied values.
    /// </summary>
    /// <param name="course_ID">The identification of the <see cref="StudyPlanning.DAL.Courses.Course"/>.</param>
    /// <param name="part">The number of the course part.</param>
    public CoursePart(Guid course_ID, int part)
    {
      this._Course_ID = course_ID;
      this._Part = part;
    }

    #endregion //Constructors

  }
}
```

1.6.1 CourseGrab

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.Courses
{
  /// <summary>
  /// Represents a course version.
  /// </summary>
  public class CourseGrab : StudyPlanning.DAL.DbObject
  {
    #region Private Properties
    private DalGuid _CourseGrab_ID = new DalGuid(false);
    private DalGuid _CourseVersion_ID = new DalGuid(false);
    private DalGuid _Course_ID = new DalGuid(false);
    private DalString _Number = new DalString(false);
    private DalInt _DtuVersion = new DalInt(true);
    private DalStringLocalizable _Schedule = new DalStringLocalizable(true);
    private DalString _Duration = new DalString(true);
    private DalStringLocalizable _ExaminationPlacement = new DalStringLocalizable(true);
    private DalStringLocalizable _ExaminationAids = new DalStringLocalizable(true);
    private DalString _PreviousCourseNumbers = new DalString(true);
    private DalString _MandatoryPrerequisites = new DalString(true);
    private DalString _TechnicalPrerequisites = new DalString(true);
    private DalString _DesirablePrerequisites = new DalString(true);
    private DalString _ExternalInstitutions = new DalString(true);
    private DalDateTime _LastUpdated = new DalDateTime(true);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Initializes a new instance of the <see cref="StudyPlanning.DAL.Courses.CourseGrab"/>
    /// class.
    /// </summary>
    public CourseGrab()
    {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the ID of the course grab.
    /// </summary>
    public DalGuid CourseGrab_ID
    {
      get { return _CourseGrab_ID; }
    }

    /// <summary>
    /// Gets the ID of the course version.
    /// </summary>
    public DalGuid CourseVersion_ID
    {
      get { return _CourseVersion_ID; }
    }
```

```
/// <summary>
/// Gets the ID of the course.
/// </summary>
public DalGuid Course_ID
{
  get { return _Course_ID; }
}

/// <summary>
/// Gets the number of the course.
/// </summary>
public DalString Number
{
  get { return _Number; }
}

/// <summary>
/// Gets the DTU version number of the course.
/// </summary>
public DalInt DtuVersion
{
  get { return _DtuVersion; }
}

/// <summary>
/// Gets the Schedule attribute.
/// </summary>
public DalStringLocalizable Schedule
{
  get { return _Schedule; }
}

/// <summary>
/// Gets the Duration attribute.
/// </summary>
public DalString Duration
{
  get { return _Duration; }
}

/// <summary>
/// Gets the ExaminationPlacement attribute.
/// </summary>
public DalStringLocalizable ExaminationPlacement
{
  get { return _ExaminationPlacement; }
}

/// <summary>
/// Gets the ExaminationAids attribute.
/// </summary>
public DalStringLocalizable ExaminationAids
{
  get { return _ExaminationAids; }
}

/// <summary>
/// Gets the PreviousCourseNumbers attribute.
/// </summary>
public DalString PreviousCourseNumbers
{
  get { return _PreviousCourseNumbers; }
}

/// <summary>
/// Gets the MandatoryPrerequisites attribute.
/// </summary>
public DalString MandatoryPrerequisites
{
  get { return _MandatoryPrerequisites; }
}

/// <summary>
/// Gets the TechnicalPrerequisites attribute.
/// </summary>
public DalString TechnicalPrerequisites
{
  get { return _TechnicalPrerequisites; }
}

/// <summary>
/// Gets the DesirablePrerequisites attribute.
/// </summary>
public DalString DesirablePrerequisites
{
  get { return _DesirablePrerequisites; }
}

/// <summary>
/// Gets the ExternalInstitutions attribute.
/// </summary>
public DalString ExternalInstitutions
{
  get { return _ExternalInstitutions; }
}

/// <summary>
/// Gets the LastUpdated attribute.
```

```
/// </summary>
public DalDateTime LastUpdated
{
  get { return _LastUpdated; }
}

/// <summary>
/// Gets the data log of the course.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Methods

/// <summary>
/// Creates a new course grab in the database using the values of the properties
/// of the current instance.
/// </summary>
public void Create()
{
  try
  {
    this.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }

  SqlCommand objCommand = new SqlCommand("CourseGrab_Insert", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, false);

  try
  {
    objConnection.Open();
    objCommand.ExecuteNonQuery();
    objConnection.Close();
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
}

#region Retrieve Methods

/// <summary>
/// Retrieves the course grab from the database using the value of the
/// CourseGrab_ID property of the current instance.
/// </summary>
public void Retrieve()
{
  try
  {
    this.CourseGrab_ID.Validate("CourseGrab_ID");
  }
  catch (DalException excp)
  {
    throw excp;
  }

  try
  {
    RetrieveExecute(CourseGrab_ID.Value, this, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
}

/// <summary>
/// Retrieves the course grab with the specified identifier from the database.
/// </summary>
/// <param name="courseGrab_ID">The globally unique identifier of the course grab.</param>
public static CourseGrab Retrieve(Guid courseGrab_ID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  CourseGrab objCg = new CourseGrab();

  try
  {
    RetrieveExecute(courseGrab_ID, objCg, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }

  return objCg;
}
```

```
private static void RetrieveExecute(
  Guid courseGrab_ID,
  CourseGrab objCg,
  SqlConnection sqlConnection)
{
  SqlCommand objCommand = new SqlCommand("CourseGrab_Select", sqlConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  SqlParameter objParam;

  objParam = objCommand.Parameters.Add("@CourseGrab_ID", SqlDbType.UniqueIdenti↓er, 16);
  objParam.Value = courseGrab_ID;

  DataSet objDataSet = new DataSet("CourseGrab");
  SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

  try
  {
    sqlConnection.Open();
    objAdap.Fill(objDataSet, "CourseGrab");
    sqlConnection.Close();
    DataRow objDataRow = objDataSet.Tables["CourseGrab"].Rows[0];

    objCg.CourseGrab_ID.Value = (Guid)objDataRow["CourseGrab_ID"];
    objCg.CourseVersion_ID.Value = (Guid)objDataRow["CourseVersion_ID"];
    objCg.Course_ID.Value = (Guid)objDataRow["Course_ID"];
    objCg.Number.Value = (string)objDataRow["Number"];

    if (objDataRow["DtuVersion"].Equals(System.DBNull.Value))
      objCg.DtuVersion.IsNull = true;
    else
      objCg.DtuVersion.Value = (int)objDataRow["DtuVersion"];

    if (objDataRow["Schedule"].Equals(System.DBNull.Value))
      objCg.Schedule.IsNull = true;
    else
      objCg.Schedule.LoadXml((string)objDataRow["Schedule"]);

    if (objDataRow["Duration"].Equals(System.DBNull.Value))
      objCg.Duration.IsNull = true;
    else
      objCg.Duration.Value = (string)objDataRow["Duration"];

    if (objDataRow["ExaminationPlacement"].Equals(System.DBNull.Value))
      objCg.ExaminationPlacement.IsNull = true;
    else
      objCg.ExaminationPlacement.LoadXml((string)objDataRow["ExaminationPlacement"]);

    if (objDataRow["ExaminationAids"].Equals(System.DBNull.Value))
      objCg.ExaminationAids.IsNull = true;
    else
      objCg.ExaminationAids.LoadXml((string)objDataRow["ExaminationAids"]);

    if (objDataRow["PreviousCourseNumbers"].Equals(System.DBNull.Value))
      objCg.PreviousCourseNumbers.IsNull = true;
    else
      objCg.PreviousCourseNumbers.Value = (string)objDataRow["PreviousCourseNumbers"];

    if (objDataRow["MandatoryPrerequisites"].Equals(System.DBNull.Value))
      objCg.MandatoryPrerequisites.IsNull = true;
    else
      objCg.MandatoryPrerequisites.Value = (string)objDataRow["MandatoryPrerequisites"];

    if (objDataRow["TechnicalPrerequisites"].Equals(System.DBNull.Value))
      objCg.TechnicalPrerequisites.IsNull = true;
    else
      objCg.TechnicalPrerequisites.Value = (string)objDataRow["TechnicalPrerequisites"];

    if (objDataRow["DesirablePrerequisites"].Equals(System.DBNull.Value))
      objCg.DesirablePrerequisites.IsNull = true;
    else
      objCg.DesirablePrerequisites.Value = (string)objDataRow["DesirablePrerequisites"];

    if (objDataRow["ExternalInstitutions"].Equals(System.DBNull.Value))
      objCg.ExternalInstitutions.IsNull = true;
    else
      objCg.ExternalInstitutions.Value = (string)objDataRow["ExternalInstitutions"];

    if (objDataRow["LastUpdated"].Equals(System.DBNull.Value))
      objCg.LastUpdated.IsNull = true;
    else
      objCg.LastUpdated.Value = (DateTime)objDataRow["LastUpdated"];

    objCg.Log.SetValues(objDataRow);
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
}

#endregion //Retrive Methods

/// <summary>
/// Updates the current course grab in the database using the original
/// course grab to resolve possible concurrency issues.
/// </summary>
/// <param name="originalCourseGrab">The original <see cref="StudyPlanning.DAL.Courses.CourseGrab"/> object.</param>
```

```
/// <returns><strong>true</strong> if the update was succesfully executed; otherwise, <strong>false</strong>.</returns>
public bool Update(CourseGrab originalCourseGrab)
{
  try
  {
    this.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }

  bool blnResult = false;

  SqlCommand objCommand = new SqlCommand("CourseGrab_Update", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, false);
  AddParameters(originalCourseGrab, objCommand, true);

  int rowsA↑ected;

  try
  {
    objConnection.Open();
    rowsA↑ected = objCommand.ExecuteNonQuery();
    objConnection.Close();

    if (rowsA↑ected > 0)
      blnResult = true;
  }
  catch(SqlException excp)
  {
    throw new DalException(excp);
  }

  return blnResult;
}

/// <summary>
/// Deletes the current course grab from the database.
/// </summary>
/// <returns><strong>true</strong> if the deletion is succesfully executed; otherwise, <strong>false</strong>.</returns>
public bool Delete()
{
  bool blnResult = false;

  try
  {
    this.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }

  SqlCommand objCommand = new SqlCommand("CourseGrab_Delete", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, true);

  try
  {
    objConnection.Open();
    int rowsA↑ected = objCommand.ExecuteNonQuery();
    objConnection.Close();

    if (rowsA↑ected > 0)
      blnResult = true;
  }
  catch(SqlException objEx)
  {
    throw new DalException(objEx);
  }

  return blnResult;
}

/// <summary>
/// Adds the relevant parameters to the specified <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see> object
/// setting the values to ones of the specified <see cref="StudyPlanning.DAL.Courses.CourseGrab"/> object.
/// </summary>
/// <param name="courseGrab">
/// The <see cref="StudyPlanning.DAL.Courses.CourseGrab"/> object containing the values that the parameters have to be
/// initialized to.
/// </param>
/// <param name="objCommand">
/// The <see cref="System.Data.SqlClient.SqlCommand"/> object to which parameters should be added.
/// </param>
/// <param name="isOriginal">
/// <strong>true</strong> if the parameters to be added represent original
/// data (for handling concurrent programming issues); otherwise, <strong>false</strong>.</param>
/// <returns>The <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see>
/// object with the relevant parameters added.</returns>
private static void AddParameters(CourseGrab courseGrab, SqlCommand objCommand, bool isOriginal)
{
  string paramName = "";
  SqlParameter objParam;
```

```
//CourseGrab_ID
paramName = "CourseGrab_ID";
objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

if (isOriginal)
  objParam.ParameterName = "@Original_" + paramName;

objParam.Value = courseGrab.CourseGrab_ID.Value;

//CourseVersion_ID
paramName = "CourseVersion_ID";
objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

if (isOriginal)
  objParam.ParameterName = "@Original_" + paramName;

objParam.Value = courseGrab.CourseVersion_ID.Value;

//Course_ID
paramName = "Course_ID";
objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

if (isOriginal)
  objParam.ParameterName = "@Original_" + paramName;

objParam.Value = courseGrab.Course_ID.Value;

//Number
paramName = "Number";
objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.VarChar, 20);

if (isOriginal)
  objParam.ParameterName = "@Original_" + paramName;

objParam.Value = courseGrab.Number.Value;

//DtuVersion
paramName = "DtuVersion";
objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.Int, 4);

if (isOriginal)
  objParam.ParameterName = "@Original_" + paramName;

if (courseGrab.DtuVersion.IsNull)
  objParam.Value = System.DBNull.Value;
else
  objParam.Value = courseGrab.DtuVersion.Value;

//Schedule
paramName = "Schedule";
objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.Text, 16);

if (isOriginal)
  objParam.ParameterName = "@Original_" + paramName;

if (courseGrab.Schedule.IsNull)
  objParam.Value = System.DBNull.Value;
else
  objParam.Value = courseGrab.Schedule.Value;

//Duration
paramName = "Duration";
objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.Text, 16);

if (isOriginal)
  objParam.ParameterName = "@Original_" + paramName;

if (courseGrab.Duration.IsNull)
  objParam.Value = System.DBNull.Value;
else
  objParam.Value = courseGrab.Duration.Value;

//ExaminationPlacement
paramName = "ExaminationPlacement";
objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.Text, 16);

if (isOriginal)
  objParam.ParameterName = "@Original_" + paramName;

if (courseGrab.ExaminationPlacement.IsNull)
  objParam.Value = System.DBNull.Value;
else
  objParam.Value = courseGrab.ExaminationPlacement.Value;

//ExaminationAids
paramName = "ExaminationAids";
objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.Text, 16);

if (isOriginal)
  objParam.ParameterName = "@Original_" + paramName;

if (courseGrab.ExaminationAids.IsNull)
  objParam.Value = System.DBNull.Value;
else
  objParam.Value = courseGrab.ExaminationAids.Value;

//PreviousCourseNumbers
paramName = "PreviousCourseNumbers";
objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.Text, 16);
```

```
    if (isOriginal)
      objParam.ParameterName = "@Original_" + paramName;

    if (courseGrab.PreviousCourseNumbers.IsNull)
      objParam.Value = System.DBNull.Value;
    else
      objParam.Value = courseGrab.PreviousCourseNumbers.Value;

    //MandatoryPrerequisites
    paramName = "MandatoryPrerequisites";
    objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.Text, 16);

    if (isOriginal)
      objParam.ParameterName = "@Original_" + paramName;

    if (courseGrab.MandatoryPrerequisites.IsNull)
      objParam.Value = System.DBNull.Value;
    else
      objParam.Value = courseGrab.MandatoryPrerequisites.Value;

    //TechnicalPrerequisites
    paramName = "TechnicalPrerequisites";
    objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.Text, 16);

    if (isOriginal)
      objParam.ParameterName = "@Original_" + paramName;

    if (courseGrab.TechnicalPrerequisites.IsNull)
      objParam.Value = System.DBNull.Value;
    else
      objParam.Value = courseGrab.TechnicalPrerequisites.Value;

    //DesirablePrerequisites
    paramName = "DesirablePrerequisites";
    objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.Text, 16);

    if (isOriginal)
      objParam.ParameterName = "@Original_" + paramName;

    if (courseGrab.DesirablePrerequisites.IsNull)
      objParam.Value = System.DBNull.Value;
    else
      objParam.Value = courseGrab.DesirablePrerequisites.Value;

    //ExternalInstitutions
    paramName = "ExternalInstitutions";
    objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.Text, 16);

    if (isOriginal)
      objParam.ParameterName = "@Original_" + paramName;

    if (courseGrab.ExternalInstitutions.IsNull)
      objParam.Value = System.DBNull.Value;
    else
      objParam.Value = courseGrab.ExternalInstitutions.Value;

    //LastUpdated
    paramName = "LastUpdated";
    objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.DateTime, 8);

    if (isOriginal)
      objParam.ParameterName = "@Original_" + paramName;

    if (courseGrab.LastUpdated.IsNull)
      objParam.Value = System.DBNull.Value;
    else
      objParam.Value = courseGrab.LastUpdated.Value;

    //Log
    objCommand = courseGrab.Log.AddParameters(objCommand, isOriginal);
  }

  /// <summary>
  /// Validates the current <see cref="StudyPlanning.DAL.Courses.CourseGrab"/> object. If the
  /// value of either of the private properties is null a
  /// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
  /// </summary>
  private void Validate()
  {
    try
    {
      _CourseGrab_ID.Validate("CourseGrab_ID");
      _CourseVersion_ID.Validate("CourseVersion_ID");
      _Course_ID.Validate("Course_ID");
      _Number.Validate("Number");
      _DtuVersion.Validate("DtuVersion");
      _Schedule.Validate("Schedule");
      _Duration.Validate("Duration");
      _ExaminationPlacement.Validate("ExaminationPlacement");
      _ExaminationAids.Validate("ExaminationAids");
      _PreviousCourseNumbers.Validate("PreviousCourseNumbers");
      _MandatoryPrerequisites.Validate("MandatoryPrerequisites");
      _TechnicalPrerequisites.Validate("TechnicalPrerequisites");
      _DesirablePrerequisites.Validate("DesirablePrerequisites");
      _ExternalInstitutions.Validate("ExternalInstitutions");
      _LastUpdated.Validate("LastUpdated");
      _Log.Validate();
    }
```

```
      catch (DalException excp)
      {
        throw excp;
      }
    }

    #endregion //Methods
  }
}
```

1.6.2 CourseVersion

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.Courses
{
  /// <summary>
  /// Represents a course version.
  /// </summary>
  public class CourseVersion : StudyPlanning.DAL.DbObject
  {
    #region Private Properties
    private DalGuid _CourseVersion_ID = new DalGuid(false);
    private DalGuid _Course_ID = new DalGuid(false);
    private DalInt _Version = new DalInt(false);
    private DalString _Number = new DalString(false);
    private DalStringLocalizable _Name = new DalStringLocalizable(false);
    private DalString _Language_ID = new DalString(false);
    private DalStringLocalizable _TeachingForm = new DalStringLocalizable(true);
    private DalInt _Parts = new DalInt(false);
    private DalInt _AssessmentType_ID = new DalInt(false);
    private DalStringLocalizable _EvaluationFormDescription =
      new DalStringLocalizable(true);
    private DalInt _ParticipantLimitationMin = new DalInt(true);
    private DalInt _ParticipantLimitationMax = new DalInt(true);
    private DalStringLocalizable _Objective = new DalStringLocalizable(true);
    private DalStringLocalizable _Contents = new DalStringLocalizable(false);
    private DalStringLocalizable _Remark = new DalStringLocalizable(true);
    private DalString _Url = new DalString(true);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Initializes a new instance of the <see cref="StudyPlanning.DAL.Courses.CourseVersion"/>
    /// class.
    /// </summary>
    public CourseVersion()
    {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the ID of the course version.
    /// </summary>
    public DalGuid CourseVersion_ID
    {
      get { return _CourseVersion_ID; }
    }

    /// <summary>
    /// Gets the ID of the course.
    /// </summary>
    public DalGuid Course_ID
    {
      get { return _Course_ID; }
    }

    /// <summary>
    /// Gets the version number of the course.
    /// </summary>
    public DalInt Version
    {
      get { return _Version; }
    }

    /// <summary>
    /// Gets the number of the course.
    /// </summary>
    public DalString Number
    {
      get { return _Number; }
    }

    /// <summary>
    /// Gets the name of the course.
    /// </summary>
```

```
public DalStringLocalizable Name
{
  get { return _Name; }
}

/// <summary>
/// Gets the language ID of the course.
/// </summary>
public DalString Language_ID
{
  get { return _Language_ID; }
}

/// <summary>
/// Gets the teaching form of the course.
/// </summary>
/// <value>A string containing a description of the teaching form of the course.</value>
public DalStringLocalizable TeachingForm
{
  get { return _TeachingForm; }
}

/// <summary>
/// Gets the number of parts in which the course is taught.
/// </summary>
public DalInt Parts
{
  get { return _Parts; }
}

/// <summary>
/// Gets the assessment ID of the course.
/// </summary>
public DalInt AssessmentType_ID
{
  get { return _AssessmentType_ID; }
}

/// <summary>
/// Gets the evaluation form description of the course.
/// </summary>
public DalStringLocalizable EvaluationFormDescription
{
  get { return _EvaluationFormDescription; }
}

/// <summary>
/// Gets the minimum number of participants of the course.
/// </summary>
public DalInt ParticipantLimitationMin
{
  get { return _ParticipantLimitationMin; }
}

/// <summary>
/// Gets the maximum number of participants of the course.
/// </summary>
public DalInt ParticipantLimitationMax
{
  get { return _ParticipantLimitationMax; }
}

/// <summary>
/// Gets the objective description of the course.
/// </summary>
public DalStringLocalizable Objective
{
  get { return _Objective; }
}

/// <summary>
/// Gets the contents description of the course.
/// </summary>
public DalStringLocalizable Contents
{
  get { return _Contents; }
}

/// <summary>
/// Gets the remarks of the course.
/// </summary>
public DalStringLocalizable Remark
{
  get { return _Remark; }
}

/// <summary>
/// Gets the URL of the course.
/// </summary>
public DalString Url
{
  get { return _Url; }
}

/// <summary>
/// Gets the data log of the course.
/// </summary>
public DataLog Log
{
```

```
    get { return _Log; }
  }

  #endregion //Public Properties

  #region Methods

  /// <summary>
  /// Creates a new course version in the database using the values of the properties
  /// of the current instance.
  /// </summary>
  public void Create()
  {
    try
    {
      this.Validate();
    }
    catch (DalException excp)
    {
      throw excp;
    }

    SqlCommand objCommand = new SqlCommand("CourseVersion_Insert", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, false);

    try
    {
      objConnection.Open();
      objCommand.ExecuteNonQuery();
      objConnection.Close();
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
  }

  #region Retrieve Methods

  /// <summary>
  /// Retrieves the course version from the database using the value of the
  /// CourseVersion_ID property of the current instance.
  /// </summary>
  public void Retrieve()
  {
    try
    {
      this.CourseVersion_ID.Validate("CourseVersion");
    }
    catch (DalException excp)
    {
      throw excp;
    }

    try
    {
      RetrieveExecute(CourseVersion_ID.Value, this, objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }
  }

  /// <summary>
  /// Retrieves the course version with the specified identifier from the database.
  /// </summary>
  /// <param name="courseVersion_ID">The globally unique identifier of the course version.</param>
  public static CourseVersion Retrieve(Guid courseVersion_ID)
  {
    string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
    SqlConnection objConnection = new SqlConnection(connString);

    CourseVersion objCV = new CourseVersion();

    try
    {
      RetrieveExecute(courseVersion_ID, objCV, objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }

    return objCV;
  }

  private static void RetrieveExecute(
    Guid courseVersion_ID,
    CourseVersion objCv,
    SqlConnection sqlConnection)
  {

    SqlCommand objCommand = new SqlCommand("CourseVersion_Select", sqlConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam;
```

```
    objParam = objCommand.Parameters.Add("@CourseVersion_ID", SqlDbType.UniqueIdenti↓er, 16);
    objParam.Value = courseVersion_ID;

    DataSet objDataSet = new DataSet("CourseVersion");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
      sqlConnection.Open();
      objAdap.Fill(objDataSet, "CourseVersion");
      sqlConnection.Close();
      DataRow objDataRow = objDataSet.Tables["CourseVersion"].Rows[0];

      objCv.CourseVersion_ID.Value = (Guid)objDataRow["CourseVersion_ID"];
      objCv.Course_ID.Value = (Guid)objDataRow["Course_ID"];
      objCv.Version.Value = (int)objDataRow["Version"];
      objCv.Number.Value = (string)objDataRow["Number"];
      objCv.Name.LoadXml(objDataRow["Name"].ToString());
      objCv.Language_ID.Value = (string)objDataRow["Language_ID"];

      if (objDataRow["TeachingForm"].Equals(System.DBNull.Value))
        objCv.TeachingForm.IsNull = true;
      else
        objCv.TeachingForm.LoadXml(objDataRow["TeachingForm"].ToString());

      objCv.Parts.Value = (int)objDataRow["Parts"];
      objCv.AssessmentType_ID.Value = (int)objDataRow["AssessmentType_ID"];

      if (objDataRow["EvaluationFormDescription"].Equals(System.DBNull.Value))
        objCv.EvaluationFormDescription.IsNull = true;
      else
        objCv.EvaluationFormDescription.LoadXml(objDataRow["EvaluationFormDescription"].ToString());

      if (objDataRow["ParticipantLimitationMin"].Equals(System.DBNull.Value))
        objCv.ParticipantLimitationMin.IsNull = true;
      else
        objCv.ParticipantLimitationMin.Value = (int)objDataRow["ParticipantLimitationMin"];

      if (objDataRow["ParticipantLimitationMax"].Equals(System.DBNull.Value))
        objCv.ParticipantLimitationMax.IsNull = true;
      else
        objCv.ParticipantLimitationMax.Value = (int)objDataRow["ParticipantLimitationMax"];

      if (objDataRow["Objective"].Equals(System.DBNull.Value))
        objCv.Remark.IsNull = true;
      else
        objCv.Remark.LoadXml(objDataRow["Objective"].ToString());

      objCv.Contents.LoadXml(objDataRow["Contents"].ToString());

      if (objDataRow["Remark"].Equals(System.DBNull.Value))
        objCv.Remark.IsNull = true;
      else
        objCv.Remark.LoadXml(objDataRow["Remark"].ToString());

      if (objDataRow["Url"].Equals(System.DBNull.Value))
        objCv.Url.IsNull = true;
      else
        objCv.Url.Value = (string)objDataRow["Url"];

      objCv.Log.SetValues(objDataRow);
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
  }

  #endregion //Retrive Methods

  /// <summary>
  /// Updates the current course version in the database using the original
  /// course to resolve possible concurrency issues.
  /// </summary>
  /// <param name="originalCourseVersion">The original <see cref="StudyPlanning.DAL.Courses.CourseVersion"/> object.</param>
  /// <returns><strong>true</strong> if the update was succesfully executed; otherwise, <strong>false</strong>.</returns>
  public bool Update(CourseVersion originalCourseVersion)
  {
    try
    {
      this.Validate();
    }
    catch (DalException excp)
    {
      throw excp;
    }

    bool blnResult = false;

    SqlCommand objCommand = new SqlCommand("CourseVersion_Update", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, false);
    AddParameters(originalCourseVersion, objCommand, true);

    int rowsA↑ected;

    try
```

```
    {
      objConnection.Open();
      rowsA↑ected = objCommand.ExecuteNonQuery();
      objConnection.Close();

      if (rowsA↑ected > 0)
        blnResult = true;
    }
    catch(SqlException excp)
    {
      throw new DalException(excp);
    }

    return blnResult;
  }

  /// <summary>
  /// Deletes the current course version from the database.
  /// </summary>
  /// <returns><strong>true</strong> if the deletion is succesfully executed; otherwise, <strong>false</strong>.</returns>
  public bool Delete()
  {
    bool blnResult = false;

    try
    {
      this.Validate();
    }
    catch (DalException excp)
    {
      throw excp;
    }

    SqlCommand objCommand = new SqlCommand("CourseVersion_Delete", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, true);

    try
    {
      objConnection.Open();
      int rowsA↑ected = objCommand.ExecuteNonQuery();
      objConnection.Close();

      if (rowsA↑ected > 0)
        blnResult = true;
    }
    catch(SqlException objEx)
    {
      throw new DalException(objEx);
    }

    return blnResult;
  }

  /// <summary>
  /// Adds the relevant parameters to the specified <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see> object
  /// setting the values to ones of the specified <see cref="StudyPlanning.DAL.Courses.CourseVersion"/> object.
  /// </summary>
  /// <param name="courseVersion">
  /// The <see cref="StudyPlanning.DAL.Courses.CourseVersion"/> object containing the values that the parameters have to be
  /// initialized to.
  /// </param>
  /// <param name="objCommand">
  /// The <see cref="System.Data.SqlClient.SqlCommand"/> object to which parameters should be added.
  /// </param>
  /// <param name="isOriginal">
  /// <strong>true</strong> if the parameters to be added represent original
  /// data (for handling concurrent programming issues); otherwise, <strong>false</strong>.</param>
  /// <returns>The <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see>
  /// object with the relevant parameters added.</returns>
  private static void AddParameters(CourseVersion courseVersion, SqlCommand objCommand, bool isOriginal)
  {
    string paramName = "";
    SqlParameter objParam;

    //CourseVersion_ID
    paramName = "CourseVersion_ID";
    objParam = objCommand.Parameters.Add("@CourseVersion_ID", SqlDbType.UniqueIdenti↓er, 16);

    if (isOriginal)
      objParam.ParameterName = "@Original_" + paramName;

    objParam.Value = courseVersion.CourseVersion_ID.Value;

    //Course_ID
    paramName = "Course_ID";
    objParam = objCommand.Parameters.Add("@Course_ID", SqlDbType.UniqueIdenti↓er, 16);

    if (isOriginal)
      objParam.ParameterName = "@Original_" + paramName;

    objParam.Value = courseVersion.Course_ID.Value;

    //Version
    paramName = "Version";
    objParam = objCommand.Parameters.Add("@Version", SqlDbType.Int, 4);

    if (isOriginal)
```

```
    objParam.ParameterName = "@Original_" + paramName;

objParam.Value = courseVersion.Version.Value;

//Number
paramName = "Number";
objParam = objCommand.Parameters.Add("@Number", SqlDbType.VarChar, 20);

if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

objParam.Value = courseVersion.Number.Value;

//Name
paramName = "Name";
objParam = objCommand.Parameters.Add("@Name", SqlDbType.Text);

if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

objParam.Value = courseVersion.Name.Value;

//Language_ID
paramName = "Language_ID";
objParam = objCommand.Parameters.Add("@Language_ID", SqlDbType.Char, 2);

if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

objParam.Value = courseVersion.Language_ID.Value;

//TeachingForm
paramName = "TeachingForm";
objParam = objCommand.Parameters.Add("@TeachingForm", SqlDbType.Text);

if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

objParam.Value = courseVersion.TeachingForm.Value;

//Parts
paramName = "Parts";
objParam = objCommand.Parameters.Add("@Parts", SqlDbType.Int, 4);

if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

objParam.Value = courseVersion.Parts.Value;

//AssessmentType_ID
paramName = "AssessmentType_ID";
objParam = objCommand.Parameters.Add("@AssessmentType_ID", SqlDbType.Int, 4);

if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

objParam.Value = courseVersion.AssessmentType_ID.Value;

//EvaluationFormDescription
paramName = "EvaluationFormDescription";
objParam = objCommand.Parameters.Add("@EvaluationFormDescription", SqlDbType.Text);

if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

if (courseVersion.EvaluationFormDescription.IsNull)
    objParam.Value = System.DBNull.Value;
else
    objParam.Value = courseVersion.EvaluationFormDescription.Value;

//ParticipantLimitationMin
paramName = "ParticipantLimitationMin";
objParam = objCommand.Parameters.Add("@ParticipantLimitationMin", SqlDbType.Int, 4);

if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

if (courseVersion.ParticipantLimitationMin.IsNull)
    objParam.Value = System.DBNull.Value;
else
    objParam.Value = courseVersion.ParticipantLimitationMin.Value;

//ParticipantLimitationMax
paramName = "ParticipantLimitationMax";
objParam = objCommand.Parameters.Add("@ParticipantLimitationMax", SqlDbType.Int, 4);

if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

if (courseVersion.ParticipantLimitationMax.IsNull)
    objParam.Value = System.DBNull.Value;
else
    objParam.Value = courseVersion.ParticipantLimitationMax.Value;

//Objective
paramName = "Objective";
objParam = objCommand.Parameters.Add("@Objective", SqlDbType.Text);

if (isOriginal)
```

```
    objParam.ParameterName = "@Original_" + paramName;

  if (courseVersion.Objective.IsNull)
    objParam.Value = System.DBNull.Value;
  else
    objParam.Value = courseVersion.Objective.Value;

  //Contents
  paramName = "Contents";
  objParam = objCommand.Parameters.Add("@Contents", SqlDbType.Text);

  if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

  objParam.Value = courseVersion.Contents.Value;

  //Remark
  paramName = "Remark";
  objParam = objCommand.Parameters.Add("@Remark", SqlDbType.Text);

  if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

  if (courseVersion.Remark.IsNull)
    objParam.Value = System.DBNull.Value;
  else
    objParam.Value = courseVersion.Remark.Value;

  //Url
  paramName = "Url";
  objParam = objCommand.Parameters.Add("@Url", SqlDbType.VarChar, 100);

  if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

  if (courseVersion.Url.IsNull)
    objParam.Value = System.DBNull.Value;
  else
    objParam.Value = courseVersion.Url.Value;

  //Log
  objCommand = courseVersion.Log.AddParameters(objCommand, isOriginal);
}

/// <summary>
/// Validates the current <see cref="StudyPlanning.DAL.Courses.CourseVersion"/> object. If the
/// value of either of the private properties is null a
/// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
/// </summary>
private void Validate()
{
  try
  {
    _CourseVersion_ID.Validate("CourseVersion_ID");
    _Course_ID.Validate("Course_ID");
    _Version.Validate("Version");
    _Number.Validate("Number");
    _Name.Validate("Name");
    _Language_ID.Validate("Language_ID");
    _TeachingForm.Validate("TeachingForm");
    _Parts.Validate("Parts");
    _AssessmentType_ID.Validate("AssessmentType_ID");
    _EvaluationFormDescription.Validate("EvaluationFormDescription");
    _ParticipantLimitationMin.Validate("ParticipantLimitationMin");
    _ParticipantLimitationMax.Validate("ParticipantLimitationMax");
    _Objective.Validate("Objective");
    _Contents.Validate("Contents");
    _Remark.Validate("Remark");
    _Url.Validate("Url");
    _Log.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }
}

/// <summary>
/// Retrives the course version ID of the newest version of the specified
/// course ID.
/// </summary>
/// <param name="course_ID">The course for which to retrieve the course version ID of the newest version.</param>
/// <returns>A course version ID of the newest version.</returns>
public static Guid GetNewestVersion(Guid course_ID)
{
  Guid courseVersion_ID;

  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  SqlCommand objCommand = new SqlCommand("CourseVersion_GetNewestVersionFromCourseID", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  SqlParameter objParam;

  objParam = objCommand.Parameters.Add("@Course_ID", SqlDbType.UniqueIdenti↓er, 16);
  objParam.Value = course_ID;

  objParam = objCommand.Parameters.Add("@CourseVersion_ID", SqlDbType.UniqueIdenti↓er, 16);
```

```
      objParam.Direction = ParameterDirection.Output;

      try
      {
        objConnection.Open();
        objCommand.ExecuteNonQuery();
        courseVersion_ID = (Guid)objCommand.Parameters["@CourseVersion_ID"].Value;
        objConnection.Close();
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }

      return courseVersion_ID;
    }

    /// <summary>
    /// Retrieves the course versions of the specified course.
    /// </summary>
    /// <param name="course_ID">The course for which to retrieve course versions.</param>
    /// <returns>An array containing IDs of the course versions of the specified course.</returns>
    public static Guid[] GetVersions(Guid course_ID)
    {
      Guid[] versions;

      string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
      SqlConnection objConnection = new SqlConnection(connString);

      SqlCommand objCommand = new SqlCommand("CourseVersion_GetVersions", objConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      SqlParameter objParam;

      objParam = objCommand.Parameters.Add("@Course_ID", SqlDbType.UniqueIdenti↓er, 16);
      objParam.Value = course_ID;

      DataSet objDataSet = new DataSet("Versions");
      SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

      try
      {
        objConnection.Open();
        objAdap.Fill(objDataSet, "Versions");
        objConnection.Close();

        int numRows = objDataSet.Tables["Versions"].Rows.Count;
        versions = new Guid[numRows];

        for(int i=0; i<numRows; i++)
        {
          versions[i] = (Guid)objDataSet.Tables["Versions"].Rows[i]["CourseVersion_ID"];
        }
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }

      return versions;
    }

    #endregion //Methods
  }
}
```

1.6.3 Department

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.Courses
{
  /// <summary>
  /// Represents a course department.
  /// </summary>
  public class Department : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalGuid _Course_Department_ID = new DalGuid(false);
    private DalGuid _Course_ID = new DalGuid(false);
    private DalInt _Department_ID = new DalInt(false);
    private DalBool _Responsible = new DalBool(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Initializes a new instance of the <see cref="StudyPlanning.DAL.Courses.Department"/>
    /// class.
```

```
/// </summary>
public Department()
{
  _Log = new DataLog();
}

#endregion

#region Public Properties

/// <summary>
/// Gets the ID of the course department.
/// </summary>
public DalGuid Course_Department_ID
{
  get { return _Course_Department_ID; }
}

/// <summary>
/// Gets the ID of the course version.
/// </summary>
public DalGuid Course_ID
{
  get { return _Course_ID; }
}

/// <summary>
/// Gets the ID of the department.
/// </summary>
public DalInt Department_ID
{
  get { return _Department_ID; }
}

/// <summary>
/// Gets the Responsible attribute.
/// </summary>
public DalBool Responsible
{
  get { return _Responsible; }
}

/// <summary>
/// Gets the data log of the course.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Methods

/// <summary>
/// Creates a new course department in the database using the values of the properties
/// of the current instance.
/// </summary>
public void Create()
{
  try
  {
    this.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }

  SqlCommand objCommand = new SqlCommand("Course_Department_Insert", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, false);

  try
  {
    objConnection.Open();
    objCommand.ExecuteNonQuery();
    objConnection.Close();
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
}

#region Retrieve Methods

/// <summary>
/// Retrieves the course department from the database using the value of the
/// Course_Department_ID property of the current instance.
/// </summary>
public void Retrieve()
{
  try
  {
    this._Course_Department_ID.Validate("Course_Department_ID");
  }
  catch (DalException excp)
```

```
    {
      throw excp;
    }

    try
    {
        RetrieveExecute(_Course_Department_ID.Value, this, objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }
}

/// <summary>
/// Retrieves the course department with the specified identifier from the database.
/// </summary>
/// <param name="course_Department_ID">The globally unique identifier of the course department.</param>
public static Department Retrieve(Guid course_Department_ID)
{
    string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
    SqlConnection objConnection = new SqlConnection(connString);

    Department objLec = new Department();

    try
    {
      RetrieveExecute(course_Department_ID, objLec, objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }

    return objLec;
}

private static void RetrieveExecute(
    Guid course_Department_ID,
    Department objDepartment,
    SqlConnection sqlConnection)
{
    SqlCommand objCommand = new SqlCommand("Course_Department_Select", sqlConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam;

    objParam = objCommand.Parameters.Add("@Course_Department_ID", SqlDbType.UniqueIdenti↓er, 16);
    objParam.Value = course_Department_ID;

    DataSet objDataSet = new DataSet("Course_Department");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
      sqlConnection.Open();
      objAdap.Fill(objDataSet, "Course_Department");
      sqlConnection.Close();
      DataRow objDataRow = objDataSet.Tables["Course_Department"].Rows[0];

      objDepartment.Course_Department_ID.Value = (Guid)objDataRow["Course_Department_ID"];
      objDepartment.Course_ID.Value = (Guid)objDataRow["Course_ID"];
      objDepartment.Department_ID.Value = (int)objDataRow["Department_ID"];
      objDepartment.Responsible.Value = (bool)objDataRow["Responsible"];

      objDepartment.Log.SetValues(objDataRow);
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
}

#endregion //Retrive Methods

/// <summary>
/// Updates the current course department in the database using the original
/// course evaluation form to resolve possible concurrency issues.
/// </summary>
/// <param name="originalDepartment">The original <see cref="StudyPlanning.DAL.Courses.Department"/> object.</param>
/// <returns><strong>true</strong> if the update was succesfully executed; otherwise, <strong>false</strong>.</returns>
public bool Update(Department originalDepartment)
{
    try
    {
      this.Validate();
    }
    catch (DalException excp)
    {
      throw excp;
    }

    bool blnResult = false;

    SqlCommand objCommand = new SqlCommand("Course_Department_Update", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, false);
    AddParameters(originalDepartment, objCommand, true);
```

```
    int rowsA↑ected;

    try
    {
      objConnection.Open();
      rowsA↑ected = objCommand.ExecuteNonQuery();
      objConnection.Close();

      if (rowsA↑ected > 0)
        blnResult = true;
    }
    catch(SqlException excp)
    {
      throw new DalException(excp);
    }

    return blnResult;
  }

  /// <summary>
  /// Deletes the current course department form from the database.
  /// </summary>
  /// <returns><strong>true</strong> if the deletion is succesfully executed; otherwise, <strong>false</strong>.</returns>
  public bool Delete()
  {
    bool blnResult = false;

    try
    {
      this.Validate();
    }
    catch (DalException excp)
    {
      throw excp;
    }

    SqlCommand objCommand = new SqlCommand("Course_Department_Delete", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, true);

    try
    {
      objConnection.Open();
      int rowsA↑ected = objCommand.ExecuteNonQuery();
      objConnection.Close();

      if (rowsA↑ected > 0)
        blnResult = true;
    }
    catch(SqlException objEx)
    {
      throw new DalException(objEx);
    }

    return blnResult;
  }

  /// <summary>
  /// Retrieves the list of departments which offer the specified course.
  /// </summary>
  /// <param name="course_ID">The ID of the course for which to retrieve the
  /// list of departments.</param>
  /// <returns>An array containing IDs of departments.</returns>
  public static int[] GetDepartments(Guid course_ID)
  {
    int[] departments;

    string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
    SqlConnection objConnection = new SqlConnection(connString);

    SqlCommand objCommand = new SqlCommand("Course_Department_GetDepartments", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam;

    objParam = objCommand.Parameters.Add("@Course_ID", SqlDbType.UniqueIdenti↓er, 16);
    objParam.Value = course_ID;

    DataSet objDataSet = new DataSet("Departments");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
      objConnection.Open();
      objAdap.Fill(objDataSet, "Departments");
      objConnection.Close();

      int numRows = objDataSet.Tables["Departments"].Rows.Count;
      departments = new int[numRows];

      for(int i=0; i<numRows; i++)
      {
        departments[i] = (int)objDataSet.Tables["Departments"].Rows[i]["Department_ID"];
      }
    }
    catch (SqlException objExc)
    {
```

```
            throw new DalException(objExc);
        }

        return departments;
    }

    /// <summary>
    /// Adds the relevant parameters to the specified <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see> object
    /// setting the values to ones of the specified <see cref="StudyPlanning.DAL.Courses.Department"/> object.
    /// </summary>
    /// <param name="department">
    /// The <see cref="StudyPlanning.DAL.Courses.Department"/> object containing the values that the parameters have to be
    /// initialized to.
    /// </param>
    /// <param name="objCommand">
    /// The <see cref="System.Data.SqlClient.SqlCommand"/> object to which parameters should be added.
    /// </param>
    /// <param name="isOriginal">
    /// <strong>true</strong> if the parameters to be added represent original
    /// data (for handling concurrent programming issues); otherwise, <strong>false</strong>.</param>
    /// <returns>The <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see>
    /// object with the relevant parameters added.</returns>
    private static void AddParameters(Department department, SqlCommand objCommand, bool isOriginal)
    {
        string paramName = "";
        SqlParameter objParam;

        //Course_Department_ID
        paramName = "Course_Department_ID";
        objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

        if (isOriginal)
            objParam.ParameterName = "@Original_" + paramName;

        objParam.Value = department.Course_Department_ID.Value;

        //Course_ID
        paramName = "Course_ID";
        objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

        if (isOriginal)
            objParam.ParameterName = "@Original_" + paramName;

        objParam.Value = department.Course_ID.Value;

        //Department_ID
        paramName = "Department_ID";
        objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.Int, 4);

        if (isOriginal)
            objParam.ParameterName = "@Original_" + paramName;

        objParam.Value = department.Department_ID.Value;

        //Responsible
        paramName = "Responsible";
        objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.Bit, 1);

        if (isOriginal)
            objParam.ParameterName = "@Original_" + paramName;

        objParam.Value = department.Responsible.Value;

        //Log
        objCommand = department.Log.AddParameters(objCommand, isOriginal);
    }

    /// <summary>
    /// Validates the current <see cref="StudyPlanning.DAL.Courses.Department"/> object. If the
    /// value of either of the private properties is null a
    /// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
    /// </summary>
    private void Validate()
    {
        try
        {
            _Course_Department_ID.Validate("Course_Department_ID");
            _Course_ID.Validate("Course_ID");
            _Department_ID.Validate("Department_ID");
            _Responsible.Validate("Responsible");
            _Log.Validate();
        }
        catch (DalException excp)
        {
            throw excp;
        }
    }

    #endregion //Methods
  }
}
```

1.6.4 EvaluationForm

```
using System;
using System.Data;
```

```
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.Courses
{
  /// <summary>
  /// Represents a course evaluation form.
  /// </summary>
  public class EvaluationForm : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalGuid _Course_EvaluationForm_ID = new DalGuid(false);
    private DalGuid _CourseVersion_ID = new DalGuid(false);
    private DalInt _EvaluationForm_ID = new DalInt(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Initializes a new instance of the <see cref="StudyPlanning.DAL.Courses.EvaluationForm"/>
    /// class.
    /// </summary>
    public EvaluationForm()
    {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the ID of the course evaluation form.
    /// </summary>
    public DalGuid Course_EvaluationForm_ID
    {
      get { return _Course_EvaluationForm_ID; }
    }

    /// <summary>
    /// Gets the ID of the course version.
    /// </summary>
    public DalGuid CourseVersion_ID
    {
      get { return _CourseVersion_ID; }
    }

    /// <summary>
    /// Gets the ID of the evaluation form.
    /// </summary>
    public DalInt EvaluationForm_ID
    {
      get { return _EvaluationForm_ID; }
    }

    /// <summary>
    /// Gets the data log of the course.
    /// </summary>
    public DataLog Log
    {
      get { return _Log; }
    }

    #endregion //Public Properties

    #region Methods

    /// <summary>
    /// Creates a new course evaluation form in the database using the values of the properties
    /// of the current instance.
    /// </summary>
    public void Create()
    {
      try
      {
        this.Validate();
      }
      catch (DalException excp)
      {
        throw excp;
      }

      SqlCommand objCommand = new SqlCommand("Course_EvaluationForm_Insert", objConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      AddParameters(this, objCommand, false);

      try
      {
        objConnection.Open();
        objCommand.ExecuteNonQuery();
        objConnection.Close();
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
```

```
    }
  }

  #region Retrieve Methods

  /// <summary>
  /// Retrieves the course evaluation form from the database using the value of the
  /// Course_EvaluationForm_ID property of the current instance.
  /// </summary>
  public void Retrieve()
  {
    try
    {
      this._Course_EvaluationForm_ID.Validate("Course_EvaluationForm_ID");
    }
    catch (DalException excp)
    {
      throw excp;
    }

    try
    {
        RetrieveExecute(_Course_EvaluationForm_ID.Value, this, objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }
  }

  /// <summary>
  /// Retrieves the course evaluation form with the specified identifier from the database.
  /// </summary>
  /// <param name="course_EvaluationForm_ID">The globally unique identifier of the course evaluation form.</param>
  public static EvaluationForm Retrieve(Guid course_EvaluationForm_ID)
  {
    string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
    SqlConnection objConnection = new SqlConnection(connString);

    EvaluationForm objEf = new EvaluationForm();

    try
    {
      RetrieveExecute(course_EvaluationForm_ID, objEf, objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }

    return objEf;
  }

  private static void RetrieveExecute(
    Guid course_EvaluationForm_ID,
    EvaluationForm objEf,
    SqlConnection sqlConnection)
  {
    SqlCommand objCommand = new SqlCommand("Course_EvaluationForm_Select", sqlConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam;

    objParam = objCommand.Parameters.Add("@Course_EvaluationForm_ID", SqlDbType.UniqueIdenti↓er, 16);
    objParam.Value = course_EvaluationForm_ID;

    DataSet objDataSet = new DataSet("Course_EvaluationForm");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
      sqlConnection.Open();
      objAdap.Fill(objDataSet, "Course_EvaluationForm");
      sqlConnection.Close();
      DataRow objDataRow = objDataSet.Tables["Course_EvaluationForm"].Rows[0];

      objEf.Course_EvaluationForm_ID.Value = (Guid)objDataRow["Course_EvaluationForm_ID"];
      objEf.CourseVersion_ID.Value = (Guid)objDataRow["CourseVersion_ID"];
      objEf.EvaluationForm_ID.Value = (int)objDataRow["EvaluationForm_ID"];

      objEf.Log.SetValues(objDataRow);
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
  }

  #endregion //Retrive Methods

  /// <summary>
  /// Updates the current course evaluation form in the database using the original
  /// course evaluation form to resolve possible concurrency issues.
  /// </summary>
  /// <param name="originalEvaluationForm">The original <see cref="StudyPlanning.DAL.Courses.EvaluationForm"/> object.</param>
  /// <returns><strong>true</strong> if the update was succesfully executed; otherwise, <strong>false</strong>.</returns>
  public bool Update(EvaluationForm originalEvaluationForm)
  {
    try
```

```
    {
      this.Validate();
    }
    catch (DalException excp)
    {
      throw excp;
    }

    bool blnResult = false;

    SqlCommand objCommand = new SqlCommand("Course_EvaluationForm_Update", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, false);
    AddParameters(originalEvaluationForm, objCommand, true);

    int rowsA↑ected;

    try
    {
      objConnection.Open();
      rowsA↑ected = objCommand.ExecuteNonQuery();
      objConnection.Close();

      if (rowsA↑ected > 0)
        blnResult = true;
    }
    catch(SqlException excp)
    {
      throw new DalException(excp);
    }

    return blnResult;
  }

  /// <summary>
  /// Deletes the current course evaluation form from the database.
  /// </summary>
  /// <returns><strong>true</strong> if the deletion is succesfully executed; otherwise, <strong>false</strong>.</returns>
  public bool Delete()
  {
    bool blnResult = false;

    try
    {
      this.Validate();
    }
    catch (DalException excp)
    {
      throw excp;
    }

    SqlCommand objCommand = new SqlCommand("Course_EvaluationForm_Delete", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, true);

    try
    {
      objConnection.Open();
      int rowsA↑ected = objCommand.ExecuteNonQuery();
      objConnection.Close();

      if (rowsA↑ected > 0)
        blnResult = true;
    }
    catch(SqlException objEx)
    {
      throw new DalException(objEx);
    }

    return blnResult;
  }

  /// <summary>
  /// Adds the relevant parameters to the specified <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see> object
  /// setting the values to ones of the specified <see cref="StudyPlanning.DAL.Courses.EvaluationForm"/> object.
  /// </summary>
  /// <param name="evaluationForm">
  /// The <see cref="StudyPlanning.DAL.Courses.EvaluationForm"/> object containing the values that the parameters have to be
  /// initialized to.
  /// </param>
  /// <param name="objCommand">
  /// The <see cref="System.Data.SqlClient.SqlCommand"/> object to which parameters should be added.
  /// </param>
  /// <param name="isOriginal">
  /// <strong>true</strong> if the parameters to be added represent original
  /// data (for handling concurrent programming issues); otherwise, <strong>false</strong>.</param>
  /// <returns>The <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see>
  /// object with the relevant parameters added.</returns>
  private static void AddParameters(EvaluationForm evaluationForm, SqlCommand objCommand, bool isOriginal)
  {
    string paramName = "";
    SqlParameter objParam;

    //Course_EvaluationForm_ID
    paramName = "Course_EvaluationForm_ID";
    objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);
```

```
      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      objParam.Value = evaluationForm.Course_EvaluationForm_ID.Value;

      //CourseVersion_ID
      paramName = "CourseVersion_ID";
      objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      objParam.Value = evaluationForm.CourseVersion_ID.Value;

      //EvaluationForm_ID
      paramName = "EvaluationForm_ID";
      objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.Int, 4);

      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      objParam.Value = evaluationForm.EvaluationForm_ID.Value;

      //Log
      objCommand = evaluationForm.Log.AddParameters(objCommand, isOriginal);
    }

    /// <summary>
    /// Validates the current <see cref="StudyPlanning.DAL.Courses.EvaluationForm"/> object. If the
    /// value of either of the private properties is null a
    /// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
    /// </summary>
    private void Validate()
    {
      try
      {
        _Course_EvaluationForm_ID.Validate("Course_EvaluationForm_ID");
        _CourseVersion_ID.Validate("CourseVersion_ID");
        _EvaluationForm_ID.Validate("EvaluationForm_ID");
        _Log.Validate();
      }
      catch (DalException excp)
      {
        throw excp;
      }
    }

    #endregion //Methods
  }
}
```

1.6.5 Keyword

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.Courses
{
  /// <summary>
  /// Represents a course evaluation form.
  /// </summary>
  public class Keyword : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalGuid _Course_Keyword_ID = new DalGuid(false);
    private DalGuid _CourseVersion_ID = new DalGuid(false);
    private DalGuid _Keyword_ID = new DalGuid(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Initializes a new instance of the <see cref="StudyPlanning.DAL.Courses.Keyword"/>
    /// class.
    /// </summary>
    public Keyword()
    {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the ID of the course keyword.
    /// </summary>
    public DalGuid Course_Keyword_ID
    {
      get { return _Course_Keyword_ID; }
```

```
}

/// <summary>
/// Gets the ID of the course version.
/// </summary>
public DalGuid CourseVersion_ID
{
  get { return _CourseVersion_ID; }
}

/// <summary>
/// Gets the ID of the keyword.
/// </summary>
public DalGuid Keyword_ID
{
  get { return _Keyword_ID; }
}

/// <summary>
/// Gets the data log of the course.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Methods

/// <summary>
/// Creates a new course keyword in the database using the values of the properties
/// of the current instance.
/// </summary>
public void Create()
{
  try
  {
    this.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }

  SqlCommand objCommand = new SqlCommand("Course_Keyword_Insert", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, false);

  try
  {
    objConnection.Open();
    objCommand.ExecuteNonQuery();
    objConnection.Close();
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
}

#region Retrieve Methods

/// <summary>
/// Retrieves the course keyword from the database using the value of the
/// Course_Keyword_ID property of the current instance.
/// </summary>
public void Retrieve()
{
  try
  {
    this._Course_Keyword_ID.Validate("Course_Keyword_ID");
  }
  catch (DalException excp)
  {
    throw excp;
  }

  try
  {
      RetrieveExecute(_Course_Keyword_ID.Value, this, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
}

/// <summary>
/// Retrieves the course keyword with the specified identifier from the database.
/// </summary>
/// <param name="course_Keyword_ID">The globally unique identifier of the course keyword.</param>
public static Keyword Retrieve(Guid course_Keyword_ID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  Keyword objEf = new Keyword();
```

```
  try
  {
    RetrieveExecute(course_Keyword_ID, objEf, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }

  return objEf;
}

private static void RetrieveExecute(
  Guid course_Keyword_ID,
  Keyword objKeyword,
  SqlConnection sqlConnection)
{
  SqlCommand objCommand = new SqlCommand("Course_Keyword_Select", sqlConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  SqlParameter objParam;

  objParam = objCommand.Parameters.Add("@Course_Keyword_ID", SqlDbType.UniqueIdenti↓er, 16);
  objParam.Value = course_Keyword_ID;

  DataSet objDataSet = new DataSet("Course_Keyword");
  SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

  try
  {
    sqlConnection.Open();
    objAdap.Fill(objDataSet, "Course_Keyword");
    sqlConnection.Close();
    DataRow objDataRow = objDataSet.Tables["Course_Keyword"].Rows[0];

    objKeyword.Course_Keyword_ID.Value = (Guid)objDataRow["Course_Keyword_ID"];
    objKeyword.CourseVersion_ID.Value = (Guid)objDataRow["CourseVersion_ID"];
    objKeyword.Keyword_ID.Value = (Guid)objDataRow["Keyword_ID"];

    objKeyword.Log.SetValues(objDataRow);
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
}

#endregion //Retrive Methods

/// <summary>
/// Updates the current course keyword in the database using the original
/// course evaluation form to resolve possible concurrency issues.
/// </summary>
/// <param name="originalKeyword">The original <see cref="StudyPlanning.DAL.Courses.Keyword"/> object.</param>
/// <returns><strong>true</strong> if the update was succesfully executed; otherwise, <strong>false</strong>.</returns>
public bool Update(Keyword originalKeyword)
{
  bool blnResult = false;

  SqlCommand objCommand = new SqlCommand("Course_Keyword_Update", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, false);
  AddParameters(originalKeyword, objCommand, true);

  int rowsA↑ected;

  try
  {
    objConnection.Open();
    rowsA↑ected = objCommand.ExecuteNonQuery();
    objConnection.Close();

    if (rowsA↑ected > 0)
      blnResult = true;
  }
  catch(SqlException excp)
  {
    throw new DalException(excp);
  }

  return blnResult;
}

/// <summary>
/// Deletes the current course keyword form from the database.
/// </summary>
/// <returns><strong>true</strong> if the deletion is succesfully executed; otherwise, <strong>false</strong>.</returns>
public bool Delete()
{
  bool blnResult = false;

  try
  {
    this.Validate();
  }
  catch (DalException excp)
  {
```

```
      throw excp;
    }

    SqlCommand objCommand = new SqlCommand("Course_Keyword_Delete", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, true);

    try
    {
      objConnection.Open();
      int rowsA↑ected = objCommand.ExecuteNonQuery();
      objConnection.Close();

      if (rowsA↑ected > 0)
        blnResult = true;
    }
    catch(SqlException objEx)
    {
      throw new DalException(objEx);
    }

    return blnResult;
  }

  /// <summary>
  /// Adds the relevant parameters to the specified <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see> object
  /// setting the values to ones of the specified <see cref="StudyPlanning.DAL.Courses.Keyword"/> object.
  /// </summary>
  /// <param name="keyword">
  /// The <see cref="StudyPlanning.DAL.Courses.Keyword"/> object containing the values that the parameters have to be
  /// initialized to.
  /// </param>
  /// <param name="objCommand">
  /// The <see cref="System.Data.SqlClient.SqlCommand"/> object to which parameters should be added.
  /// </param>
  /// <param name="isOriginal">
  /// <strong>true</strong> if the parameters to be added represent original
  /// data (for handling concurrent programming issues); otherwise, <strong>false</strong>.</param>
  /// <returns>The <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see>
  /// object with the relevant parameters added.</returns>
  private static void AddParameters(Keyword keyword, SqlCommand objCommand, bool isOriginal)
  {
    string paramName = "";
    SqlParameter objParam;

    //Course_Keyword_ID
    paramName = "Course_Keyword_ID";
    objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

    if (isOriginal)
      objParam.ParameterName = "@Original_" + paramName;

    objParam.Value = keyword.Course_Keyword_ID.Value;

    //CourseVersion_ID
    paramName = "CourseVersion_ID";
    objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

    if (isOriginal)
      objParam.ParameterName = "@Original_" + paramName;

    objParam.Value = keyword.CourseVersion_ID.Value;

    //Keyword_ID
    paramName = "Keyword_ID";
    objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

    if (isOriginal)
      objParam.ParameterName = "@Original_" + paramName;

    objParam.Value = keyword.Keyword_ID.Value;

    //Log
    objCommand = keyword.Log.AddParameters(objCommand, isOriginal);
  }

  /// <summary>
  /// Validates the current <see cref="StudyPlanning.DAL.Courses.Keyword"/> object. If the
  /// value of either of the private properties is null a
  /// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
  /// </summary>
  private void Validate()
  {
    try
    {
      _Course_Keyword_ID.Validate("Course_Keyword_ID");
      _CourseVersion_ID.Validate("CourseVersion_ID");
      _Keyword_ID.Validate("Keyword_ID");
      _Log.Validate();
    }
    catch (DalException excp)
    {
      throw excp;
    }
  }

  /// <summary>
  /// Retrieves a list of courses associated to the specified keyword.
```

```
    /// </summary>
    /// <param name="keyword_ID">The ID of the keyword for which to search for courses.</param>
    /// <returns>An array containing IDs of the course verions associated to the specified keyword.</returns>
    public static Guid[] GetCoursesFromKeyword(Guid keyword_ID)
    {
      Guid[] courses;

      string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
      SqlConnection objConnection = new SqlConnection(connString);

      SqlCommand objCommand = new SqlCommand("Course_Keyword_GetCoursesFromKeyword", objConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      SqlParameter objParam;

      objParam = objCommand.Parameters.Add("@Keyword_ID", SqlDbType.UniqueIdenti↓er, 16);
      objParam.Value = keyword_ID;

      DataSet objDataSet = new DataSet("Courses");
      SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

      try
      {
        objConnection.Open();
        objAdap.Fill(objDataSet, "Courses");
        objConnection.Close();

        int numRows = objDataSet.Tables["Courses"].Rows.Count;
        courses = new Guid[numRows];

        for(int i=0; i<numRows; i++)
        {
          courses[i] = (Guid)objDataSet.Tables["Courses"].Rows[i]["CourseVersion_ID"];
        }
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }

      return courses;
    }

    #endregion //Methods
  }
}
```

1.6.6 Lecturer

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.Courses
{
  /// <summary>
  /// Represents a lecturer of a course.
  /// </summary>
  public class Lecturer : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalGuid _Course_Lecturer_ID = new DalGuid(false);
    private DalGuid _CourseVersion_ID = new DalGuid(false);
    private DalGuid _Lecturer_ID = new DalGuid(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Initializes a new instance of the <see cref="StudyPlanning.DAL.Courses.Lecturer"/>
    /// class.
    /// </summary>
    public Lecturer()
    {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the ID of the course lecturer.
    /// </summary>
    public DalGuid Course_Lecturer_ID
    {
      get { return _Course_Lecturer_ID; }
    }

    /// <summary>
    /// Gets the ID of the course version.
    /// </summary>
```

```
public DalGuid CourseVersion_ID
{
  get { return _CourseVersion_ID; }
}

/// <summary>
/// Gets the ID of the lecturer.
/// </summary>
public DalGuid Lecturer_ID
{
  get { return _Lecturer_ID; }
}

/// <summary>
/// Gets the data log of the course.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Methods

/// <summary>
/// Creates a new course lecturer in the database using the values of the properties
/// of the current instance.
/// </summary>
public void Create()
{
  try
  {
    this.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }

  SqlCommand objCommand = new SqlCommand("Course_Lecturer_Insert", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, false);

  try
  {
    objConnection.Open();
    objCommand.ExecuteNonQuery();
    objConnection.Close();
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
}

#region Retrieve Methods

/// <summary>
/// Retrieves the course lecturer from the database using the value of the
/// Course_Lecturer_ID property of the current instance.
/// </summary>
public void Retrieve()
{
  try
  {
    this._Course_Lecturer_ID.Validate("Course_Lecturer_ID");
  }
  catch (DalException excp)
  {
    throw excp;
  }

  try
  {
      RetrieveExecute(_Course_Lecturer_ID.Value, this, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
}

/// <summary>
/// Retrieves the course lecturer with the specified identifier from the database.
/// </summary>
/// <param name="course_Lecturer_ID">The globally unique identifier of the course lecturer.</param>
public static Lecturer Retrieve(Guid course_Lecturer_ID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  Lecturer objLec = new Lecturer();

  try
  {
    RetrieveExecute(course_Lecturer_ID, objLec, objConnection);
  }
```

```
    catch (DalException objExc)
    {
      throw objExc;
    }

    return objLec;
  }

  private static void RetrieveExecute(
    Guid course_Lecturer_ID,
    Lecturer objLecturer,
    SqlConnection sqlConnection)
  {
    SqlCommand objCommand = new SqlCommand("Course_Lecturer_Select", sqlConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam;

    objParam = objCommand.Parameters.Add("@Course_Lecturer_ID", SqlDbType.UniqueIdenti↓er, 16);
    objParam.Value = course_Lecturer_ID;

    DataSet objDataSet = new DataSet("Course_Lecturer");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
      sqlConnection.Open();
      objAdap.Fill(objDataSet, "Course_Lecturer");
      sqlConnection.Close();
      DataRow objDataRow = objDataSet.Tables["Course_Lecturer"].Rows[0];

      objLecturer.Course_Lecturer_ID.Value = (Guid)objDataRow["Course_Lecturer_ID"];
      objLecturer.CourseVersion_ID.Value = (Guid)objDataRow["CourseVersion_ID"];
      objLecturer.Lecturer_ID.Value = (Guid)objDataRow["Lecturer_ID"];

      objLecturer.Log.SetValues(objDataRow);
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
  }

  #endregion //Retrive Methods

  /// <summary>
  /// Updates the current course lecturer in the database using the original
  /// course evaluation form to resolve possible concurrency issues.
  /// </summary>
  /// <param name="originalLecturer">The original <see cref="StudyPlanning.DAL.Courses.Lecturer"/> object.</param>
  /// <returns><strong>true</strong> if the update was succesfully executed; otherwise, <strong>false</strong>.</returns>
  public bool Update(Lecturer originalLecturer)
  {
    try
    {
      this.Validate();
    }
    catch (DalException excp)
    {
      throw excp;
    }

    bool blnResult = false;

    SqlCommand objCommand = new SqlCommand("Course_Lecturer_Update", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, false);
    AddParameters(originalLecturer, objCommand, true);

    int rowsA↑ected;

    try
    {
      objConnection.Open();
      rowsA↑ected = objCommand.ExecuteNonQuery();
      objConnection.Close();

      if (rowsA↑ected > 0)
        blnResult = true;
    }
    catch(SqlException excp)
    {
      throw new DalException(excp);
    }

    return blnResult;
  }

  /// <summary>
  /// Deletes the current course lecturer form from the database.
  /// </summary>
  /// <returns><strong>true</strong> if the deletion is succesfully executed; otherwise, <strong>false</strong>.</returns>
  public bool Delete()
  {
    bool blnResult = false;

    try
    {
```

```
      this.Validate();
    }
    catch (DalException excp)
    {
      throw excp;
    }

    SqlCommand objCommand = new SqlCommand("Course_Lecturer_Delete", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, true);

    try
    {
      objConnection.Open();
      int rowsA↑ected = objCommand.ExecuteNonQuery();
      objConnection.Close();

      if (rowsA↑ected > 0)
        blnResult = true;
    }
    catch(SqlException objEx)
    {
      throw new DalException(objEx);
    }

    return blnResult;
  }

  /// <summary>
  /// Retrieves the list of lecturers which are contact persons of the specified course version.
  /// </summary>
  /// <param name="courseVersion_ID">The ID of the course version for which to retrieve the
  /// list of modules.</param>
  /// <returns>An array containing IDs of lecturers.</returns>
  public static Guid[] GetLecturers(Guid courseVersion_ID)
  {
    Guid[] lecturers;

    string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
    SqlConnection objConnection = new SqlConnection(connString);

    SqlCommand objCommand = new SqlCommand("Course_Lecturer_GetLecturers", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam;

    objParam = objCommand.Parameters.Add("@CourseVersion_ID", SqlDbType.UniqueIdenti↓er, 16);
    objParam.Value = courseVersion_ID;

    DataSet objDataSet = new DataSet("Lecturers");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
      objConnection.Open();
      objAdap.Fill(objDataSet, "Lecturers");
      objConnection.Close();

      int numRows = objDataSet.Tables["Lecturers"].Rows.Count;
      lecturers = new Guid[numRows];

      for(int i=0; i<numRows; i++)
      {
        lecturers[i] = (Guid)objDataSet.Tables["Lecturers"].Rows[i]["Lecturer_ID"];
      }
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }

    return lecturers;
  }

  /// <summary>
  /// Adds the relevant parameters to the specified <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see> object
  /// setting the values to ones of the specified <see cref="StudyPlanning.DAL.Courses.Lecturer"/> object.
  /// </summary>
  /// <param name="lecturer">
  /// The <see cref="StudyPlanning.DAL.Courses.Lecturer"/> object containing the values that the parameters have to be
  /// initialized to.
  /// </param>
  /// <param name="objCommand">
  /// The <see cref="System.Data.SqlClient.SqlCommand"/> object to which parameters should be added.
  /// </param>
  /// <param name="isOriginal">
  /// <strong>true</strong> if the parameters to be added represent original
  /// data (for handling concurrent programming issues); otherwise, <strong>false</strong>.</param>
  /// <returns>The <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see>
  /// object with the relevant parameters added.</returns>
  private static void AddParameters(Lecturer lecturer, SqlCommand objCommand, bool isOriginal)
  {
    string paramName = "";
    SqlParameter objParam;

    //Course_Lecturer_ID
    paramName = "Course_Lecturer_ID";
    objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);
```

```
      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      objParam.Value = lecturer.Course_Lecturer_ID.Value;

      //CourseVersion_ID
      paramName = "CourseVersion_ID";
      objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      objParam.Value = lecturer.CourseVersion_ID.Value;

      //Lecturer_ID
      paramName = "Lecturer_ID";
      objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      objParam.Value = lecturer.Lecturer_ID.Value;

      //Log
      objCommand = lecturer.Log.AddParameters(objCommand, isOriginal);
    }

    /// <summary>
    /// Validates the current <see cref="StudyPlanning.DAL.Courses.Lecturer"/> object. If the
    /// value of either of the private properties is null a
    /// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
    /// </summary>
    private void Validate()
    {
      try
      {
        _Course_Lecturer_ID.Validate("Course_Lecturer_ID");
        _CourseVersion_ID.Validate("CourseVersion_ID");
        _Lecturer_ID.Validate("Lecturer_ID");
        _Log.Validate();
      }
      catch (DalException excp)
      {
        throw excp;
      }
    }

    #endregion //Methods
  }
}
```

1.6.7 Period

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.Courses
{
  /// <summary>
  /// Represents a course period.
  /// </summary>
  public class Period : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalGuid _Course_Period_ID = new DalGuid(false);
    private DalGuid _CourseVersion_ID = new DalGuid(false);
    private DalGuid _Period_ID = new DalGuid(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Initializes a new instance of the <see cref="StudyPlanning.DAL.Courses.Period"/>
    /// class.
    /// </summary>
    public Period()
    {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the ID of the course period.
    /// </summary>
    public DalGuid Course_Period_ID
    {
      get { return _Course_Period_ID; }
```

```
}

/// <summary>
/// Gets the ID of the course version.
/// </summary>
public DalGuid CourseVersion_ID
{
  get { return _CourseVersion_ID; }
}

/// <summary>
/// Gets the ID of the period.
/// </summary>
public DalGuid Period_ID
{
  get { return _Period_ID; }
}

/// <summary>
/// Gets the data log of the course.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Methods

/// <summary>
/// Creates a new course period in the database using the values of the properties
/// of the current instance.
/// </summary>
public void Create()
{
  try
  {
    this.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }

  SqlCommand objCommand = new SqlCommand("Course_Period_Insert", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, false);

  try
  {
    objConnection.Open();
    objCommand.ExecuteNonQuery();
    objConnection.Close();
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
}

#region Retrieve Methods

/// <summary>
/// Retrieves the course period from the database using the value of the
/// Course_Period_ID property of the current instance.
/// </summary>
public void Retrieve()
{
  try
  {
    this._Course_Period_ID.Validate("Course_Period_ID");
  }
  catch (DalException excp)
  {
    throw excp;
  }

  try
  {
      RetrieveExecute(_Course_Period_ID.Value, this, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
}

/// <summary>
/// Retrieves the course period with the specified identifier from the database.
/// </summary>
/// <param name="course_Period_ID">The globally unique identifier of the course period.</param>
public static Period Retrieve(Guid course_Period_ID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  Period objPeriod = new Period();
```

```
    try
    {
      RetrieveExecute(course_Period_ID, objPeriod, objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }

    return objPeriod;
  }

  private static void RetrieveExecute(
    Guid course_Period_ID,
    Period objPeriod,
    SqlConnection sqlConnection)
  {
    SqlCommand objCommand = new SqlCommand("Course_Period_Select", sqlConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam;

    objParam = objCommand.Parameters.Add("@Course_Period_ID", SqlDbType.UniqueIdenti↓er, 16);
    objParam.Value = course_Period_ID;

    DataSet objDataSet = new DataSet("Course_Period");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
      sqlConnection.Open();
      objAdap.Fill(objDataSet, "Course_Period");
      sqlConnection.Close();
      DataRow objDataRow = objDataSet.Tables["Course_Period"].Rows[0];

      objPeriod.Course_Period_ID.Value = (Guid)objDataRow["Course_Period_ID"];
      objPeriod.CourseVersion_ID.Value = (Guid)objDataRow["CourseVersion_ID"];
      objPeriod.Period_ID.Value = (Guid)objDataRow["Period_ID"];

      objPeriod.Log.SetValues(objDataRow);
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
  }

  #endregion //Retrive Methods

  /// <summary>
  /// Updates the current course period in the database using the original
  /// course module to resolve possible concurrency issues.
  /// </summary>
  /// <param name="originalPeriod">The original <see cref="StudyPlanning.DAL.Courses.Period"/> object.</param>
  /// <returns><strong>true</strong> if the update was succesfully executed; otherwise, <strong>false</strong>.</returns>
  public bool Update(Period originalPeriod)
  {
    try
    {
      this.Validate();
    }
    catch (DalException excp)
    {
      throw excp;
    }

    bool blnResult = false;

    SqlCommand objCommand = new SqlCommand("Course_Period_Update", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, false);
    AddParameters(originalPeriod, objCommand, true);

    int rowsA↑ected;

    try
    {
      objConnection.Open();
      rowsA↑ected = objCommand.ExecuteNonQuery();
      objConnection.Close();

      if (rowsA↑ected > 0)
        blnResult = true;
    }
    catch(SqlException excp)
    {
      throw new DalException(excp);
    }

    return blnResult;
  }

  /// <summary>
  /// Deletes the current course period from the database.
  /// </summary>
  /// <returns><strong>true</strong> if the deletion is succesfully executed; otherwise, <strong>false</strong>.</returns>
  public bool Delete()
```

```
{
  bool blnResult = false;

  try
  {
    this.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }

  SqlCommand objCommand = new SqlCommand("Course_Period_Delete", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, true);

  try
  {
    objConnection.Open();
    int rowsA↑ected = objCommand.ExecuteNonQuery();
    objConnection.Close();

    if (rowsA↑ected > 0)
      blnResult = true;
  }
  catch(SqlException objEx)
  {
    throw new DalException(objEx);
  }

  return blnResult;
}

/// <summary>
/// Adds the relevant parameters to the specified <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see> object
/// setting the values to ones of the specified <see cref="StudyPlanning.DAL.Courses.Period"/> object.
/// </summary>
/// <param name="period">
/// The <see cref="StudyPlanning.DAL.Courses.Period"/> object containing the values that the parameters have to be
/// initialized to.
/// </param>
/// <param name="objCommand">
/// The <see cref="System.Data.SqlClient.SqlCommand"/> object to which parameters should be added.
/// </param>
/// <param name="isOriginal">
/// <strong>true</strong> if the parameters to be added represent original
/// data (for handling concurrent programming issues); otherwise, <strong>false</strong>.</param>
/// <returns>The <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see>
/// object with the relevant parameters added.</returns>
private static void AddParameters(Period period, SqlCommand objCommand, bool isOriginal)
{
  string paramName = "";
  SqlParameter objParam;

  //Course_Period_ID
  paramName = "Course_Period_ID";
  objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

  if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

  objParam.Value = period.Course_Period_ID.Value;

  //CourseVersion_ID
  paramName = "CourseVersion_ID";
  objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

  if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

  objParam.Value = period.CourseVersion_ID.Value;

  //Period_ID
  paramName = "Period_ID";
  objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

  if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

  objParam.Value = period.Period_ID.Value;

  //Log
  objCommand = period.Log.AddParameters(objCommand, isOriginal);
}

/// <summary>
/// Validates the current <see cref="StudyPlanning.DAL.Courses.Period"/> object. If the
/// value of either of the private properties is null a
/// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
/// </summary>
private void Validate()
{
  try
  {
    _Course_Period_ID.Validate("Course_Period_ID");
    _CourseVersion_ID.Validate("CourseVersion_ID");
    _Period_ID.Validate("Period_ID");
    _Log.Validate();
```

```
      }
      catch (DalException excp)
      {
        throw excp;
      }
    }

    /// <summary>
    /// Retrieves the periods in which a given course version is taught.
    /// </summary>
    /// <param name="courseVersion_ID">Identification of the <see cref="StudyPlanning.DAL.Courses.CourseVersion"/>
    /// for which to retrieve the periods.</param>
    /// <param name="part">The course version part.</param>
    /// <returns>An array containing the <see cref="StudyPlanning.DAL.Period"/> identifications.</returns>
    public static Guid[] GetPeriods(Guid courseVersion_ID, int part)
    {
      string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
      SqlConnection objConnection = new SqlConnection(connString);

      SqlCommand objCommand = new SqlCommand("Course_Period_GetPeriods", objConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      SqlParameter objParam;

      objParam = objCommand.Parameters.Add("@CourseVersion_ID", SqlDbType.UniqueIdenti↓er, 16);
      objParam.Value = courseVersion_ID;

      objParam = objCommand.Parameters.Add("@Part", SqlDbType.Int);
      objParam.Value = part;

      DataSet objDataSet = new DataSet("Result");
      SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

      try
      {
        objConnection.Open();
        objAdap.Fill(objDataSet, "Result");
        objConnection.Close();
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }
      int intRows = objDataSet.Tables["Result"].Rows.Count;
      Guid[] periods = new Guid[intRows];

      for(int i=0; i < intRows; i++)
      {
        periods[i] = (Guid)objDataSet.Tables["Result"].Rows[i]["Period_ID"];
      }
      return periods;
    }

    /// <summary>
    /// Retrieves the Course_Period_ID for a given courseversion, part and period.
    /// </summary>
    /// <param name="courseVersion_ID">Identification of the <see cref="StudyPlanning.DAL.Courses.CourseVersion"/>.</param>
    /// <param name="part">The course version part.</param>
    /// <param name="period">Identification of the <see cref="StudyPlanning.DAL.Period"/>.</param>
    /// <returns>A Course_Period_ID.</returns>
    public static Guid GetID(Guid courseVersion_ID, int part, Guid period)
    {
      string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
      SqlConnection objConnection = new SqlConnection(connString);

      SqlCommand objCommand = new SqlCommand("Course_Period_GetID", objConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      SqlParameter objParam;

      objParam = objCommand.Parameters.Add("@CourseVersion_ID", SqlDbType.UniqueIdenti↓er, 16);
      objParam.Value = courseVersion_ID;

      objParam = objCommand.Parameters.Add("@Part", SqlDbType.Int);
      objParam.Value = part;

      objParam = objCommand.Parameters.Add("@Period_ID", SqlDbType.UniqueIdenti↓er);
      objParam.Value = period;

      DataSet objDataSet = new DataSet("Result");
      SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

      try
      {
        objConnection.Open();
        objAdap.Fill(objDataSet, "Result");
        objConnection.Close();
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }
      return (Guid)objDataSet.Tables["Result"].Rows[0]["Course_Period_ID"];
    }

    #endregion //Methods
  }
}
```

1.6.8 PeriodModule

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.Courses
{
  /// <summary>
  /// Represents a course module.
  /// </summary>
  public class PeriodModule : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalGuid _Course_Period_Module_ID = new DalGuid(false);
    private DalGuid _Course_Period_ID = new DalGuid(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Initializes a new instance of the <see cref="StudyPlanning.DAL.Courses.PeriodModule"/>
    /// class.
    /// </summary>
    public PeriodModule()
    {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the ID of the course module.
    /// </summary>
    public DalGuid Course_Period_Module_ID
    {
      get { return _Course_Period_Module_ID; }
    }

    /// <summary>
    /// Gets the ID the course period.
    /// </summary>
    public DalGuid Course_Period_ID
    {
      get { return _Course_Period_ID; }
    }

    /// <summary>
    /// Gets the data log of the course.
    /// </summary>
    public DataLog Log
    {
      get { return _Log; }
    }

    #endregion //Public Properties

    #region Methods

    /// <summary>
    /// Creates a new course module in the database using the values of the properties
    /// of the current instance.
    /// </summary>
    public void Create()
    {
      try
      {
        this.Validate();
      }
      catch (DalException excp)
      {
        throw excp;
      }

      SqlCommand objCommand = new SqlCommand("Course_Period_Module_Insert", objConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      AddParameters(this, objCommand, false);

      try
      {
        objConnection.Open();
        objCommand.ExecuteNonQuery();
        objConnection.Close();
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }
    }

    #region Retrieve Methods
```

```
/// <summary>
/// Retrieves the course version from the database using the value of the
/// Course_StudyType_ID property of the current instance.
/// </summary>
public void Retrieve()
{
  try
  {
    this._Course_Period_Module_ID.Validate("Course_Period_Module_ID");
  }
  catch (DalException excp)
  {
    throw excp;
  }

  try
  {
      RetrieveExecute(_Course_Period_Module_ID.Value, this, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
}

/// <summary>
/// Retrieves the course period module with the specified identifier from the database.
/// </summary>
/// <param name="course_Period_Module_ID">The globally unique identifier of the course period module.</param>
public static PeriodModule Retrieve(Guid course_Period_Module_ID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  PeriodModule objModule = new PeriodModule();

  try
  {
    RetrieveExecute(course_Period_Module_ID, objModule, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }

  return objModule;
}

private static void RetrieveExecute(
  Guid course_Period_Module_ID,
  PeriodModule objModule,
  SqlConnection sqlConnection)
{
  SqlCommand objCommand = new SqlCommand("Course_Period_Module_Select", sqlConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  SqlParameter objParam;

  objParam = objCommand.Parameters.Add("@Course_Period_Module_ID", SqlDbType.UniqueIdenti↓er, 16);
  objParam.Value = course_Period_Module_ID;

  DataSet objDataSet = new DataSet("Result");
  SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

  try
  {
    sqlConnection.Open();
    objAdap.Fill(objDataSet, "Result");
    sqlConnection.Close();
    DataRow objDataRow = objDataSet.Tables["Result"].Rows[0];

    objModule.Course_Period_Module_ID.Value = (Guid)objDataRow["Course_Period_Module_ID"];
    objModule.Course_Period_ID.Value = (Guid)objDataRow["Course_Period_ID"];

    objModule.Log.SetValues(objDataRow);
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
}

#endregion //Retrive Methods

/// <summary>
/// Updates the current course period module in the database using the original
/// course module to resolve possible concurrency issues.
/// </summary>
/// <param name="originalModule">The original <see cref="StudyPlanning.DAL.Courses.PeriodModule"/> object.</param>
/// <returns><strong>true</strong> if the update was succesfully executed; otherwise, <strong>false</strong>.</returns>
public bool Update(PeriodModule originalModule)
{
  bool blnResult = false;

  try
  {
    this.Validate();
  }
```

```
    catch (DalException excp)
    {
      throw excp;
    }

    SqlCommand objCommand = new SqlCommand("Course_Period_Module_Update", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, false);
    AddParameters(originalModule, objCommand, true);

    int rowsA↑ected;

    try
    {
      objConnection.Open();
      rowsA↑ected = objCommand.ExecuteNonQuery();
      objConnection.Close();

      if (rowsA↑ected > 0)
        blnResult = true;
    }
    catch(SqlException excp)
    {
      throw new DalException(excp);
    }

    return blnResult;
  }

  /// <summary>
  /// Deletes the current course module from the database.
  /// </summary>
  /// <returns><strong>true</strong> if the deletion is succesfully executed; otherwise, <strong>false</strong>.</returns>
  public bool Delete()
  {
    bool blnResult = false;

    try
    {
      this.Validate();
    }
    catch (DalException excp)
    {
      throw excp;
    }

    SqlCommand objCommand = new SqlCommand("Course_Period_Module_Delete", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, true);

    try
    {
      objConnection.Open();
      int rowsA↑ected = objCommand.ExecuteNonQuery();
      objConnection.Close();

      if (rowsA↑ected > 0)
        blnResult = true;
    }
    catch(SqlException objEx)
    {
      throw new DalException(objEx);
    }

    return blnResult;
  }

  /// <summary>
  /// Adds the relevant parameters to the specified <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see> object
  /// setting the values to ones of the specified <see cref="StudyPlanning.DAL.Courses.PeriodModule"/> object.
  /// </summary>
  /// <param name="module">
  /// The <see cref="StudyPlanning.DAL.Courses.PeriodModule"/> object containing the values that the parameters have to be
  /// initialized to.
  /// </param>
  /// <param name="objCommand">
  /// The <see cref="System.Data.SqlClient.SqlCommand"/> object to which parameters should be added.
  /// </param>
  /// <param name="isOriginal">
  /// <strong>true</strong> if the parameters to be added represent original
  /// data (for handling concurrent programming issues); otherwise, <strong>false</strong>.</param>
  /// <returns>The <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see>
  /// object with the relevant parameters added.</returns>
  private static void AddParameters(PeriodModule module, SqlCommand objCommand, bool isOriginal)
  {
    string paramName = "";
    SqlParameter objParam;

    //Course_Module_ID
    paramName = "Course_Period_Module_ID";
    objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

    if (isOriginal)
      objParam.ParameterName = "@Original_" + paramName;

    objParam.Value = module.Course_Period_Module_ID.Value;
```

```
      //Course_Period_ID
      paramName = "Course_Period_ID";
      objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      objParam.Value = module.Course_Period_ID.Value;

      //Log
      objCommand = module.Log.AddParameters(objCommand, isOriginal);
    }

    /// <summary>
    /// Validates the current <see cref="StudyPlanning.DAL.Courses.PeriodModule"/> object. If the
    /// value of either of the private properties is null a
    /// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
    /// </summary>
    private void Validate()
    {
      try
      {
        _Course_Period_Module_ID.Validate("Course_Period_Module_ID");
        _Course_Period_ID.Validate("Course_Period_ID");
        _Log.Validate();
      }
      catch (DalException excp)
      {
        throw excp;
      }
    }

    /// <summary>
    /// Retrieves the Course_Period_Module_ID for a given Course_Period_ID.
    /// </summary>
    /// <param name="course_Period_ID">Identification of a <see cref="StudyPlanning.DAL.Courses.Period"/>.</param>
    /// <returns>An array of Course_Period_Module_IDs.</returns>
    public static Guid[] GetID(Guid course_Period_ID)
    {
      string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
      SqlConnection objConnection = new SqlConnection(connString);

      SqlCommand objCommand = new SqlCommand("Course_Period_Module_GetID", objConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      SqlParameter objParam;

      objParam = objCommand.Parameters.Add("@Course_Period_ID", SqlDbType.UniqueIdenti↓er, 16);
      objParam.Value = course_Period_ID;

      DataSet objDataSet = new DataSet("Result");
      SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

      try
      {
        objConnection.Open();
        objAdap.Fill(objDataSet, "Result");
        objConnection.Close();
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }
      int numRows = objDataSet.Tables["Result"].Rows.Count;
      Guid[] ids = new Guid[numRows];
      for (int i=0; i < numRows; i++)
      {
        ids[i] = (Guid)objDataSet.Tables["Result"].Rows[i]["Course_Period_Module_ID"];
      }
      return ids;
    }

    #endregion //Methods
  }
}
```

1.6.9 PeriodModuleItem

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.Courses
{
  /// <summary>
  /// Represents a course module.
  /// </summary>
  public class PeriodModuleItem : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalGuid _Course_Period_ModuleItem_ID = new DalGuid(false);
    private DalGuid _Course_Period_Module_ID = new DalGuid(false);
    private DalInt _Module_ID = new DalInt(false);
    private DataLog _Log;
```

```
#endregion //Private Properties

#region Constructors

/// <summary>
/// Initializes a new instance of the <see cref="StudyPlanning.DAL.Courses.PeriodModuleItem"/>
/// class.
/// </summary>
public PeriodModuleItem()
{
  _Log = new DataLog();
}

#endregion

#region Public Properties

/// <summary>
/// Gets the ID of the course period module item.
/// </summary>
public DalGuid Course_Period_ModuleItem_ID
{
  get { return _Course_Period_ModuleItem_ID; }
}

/// <summary>
/// Gets the ID of the course period module.
/// </summary>
public DalGuid Course_Period_Module_ID
{
  get { return _Course_Period_Module_ID; }
}

/// <summary>
/// Gets the ID of the module
/// </summary>
public DalInt Module_ID
{
  get { return _Module_ID; }
}

/// <summary>
/// Gets the data log of the course.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Methods

/// <summary>
/// Creates a new course period module item in the database using the values of the properties
/// of the current instance.
/// </summary>
public void Create()
{
  try
  {
    this.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }

  SqlCommand objCommand = new SqlCommand("Course_Period_ModuleItem_Insert", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, false);

  try
  {
    objConnection.Open();
    objCommand.ExecuteNonQuery();
    objConnection.Close();
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
}

#region Retrieve Methods

/// <summary>
/// Retrieves the course period module item from the database using the value of the
/// Course_StudyType_ID property of the current instance.
/// </summary>
public void Retrieve()
{
  try
  {
    this._Course_Period_ModuleItem_ID.Validate("Course_Period_ModuleItem_ID");
  }
  catch (DalException excp)
```

```
    {
      throw excp;
    }

    try
    {
        RetrieveExecute(_Course_Period_ModuleItem_ID.Value, this, objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }
}

/// <summary>
/// Retrieves the course period module with the specified identifier from the database.
/// </summary>
/// <param name="course_Period_ModuleItem_ID">The globally unique identifier of the course period module item.</param>
public static PeriodModuleItem Retrieve(Guid course_Period_ModuleItem_ID)
{
    string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
    SqlConnection objConnection = new SqlConnection(connString);

    PeriodModuleItem objModule = new PeriodModuleItem();

    try
    {
      RetrieveExecute(course_Period_ModuleItem_ID, objModule, objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }

    return objModule;
}

private static void RetrieveExecute(
    Guid course_Period_ModuleItem_ID,
    PeriodModuleItem objModule,
    SqlConnection sqlConnection)
{
    SqlCommand objCommand = new SqlCommand("Course_Period_ModuleItem_Select", sqlConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam;

    objParam = objCommand.Parameters.Add("@Course_Period_ModuleItem_ID", SqlDbType.UniqueIdenti↓er, 16);
    objParam.Value = course_Period_ModuleItem_ID;

    DataSet objDataSet = new DataSet("Result");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
      sqlConnection.Open();
      objAdap.Fill(objDataSet, "Result");
      sqlConnection.Close();
      DataRow objDataRow = objDataSet.Tables["Result"].Rows[0];

      objModule.Course_Period_ModuleItem_ID.Value = (Guid)objDataRow["Course_Period_ModuleItem_ID"];
      objModule.Course_Period_Module_ID.Value = (Guid)objDataRow["Course_Period_Module_ID"];
      objModule.Module_ID.Value = (int)objDataRow["Module_ID"];

      objModule.Log.SetValues(objDataRow);
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
}

#endregion //Retrive Methods

/// <summary>
/// Updates the current course period module item in the database using the original
/// course module to resolve possible concurrency issues.
/// </summary>
/// <param name="originalModule">The original <see cref="StudyPlanning.DAL.Courses.PeriodModuleItem"/> object.</param>
/// <returns><strong>true</strong> if the update was succesfully executed; otherwise, <strong>false</strong>.</returns>
public bool Update(PeriodModuleItem originalModule)
{
    bool blnResult = false;

    try
    {
      this.Validate();
    }
    catch (DalException excp)
    {
      throw excp;
    }

    SqlCommand objCommand = new SqlCommand("Course_Period_ModuleItem_Update", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, false);
    AddParameters(originalModule, objCommand, true);
```

```
  int rowsA↑ected;

  try
  {
    objConnection.Open();
    rowsA↑ected = objCommand.ExecuteNonQuery();
    objConnection.Close();

    if (rowsA↑ected > 0)
      blnResult = true;
  }
  catch(SqlException excp)
  {
    throw new DalException(excp);
  }

  return blnResult;
}

/// <summary>
/// Deletes the current course period module item from the database.
/// </summary>
/// <returns><strong>true</strong> if the deletion is succesfully executed; otherwise, <strong>false</strong>.</returns>
public bool Delete()
{
  bool blnResult = false;

  try
  {
    this.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }

  SqlCommand objCommand = new SqlCommand("Course_Period_ModuleItem_Delete", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, true);

  try
  {
    objConnection.Open();
    int rowsA↑ected = objCommand.ExecuteNonQuery();
    objConnection.Close();

    if (rowsA↑ected > 0)
      blnResult = true;
  }
  catch(SqlException objEx)
  {
    throw new DalException(objEx);
  }

  return blnResult;
}

/// <summary>
/// Adds the relevant parameters to the specified <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see> object
/// setting the values to ones of the specified <see cref="StudyPlanning.DAL.Courses.PeriodModuleItem"/> object.
/// </summary>
/// <param name="module">
/// The <see cref="StudyPlanning.DAL.Courses.PeriodModuleItem"/> object containing the values that the parameters have to be
/// initialized to.
/// </param>
/// <param name="objCommand">
/// The <see cref="System.Data.SqlClient.SqlCommand"/> object to which parameters should be added.
/// </param>
/// <param name="isOriginal">
/// <strong>true</strong> if the parameters to be added represent original
/// data (for handling concurrent programming issues); otherwise, <strong>false</strong>.</param>
/// <returns>The <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see>
/// object with the relevant parameters added.</returns>
private static void AddParameters(PeriodModuleItem module, SqlCommand objCommand, bool isOriginal)
{
  string paramName = "";
  SqlParameter objParam;

  //Course_Period_ModuleItem_ID
  paramName = "Course_Period_ModuleItem_ID";
  objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

  if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

  objParam.Value = module.Course_Period_ModuleItem_ID.Value;

  //Course_Period_Module_ID
  paramName = "Course_Period_Module_ID";
  objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

  if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

  objParam.Value = module.Course_Period_Module_ID.Value;

  //Module_ID
  paramName = "Module_ID";
```

```
      objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.Int);

      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      objParam.Value = module.Module_ID.Value;

      //Log
      objCommand = module.Log.AddParameters(objCommand, isOriginal);
    }

    /// <summary>
    /// Validates the current <see cref="StudyPlanning.DAL.Courses.PeriodModuleItem"/> object. If the
    /// value of either of the private properties is null a
    /// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
    /// </summary>
    private void Validate()
    {
      try
      {
        _Course_Period_ModuleItem_ID.Validate("Course_Period_ModuleItem_ID");
        _Course_Period_Module_ID.Validate("Course_Period_Module_ID");
        _Module_ID.Validate("Module_ID");
        _Log.Validate();
      }
      catch (DalException excp)
      {
        throw excp;
      }
    }

    /// <summary>
    /// Retrieves the Course_Period_ModuleItem_IDs for a given Course_Period_Module_ID.
    /// </summary>
    /// <param name="course_Period_Module_ID">Identification of a <see cref="StudyPlanning.DAL.Courses.PeriodModule"/>.</param>
    /// <returns>An array of Course_Period_Module_IDs.</returns>
    public static Guid[] GetID(Guid course_Period_Module_ID)
    {
      string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
      SqlConnection objConnection = new SqlConnection(connString);

      SqlCommand objCommand = new SqlCommand("Course_Period_ModuleItem_GetID", objConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      SqlParameter objParam;

      objParam = objCommand.Parameters.Add("@Course_Period_Module_ID", SqlDbType.UniqueIdenti↓er, 16);
      objParam.Value = course_Period_Module_ID;

      DataSet objDataSet = new DataSet("Result");
      SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

      try
      {
        objConnection.Open();
        objAdap.Fill(objDataSet, "Result");
        objConnection.Close();
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }
      int numRows = objDataSet.Tables["Result"].Rows.Count;
      Guid[] ids = new Guid[numRows];
      for (int i=0; i < numRows; i++)
      {
        ids[i] = (Guid)objDataSet.Tables["Result"].Rows[i]["Course_Period_ModuleItem_ID"];
      }
      return ids;
    }

    #endregion //Methods
  }
}
```

1.6.10 Point

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.Courses
{
  /// <summary>
  /// Represents a point of a course part.
  /// </summary>
  public class Point : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalGuid _Course_Point_ID = new DalGuid(false);
    private DalGuid _CourseVersion_ID = new DalGuid(false);
    private DalInt _Part = new DalInt(false);
    private DalGuid _Point_ID = new DalGuid(true);
    private DataLog _Log;
```

```
#endregion //Private Properties

#region Constructors

/// <summary>
/// Initializes a new instance of the <see cref="StudyPlanning.DAL.Courses.Point"/>
/// class.
/// </summary>
public Point()
{
  _Log = new DataLog();
}

#endregion

#region Public Properties

/// <summary>
/// Gets the ID of the point of the course part.
/// </summary>
public DalGuid Course_Point_ID
{
  get { return _Course_Point_ID; }
}

/// <summary>
/// Gets the ID of the course version.
/// </summary>
public DalGuid CourseVersion_ID
{
  get { return _CourseVersion_ID; }
}

/// <summary>
/// Gets the number of the course part.
/// </summary>
public DalInt Part
{
  get { return _Part; }
}

/// <summary>
/// Gets the ID of the point.
/// </summary>
public DalGuid Point_ID
{
  get { return _Point_ID; }
}

/// <summary>
/// Gets the data log of the course.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Methods

/// <summary>
/// Creates a new point of a course part in the database using the values of the properties
/// of the current instance.
/// </summary>
public void Create()
{
  try
  {
    this.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }

  SqlCommand objCommand = new SqlCommand("Course_Point_Insert", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, false);

  try
  {
    objConnection.Open();
    objCommand.ExecuteNonQuery();
    objConnection.Close();
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
}

#region Retrieve Methods

/// <summary>
/// Retrieves the point of a course part from the database using the value of the
/// Course_Point_ID property of the current instance.
```

```
/// </summary>
public void Retrieve()
{
  try
  {
    this._Course_Point_ID.Validate("Course_Point_ID");
  }
  catch (DalException excp)
  {
    throw excp;
  }

  try
  {
      RetrieveExecute(_Course_Point_ID.Value, this, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
}

/// <summary>
/// Retrieves the point of a course part with the specified identifier from the database.
/// </summary>
/// <param name="course_Point_ID">The globally unique identifier of the point of a course part.</param>
public static Point Retrieve(Guid course_Point_ID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  Point objRp = new Point();

  try
  {
    RetrieveExecute(course_Point_ID, objRp, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }

  return objRp;
}

private static void RetrieveExecute(
  Guid course_Point_ID,
  Point objRp,
  SqlConnection sqlConnection)
{
  SqlCommand objCommand = new SqlCommand("Course_Point_Select", sqlConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  SqlParameter objParam;

  objParam = objCommand.Parameters.Add("@Course_Point_ID", SqlDbType.UniqueIdenti↓er, 16);
  objParam.Value = course_Point_ID;

  DataSet objDataSet = new DataSet("Course_Point");
  SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

  try
  {
    sqlConnection.Open();
    objAdap.Fill(objDataSet, "Point");
    sqlConnection.Close();
    DataRow objDataRow = objDataSet.Tables["Point"].Rows[0];

    objRp.Course_Point_ID.Value = (Guid)objDataRow["Course_Point_ID"];
    objRp.CourseVersion_ID.Value = (Guid)objDataRow["CourseVersion_ID"];
    objRp.Part.Value = (int)objDataRow["Part"];
    objRp.Point_ID.Value = (Guid)objDataRow["Point_ID"];
    objRp.Log.SetValues(objDataRow);
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
}

#endregion //Retrive Methods

/// <summary>
/// Updates the current point of a course part in the database using the original
/// point of a course part to resolve possible concurrency issues.
/// </summary>
/// <param name="originalPoint">The original <see cref="StudyPlanning.DAL.Courses.Point"/> object.</param>
/// <returns><strong>true</strong> if the update was succesfully executed; otherwise, <strong>false</strong>.</returns>
public bool Update(Point originalPoint)
{
  bool blnResult = false;

  SqlCommand objCommand = new SqlCommand("Course_Point_Update", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, false);
  AddParameters(originalPoint, objCommand, true);

  int rowsA↑ected;
```

```
    try
    {
      objConnection.Open();
      rowsA↑ected = objCommand.ExecuteNonQuery();
      objConnection.Close();

      if (rowsA↑ected > 0)
        blnResult = true;
    }
    catch(SqlException excp)
    {
      throw new DalException(excp);
    }

    return blnResult;
  }

  /// <summary>
  /// Deletes the current point of a course part from the database.
  /// </summary>
  /// <returns><strong>true</strong> if the deletion is succesfully executed; otherwise, <strong>false</strong>.</returns>
  public bool Delete()
  {
    bool blnResult = false;

    try
    {
      this.Validate();
    }
    catch (DalException excp)
    {
      throw excp;
    }

    SqlCommand objCommand = new SqlCommand("Course_Point_Delete", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, true);

    try
    {
      objConnection.Open();
      int rowsA↑ected = objCommand.ExecuteNonQuery();
      objConnection.Close();

      if (rowsA↑ected > 0)
        blnResult = true;
    }
    catch(SqlException objEx)
    {
      throw new DalException(objEx);
    }

    return blnResult;
  }

  /// <summary>
  /// Adds the relevant parameters to the specified <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see> object
  /// setting the values to ones of the specified <see cref="StudyPlanning.DAL.Courses.Point"/> object.
  /// </summary>
  /// <param name="point">
  /// The <see cref="StudyPlanning.DAL.Courses.Point"/> object containing the values that the parameters have to be
  /// initialized to.
  /// </param>
  /// <param name="objCommand">
  /// The <see cref="System.Data.SqlClient.SqlCommand"/> object to which parameters should be added.
  /// </param>
  /// <param name="isOriginal">
  /// <strong>true</strong> if the parameters to be added represent original
  /// data (for handling concurrent programming issues); otherwise, <strong>false</strong>.</param>
  /// <returns>The <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see>
  /// object with the relevant parameters added.</returns>
  private static void AddParameters(Point point, SqlCommand objCommand, bool isOriginal)
  {
    string paramName = "";
    SqlParameter objParam;

    //Course_Point_ID
    paramName = "Course_Point_ID";
    objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

    if (isOriginal)
      objParam.ParameterName = "@Original_" + paramName;

    objParam.Value = point.Course_Point_ID.Value;

    //CourseVersion_ID
    paramName = "CourseVersion_ID";
    objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

    if (isOriginal)
      objParam.ParameterName = "@Original_" + paramName;

    objParam.Value = point.CourseVersion_ID.Value;

    //Part
    paramName = "Part";
    objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.Int, 4);
```

```
      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      objParam.Value = point.Part.Value;

      //Point_ID
      paramName = "Point_ID";
      objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      if (point.Point_ID.IsNull)
        objParam.Value = System.DBNull.Value;
      else
        objParam.Value = point.Point_ID.Value;

      //Log
      objCommand = point.Log.AddParameters(objCommand, isOriginal);
    }

    /// <summary>
    /// Validates the current <see cref="StudyPlanning.DAL.Courses.Point"/> object. If the
    /// value of either of the private properties is null a
    /// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
    /// </summary>
    private void Validate()
    {
      try
      {
        _Course_Point_ID.Validate("Course_Point_ID");
        _CourseVersion_ID.Validate("CourseVersion_ID");
        _Part.Validate("Part");
        _Point_ID.Validate("Point_ID");
        _Log.Validate();
      }
      catch (DalException excp)
      {
        throw excp;
      }
    }

    /// <summary>
    /// Retrives the point ID of the specified part of the specifid course version.
    /// </summary>
    /// <param name="courseVersion_ID">The ID of the course version for which to retrieve the point ID.</param>
    /// <param name="part">The part for which to retrieve the point ID.</param>
    /// <returns>A point ID of the specified part of the specified course version.</returns>
    public static Guid GetPoint(Guid courseVersion_ID, int part)
    {
      Guid point_ID;

      string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
      SqlConnection objConnection = new SqlConnection(connString);

      SqlCommand objCommand = new SqlCommand("Course_Point_GetPoint", objConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      SqlParameter objParam;

      objParam = objCommand.Parameters.Add("@CourseVersion_ID", SqlDbType.UniqueIdenti↓er, 16);
      objParam.Value = courseVersion_ID;

      objParam = objCommand.Parameters.Add("@Part", SqlDbType.Int, 4);
      objParam.Value = part;

      objParam = objCommand.Parameters.Add("@Point_ID", SqlDbType.UniqueIdenti↓er, 16);
      objParam.Direction = ParameterDirection.Output;

      try
      {
        objConnection.Open();
        objCommand.ExecuteNonQuery();
        point_ID = (Guid)objCommand.Parameters["@Point_ID"].Value;
        objConnection.Close();
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }

      return point_ID;
    }

    #endregion //Methods
  }
}
```

1.6.11 RecommendedPlacement

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;
```

```
namespace StudyPlanning.DAL.Courses
{
  /// <summary>
  /// Represents a recommended placement of a course.
  /// </summary>
  public class RecommendedPlacement : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalGuid _Course_RecommendedPlacement_ID = new DalGuid(false);
    private DalGuid _CourseVersion_ID = new DalGuid(false);
    private DalInt _StudyType_ID = new DalInt(false);
    private DalGuid _Point_ID = new DalGuid(true);
    private DalInt _RecommendedPlacementConcept_ID = new DalInt(true);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Initializes a new instance of the <see cref="StudyPlanning.DAL.Courses.RecommendedPlacement"/>
    /// class.
    /// </summary>
    public RecommendedPlacement()
    {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the ID of the recommended placement.
    /// </summary>
    public DalGuid Course_RecommendedPlacement_ID
    {
      get { return _Course_RecommendedPlacement_ID; }
    }

    /// <summary>
    /// Gets the ID of the course version.
    /// </summary>
    public DalGuid CourseVersion_ID
    {
      get { return _CourseVersion_ID; }
    }

    /// <summary>
    /// Gets the ID of the study type.
    /// </summary>
    public DalInt StudyType_ID
    {
      get { return _StudyType_ID; }
    }

    /// <summary>
    /// Gets the ID of the point.
    /// </summary>
    public DalGuid Point_ID
    {
      get { return _Point_ID; }
    }

    /// <summary>
    /// Gets the ID of the recommended placement concept.
    /// </summary>
    public DalInt RecommendedPlacementConcept_ID
    {
      get { return _RecommendedPlacementConcept_ID; }
    }

    /// <summary>
    /// Gets the data log of the course.
    /// </summary>
    public DataLog Log
    {
      get { return _Log; }
    }

    #endregion //Public Properties

    #region Methods

    /// <summary>
    /// Creates a new recommended placement in the database using the values of the properties
    /// of the current instance.
    /// </summary>
    public void Create()
    {
      try
      {
        this.Validate();
      }
      catch (DalException excp)
      {
        throw excp;
      }
```

```
    SqlCommand objCommand = new SqlCommand("Course_RecommendedPlacement_Insert", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, false);

    try
    {
      objConnection.Open();
      objCommand.ExecuteNonQuery();
      objConnection.Close();
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
  }

  #region Retrieve Methods

  /// <summary>
  /// Retrieves the recommended placement from the database using the value of the
  /// Course_RecommendedPlacement_ID property of the current instance.
  /// </summary>
  public void Retrieve()
  {
    try
    {
      this._Course_RecommendedPlacement_ID.Validate("Course_RecommendedPlacement_ID");
    }
    catch (DalException excp)
    {
      throw excp;
    }

    try
    {
        RetrieveExecute(_Course_RecommendedPlacement_ID.Value, this, objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }
  }

  /// <summary>
  /// Retrieves the recommended placement with the specified identifier from the database.
  /// </summary>
  /// <param name="course_RecommendedPlacement_ID">The globally unique identifier of the recommended placement.</param>
  public static RecommendedPlacement Retrieve(Guid course_RecommendedPlacement_ID)
  {
    string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
    SqlConnection objConnection = new SqlConnection(connString);

    RecommendedPlacement objRp = new RecommendedPlacement();

    try
    {
      RetrieveExecute(course_RecommendedPlacement_ID, objRp, objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }

    return objRp;
  }

  private static void RetrieveExecute(
    Guid course_RecommendedPlacement_ID,
    RecommendedPlacement objRp,
    SqlConnection sqlConnection)
  {
    SqlCommand objCommand = new SqlCommand("Course_RecommendedPlacement_Select", sqlConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam;

    objParam = objCommand.Parameters.Add("@Course_RecommendedPlacement_ID", SqlDbType.UniqueIdenti↓er, 16);
    objParam.Value = course_RecommendedPlacement_ID;

    DataSet objDataSet = new DataSet("Course_RecommendedPlacement");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
      sqlConnection.Open();
      objAdap.Fill(objDataSet, "RecommendedPlacement");
      sqlConnection.Close();
      DataRow objDataRow = objDataSet.Tables["RecommendedPlacement"].Rows[0];

      objRp.Course_RecommendedPlacement_ID.Value = (Guid)objDataRow["Course_RecommendedPlacement_ID"];
      objRp.CourseVersion_ID.Value = (Guid)objDataRow["CourseVersion_ID"];
      objRp.StudyType_ID.Value = (int)objDataRow["StudyType_ID"];

      if (objDataRow["Point_ID"].Equals(System.DBNull.Value))
        objRp.Point_ID.IsNull = true;
      else
        objRp.Point_ID.Value = (Guid)objDataRow["Point_ID"];
```

```
      if (objDataRow["RecommendedPlacementConcept_ID"].Equals(System.DBNull.Value))
        objRp.RecommendedPlacementConcept_ID.IsNull = true;
      else
        objRp.RecommendedPlacementConcept_ID.Value = (int)objDataRow["RecommendedPlacementConcept_ID"];

      objRp.Log.SetValues(objDataRow);
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
  }

  #endregion //Retrive Methods

  /// <summary>
  /// Updates the current recommended placement in the database using the original
  /// recommended placement to resolve possible concurrency issues.
  /// </summary>
  /// <param name="originalRecommendedPlacement">The original <see cref="StudyPlanning.DAL.Courses.RecommendedPlacement"/> object.</
    param>
  /// <returns><strong>true</strong> if the update was succesfully executed; otherwise, <strong>false</strong>.</returns>
  public bool Update(RecommendedPlacement originalRecommendedPlacement)
  {
    bool blnResult = false;

    SqlCommand objCommand = new SqlCommand("Course_RecommendedPlacement_Update", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, false);
    AddParameters(originalRecommendedPlacement, objCommand, true);

    int rowsA↑ected;

    try
    {
      objConnection.Open();
      rowsA↑ected = objCommand.ExecuteNonQuery();
      objConnection.Close();

      if (rowsA↑ected > 0)
        blnResult = true;
    }
    catch(SqlException excp)
    {
      throw new DalException(excp);
    }

    return blnResult;
  }

  /// <summary>
  /// Deletes the current recommended placement from the database.
  /// </summary>
  /// <returns><strong>true</strong> if the deletion is succesfully executed; otherwise, <strong>false</strong>.</returns>
  public bool Delete()
  {
    bool blnResult = false;

    try
    {
      this.Validate();
    }
    catch (DalException excp)
    {
      throw excp;
    }

    SqlCommand objCommand = new SqlCommand("Course_RecommendedPlacement_Delete", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, true);

    try
    {
      objConnection.Open();
      int rowsA↑ected = objCommand.ExecuteNonQuery();
      objConnection.Close();

      if (rowsA↑ected > 0)
        blnResult = true;
    }
    catch(SqlException objEx)
    {
      throw new DalException(objEx);
    }

    return blnResult;
  }

  /// <summary>
  /// Adds the relevant parameters to the specified <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see> object
  /// setting the values to ones of the specified <see cref="StudyPlanning.DAL.Courses.RecommendedPlacement"/> object.
  /// </summary>
  /// <param name="recommendedPlacement">
  /// The <see cref="StudyPlanning.DAL.Courses.RecommendedPlacement"/> object containing the values that the parameters have to be
  /// initialized to.
  /// </param>
```

```
/// <param name="objCommand">
/// The <see cref="System.Data.SqlClient.SqlCommand"/> object to which parameters should be added.
/// </param>
/// <param name="isOriginal">
/// <strong>true</strong> if the parameters to be added represent original
/// data (for handling concurrent programming issues); otherwise, <strong>false</strong>.</param>
/// <returns>The <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see>
/// object with the relevant parameters added.</returns>
private static void AddParameters(RecommendedPlacement recommendedPlacement, SqlCommand objCommand, bool
  isOriginal)
{
  string paramName = "";
  SqlParameter objParam;

  //Course_RecommendedPlacement_ID
  paramName = "Course_RecommendedPlacement_ID";
  objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

  if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

  objParam.Value = recommendedPlacement.Course_RecommendedPlacement_ID.Value;

  //CourseVersion_ID
  paramName = "CourseVersion_ID";
  objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

  if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

  objParam.Value = recommendedPlacement.CourseVersion_ID.Value;

  //StudyType_ID
  paramName = "StudyType_ID";
  objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.Int, 4);

  if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

  objParam.Value = recommendedPlacement.StudyType_ID.Value;

  //Point_ID
  paramName = "Point_ID";
  objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

  if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

  if (recommendedPlacement.Point_ID.IsNull)
    objParam.Value = System.DBNull.Value;
  else
    objParam.Value = recommendedPlacement.Point_ID.Value;

  //RecommendedPlacementConcept_ID
  paramName = "RecommendedPlacementConcept_ID";
  objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.Int, 4);

  if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

  if (recommendedPlacement.RecommendedPlacementConcept_ID.IsNull)
    objParam.Value = System.DBNull.Value;
  else
    objParam.Value = recommendedPlacement.RecommendedPlacementConcept_ID.Value;

  //Log
  objCommand = recommendedPlacement.Log.AddParameters(objCommand, isOriginal);
}

/// <summary>
/// Validates the current <see cref="StudyPlanning.DAL.Courses.RecommendedPlacement"/> object. If the
/// value of either of the private properties is null a
/// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
/// </summary>
private void Validate()
{
  try
  {
    _Course_RecommendedPlacement_ID.Validate("Course_RecommendedPlacement_ID");
    _CourseVersion_ID.Validate("CourseVersion_ID");
    _StudyType_ID.Validate("StudyType_ID");
    _Point_ID.Validate("Point_ID");
    _RecommendedPlacementConcept_ID.Validate("RecommendedPlacementConcept_ID");
    _Log.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }
}

/// <summary>
/// Retrieves the Course_RecommendedPlacement_IDs for a given CourseVersion_ID and StudyType_ID.
/// </summary>
/// <param name="courseVersion_ID">Identification of the CourseVersion.</param>
/// <param name="studyType_ID">Identification of the StudyType.</param>
/// <returns>A Course_RecommendedPlacement_ID.</returns>
public static Guid GetID(Guid courseVersion_ID, int studyType_ID)
{
```

```
      string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
      SqlConnection objConnection = new SqlConnection(connString);

      SqlCommand objCommand = new SqlCommand("Course_RecommendedPlacement_GetID", objConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      SqlParameter objParam;

      objParam = objCommand.Parameters.Add("@CourseVersion_ID", SqlDbType.UniqueIdenti↓er, 16);
      objParam.Value = courseVersion_ID;

      objParam = objCommand.Parameters.Add("@StudyType_ID", SqlDbType.Int);
      objParam.Value = studyType_ID;

      DataSet objDataSet = new DataSet("Result");
      SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

      try
      {
        objConnection.Open();
        objAdap.Fill(objDataSet, "Result");
        objConnection.Close();
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }
      return (Guid)objDataSet.Tables["Result"].Rows[0]["Course_RecommendedPlacement_ID"];
    }

    #endregion //Methods
  }
}
```

1.6.12 RelationCourse

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.Courses
{
  /// <summary>
  /// Represents a relation (of some type) between to courses.
  /// </summary>
  public class RelationCourse : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalGuid _Course_RelationCourse_ID = new DalGuid(false);
    private DalGuid _CourseVersion_ID = new DalGuid(false);
    private DalInt _Course_RelationCourseType_ID = new DalInt(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Initializes a new instance of the <see cref="StudyPlanning.DAL.Courses.RelationCourse"/>
    /// class.
    /// </summary>
    public RelationCourse()
    {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the ID of the course evaluation form.
    /// </summary>
    public DalGuid Course_RelationCourse_ID
    {
      get { return _Course_RelationCourse_ID; }
    }

    /// <summary>
    /// Gets the ID of the course version.
    /// </summary>
    public DalGuid CourseVersion_ID
    {
      get { return _CourseVersion_ID; }
    }

    /// <summary>
    /// Gets the ID of the evaluation form.
    /// </summary>
    public DalInt Course_RelationCourseType_ID
    {
      get { return _Course_RelationCourseType_ID; }
    }
```

```
/// <summary>
/// Gets the data log of the course.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Methods

#region Retrieve Methods

/// <summary>
/// Retrieves the relation course from the database using the value of the
/// Course_RelationCourse_ID property of the current instance.
/// </summary>
public void Retrieve()
{
  try
  {
    this._Course_RelationCourse_ID.Validate("Course_RelationCourse_ID");
  }
  catch (DalException excp)
  {
    throw excp;
  }

  try
  {
      RetrieveExecute(_Course_RelationCourse_ID.Value, this, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
}

/// <summary>
/// Retrieves the relation course with the specified identifier from the database.
/// </summary>
/// <param name="course_RelationCourse_ID">The globally unique identifier of the relation course.</param>
public static RelationCourse Retrieve(Guid course_RelationCourse_ID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  RelationCourse objRc = new RelationCourse();

  try
  {
    RetrieveExecute(course_RelationCourse_ID, objRc, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }

  return objRc;
}

private static void RetrieveExecute(
  Guid course_RelationCourse_ID,
  RelationCourse objRc,
  SqlConnection sqlConnection)
{
  SqlCommand objCommand = new SqlCommand("Course_RelationCourse_Select", sqlConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  SqlParameter objParam;

  objParam = objCommand.Parameters.Add("@Course_RelationCourse_ID", SqlDbType.UniqueIdenti↓er, 16);
  objParam.Value = course_RelationCourse_ID;

  DataSet objDataSet = new DataSet("Course_RelationCourse");
  SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

  try
  {
    sqlConnection.Open();
    objAdap.Fill(objDataSet, "Course_RelationCourse");
    sqlConnection.Close();
    DataRow objDataRow = objDataSet.Tables["Course_RelationCourse"].Rows[0];

    objRc.Course_RelationCourse_ID.Value = (Guid)objDataRow["Course_RelationCourse_ID"];
    objRc.CourseVersion_ID.Value = (Guid)objDataRow["CourseVersion_ID"];
    objRc.Course_RelationCourseType_ID.Value = (int)objDataRow["Course_RelationCourseType_ID"];

    objRc.Log.SetValues(objDataRow);
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
}

/// <summary>
/// Retrieves a list of Course_RelationCourse_ID associated with
```

```
    /// the <see cref="StudyPlanning.DAL.Courses.CourseVersion"/>.
    /// </summary>
    /// <param name="courseVersion_ID">Identification of the <see cref="StudyPlanning.DAL.Courses.CourseVersion"/>.</param>
    /// <param name="course_RelationCourseType_ID">Identification of the relation type to retrieve.</param>
    public static Guid[] GetIDFromVersion(Guid courseVersion_ID, int course_RelationCourseType_ID)
    {
      string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
      SqlConnection objConnection = new SqlConnection(connString);

      SqlCommand objCommand = new SqlCommand("Course_RelationCourse_GetIDFromVersion", objConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      SqlParameter objParam = objCommand.Parameters.Add("@CourseVersion_ID", SqlDbType.UniqueIdenti↓er);
      objParam.Value = courseVersion_ID;

      objParam = objCommand.Parameters.Add("@Course_RelationCourseType_ID", SqlDbType.Int);
      objParam.Value = course_RelationCourseType_ID;

      DataSet objDataSet = new DataSet("Course");
      SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

      try
      {
        objConnection.Open();
        objAdap.Fill(objDataSet, "Course");
        objConnection.Close();
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }
      int intRows = objDataSet.Tables["Course"].Rows.Count;
      Guid[] ids = new Guid[intRows];

      for (int i=0; i < intRows; i++)
      {
        ids[i] = (Guid)objDataSet.Tables["Course"].Rows[i]["Course_RelationCourse_ID"];
      }

      return ids;
    }

    #endregion //Retrive Methods

    #endregion //Methods
  }
}
```

1.6.13 RelationCourseItem

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.Courses
{
  /// <summary>
  /// Represents an item of a relation between to courses (a relation
  /// is composed of one or more items).
  /// </summary>
  public class RelationCourseItem : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalGuid _Course_RelationCourseItem_ID = new DalGuid(false);
    private DalGuid _Course_RelationCourse_ID = new DalGuid(false);
    private DalGuid _Course_ID = new DalGuid(true);
    private DalString _CourseNumber = new DalString(true);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Initializes a new instance of the <see cref="StudyPlanning.DAL.Courses.RelationCourseItem"/>
    /// class.
    /// </summary>
    public RelationCourseItem()
    {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the ID of the relation course item.
    /// </summary>
    public DalGuid Course_RelationCourseItem_ID
    {
      get { return _Course_RelationCourseItem_ID; }
    }
```

```
/// <summary>
/// Gets the ID of the relation course.
/// </summary>
public DalGuid Course_RelationCourse_ID
{
  get { return _Course_RelationCourse_ID; }
}

/// <summary>
/// Gets the ID of the course ID.
/// </summary>
public DalGuid Course_ID
{
  get { return _Course_ID; }
}

/// <summary>
/// Gets the number of the course.
/// </summary>
public DalString CourseNumber
{
  get { return _CourseNumber; }
}

/// <summary>
/// Gets the data log of the course.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Methods

#region Retrieve Methods

/// <summary>
/// Retrieves the relation course item from the database using the value of the
/// Course_RelationCourseItem_ID property of the current instance.
/// </summary>
public void Retrieve()
{
  try
  {
    this._Course_RelationCourseItem_ID.Validate("Course_RelationCourseItem_ID");
  }
  catch (DalException excp)
  {
    throw excp;
  }

  try
  {
      RetrieveExecute(_Course_RelationCourseItem_ID.Value, this, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
}

/// <summary>
/// Retrieves the relation course item with the specified identifier from the database.
/// </summary>
/// <param name="course_RelationCourseItem_ID">The globally unique identifier of the relation course item.</param>
public static RelationCourseItem Retrieve(Guid course_RelationCourseItem_ID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  RelationCourseItem objRci = new RelationCourseItem();

  try
  {
    RetrieveExecute(course_RelationCourseItem_ID, objRci, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }

  return objRci;
}

private static void RetrieveExecute(
  Guid course_RelationCourseItem_ID,
  RelationCourseItem objRci,
  SqlConnection sqlConnection)
{
  SqlCommand objCommand = new SqlCommand("Course_RelationCourseItem_Select", sqlConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  SqlParameter objParam;

  objParam = objCommand.Parameters.Add("@Course_RelationCourseItem_ID", SqlDbType.UniqueIdenti↓er, 16);
  objParam.Value = course_RelationCourseItem_ID;
```

```
      DataSet objDataSet = new DataSet("Course_RelationCourseItem");
      SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

      try
      {
        sqlConnection.Open();
        objAdap.Fill(objDataSet, "Course_RelationCourseItem");
        sqlConnection.Close();
        DataRow objDataRow = objDataSet.Tables["Course_RelationCourseItem"].Rows[0];

        objRci.Course_RelationCourseItem_ID.Value = (Guid)objDataRow["Course_RelationCourseItem_ID"];
        objRci.Course_RelationCourse_ID.Value = (Guid)objDataRow["Course_RelationCourse_ID"];

        if (objDataRow["Course_ID"].Equals(System.DBNull.Value))
          objRci.Course_ID.IsNull = true;
        else
          objRci.Course_ID.Value = (Guid)objDataRow["Course_ID"];

        if (objDataRow["CourseNumber"].Equals(System.DBNull.Value))
          objRci.CourseNumber.IsNull = true;
        else
          objRci.CourseNumber.Value = (string)objDataRow["CourseNumber"];

        objRci.Log.SetValues(objDataRow);
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }
    }

    /// <summary>
    /// Retrieves a list of Course_IDs associated with
    /// a Course_RelationCourse object.
    /// </summary>
    /// <param name="course_RelationCourse_ID">Identification of the
    /// <see cref="StudyPlanning.DAL.Courses.RelationCourse"/>.</param>
    public static Guid[] GetCourses(Guid course_RelationCourse_ID)
    {
      string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
      SqlConnection objConnection = new SqlConnection(connString);

      SqlCommand objCommand = new SqlCommand("Course_RelationCourseItem_GetCourses", objConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      SqlParameter objParam = objCommand.Parameters.Add("@Course_RelationCourse_ID", SqlDbType.UniqueIdenti↓er);
      objParam.Value = course_RelationCourse_ID;

      DataSet objDataSet = new DataSet("Course");
      SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

      try
      {
        objConnection.Open();
        objAdap.Fill(objDataSet, "Course");
        objConnection.Close();
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }
      int intRows = objDataSet.Tables["Course"].Rows.Count;
      Guid[] ids = new Guid[intRows];

      for (int i=0; i < intRows; i++)
      {
        ids[i] = (Guid)objDataSet.Tables["Course"].Rows[i]["Course_ID"];
      }

      return ids;
    }

    #endregion //Retrive Methods

    #endregion //Methods
  }
}
```

1.6.14 StudyType

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.Courses
{
  /// <summary>
  /// Represents a course study type.
  /// </summary>
  public class StudyType : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalGuid _Course_StudyType_ID = new DalGuid(false);
```

```
private DalGuid _CourseVersion_ID = new DalGuid(false);
private DalInt _StudyType_ID = new DalInt(true);
private DalInt _Course_StudyTypeCategory_ID = new DalInt(true);
private DataLog _Log;

#endregion //Private Properties

#region Constructors

/// <summary>
/// Initializes a new instance of the <see cref="StudyPlanning.DAL.Courses.StudyType"/>
/// class.
/// </summary>
public StudyType()
{
  _Log = new DataLog();
}

#endregion

#region Public Properties

/// <summary>
/// Gets the ID of the course study type.
/// </summary>
public DalGuid Course_StudyType_ID
{
  get { return _Course_StudyType_ID; }
}

/// <summary>
/// Gets the ID of the course version.
/// </summary>
public DalGuid CourseVersion_ID
{
  get { return _CourseVersion_ID; }
}

/// <summary>
/// Gets the ID of the study type.
/// </summary>
public DalInt StudyType_ID
{
  get { return _StudyType_ID; }
}

/// <summary>
/// Gets the ID of the study type category.
/// </summary>
public DalInt Course_StudyTypeCategory_ID
{
  get { return _Course_StudyTypeCategory_ID; }
}

/// <summary>
/// Gets the data log of the course.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Methods

/// <summary>
/// Creates a new course study type in the database using the values of the properties
/// of the current instance.
/// </summary>
public void Create()
{
  try
  {
    this.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }

  SqlCommand objCommand = new SqlCommand("Course_StudyType_Insert", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, false);

  try
  {
    objConnection.Open();
    objCommand.ExecuteNonQuery();
    objConnection.Close();
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
}

#region Retrieve Methods
```

```
/// <summary>
/// Retrieves the course version from the database using the value of the
/// Course_StudyType_ID property of the current instance.
/// </summary>
public void Retrieve()
{
  try
  {
    this._Course_StudyType_ID.Validate("Course_StudyType_ID");
  }
  catch (DalException excp)
  {
    throw excp;
  }

  try
  {
      RetrieveExecute(_Course_StudyType_ID.Value, this, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
}

/// <summary>
/// Retrieves the course study type with the specified identifier from the database.
/// </summary>
/// <param name="course_StudyType_ID">The globally unique identifier of the course study type.</param>
public static StudyType Retrieve(Guid course_StudyType_ID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  StudyType objStudyType = new StudyType();

  try
  {
    RetrieveExecute(course_StudyType_ID, objStudyType, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }

  return objStudyType;
}

private static void RetrieveExecute(
  Guid course_StudyType_ID,
  StudyType objStudyType,
  SqlConnection sqlConnection)
{
  SqlCommand objCommand = new SqlCommand("Course_StudyType_Select", sqlConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  SqlParameter objParam;

  objParam = objCommand.Parameters.Add("@Course_StudyType_ID", SqlDbType.UniqueIdenti↓er, 16);
  objParam.Value = course_StudyType_ID;

  DataSet objDataSet = new DataSet("Course_StudyType");
  SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

  try
  {
    sqlConnection.Open();
    objAdap.Fill(objDataSet, "Course_StudyType");
    sqlConnection.Close();
    DataRow objDataRow = objDataSet.Tables["Course_StudyType"].Rows[0];

    objStudyType.Course_StudyType_ID.Value = (Guid)objDataRow["Course_StudyType_ID"];
    objStudyType.CourseVersion_ID.Value = (Guid)objDataRow["CourseVersion_ID"];

    if (objDataRow["StudyType_ID"].Equals(System.DBNull.Value))
      objStudyType.StudyType_ID.IsNull = true;
    else
      objStudyType.StudyType_ID.Value = (int)objDataRow["StudyType_ID"];

    if (objDataRow["Course_StudyTypeCategory_ID"].Equals(System.DBNull.Value))
      objStudyType.Course_StudyTypeCategory_ID.IsNull = true;
    else
      objStudyType.Course_StudyTypeCategory_ID.Value = (int)objDataRow["Course_StudyTypeCategory_ID"];

    objStudyType.Log.SetValues(objDataRow);
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
}

#endregion //Retrive Methods

/// <summary>
/// Updates the current course version in the database using the original
/// course to resolve possible concurrency issues.
/// </summary>
```

```
/// <param name="originalStudyType">The original <see cref="StudyPlanning.DAL.Courses.StudyType"/> object.</param>
/// <returns><strong>true</strong> if the update was succesfully executed; otherwise, <strong>false</strong>.</returns>
public bool Update(StudyType originalStudyType)
{
  try
  {
    this.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }

  bool blnResult = false;

  SqlCommand objCommand = new SqlCommand("Course_StudyType_Update", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, false);
  AddParameters(originalStudyType, objCommand, true);

  int rowsA↑ected;

  try
  {
    objConnection.Open();
    rowsA↑ected = objCommand.ExecuteNonQuery();
    objConnection.Close();

    if (rowsA↑ected > 0)
      blnResult = true;
  }
  catch(SqlException excp)
  {
    throw new DalException(excp);
  }

  return blnResult;
}

/// <summary>
/// Deletes the current course version from the database.
/// </summary>
/// <returns><strong>true</strong> if the deletion is succesfully executed; otherwise, <strong>false</strong>.</returns>
public bool Delete()
{
  bool blnResult = false;

  try
  {
    this.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }

  SqlCommand objCommand = new SqlCommand("Course_StudyType_Delete", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, true);

  try
  {
    objConnection.Open();
    int rowsA↑ected = objCommand.ExecuteNonQuery();
    objConnection.Close();

    if (rowsA↑ected > 0)
      blnResult = true;
  }
  catch(SqlException objEx)
  {
    throw new DalException(objEx);
  }

  return blnResult;
}

/// <summary>
/// Adds the relevant parameters to the specified <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see> object
/// setting the values to ones of the specified <see cref="StudyPlanning.DAL.Courses.StudyType"/> object.
/// </summary>
/// <param name="studyType">
/// The <see cref="StudyPlanning.DAL.Courses.StudyType"/> object containing the values that the parameters have to be
/// initialized to.
/// </param>
/// <param name="objCommand">
/// The <see cref="System.Data.SqlClient.SqlCommand"/> object to which parameters should be added.
/// </param>
/// <param name="isOriginal">
/// <strong>true</strong> if the parameters to be added represent original
/// data (for handling concurrent programming issues); otherwise, <strong>false</strong>.</param>
/// <returns>The <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see>
/// object with the relevant parameters added.</returns>
private static void AddParameters(StudyType studyType, SqlCommand objCommand, bool isOriginal)
{
  string paramName = "";
  SqlParameter objParam;
```

```
    //Course_StudyType_ID
    paramName = "Course_StudyType_ID";
    objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

    if (isOriginal)
      objParam.ParameterName = "@Original_" + paramName;

    objParam.Value = studyType.Course_StudyType_ID.Value;

    //CourseVersion_ID
    paramName = "CourseVersion_ID";
    objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

    if (isOriginal)
      objParam.ParameterName = "@Original_" + paramName;

    objParam.Value = studyType.CourseVersion_ID.Value;

    //StudyType_ID
    paramName = "StudyType_ID";
    objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.Int, 4);

    if (isOriginal)
      objParam.ParameterName = "@Original_" + paramName;

    if (studyType.StudyType_ID.IsNull)
      objParam.Value = System.DBNull.Value;
    else
      objParam.Value = studyType.StudyType_ID.Value;

    //Course_StudyTypeCategory_ID
    paramName = "Course_StudyTypeCategory_ID";
    objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.Int, 4);

    if (isOriginal)
      objParam.ParameterName = "@Original_" + paramName;

    if (studyType.Course_StudyTypeCategory_ID.IsNull)
      objParam.Value = System.DBNull.Value;
    else
      objParam.Value = studyType.Course_StudyTypeCategory_ID.Value;

    //Log
    objCommand = studyType.Log.AddParameters(objCommand, isOriginal);
  }

  /// <summary>
  /// Validates the current <see cref="StudyPlanning.DAL.Courses.StudyType"/> object. If the
  /// value of either of the private properties is null a
  /// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
  /// </summary>
  private void Validate()
  {
    try
    {
      _Course_StudyType_ID.Validate("Course_StudyType_ID");
      _CourseVersion_ID.Validate("CourseVersion_ID");
      _StudyType_ID.Validate("StudyType_ID");
      _Course_StudyTypeCategory_ID.Validate("Course_StudyTypeCategory_ID");
      _Log.Validate();
    }
    catch (DalException excp)
    {
      throw excp;
    }
  }

  /// <summary>
  /// Retrieves a list of study types to which a given course applies.
  /// </summary>
  /// <param name="courseVersion_ID">Identification of the the course version.</param>
  /// <returns>A possibly empty array of study type identifications.</returns>
  public static int[] GetStudyTypes(Guid courseVersion_ID)
  {
    int[] studytypes = null;

    string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
    SqlConnection objConnection = new SqlConnection(connString);

    SqlCommand objCommand = new SqlCommand("Course_StudyType_GetStudyTypes", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam;

    objParam = objCommand.Parameters.Add("@CourseVersion_ID", SqlDbType.UniqueIdenti↓er, 16);
    objParam.Value = courseVersion_ID;

    DataSet objDataSet = new DataSet("Result");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
      objConnection.Open();
      objAdap.Fill(objDataSet, "Result");
      objConnection.Close();

      int numRows = objDataSet.Tables["Result"].Rows.Count;
      studytypes = new int[numRows];
```

```
        for(int i=0; i<numRows; i++)
        {
          studytypes[i] = (int)objDataSet.Tables["Result"].Rows[i]["StudyType_ID"];
        }
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }
      return studytypes;
    }

    #endregion //Methods
  }
}
```

1.6.15 StudyTypeCategory

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.Courses
{
  /// <summary>
  /// Represents a course study type.
  /// </summary>
  public class StudyTypeCategory : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalInt _Course_StudyTypeCategory_ID = new DalInt(false);
    private DalInt _StudyType_ID = new DalInt(false);
    private DalStringLocalizable _Name = new DalStringLocalizable(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Initializes a new instance of the <see cref="StudyPlanning.DAL.Courses.StudyTypeCategory"/>
    /// class.
    /// </summary>
    public StudyTypeCategory()
    {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the ID of the course study type.
    /// </summary>
    public DalInt Course_StudyTypeCategory_ID
    {
      get { return _Course_StudyTypeCategory_ID; }
    }

    /// <summary>
    /// Gets the ID of the study type.
    /// </summary>
    public DalInt StudyType_ID
    {
      get { return _StudyType_ID; }
    }

    /// <summary>
    /// Gets the name of the category.
    /// </summary>
    public DalStringLocalizable Name
    {
      get { return _Name; }
    }

    /// <summary>
    /// Gets the data log of the course.
    /// </summary>
    public DataLog Log
    {
      get { return _Log; }
    }

    #endregion //Public Properties

    #region Methods

    #region Retrieve Methods

    /// <summary>
    /// Retrieves the course version from the database using the value of the
```

```
    /// Course_StudyTypeCategory_ID property of the current instance.
    /// </summary>
    public void Retrieve()
    {
      try
      {
        this._Course_StudyTypeCategory_ID.Validate("Course_StudyTypeCategory_ID");
      }
      catch (DalException excp)
      {
        throw excp;
      }

      try
      {
          RetrieveExecute(_Course_StudyTypeCategory_ID.Value, this, objConnection);
      }
      catch (DalException objExc)
      {
        throw objExc;
      }
    }

    /// <summary>
    /// Retrieves the course study type with the specified identifier from the database.
    /// </summary>
    /// <param name="course_StudyTypeCategory_ID">The globally unique identifier of the course study type.</param>
    public static StudyTypeCategory Retrieve(int course_StudyTypeCategory_ID)
    {
      string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
      SqlConnection objConnection = new SqlConnection(connString);

      StudyTypeCategory objStudyTypeCategory = new StudyTypeCategory();

      try
      {
        RetrieveExecute(course_StudyTypeCategory_ID, objStudyTypeCategory, objConnection);
      }
      catch (DalException objExc)
      {
        throw objExc;
      }

      return objStudyTypeCategory;
    }

    private static void RetrieveExecute(
      int course_StudyTypeCategory_ID,
      StudyTypeCategory objStudyTypeCategory,
      SqlConnection sqlConnection)
    {
      SqlCommand objCommand = new SqlCommand("Course_StudyTypeCategory_Select", sqlConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      SqlParameter objParam;

      objParam = objCommand.Parameters.Add("@Course_StudyTypeCategory_ID", SqlDbType.Int, 4);
      objParam.Value = course_StudyTypeCategory_ID;

      DataSet objDataSet = new DataSet("Course_StudyTypeCategory");
      SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

      try
      {
        sqlConnection.Open();
        objAdap.Fill(objDataSet, "Course_StudyTypeCategory");
        sqlConnection.Close();
        DataRow objDataRow = objDataSet.Tables["Course_StudyTypeCategory"].Rows[0];

        objStudyTypeCategory.Course_StudyTypeCategory_ID.Value = (int)objDataRow["Course_StudyTypeCategory_ID"];
        objStudyTypeCategory.StudyType_ID.Value = (int)objDataRow["StudyType_ID"];
        objStudyTypeCategory.Name.LoadXml(objDataRow["Name"].ToString());
        objStudyTypeCategory.Log.SetValues(objDataRow);
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }
    }

    #endregion //Retrive Methods

    #endregion //Methods
  }
}
```

1.7 Department

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL
```

```
{
  /// <summary>
  /// Represents a department.
  /// </summary>
  public class Department : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalInt _Department_ID = new DalInt(false);
    private DalGuid _ContactData_ID = new DalGuid(true);
    private DalStringLocalizable _Name = new DalStringLocalizable(false);
    private DalInt _Number = new DalInt(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.Department" />
    /// class.
    /// </summary>
    public Department()
    {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the ID of the department.
    /// </summary>
    public DalInt Department_ID
    {
      get { return _Department_ID; }
    }

    /// <summary>
    /// Gets the contact data ID of the department.
    /// </summary>
    public DalGuid ContactData_ID
    {
      get { return _ContactData_ID; }
    }

    /// <summary>
    /// Gets the name of the department.
    /// </summary>
    public DalStringLocalizable Name
    {
      get { return _Name; }
    }

    /// <summary>
    /// Gets the number of the department.
    /// </summary>
    public DalInt Number
    {
      get { return _Number; }
    }

    /// <summary>
    /// Gets the data log of the department.
    /// </summary>
    public DataLog Log
    {
      get { return _Log; }
    }

    #endregion //Public Properties

    #region Methods

    #region Retrieve Methods

    /// <summary>
    /// Retrieves the department from the database using the value of the
    /// Department_ID property of the current instance.
    /// </summary>
    public void Retrieve()
    {
      try
      {
        this._Department_ID.Validate("Department_ID");
      }
      catch (DalException excp)
      {
        throw excp;
      }

      try
      {
        RetrieveExecute(_Department_ID.Value, this, objConnection);
      }
      catch (DalException objExc)
      {
        throw objExc;
```

```
    }
}

/// <summary>
/// Retrieves the department with the specified identifier from the database.
/// </summary>
/// <param name="department_ID">The globally unique identifier of the department.</param>
public static Department Retrieve(int department_ID)
{
    string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
    SqlConnection objConnection = new SqlConnection(connString);

    Department objDepartment = new Department();

    try
    {
        RetrieveExecute(department_ID, objDepartment, objConnection);
    }
    catch (DalException objExc)
    {
        throw objExc;
    }

    return objDepartment;
}

private static void RetrieveExecute(
    int department_ID,
    Department objDepartment,
    SqlConnection sqlConnection)
{
    SqlCommand objCommand = new SqlCommand("Department_Select", sqlConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam;

    objParam = objCommand.Parameters.Add("@Department_ID", SqlDbType.Int, 4);
    objParam.Value = department_ID;

    DataSet objDataSet = new DataSet("Department");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
        sqlConnection.Open();
        objAdap.Fill(objDataSet, "Department");
        sqlConnection.Close();
        DataRow objDataRow = objDataSet.Tables["Department"].Rows[0];

        objDepartment.Department_ID.Value = (int)objDataRow["Department_ID"];

        if (!objDataRow["ContactData_ID"].Equals(System.DBNull.Value))
            objDepartment.ContactData_ID.Value = (Guid)objDataRow["ContactData_ID"];

        objDepartment.Name.LoadXml(objDataRow["Name"].ToString());
        objDepartment.Number.Value = (int)objDataRow["Number"];
        objDepartment.Log.SetValues(objDataRow);
    }
    catch (SqlException objExc)
    {
        throw new DalException(objExc);
    }
}

#endregion //Retrive Methods

/// <summary>
/// Gets the integer ID of the department having the specified
/// number.
/// </summary>
/// <param name="number">The number of the department.</param>
/// <returns>The integer ID of the department.</returns>
public static int GetDepartmentID(int number)
{
    int departmentID;

    string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
    SqlConnection objConnection = new SqlConnection(connString);

    SqlCommand objCommand = new SqlCommand("Department_GetIDFromNumber", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam;

    objParam = objCommand.Parameters.Add("@Number", SqlDbType.Int, 4);
    objParam.Value = number;

    objParam = objCommand.Parameters.Add("@Department_ID", SqlDbType.Int, 4);
    objParam.Direction = ParameterDirection.Output;

    try
    {
        objConnection.Open();
        //SqlDataReader objReader = objCommand.ExecuteReader();
        //departmentID = (int)objReader["Department_ID"];
        objCommand.ExecuteNonQuery();
        departmentID = Convert.ToInt32(objCommand.Parameters["@Department_ID"].Value);
        objConnection.Close();
        return departmentID;
```

```
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }
    }

    /// <summary>
    /// Gets the IDs of all the departments in the system.
    /// </summary>
    /// <returns>An array containing the IDs of the departments.</returns>
    public static int[] GetDepartments()
    {
      int[] departments;

      string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
      SqlConnection objConnection = new SqlConnection(connString);

      SqlCommand objCommand = new SqlCommand("Department_GetDepartments", objConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      DataSet objDataSet = new DataSet("Departments");
      SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

      try
      {
        objConnection.Open();
        objAdap.Fill(objDataSet, "Departments");
        objConnection.Close();

        int numRows = objDataSet.Tables["Departments"].Rows.Count;
        departments = new int[numRows];

        for(int i=0; i<numRows; i++)
        {
          departments[i] = (int)objDataSet.Tables["Departments"].Rows[i]["Department_ID"];
        }
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }

      return departments;
    }

    #endregion //Methods
  }
}
```

1.8 Gender

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL
{
  /// <summary>
  /// Represents a studyPlanCriterion.
  /// </summary>
  public class Gender : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalInt _Gender_ID = new DalInt(false);
    private DalStringLocalizable _Name = new DalStringLocalizable(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.Gender"/>
    /// class.
    /// </summary>
    public Gender() {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the Gender_ID.
    /// </summary>
    /// <value></value>
    public DalInt Gender_ID
    {
      get { return _Gender_ID; }
    }
```

```
/// <summary>
/// Gets the name of the gender.
/// </summary>
public DalStringLocalizable Name
{
  get { return _Name; }
}

/// <summary>
/// Gets the data log of the course.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Public Methods

#region Retrieve Methods

/// <summary>
/// Retrieves the Gender from the database using the value of the
/// Gender_ID property of the current instance
/// </summary>
public void Retrieve()
{
  try
  {
    this._Gender_ID.Validate("Gender_ID");
  }
  catch (DalException e)
  {
    throw e;
  }

  try
  {
    RetrieveExecute(_Gender_ID.Value, this, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
}

/// <summary>
/// Retrieves the Gender with the specified identifier from the database.
/// </summary>
/// <param name="gender_ID"></param>
public static Gender Retrieve(int gender_ID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  Gender objGender = new Gender();

  try
  {
    RetrieveExecute(gender_ID, objGender, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
  return objGender;
}

private static void RetrieveExecute(
  int gender_ID,
  Gender gender,
  SqlConnection sqlConnection)
{
  DataRow objDataRow;

  SqlCommand objCommand = new SqlCommand("Gender_Select", sqlConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  SqlParameter objParam;

  objParam = objCommand.Parameters.Add("@Gender_ID", SqlDbType.Int);
  objParam.Value = gender_ID;

  DataSet objDataSet = new DataSet("Gender");
  SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

  try
  {
    sqlConnection.Open();
    objAdap.Fill(objDataSet, "Gender");
    sqlConnection.Close();
    objDataRow = objDataSet.Tables["Gender"].Rows[0];
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
```

```
      gender._Gender_ID.Value = (int)objDataRow["Gender_ID"];
      gender._Name.LoadXml((string)objDataRow["Name"]);
      gender._Log.SetValues(objDataRow);
    }

    #endregion //Retrieve Methods

    #endregion //Public Methods

  }
}
```

1.9 Grade

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL
{
  /// <summary>
  /// Represents a way whereupon a course or a project can be assessed.
  /// </summary>
  public class Grade : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalInt _Grade_ID = new DalInt(false);
    private DalInt _Number = new DalInt(false);
    private DalString _Name = new DalString(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.Grade" />
    /// class.
    /// </summary>
    public Grade()
    {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the ID of the grade.
    /// </summary>
    public DalInt Grade_ID
    {
      get { return _Grade_ID; }
    }

    /// <summary>
    /// Gets the numeric representation of the grade number.
    /// </summary>
    public DalInt Number
    {
      get { return _Number; }
    }

    /// <summary>
    /// Gets the string representation of the grade number.
    /// </summary>
    public DalString Name
    {
      get { return _Name; }
    }

    /// <summary>
    /// Gets the data log of the grade.
    /// </summary>
    public DataLog Log
    {
      get { return _Log; }
    }

    #endregion //Public Properties

    #region Methods

    #region Retrieve Methods

    /// <summary>
    /// Retrieves the grade from the database using the value of the
    /// Grade_ID property of the current instance.
    /// </summary>
    public void Retrieve()
    {
```

```
      try
      {
        this._Grade_ID.Validate("Grade_ID");
      }
      catch (DalException excp)
      {
        throw excp;
      }

      try
      {
        RetrieveExecute(_Grade_ID.Value, this, objConnection);
      }
      catch (DalException objExc)
      {
        throw objExc;
      }
    }

    /// <summary>
    /// Retrieves the grade with the specified identifier from the database.
    /// </summary>
    /// <param name="grade_ID">The globally unique identifier of the grade.</param>
    public static Grade Retrieve(int grade_ID)
    {
      string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
      SqlConnection objConnection = new SqlConnection(connString);

      Grade objGrade = new Grade();

      try
      {
        RetrieveExecute(grade_ID, objGrade, objConnection);
      }
      catch (DalException objExc)
      {
        throw objExc;
      }

      return objGrade;
    }

    private static void RetrieveExecute(
      int grade_ID,
      Grade objGrade,
      SqlConnection sqlConnection)
    {
      SqlCommand objCommand = new SqlCommand("Grade_Select", sqlConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      SqlParameter objParam;

      objParam = objCommand.Parameters.Add("@Grade_ID", SqlDbType.Int, 4);
      objParam.Value = grade_ID;

      DataSet objDataSet = new DataSet("Grade");
      SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

      try
      {
        sqlConnection.Open();
        objAdap.Fill(objDataSet, "Grade");
        sqlConnection.Close();
        DataRow objDataRow = objDataSet.Tables["Grade"].Rows[0];

        objGrade.Grade_ID.Value = (int)objDataRow["Grade_ID"];
        objGrade.Number.Value = (int)objDataRow["Number"];
        objGrade.Name.Value = (string)objDataRow["Name"];
        objGrade.Log.SetValues(objDataRow);
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }
    }

    #endregion //Retrive Methods

    #endregion //Methods
  }
}
```

1.10 Keyword

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL
{
  /// <summary>
  /// Represents a keyword.
  /// </summary>
```

```
public class Keyword : StudyPlanning.DAL.DbObject
{
  #region Private Properties

  private DalGuid _Keyword_ID = new DalGuid(false);
  private DalString _Name = new DalString(false);
  private DalString _Culture_ID = new DalString(false);
  private DataLog _Log;

  #endregion //Private Properties

  #region Constructors

  /// <summary>
  /// Creates a new instance of the <see cref="StudyPlanning.DAL.Keyword">Keyword</see>
  /// class.
  /// </summary>
  public Keyword() {
    _Log = new DataLog();
  }

  #endregion

  #region Public Properties

  /// <summary>
  /// Gets the ID of the keyword.
  /// </summary>
  /// <value></value>
  public DalGuid Keyword_ID
  {
    get { return _Keyword_ID; }
  }

  /// <summary>
  /// Gets the name of the keyword.
  /// </summary>
  public DalString Name
  {
    get { return _Name; }
  }

  /// <summary>
  /// Gets the culture_id of the keyword.
  /// </summary>
  public DalString Culture_ID
  {
    get { return _Culture_ID; }
  }

  /// <summary>
  /// Gets the data log of the course.
  /// </summary>
  public DataLog Log
  {
    get { return _Log; }
  }

  #endregion //Public Properties

  #region Public Methods

  #region Create Methods

  /// <summary>
  /// Creates a keyword in the database using the values of the properties
  /// of the current instance.
  /// </summary>
  public void Create()
  {
    try
    {
      _Keyword_ID.Validate("Keyword_ID");
      _Name.Validate("Name");
      _Culture_ID.Validate("Culture_ID");
      _Log.Validate();
    }
    catch (DalException excp)
    {
      throw excp;
    }

    SqlCommand objCommand = new SqlCommand("Keyword_Insert", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam;

    objParam = objCommand.Parameters.Add("@Keyword_ID", SqlDbType.UniqueIdenti↓er);
    objParam.Value = _Keyword_ID.Value;

    objParam = objCommand.Parameters.Add("@Name", SqlDbType.VarChar);
    objParam.Value = _Name.Value;

    objParam = objCommand.Parameters.Add("@Culture_ID", SqlDbType.VarChar);
    objParam.Value = _Culture_ID.Value;

    objCommand = _Log.AddParameters(objCommand, false);

    DataSet objDataSet = new DataSet("Keyword");
```

```
  SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

  try
  {
    objConnection.Open();
    objAdap.Fill(objDataSet, "Keyword");
    objConnection.Close();
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
}

#endregion

#region Retrieve Methods

/// <summary>
/// Retrieves the keyword from the database using the value of the
/// Keyword_ID property of the current instance
/// </summary>
public void Retrieve()
{
  DataRow objDataRow;

  try
  {
    _Keyword_ID.Validate("Keyword_ID");
  }
  catch (DalException excp)
  {
    throw excp;
  }

  try
  {
    objDataRow = RetrieveExecute(_Keyword_ID.Value, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
  this._Keyword_ID.Value  = (Guid)objDataRow["Keyword_ID"];
  this._Name.Value      = (string)objDataRow["Name"];
  this._Culture_ID.Value  = (string)objDataRow["Culture_ID"];
  this._Log.SetValues(objDataRow);
}

/// <summary>
/// Retrieves the keyword with the specified identifier from the database.
/// </summary>
/// <param name="keyword_ID">The globally unique identifier of the keyword.</param>
public static Keyword Retrieve(Guid keyword_ID)
{
  DataRow objDataRow;
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  Keyword objKeyword = new Keyword();

  try
  {
    objDataRow = RetrieveExecute(keyword_ID, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
  objKeyword._Keyword_ID.Value  = (Guid)objDataRow["Keyword_ID"];
  objKeyword._Name.Value      = (string)objDataRow["Name"];
  objKeyword._Culture_ID.Value  = (string)objDataRow["Culture_ID"];
  objKeyword._Log.SetValues(objDataRow);
  return objKeyword;
}

private static DataRow RetrieveExecute(
  Guid keyword_ID,
  SqlConnection sqlConnection)
{
  SqlCommand objCommand = new SqlCommand("Keyword_Select", sqlConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  SqlParameter objParam;

  objParam = objCommand.Parameters.Add("@Keyword_ID", SqlDbType.UniqueIdenti↓er, 16);
  objParam.Value = keyword_ID;

  DataSet objDataSet = new DataSet("Keyword");
  SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

  try
  {
    sqlConnection.Open();
    objAdap.Fill(objDataSet, "Keyword");
    sqlConnection.Close();
    DataRow objDataRow = objDataSet.Tables["Keyword"].Rows[0];
    return objDataRow;
  }
```

```
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
  }

  #endregion //Retrieve Methods

  #region Update Methods

  /// <summary>
  /// Updates the current keyword in the database using the original
  /// keyword to resolve possible concurrency issues.
  /// </summary>
  /// <param name="originalKeyword">The keyword <see cref="StudyPlanning.DAL.Keyword">Keyword</see> object.</param>
  /// <returns><strong>true</strong> if the update was succesfully executed; otherwise, <strong>false</strong>.</returns>
  public bool Update(Keyword originalKeyword)
  {
    bool blnResult = false;

    try
    {
      _Keyword_ID.Validate("Keyword_ID");
      _Name.Validate("Name");
      _Culture_ID.Validate("Culture_ID");
      _Log.Validate();
    }
    catch (DalException excp)
    {
      throw excp;
    }

    SqlCommand objCommand = new SqlCommand("Keyword_Update", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam;

    objParam = objCommand.Parameters.Add("@Keyword_ID", SqlDbType.UniqueIdenti↓er);
    objParam.Value = this._Keyword_ID.Value;

    objParam = objCommand.Parameters.Add("@Name", SqlDbType.VarChar);
    objParam.Value = this._Name.Value;

    objParam = objCommand.Parameters.Add("@Culture_ID", SqlDbType.VarChar);
    objParam.Value = this._Culture_ID.Value;

    objCommand = this.Log.AddParameters(objCommand, false);

    objParam = objCommand.Parameters.Add("@Original_Keyword_ID", SqlDbType.UniqueIdenti↓er);
    objParam.Value = originalKeyword._Keyword_ID.Value;

    objParam = objCommand.Parameters.Add("@Original_Name", SqlDbType.VarChar);
    objParam.Value = originalKeyword._Name.Value;

    objParam = objCommand.Parameters.Add("@Original_Culture_ID", SqlDbType.VarChar);
    objParam.Value = originalKeyword._Culture_ID.Value;

    objCommand = originalKeyword.Log.AddParameters(objCommand, true);

    int rowsA↑ected;

    try
    {
      objConnection.Open();
      rowsA↑ected = objCommand.ExecuteNonQuery();
      objConnection.Close();

      if (rowsA↑ected > 0)
        blnResult = true;
    }
    catch(SqlException excp)
    {
      throw new DalException(excp);
    }
    return blnResult;
  }

  #endregion //Update Methods

  #region Delete Methods

  /// <summary>
  /// Deletes the current keyword from the database.
  /// </summary>
  /// <returns><strong>true</strong> if the deletion is succesfully executed; otherwise, <strong>false</strong>.</returns>
  public bool Delete()
  {
    bool blnResult = false;

    try
    {
      _Keyword_ID.Validate("Keyword_ID");
      _Name.Validate("Name");
      _Culture_ID.Validate("Culture_ID");
      _Log.Validate();
    }
    catch (DalException excp)
    {
      throw excp;
```

```
        }

        SqlCommand objCommand = new SqlCommand("Keyword_Delete", objConnection);
        objCommand.CommandType = CommandType.StoredProcedure;

        SqlParameter objParam;

        objParam = objCommand.Parameters.Add("@Original_Keyword_ID", SqlDbType.UniqueIdenti↓er);
        objParam.Value = this._Keyword_ID.Value;

        objParam = objCommand.Parameters.Add("@Original_Name", SqlDbType.VarChar);
        objParam.Value = this._Name.Value;

        objParam = objCommand.Parameters.Add("@Original_Culture_ID", SqlDbType.VarChar);
        objParam.Value = this._Culture_ID.Value;

        objCommand = this.Log.AddParameters(objCommand, true);

        int rowsA↑ected;
        try
        {
          objConnection.Open();
          rowsA↑ected = objCommand.ExecuteNonQuery();
          objConnection.Close();

          if (rowsA↑ected > 0)
            blnResult = true;
        }
        catch(SqlException e)
        {
          throw new DalException(e);
        }

        return blnResult;
      }

      #endregion //Delete Methods

      /// <summary>
      /// Retrieves a list of keywords which match the specified keyword in the specified culture.
      /// </summary>
      /// <param name="keyword">The keyword for which to search for matches.</param>
      /// <param name="culture_ID">The ID of the culture in which to search for matches.</param>
      /// <returns>An array containing IDs of the matching keywords.</returns>
      public static Guid[] SearchForKeywords(string keyword, string culture_ID)
      {
        Guid[] keywords;

        string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
        SqlConnection objConnection = new SqlConnection(connString);

        SqlCommand objCommand = new SqlCommand("Keyword_SearchForKeywords", objConnection);
        objCommand.CommandType = CommandType.StoredProcedure;

        SqlParameter objParam;

        objParam = objCommand.Parameters.Add("@Keyword", SqlDbType.VarChar, 100);
        objParam.Value = keyword;

        objParam = objCommand.Parameters.Add("@Culture_ID", SqlDbType.VarChar, 100);
        objParam.Value = culture_ID;

        DataSet objDataSet = new DataSet("Keywords");
        SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

        try
        {
          objConnection.Open();
          objAdap.Fill(objDataSet, "Keywords");
          objConnection.Close();

          int numRows = objDataSet.Tables["Keywords"].Rows.Count;
          keywords = new Guid[numRows];

          for(int i=0; i<numRows; i++)
          {
            keywords[i] = (Guid)objDataSet.Tables["Keywords"].Rows[i]["Keyword_ID"];
          }
        }
        catch (SqlException objExc)
        {
          throw new DalException(objExc);
        }

        return keywords;
      }

      #endregion //Public Methods

    }
}
```

1.11 Language

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL
{
  /// <summary>
  /// Represents a language.
  /// </summary>
  public class Language : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalString _Language_ID = new DalString(false);
    private DalStringLocalizable _Name = new DalStringLocalizable(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.Language" />
    /// class.
    /// </summary>
    public Language()
    {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the ID of the language.
    /// </summary>
    public DalString Language_ID
    {
      get { return _Language_ID; }
    }

    /// <summary>
    /// Gets the name of the language.
    /// </summary>
    public DalStringLocalizable Name
    {
      get { return _Name; }
    }

    /// <summary>
    /// Gets the data log of the language.
    /// </summary>
    public DataLog Log
    {
      get { return _Log; }
    }

    #endregion //Public Properties

    #region Methods

    #region Retrieve Methods

    /// <summary>
    /// Retrieves the language from the database using the value of the
    /// Language_ID property of the current instance.
    /// </summary>
    public void Retrieve()
    {
      try
      {
        this._Language_ID.Validate("Language_ID");
      }
      catch (DalException excp)
      {
        throw excp;
      }

      try
      {
        RetrieveExecute(_Language_ID.Value, this, objConnection);
      }
      catch (DalException objExc)
      {
        throw objExc;
      }
    }

    /// <summary>
    /// Retrieves the language with the specified identifier from the database.
    /// </summary>
    /// <param name="language_ID">The globally unique identifier of the language.</param>
    public static Language Retrieve(string language_ID)
    {
      string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
      SqlConnection objConnection = new SqlConnection(connString);
```

```
      Language objLanguage = new Language();

      try
      {
        RetrieveExecute(language_ID, objLanguage, objConnection);
      }
      catch (DalException objExc)
      {
        throw objExc;
      }

      return objLanguage;
    }

    private static void RetrieveExecute(
      string language_ID,
      Language objLanguage,
      SqlConnection sqlConnection)
    {
      SqlCommand objCommand = new SqlCommand("Language_Select", sqlConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      SqlParameter objParam;

      objParam = objCommand.Parameters.Add("@Language_ID", SqlDbType.Char, 2);
      objParam.Value = language_ID;

      DataSet objDataSet = new DataSet("Language");
      SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

      try
      {
        sqlConnection.Open();
        objAdap.Fill(objDataSet, "Language");
        sqlConnection.Close();
        DataRow objDataRow = objDataSet.Tables["Language"].Rows[0];

        objLanguage.Language_ID.Value = (string)objDataRow["Language_ID"];
        objLanguage.Name.LoadXml(objDataRow["Name"].ToString());
        objLanguage.Log.SetValues(objDataRow);
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }
    }

    #endregion //Retrive Methods

    #endregion //Methods
  }
}
```

1.12 Lecturer

```
using System;
using System.Data;
using System.Data.SqlClient;

namespace StudyPlanning.DAL
{
  /// <summary>
  /// Represents a lecturer.
  /// </summary>
  public class Lecturer : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalGuid _Lecturer_ID = new DalGuid(false);
    private DalInt _CwisNumber = new DalInt(true);
    private DalGuid _Person_ID = new DalGuid(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Initializes a new instance of the <see cref="StudyPlanning.DAL.Lecturer"/>
    /// class.
    /// </summary>
    public Lecturer()
    {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the ID of lecturer.
    /// </summary>
    public DalGuid Lecturer_ID
```

```
{
  get { return _Lecturer_ID; }
}

/// <summary>
/// Gets the CWIS number of the lecturer.
/// </summary>
public DalInt CwisNumber
{
  get { return _CwisNumber; }
}

/// <summary>
/// Gets the person ID of the lecturer.
/// </summary>
public DalGuid Person_ID
{
  get { return _Person_ID; }
}

/// <summary>
/// Gets the data log of the course.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Methods

/// <summary>
/// Creates a new lecturer in the database using the values of the properties
/// of the current instance.
/// </summary>
public void Create()
{
  try
  {
    this.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }

  SqlCommand objCommand = new SqlCommand("Lecturer_Insert", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, false);

  try
  {
    objConnection.Open();
    objCommand.ExecuteNonQuery();
    objConnection.Close();
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
}

#region Retrieve Methods

/// <summary>
/// Retrieves the lecturer from the database using the value of the
/// Lecturer_ID property of the current instance.
/// </summary>
public void Retrieve()
{
  try
  {
    this._Lecturer_ID.Validate("Lecturer_ID");
  }
  catch (DalException excp)
  {
    throw excp;
  }

  try
  {
      RetrieveExecute(_Lecturer_ID.Value, this, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
}

/// <summary>
/// Retrieves the lecturer with the specified identifier from the database.
/// </summary>
/// <param name="lecturer_ID">The globally unique identifier of the lecturer.</param>
public static Lecturer Retrieve(Guid lecturer_ID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);
```

```
    Lecturer objSp = new Lecturer();

    try
    {
      RetrieveExecute(lecturer_ID, objSp, objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }

    return objSp;
  }

  private static void RetrieveExecute(
    Guid lecturer_ID,
    Lecturer objLecturer,
    SqlConnection sqlConnection)
  {
    SqlCommand objCommand = new SqlCommand("Lecturer_Select", sqlConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam;

    objParam = objCommand.Parameters.Add("@Lecturer_ID", SqlDbType.UniqueIdenti↓er, 16);
    objParam.Value = lecturer_ID;

    DataSet objDataSet = new DataSet("Lecturer");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
      sqlConnection.Open();
      objAdap.Fill(objDataSet, "Lecturer");
      sqlConnection.Close();
      DataRow objDataRow = objDataSet.Tables["Lecturer"].Rows[0];

      objLecturer.Lecturer_ID.Value = (Guid)objDataRow["Lecturer_ID"];

      if (objDataRow["CwisNumber"].Equals(System.DBNull.Value))
        objLecturer.CwisNumber.IsNull = true;
      else
        objLecturer.CwisNumber.Value = Convert.ToInt32(objDataRow["CwisNumber"]);

      objLecturer.Person_ID.Value = (Guid)objDataRow["Person_ID"];
      objLecturer.Log.SetValues(objDataRow);
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
  }

  #endregion //Retrive Methods

  /// <summary>
  /// Updates the current lecturer in the database using the original
  /// lecturer to resolve possible concurrency issues.
  /// </summary>
  /// <param name="originalLecturer">The original <see cref="StudyPlanning.DAL.Lecturer"/> object.</param>
  /// <returns><strong>true</strong> if the update was succesfully executed; otherwise, <strong>false</strong>.</returns>
  public bool Update(Lecturer originalLecturer)
  {
    try
    {
      this.Validate();
    }
    catch (DalException excp)
    {
      throw excp;
    }

    bool blnResult = false;

    SqlCommand objCommand = new SqlCommand("Lecturer_Update", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, false);
    AddParameters(originalLecturer, objCommand, true);

    int rowsA↑ected;

    try
    {
      objConnection.Open();
      rowsA↑ected = objCommand.ExecuteNonQuery();
      objConnection.Close();

      if (rowsA↑ected > 0)
        blnResult = true;
    }
    catch(SqlException excp)
    {
      throw new DalException(excp);
    }

    return blnResult;
  }
```

```
/// <summary>
/// Deletes the current lecturer from the database.
/// </summary>
/// <returns><strong>true</strong> if the deletion is succesfully executed; otherwise, <strong>false</strong>.</returns>
public bool Delete()
{
  bool blnResult = false;

  try
  {
    this.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }

  SqlCommand objCommand = new SqlCommand("Lecturer_Delete", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, true);

  try
  {
    objConnection.Open();
    int rowsA↑ected = objCommand.ExecuteNonQuery();
    objConnection.Close();

    if (rowsA↑ected > 0)
      blnResult = true;
  }
  catch(SqlException objEx)
  {
    throw new DalException(objEx);
  }

  return blnResult;
}

/// <summary>
/// Adds the relevant parameters to the specified <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see> object
/// setting the values to ones of the specified <see cref="StudyPlanning.DAL.Lecturer"/> object.
/// </summary>
/// <param name="lecturer">
/// The <see cref="StudyPlanning.DAL.Lecturer"/> object containing the values that the parameters have to be
/// initialized to.
/// </param>
/// <param name="objCommand">
/// The <see cref="System.Data.SqlClient.SqlCommand"/> object to which parameters should be added.
/// </param>
/// <param name="isOriginal">
/// <strong>true</strong> if the parameters to be added represent original
/// data (for handling concurrent programming issues); otherwise, <strong>false</strong>.</param>
/// <returns>The <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see>
/// object with the relevant parameters added.</returns>
private static void AddParameters(Lecturer lecturer, SqlCommand objCommand, bool isOriginal)
{
  string paramName = "";
  SqlParameter objParam;

  //Lecturer_ID
  paramName = "Lecturer_ID";
  objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

  if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

  objParam.Value = lecturer.Lecturer_ID.Value;

  //CwisNumber
  paramName = "CwisNumber";
  objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.Int, 4);

  if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

  if (lecturer.CwisNumber.IsNull)
    objParam.Value = System.DBNull.Value;
  else
    objParam.Value = lecturer.CwisNumber.Value;

  //Person_ID
  paramName = "Person_ID";
  objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

  if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

  objParam.Value = lecturer.Person_ID.Value;

  //Log
  objCommand = lecturer.Log.AddParameters(objCommand, isOriginal);
}

/// <summary>
/// Validates the current <see cref="StudyPlanning.DAL.Lecturer"/> object. If the
/// value of either of the private properties is null a
/// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
```

```
    /// </summary>
    private void Validate()
    {
      try
      {
        _Lecturer_ID.Validate("Lecturer_ID");
        _CwisNumber.Validate("CwisNumber");
        _Person_ID.Validate("Person_ID");
        _Log.Validate();
      }
      catch (DalException excp)
      {
        throw excp;
      }
    }

    /// <summary>
    /// Gets the ID of the lecturer having the specified CWIS number.
    /// </summary>
    /// <param name="cwisNumber">The CWIS number of the lecturer.</param>
    /// <returns>The GUID of the lecturer</returns>
    public static Guid GetIDFromCwisNumber(int cwisNumber)
    {
      Guid lecturerID;

      string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
      SqlConnection objConnection = new SqlConnection(connString);

      SqlCommand objCommand = new SqlCommand("Lecturer_GetIDFromCwisNumber", objConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      SqlParameter objParam;

      objParam = objCommand.Parameters.Add("@CwisNumber", SqlDbType.Int, 4);
      objParam.Value = cwisNumber;

      objParam = objCommand.Parameters.Add("@Lecturer_ID", SqlDbType.UniqueIdenti↓er, 16);
      objParam.Direction = ParameterDirection.Output;

      try
      {
        objConnection.Open();
        objCommand.ExecuteNonQuery();
        lecturerID = (Guid)objCommand.Parameters["@Lecturer_ID"].Value;
        objConnection.Close();
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }

      return lecturerID;
    }

    #endregion //Methods
  }
}
```

1.13 Module

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL
{
  /// <summary>
  /// Represents a module.
  /// </summary>
  public class Module : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalInt _Module_ID = new DalInt(false);
    private DalString _Name = new DalString(false);
    private DalDateTime _StartTime = new DalDateTime(false);
    private DalDateTime _EndTime = new DalDateTime(false);
    private DalInt _Weekday_ID = new DalInt(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.Module">Module</see>
    /// class.
    /// </summary>
    public Module() {
      _Log = new DataLog();
    }

    #endregion
```

```
#region Public Properties

/// <summary>
/// Gets the ID of the module.
/// </summary>
/// <value></value>
public DalInt Module_ID
{
  get { return _Module_ID; }
}

/// <summary>
/// Gets the name of the module.
/// </summary>
public DalString Name
{
  get { return _Name; }
}

/// <summary>
/// Gets the start time of the module.
/// </summary>
public DalDateTime StartTime
{
  get { return _StartTime; }
}

/// <summary>
/// Gets the end time of the module.
/// </summary>
public DalDateTime EndTime
{
  get { return _EndTime; }
}

/// <summary>
/// Gets the weekday ID of the module.
/// </summary>
public DalInt Weekday_ID
{
  get { return _Weekday_ID; }
}

/// <summary>
/// Gets the data log of the course.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Public Methods

#region Retrieve Methods

/// <summary>
/// Retrieves the module from the database using the value of the
/// Module_ID property of the current instance
/// </summary>
public void Retrieve()
{
  DataRow objDataRow;

  try
  {
    _Module_ID.Validate("Module_ID");
  }
  catch (DalException excp)
  {
    throw excp;
  }

  try
  {
    objDataRow = RetrieveExecute(_Module_ID.Value, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
  this._Module_ID.Value   = (int)objDataRow["Module_ID"];
  this._Name.Value      = (string)objDataRow["Name"];
  this._StartTime.Value   = (DateTime)objDataRow["StartTime"];
  this._EndTime.Value     = (DateTime)objDataRow["EndTime"];
  this._Weekday_ID.Value  = (int)objDataRow["Weekday_ID"];
  this.Log.Created.Value  = (DateTime)objDataRow["Created"];
  this.Log.CreatedBy.Value = (Guid)objDataRow["CreatedBy"];
  this.Log.Updated.Value  = (DateTime)objDataRow["Updated"];
  this.Log.UpdatedBy.Value = (Guid)objDataRow["UpdatedBy"];
}

/// <summary>
/// Retrieves the module with the specified identifier from the database.
/// </summary>
/// <param name="module_ID">The unique identification of the module.</param>
public static Module Retrieve(int module_ID)
```

```
    {
      DataRow objDataRow;
      string connString = System.Con|guration.Con|gurationSettings.AppSettings["sqlConnectionString"];
      SqlConnection objConnection = new SqlConnection(connString);

      Module objModule = new Module();

      try
      {
        objDataRow = RetrieveExecute(module_ID, objConnection);
      }
      catch (DalException objExc)
      {
        throw objExc;
      }
      objModule._Module_ID.Value    = (int)objDataRow["Module_ID"];
      objModule._Name.Value         = (string)objDataRow["Name"];
      objModule._StartTime.Value    = (DateTime)objDataRow["StartTime"];
      objModule._EndTime.Value      = (DateTime)objDataRow["EndTime"];
      objModule._Weekday_ID.Value   = (int)objDataRow["Weekday_ID"];
      objModule._Log.Created.Value  = (DateTime)objDataRow["Created"];
      objModule._Log.CreatedBy.Value = (Guid)objDataRow["CreatedBy"];
      objModule._Log.Updated.Value  = (DateTime)objDataRow["Updated"];
      objModule._Log.UpdatedBy.Value = (Guid)objDataRow["UpdatedBy"];
      return objModule;
    }

    private static DataRow RetrieveExecute(
      int module_ID,
      SqlConnection sqlConnection)
    {
      SqlCommand objCommand = new SqlCommand("Module_Select", sqlConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      SqlParameter objParam;

      objParam = objCommand.Parameters.Add("@Module_ID", SqlDbType.Int);
      objParam.Value = module_ID;

      DataSet objDataSet = new DataSet("Module");
      SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

      try
      {
        sqlConnection.Open();
        objAdap.Fill(objDataSet, "Module");
        sqlConnection.Close();
        DataRow objDataRow = objDataSet.Tables["Module"].Rows[0];
        return objDataRow;
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }
    }

    #endregion //Retrieve Methods

    #endregion //Public Methods

  }
}
```

1.14 Period

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL
{
  /// <summary>
  /// Represents a period.
  /// </summary>
  public class Period : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalGuid _Period_ID = new DalGuid(false);
    private DalInt _PeriodType_ID = new DalInt(false);
    private DalStringLocalizable _Name = new DalStringLocalizable(false);
    private DalDateTime _StartDate = new DalDateTime(false);
    private DalDateTime _EndDate = new DalDateTime(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.Period">Period</see>
    /// class.
    /// </summary>
```

```
public Period() {
  _Log = new DataLog();
}

#endregion

#region Public Properties

/// <summary>
/// Gets the ID of the period.
/// </summary>
/// <value></value>
public DalGuid Period_ID
{
  get { return _Period_ID; }
}

/// <summary>
/// Gets the period type ID.
/// </summary>
public DalInt PeriodType_ID
{
  get { return _PeriodType_ID; }
}

/// <summary>
/// Gets the name of the period.
/// </summary>
public DalStringLocalizable Name
{
  get { return _Name; }
}

/// <summary>
/// Gets the start date of the period.
/// </summary>
public DalDateTime StartDate
{
  get { return _StartDate; }
}

/// <summary>
/// Gets the end date of the period.
/// </summary>
public DalDateTime EndDate
{
  get { return _EndDate; }
}

/// <summary>
/// Gets the data log of the course.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Public Methods

#region Create Methods

/// <summary>
/// Creates a period in the database using the values of the properties
/// of the current instance.
/// </summary>
public void Create()
{
  try
  {
    _PeriodType_ID.Validate("PeriodType_ID");
    _Period_ID.Validate("Period_ID");
    _Name.Validate("Name");
    _StartDate.Validate("StartDate");
    _EndDate.Validate("EndDate");
    _Log.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }

  SqlCommand objCommand = new SqlCommand("Period_Insert", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  SqlParameter objParam;

  objParam = objCommand.Parameters.Add("@Period_ID", SqlDbType.UniqueIdenti↓er);
  objParam.Value = _Period_ID.Value;

  objParam = objCommand.Parameters.Add("@PeriodType_ID", SqlDbType.Int);
  objParam.Value = _PeriodType_ID.Value;

  objParam = objCommand.Parameters.Add("@Name", SqlDbType.Text);
  objParam.Value = _Name.Xml;

  objParam = objCommand.Parameters.Add("@StartDate", SqlDbType.DateTime);
  objParam.Value = _StartDate.Value;
```

```
    objParam = objCommand.Parameters.Add("@EndDate", SqlDbType.DateTime);
    objParam.Value = _EndDate.Value;

    objCommand = _Log.AddParameters(objCommand, false);

    DataSet objDataSet = new DataSet("Period");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
      objConnection.Open();
      objAdap.Fill(objDataSet, "Period");
      objConnection.Close();
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
}

#endregion

#region Retrieve Methods

/// <summary>
/// Retrieves the period from the database using the value of the
/// Period_ID property of the current instance
/// </summary>
public void Retrieve()
{
    DataRow objDataRow;
    try
    {
      _Period_ID.Validate("Period_ID");
    }
    catch (DalException excp)
    {
      throw excp;
    }

    try
    {
      objDataRow = RetrieveExecute(_Period_ID.Value, objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }
    this._Period_ID.Value    = (Guid)objDataRow["Period_ID"];
    this._PeriodType_ID.Value = System.Convert.ToInt32(objDataRow["PeriodType_ID"]);
    this._Name.LoadXml((string)objDataRow["Name"]);
    this._StartDate.Value    = (DateTime)objDataRow["StartDate"];
    this._EndDate.Value      = (DateTime)objDataRow["EndDate"];
    this._Log.SetValues(objDataRow);
}

/// <summary>
/// Retrieves the period with the specified identifier from the database.
/// </summary>
/// <param name="period_ID">The globally unique identifier of the period.</param>
public static Period Retrieve(Guid period_ID)
{
    DataRow objDataRow;
    string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
    SqlConnection objConnection = new SqlConnection(connString);

    Period objPeriod = new Period();

    try
    {
      objDataRow = RetrieveExecute(period_ID, objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }
    objPeriod._Period_ID.Value    = (Guid)objDataRow["Period_ID"];
    objPeriod._PeriodType_ID.Value = (int)objDataRow["PeriodType_ID"];
    objPeriod._Name.LoadXml((string)objDataRow["Name"]);
    objPeriod._StartDate.Value    = (DateTime)objDataRow["StartDate"];
    objPeriod._EndDate.Value      = (DateTime)objDataRow["EndDate"];
    objPeriod._Log.SetValues(objDataRow);
    return objPeriod;
}

private static DataRow RetrieveExecute(
    Guid period_ID,
    SqlConnection sqlConnection)
{
    SqlCommand objCommand = new SqlCommand("Period_Select", sqlConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam;

    objParam = objCommand.Parameters.Add("@Period_ID", SqlDbType.UniqueIdenti↓er, 16);
    objParam.Value = period_ID;

    DataSet objDataSet = new DataSet("Period");
```

```
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
      sqlConnection.Open();
      objAdap.Fill(objDataSet, "Period");
      sqlConnection.Close();
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
    DataRow objDataRow = objDataSet.Tables["Period"].Rows[0];
    return objDataRow;
  }

  #endregion //Retrieve Methods

  #region Update Methods

  /// <summary>
  /// Updates the current period in the database using the original
  /// period to resolve possible concurrency issues.
  /// </summary>
  /// <param name="originalPeriod">The period <see cref="StudyPlanning.DAL.Period">Period</see> object.</param>
  /// <returns><strong>true</strong> if the update was succesfully executed; otherwise, <strong>false</strong>.</returns>
  public bool Update(Period originalPeriod)
  {

    try
    {
      _Period_ID.Validate("Period_ID");
      _PeriodType_ID.Validate("PeriodType_ID");
      _Name.Validate("Name");
      _StartDate.Validate("StartDate");
      _EndDate.Validate("EndDate");
      _Log.Validate();
    }
    catch (DalException excp)
    {
      throw excp;
    }

    bool blnResult = false;

    SqlCommand objCommand = new SqlCommand("Period_Update", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam;

    objParam = objCommand.Parameters.Add("@Period_ID", SqlDbType.UniqueIdenti↓er);
    objParam.Value = this._Period_ID.Value;

    objParam = objCommand.Parameters.Add("@PeriodType_ID", SqlDbType.Int);
    objParam.Value = this._PeriodType_ID.Value;

    objParam = objCommand.Parameters.Add("@Name", SqlDbType.Text);
    objParam.Value = this._Name.Xml;

    objParam = objCommand.Parameters.Add("@StartDate", SqlDbType.DateTime);
    objParam.Value = this._StartDate.Value;

    objParam = objCommand.Parameters.Add("@EndDate", SqlDbType.DateTime);
    objParam.Value = this._EndDate.Value;

    objCommand = this._Log.AddParameters(objCommand, false);

    objParam = objCommand.Parameters.Add("@Original_Period_ID", SqlDbType.UniqueIdenti↓er);
    objParam.Value = originalPeriod._Period_ID.Value;

    objParam = objCommand.Parameters.Add("@Original_PeriodType_ID", SqlDbType.Int);
    objParam.Value = originalPeriod._PeriodType_ID.Value;

    objParam = objCommand.Parameters.Add("@Original_Name", SqlDbType.Text);
    objParam.Value = originalPeriod._Name.Xml;

    objParam = objCommand.Parameters.Add("@Original_StartDate", SqlDbType.DateTime);
    objParam.Value = originalPeriod._StartDate.Value;

    objParam = objCommand.Parameters.Add("@Original_EndDate", SqlDbType.DateTime);
    objParam.Value = originalPeriod._EndDate.Value;

    objCommand = originalPeriod._Log.AddParameters(objCommand, true);

    int rowsA↑ected;

    try
    {
      objConnection.Open();
      rowsA↑ected = objCommand.ExecuteNonQuery();
      objConnection.Close();

      if (rowsA↑ected > 0)
        blnResult = true;
    }
    catch(SqlException excp)
    {
      throw new DalException(excp);
    }
```

```
    return blnResult;
}

#endregion //Update Methods

#region Delete Methods

/// <summary>
/// Deletes the current period from the database.
/// </summary>
/// <returns><strong>true</strong> if the deletion is succesfully executed; otherwise, <strong>false</strong>.</returns>
public bool Delete()
{
    bool blnResult = false;

    try
    {
        _Period_ID.Validate("Period_ID");
    }
    catch (DalException excp)
    {
        throw excp;
    }

    SqlCommand objCommand = new SqlCommand("Period_Delete", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam;

    objParam = objCommand.Parameters.Add("@Original_Period_ID", SqlDbType.UniqueIdenti↓er);
    objParam.Value = this._Period_ID.Value;

    objParam = objCommand.Parameters.Add("@Original_PeriodType_ID", SqlDbType.Int);
    objParam.Value = this._PeriodType_ID.Value;

    objParam = objCommand.Parameters.Add("@Original_Name", SqlDbType.Text);
    objParam.Value = this._Name.Xml;

    objParam = objCommand.Parameters.Add("@Original_StartDate", SqlDbType.DateTime);
    objParam.Value = this._StartDate.Value;

    objParam = objCommand.Parameters.Add("@Original_EndDate", SqlDbType.DateTime);
    objParam.Value = this._EndDate.Value;

    objCommand = this._Log.AddParameters(objCommand, true);

    int rowsA↑ected;
    try
    {
        objConnection.Open();
        rowsA↑ected = objCommand.ExecuteNonQuery();
        objConnection.Close();

        if (rowsA↑ected > 0)
            blnResult = true;
    }
    catch(SqlException e)
    {
        throw new DalException(e);
    }
    return blnResult;
}

#endregion //Delete Methods

#region Methods

/// <summary>
/// Retrieves the identification of the next period in relation to the supplied period.
/// </summary>
/// <param name="period">Identification of the current <see cref="StudyPlanning.DAL.Period"/>.</param>
/// <returns>The identification of the next period.</returns>
public static Guid GetNextPeriod(Guid period)
{
    string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
    SqlConnection objConnection = new SqlConnection(connString);

    SqlCommand objCommand = new SqlCommand("Period_GetNextPeriod", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam;

    objParam = objCommand.Parameters.Add("@Period_ID", SqlDbType.UniqueIdenti↓er, 16);
    objParam.Value = period;

    DataSet objDataSet = new DataSet("Result");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
        objConnection.Open();
        objAdap.Fill(objDataSet, "Result");
        objConnection.Close();
    }
    catch (SqlException objExc)
    {
        throw new DalException(objExc);
    }
    return (Guid)objDataSet.Tables["Result"].Rows[0]["Period_ID"];
```

```
    }

    #endregion //Methods

    #endregion //Public Methods

  }
}
```

1.15 PeriodTypes

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.PeriodTypes
{
  /// <summary>
  /// Represents a period type.
  /// </summary>
  public class PeriodType : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalInt _PeriodType_ID = new DalInt(false);
    private DalStringLocalizable _Name = new DalStringLocalizable(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.PeriodTypes.PeriodType"/>
    /// class.
    /// </summary>
    public PeriodType() {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the ID of the periodType.
    /// </summary>
    /// <value></value>
    public DalInt PeriodType_ID
    {
      get { return _PeriodType_ID; }
    }

    /// <summary>
    /// Gets the name of the periodType.
    /// </summary>
    public DalStringLocalizable Name
    {
      get { return _Name; }
    }

    /// <summary>
    /// Gets the data log of the course.
    /// </summary>
    public DataLog Log
    {
      get { return _Log; }
    }

    #endregion //Public Properties

    #region Public Methods

    #region Retrieve Methods

    /// <summary>
    /// Retrieves the period type from the database using the value of the
    /// PeriodType_ID property of the current instance
    /// </summary>
    public void Retrieve()
    {
      DataRow objDataRow;

      try
      {
        _PeriodType_ID.Validate("PeriodType_ID");
      }
      catch (DalException excp)
      {
        throw excp;
      }

      try
```

```
      {
        objDataRow = RetrieveExecute(_PeriodType_ID.Value, objConnection);
      }
      catch (DalException objExc)
      {
        throw objExc;
      }
      this._PeriodType_ID.Value = (int)objDataRow["PeriodType_ID"];
      this._Name.LoadXml((string)objDataRow["Name"]);
      this._Log.SetValues(objDataRow);
    }

    /// <summary>
    /// Retrieves the period type with the specified identifier from the database.
    /// </summary>
    /// <param name="periodType_ID">The unique identifier of the period type.</param>
    public static PeriodType Retrieve(int periodType_ID)
    {
      DataRow objDataRow;
      string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
      SqlConnection objConnection = new SqlConnection(connString);

      PeriodType objPeriodType = new PeriodType();

      try
      {
        objDataRow = RetrieveExecute(periodType_ID, objConnection);
      }
      catch (DalException objExc)
      {
        throw objExc;
      }
      objPeriodType._PeriodType_ID.Value = (int)objDataRow["PeriodType_ID"];
      objPeriodType._Name.LoadXml((string)objDataRow["Name"]);
      objPeriodType._Log.SetValues(objDataRow);
      return objPeriodType;
    }

    private static DataRow RetrieveExecute(
      int periodType_ID,
      SqlConnection sqlConnection)
    {
      SqlCommand objCommand = new SqlCommand("PeriodType_Select", sqlConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      SqlParameter objParam;

      objParam = objCommand.Parameters.Add("@PeriodType_ID", SqlDbType.Int);
      objParam.Value = periodType_ID;

      DataSet objDataSet = new DataSet("PeriodType");
      SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

      try
      {
        sqlConnection.Open();
        objAdap.Fill(objDataSet, "PeriodType");
        sqlConnection.Close();
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }
      DataRow objDataRow = objDataSet.Tables["PeriodType"].Rows[0];
      return objDataRow;

    }

    #endregion //Retrieve Methods

    #endregion //Public Methods

  }
}
```

1.15.1 Module

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.PeriodTypes
{
  /// <summary>
  /// Represents a period type module.
  /// </summary>
  public class Module : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalInt _PeriodType_Module_ID = new DalInt(false);
    private DalInt _PeriodType_ID = new DalInt(false);
    private DalInt _Module_ID = new DalInt(false);
    private DataLog _Log;
```

```
#endregion //Private Properties

#region Constructors

/// <summary>
/// Creates a new instance of the <see cref="StudyPlanning.DAL.PeriodTypes.Module"/>
/// class.
/// </summary>
public Module() {
  _Log = new DataLog();
}

#endregion

#region Public Properties

/// <summary>
/// Gets the ID of the periodType_Module.
/// </summary>
/// <value></value>
public DalInt PeriodType_Module_ID
{
  get { return _PeriodType_Module_ID; }
}

/// <summary>
/// Gets the periodType_ID.
/// </summary>
public DalInt PeriodType_ID
{
  get { return _PeriodType_ID; }
}

/// <summary>
/// Gets the module_ID.
/// </summary>
public DalInt Module_ID
{
  get { return _Module_ID; }
}

/// <summary>
/// Gets the data log of the course.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Public Methods

#region Retrieve Methods

/// <summary>
/// Retrieves the period type module from the database using the value of the
/// PeriodType_Module_ID property of the current instance
/// </summary>
public void Retrieve()
{
  DataRow objDataRow;

  try
  {
    _PeriodType_Module_ID.Validate("PeriodType_Module_ID");
  }
  catch (DalException excp)
  {
    throw excp;
  }

  try
  {
    objDataRow = RetrieveExecute(_PeriodType_Module_ID.Value, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
  this._PeriodType_Module_ID.Value = (int)objDataRow["PeriodType_Module_ID"];
  this._PeriodType_ID.Value     = (int)objDataRow["PeriodType_ID"];
  this._Module_ID.Value       = (int)objDataRow["Module_ID"];
  this._Log.SetValues(objDataRow);
}

/// <summary>
/// Retrieves the period type module with the specified identifier from the database.
/// </summary>
/// <param name="periodType_Module_ID">The identifier of the period type module.</param>
public static Module Retrieve(int periodType_Module_ID)
{
  DataRow objDataRow;
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  Module objPeriodType_Module = new Module();

  try
```

```
      {
        objDataRow = RetrieveExecute(periodType_Module_ID, objConnection);
      }
      catch (DalException objExc)
      {
        throw objExc;
      }
      objPeriodType_Module._PeriodType_Module_ID.Value = (int)objDataRow["PeriodType_Module_ID"];
      objPeriodType_Module._PeriodType_ID.Value     = (int)objDataRow["PeriodType_ID"];
      objPeriodType_Module._Module_ID.Value       = (int)objDataRow["Module_ID"];
      objPeriodType_Module._Log.SetValues(objDataRow);
      return objPeriodType_Module;
    }

    private static DataRow RetrieveExecute(
      int periodType_Module_ID,
      SqlConnection sqlConnection)
    {
      SqlCommand objCommand = new SqlCommand("PeriodType_Module_Select", sqlConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      SqlParameter objParam;

      objParam = objCommand.Parameters.Add("@PeriodType_Module_ID", SqlDbType.Int);
      objParam.Value = periodType_Module_ID;

      DataSet objDataSet = new DataSet("PeriodType_Module");
      SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

      try
      {
        sqlConnection.Open();
        objAdap.Fill(objDataSet, "PeriodType_Module");
        sqlConnection.Close();
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }
      DataRow objDataRow = objDataSet.Tables["PeriodType_Module"].Rows[0];
      return objDataRow;
    }

    /// <summary>
    /// Retrieves the modules for a given period type.
    /// </summary>
    /// <param name="periodType_ID">Identification of the
    /// <see cref="StudyPlanning.DAL.PeriodTypes.PeriodType"/>.</param>
    /// <returns>An array of <see cref="StudyPlanning.DAL.Module"/> identifications.</returns>
    public static int[] GetModules(int periodType_ID)
    {
      string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
      SqlConnection objConnection = new SqlConnection(connString);

      SqlCommand objCommand = new SqlCommand("PeriodType_Module_GetModules", objConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      SqlParameter objParam = objCommand.Parameters.Add("@PeriodType_ID", SqlDbType.Int);
      objParam.Value = periodType_ID;

      DataSet objDataSet = new DataSet("Result");
      SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

      try
      {
        objConnection.Open();
        objAdap.Fill(objDataSet, "Result");
        objConnection.Close();
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }

      int intRows = objDataSet.Tables["Result"].Rows.Count;
      int[] modules = new int[intRows];

      for (int i=0; i < intRows; i++)
      {
        modules[i] = (int)objDataSet.Tables["Result"].Rows[i]["Module_ID"];
      }
      return modules;
    }

    #endregion //Retrieve Methods

    #endregion //Public Methods

  }
}
```

1.16 Point

```
using System;
```

```
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL
{
  /// <summary>
  /// Represents a point.
  /// </summary>
  public class Point : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalGuid _Point_ID = new DalGuid(false);
    private DalFloat _PointMin = new DalFloat(true);
    private DalFloat _PointMax = new DalFloat(true);
    private DalBool _AmsGiving = new DalBool(true);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.Point">Point</see>
    /// class.
    /// </summary>
    public Point() {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the ID of the point.
    /// </summary>
    /// <value></value>
    public DalGuid Point_ID
    {
      get { return _Point_ID; }
    }

    /// <summary>
    /// Gets the minimum value of the point.
    /// </summary>
    public DalFloat PointMin
    {
      get { return _PointMin; }
    }

    /// <summary>
    /// Gets the maximum value of the point.
    /// </summary>
    public DalFloat PointMax
    {
      get { return _PointMax; }
    }

    /// <summary>
    /// Gets the property whether the point is AMS giving or not.
    /// </summary>
    public DalBool AmsGiving
    {
      get { return _AmsGiving; }
    }

    /// <summary>
    /// Gets the data log of the course.
    /// </summary>
    public DataLog Log
    {
      get { return _Log; }
    }

    #endregion //Public Properties

    #region Public Methods

    #region Create Methods

    /// <summary>
    /// Creates a point in the database using the values of the properties
    /// of the current instance.
    /// </summary>
    public void Create()
    {
      try
      {
        _Point_ID.Validate("Point_ID");
        _PointMax.Validate("PointMax");
        _PointMin.Validate("PointMin");
        _AmsGiving.Validate("AmsGiving");
        _Log.Validate();
      }
      catch (DalException excp)
      {
        throw excp;
```

```
    }

    SqlCommand objCommand = new SqlCommand("Point_Insert", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam;

    objParam = objCommand.Parameters.Add("@Point_ID", SqlDbType.UniqueIdenti↓er);
    objParam.Value = _Point_ID.Value;

    objParam = objCommand.Parameters.Add("@PointMin", SqlDbType.Float);
    if (_PointMin.IsNull)
      objParam.Value = DBNull.Value;
    else
      objParam.Value = _PointMin.Value;

    objParam = objCommand.Parameters.Add("@PointMax", SqlDbType.Float);
    if (_PointMax.IsNull)
      objParam.Value = DBNull.Value;
    else
      objParam.Value = _PointMax.Value;

    objParam = objCommand.Parameters.Add("@AmsGiving", SqlDbType.Bit);
    if (_AmsGiving.IsNull)
      objParam.Value = DBNull.Value;
    else
      objParam.Value = _AmsGiving.Value;

    objCommand = _Log.AddParameters(objCommand, false);

    DataSet objDataSet = new DataSet("Point");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
      objConnection.Open();
      objAdap.Fill(objDataSet, "Point");
      objConnection.Close();
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
  }

  #endregion

  #region Retrieve Methods

  /// <summary>
  /// Retrieves the point from the database using the value of the
  /// Point_ID property of the current instance
  /// </summary>
  public void Retrieve()
  {
    DataRow objDataRow;

    try
    {
      _Point_ID.Validate("Point_ID");
    }
    catch (DalException excp)
    {
      throw excp;
    }

    try
    {
      objDataRow = RetrieveExecute(_Point_ID.Value, objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }
    this._Point_ID.Value = (Guid)objDataRow["Point_ID"];

    if (objDataRow["PointMin"].Equals(DBNull.Value))
      this._PointMin.IsNull = true;
    else
      this._PointMin.Value = (float)objDataRow["PointMin"];

    if (objDataRow["PointMax"].Equals(DBNull.Value))
      this._PointMax.IsNull = true;
    else
      this._PointMax.Value = (float)objDataRow["PointMax"];

    if (objDataRow["AmsGiving"].Equals(DBNull.Value))
      this.AmsGiving.IsNull = true;
    else
      this._AmsGiving.Value = (bool)objDataRow["AmsGiving"];

    this._Log.SetValues(objDataRow);
  }

  /// <summary>
  /// Retrieves the point with the specified identifier from the database.
  /// </summary>
  /// <param name="point_ID">The globally unique identifier of the point.</param>
  public static Point Retrieve(Guid point_ID)
```

```
{
  DataRow objDataRow;
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  Point objPoint = new Point();

  try
  {
    objDataRow = RetrieveExecute(point_ID, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
  objPoint.Point_ID.Value = (Guid)objDataRow["Point_ID"];

  if (objDataRow["PointMin"].Equals(DBNull.Value))
    objPoint.PointMin.IsNull = true;
  else
    objPoint.PointMin.Value = (float)objDataRow["PointMin"];

  if (objDataRow["PointMax"].Equals(DBNull.Value))
    objPoint.PointMax.IsNull = true;
  else
    objPoint.PointMax.Value = (float)objDataRow["PointMax"];

  if (objDataRow["AmsGiving"].Equals(DBNull.Value))
    objPoint.AmsGiving.IsNull = true;
  else
    objPoint.AmsGiving.Value = (bool)objDataRow["AmsGiving"];

  objPoint.Log.SetValues(objDataRow);
  return objPoint;
}

private static DataRow RetrieveExecute(
  Guid point_ID,
  SqlConnection sqlConnection)
{
  SqlCommand objCommand = new SqlCommand("Point_Select", sqlConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  SqlParameter objParam;

  objParam = objCommand.Parameters.Add("@Point_ID", SqlDbType.UniqueIdenti↓er);
  objParam.Value = point_ID;

  DataSet objDataSet = new DataSet("Point");
  SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

  try
  {
    sqlConnection.Open();
    objAdap.Fill(objDataSet, "Point");
    sqlConnection.Close();
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
  DataRow objDataRow = objDataSet.Tables["Point"].Rows[0];
  return objDataRow;
}

#endregion //Retrieve Methods

#region Update Methods

/// <summary>
/// Updates the current point in the database using the original
/// point to resolve possible concurrency issues.
/// </summary>
/// <param name="originalPoint">The point <see cref="StudyPlanning.DAL.Point">Point</see> object.</param>
/// <returns><strong>true</strong> if the update was succesfully executed; otherwise, <strong>false</strong>.</returns>
public bool Update(Point originalPoint)
{
  try
  {
    _Point_ID.Validate("Point_ID");
    _PointMax.Validate("PointMax");
    _PointMin.Validate("PointMin");
    _AmsGiving.Validate("AmsGiving");
    _Log.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }

  bool blnResult = false;

  SqlCommand objCommand = new SqlCommand("Point_Update", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  SqlParameter objParam;

  objParam = objCommand.Parameters.Add("@Point_ID", SqlDbType.UniqueIdenti↓er);
  objParam.Value = this._Point_ID.Value;
```

```
    objParam = objCommand.Parameters.Add("@PointMin", SqlDbType.Float);
    if (_PointMin.IsNull)
      objParam.Value = DBNull.Value;
    else
      objParam.Value = this.PointMin.Value;

    objParam = objCommand.Parameters.Add("@PointMax", SqlDbType.Float);
    if (_PointMax.IsNull)
      objParam.Value = DBNull.Value;
    else
      objParam.Value = this.PointMax.Value;

    objParam = objCommand.Parameters.Add("@AmsGiving", SqlDbType.Bit);
    if (_AmsGiving.IsNull)
      objParam.Value = DBNull.Value;
    else
      objParam.Value = this.AmsGiving.Value;

    objCommand = this.Log.AddParameters(objCommand, false);

    objParam = objCommand.Parameters.Add("@Original_Point_ID", SqlDbType.UniqueIdenti↓er);
    objParam.Value = originalPoint.Point_ID.Value;

    objParam = objCommand.Parameters.Add("@Original_PointMin", SqlDbType.Float);
    if (_PointMin.IsNull)
      objParam.Value = DBNull.Value;
    else
      objParam.Value = originalPoint.PointMin.Value;

    objParam = objCommand.Parameters.Add("@Original_PointMax", SqlDbType.Float);
    if (_PointMax.IsNull)
      objParam.Value = DBNull.Value;
    else
      objParam.Value = originalPoint.PointMax.Value;

    objParam = objCommand.Parameters.Add("@Original_AmsGiving", SqlDbType.Bit);
    if (_AmsGiving.IsNull)
      objParam.Value = DBNull.Value;
    else
      objParam.Value = originalPoint.AmsGiving.Value;

    objCommand = originalPoint.Log.AddParameters(objCommand, true);

    int rowsA↑ected;

    try
    {
      objConnection.Open();
      rowsA↑ected = objCommand.ExecuteNonQuery();
      objConnection.Close();

      if (rowsA↑ected > 0)
        blnResult = true;
    }
    catch(SqlException excp)
    {
      throw new DalException(excp);
    }
    return blnResult;
  }

  #endregion //Update Methods

  #region Delete Methods

  /// <summary>
  /// Deletes the current point from the database.
  /// </summary>
  /// <returns><strong>true</strong> if the deletion is succesfully executed; otherwise, <strong>false</strong>.</returns>
  public bool Delete()
  {
    bool blnResult = false;

    try
    {
      _Point_ID.Validate("Point_ID");
    }
    catch (DalException excp)
    {
      throw excp;
    }

    SqlCommand objCommand = new SqlCommand("Point_Delete", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam;

    objParam = objCommand.Parameters.Add("@Original_Point_ID", SqlDbType.UniqueIdenti↓er);
    objParam.Value = this._Point_ID.Value;

    objParam = objCommand.Parameters.Add("@Original_PointMin", SqlDbType.Float);
    if (_PointMin.IsNull)
      objParam.Value = DBNull.Value;
    else
      objParam.Value = this.PointMin.Value;

    objParam = objCommand.Parameters.Add("@Original_PointMax", SqlDbType.Float);
    if (_PointMax.IsNull)
```

```
        objParam.Value = DBNull.Value;
      else
        objParam.Value = this.PointMax.Value;

      objParam = objCommand.Parameters.Add("@Original_AmsGiving", SqlDbType.Bit);
      if (_AmsGiving.IsNull)
        objParam.Value = DBNull.Value;
      else
        objParam.Value = this.AmsGiving.Value;

      objCommand = this.Log.AddParameters(objCommand, true);

      int rowsA↑ected;
      try
      {
        objConnection.Open();
        rowsA↑ected = objCommand.ExecuteNonQuery();
        objConnection.Close();

        if (rowsA↑ected > 0)
          blnResult = true;
      }
      catch(SqlException e)
      {
        throw new DalException(e);
      }
      return blnResult;
    }

    #endregion //Delete Methods

    #endregion //Public Methods

  }
}
```

1.17 Project

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL
{
  /// <summary>
  /// Represents a studyPlanCriterion.
  /// </summary>
  public class Project : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalGuid _Project_ID = new DalGuid(false);
    private DalString _Number = new DalString(false);
    private DalStringLocalizable _Name = new DalStringLocalizable(false);
    private DalInt _ProjectType_ID = new DalInt(false);
    private DalInt _AssessmentType_ID = new DalInt(false);
    private DalGuid _Period_ID = new DalGuid(true);
    private DalGuid _Point_ID = new DalGuid(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.Project"/>
    /// class.
    /// </summary>
    public Project() {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the Project_ID.
    /// </summary>
    /// <value></value>
    public DalGuid Project_ID
    {
      get { return _Project_ID; }
    }

    /// <summary>
    /// Gets the number of the project.
    /// </summary>
    public DalString Number
    {
      get { return _Number; }
    }
```

```
/// <summary>
/// Gets the name of the project.
/// </summary>
public DalStringLocalizable Name
{
  get { return _Name; }
}

/// <summary>
/// Gets the ProjectType_ID of the project.
/// </summary>
public DalInt ProjectType_ID
{
  get { return _ProjectType_ID; }
}

/// <summary>
/// Gets the AssessmentType_ID of the project.
/// </summary>
public DalInt AssessmentType_ID
{
  get { return _AssessmentType_ID; }
}

/// <summary>
/// Gets the Period_ID of the project.
/// </summary>
public DalGuid Period_ID
{
  get { return _Period_ID; }
}

/// <summary>
/// Gets the Point_ID of the project.
/// </summary>
public DalGuid Point_ID
{
  get { return _Point_ID; }
}

/// <summary>
/// Gets the data log of the course.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Methods

/// <summary>
/// Creates a new project in the database using the values of the properties
/// of the current instance.
/// </summary>
public void Create()
{
  try
  {
    this.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }

  SqlCommand objCommand = new SqlCommand("Project_Insert", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, false);

  try
  {
    objConnection.Open();
    objCommand.ExecuteNonQuery();
    objConnection.Close();
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
}

#region Retrieve Methods

/// <summary>
/// Retrieves the project from the database using the value of the
/// Project_ID property of the current instance.
/// </summary>
public void Retrieve()
{
  try
  {
    this._Project_ID.Validate("Project_ID");
  }
  catch (DalException excp)
  {
    throw excp;
```

```
    }

    try
    {
      RetrieveExecute(_Project_ID.Value, this, objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }
  }

  /// <summary>
  /// Retrieves the project with the specified identifier from the database.
  /// </summary>
  /// <param name="project_ID">The globally unique identifier of the project.</param>
  public static Project Retrieve(Guid project_ID)
  {
    string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
    SqlConnection objConnection = new SqlConnection(connString);

    Project objPrj = new Project();

    try
    {
      RetrieveExecute(project_ID, objPrj, objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }

    return objPrj;
  }

  private static void RetrieveExecute(
    Guid project_ID,
    Project objProject,
    SqlConnection sqlConnection)
  {
    SqlCommand objCommand = new SqlCommand("Project_Select", sqlConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam;

    objParam = objCommand.Parameters.Add("@Project_ID", SqlDbType.UniqueIdenti↓er, 16);
    objParam.Value = project_ID;

    DataSet objDataSet = new DataSet("Project");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
      sqlConnection.Open();
      objAdap.Fill(objDataSet, "Project");
      sqlConnection.Close();
      DataRow objDataRow = objDataSet.Tables["Project"].Rows[0];

      objProject.Project_ID.Value = (Guid)objDataRow["Project_ID"];
      objProject.Number.Value = (string)objDataRow["Number"];
      objProject.Name.LoadXml((string)objDataRow["Name"]);
      objProject.ProjectType_ID.Value = (int)objDataRow["ProjectType_ID"];
      objProject.AssessmentType_ID.Value = (int)objDataRow["AssessmentType_ID"];

      if (objDataRow["Period_ID"].Equals(DBNull.Value))
        objProject.Period_ID.IsNull = true;
      else
        objProject.Period_ID.Value = (Guid)objDataRow["Period_ID"];

      objProject.Point_ID.Value = (Guid)objDataRow["Point_ID"];
      objProject.Log.SetValues(objDataRow);
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
  }

  #endregion //Retrive Methods

  /// <summary>
  /// Updates the current project in the database using the original
  /// project to resolve possible concurrency issues.
  /// </summary>
  /// <param name="originalProject">The original <see cref="StudyPlanning.DAL.Project"/> object.</param>
  /// <returns><strong>true</strong> if the update was succesfully executed; otherwise, <strong>false</strong>.</returns>
  public bool Update(Project originalProject)
  {
    try
    {
      this.Validate();
    }
    catch (DalException excp)
    {
      throw excp;
    }

    bool blnResult = false;
```

```
    SqlCommand objCommand = new SqlCommand("Project_Update", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, false);
    AddParameters(originalProject, objCommand, true);

    int rowsA↑ected;

    try
    {
      objConnection.Open();
      rowsA↑ected = objCommand.ExecuteNonQuery();
      objConnection.Close();

      if (rowsA↑ected > 0)
        blnResult = true;
    }
    catch(SqlException excp)
    {
      throw new DalException(excp);
    }

    return blnResult;
  }

  /// <summary>
  /// Deletes the current project from the database.
  /// </summary>
  /// <returns><strong>true</strong> if the deletion is succesfully executed; otherwise, <strong>false</strong>.</returns>
  public bool Delete()
  {
    bool blnResult = false;

    try
    {
      this.Validate();
    }
    catch (DalException excp)
    {
      throw excp;
    }

    SqlCommand objCommand = new SqlCommand("Project_Delete", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, true);

    try
    {
      objConnection.Open();
      int rowsA↑ected = objCommand.ExecuteNonQuery();
      objConnection.Close();

      if (rowsA↑ected > 0)
        blnResult = true;
    }
    catch(SqlException objEx)
    {
      throw new DalException(objEx);
    }

    return blnResult;
  }

  /// <summary>
  /// Adds the relevant parameters to the specified <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see> object
  /// setting the values to ones of the specified <see cref="StudyPlanning.DAL.Project"/> object.
  /// </summary>
  /// <param name="project">
  /// The <see cref="StudyPlanning.DAL.Project"/> object containing the values that the parameters have to be
  /// initialized to.
  /// </param>
  /// <param name="objCommand">
  /// The <see cref="System.Data.SqlClient.SqlCommand"/> object to which parameters should be added.
  /// </param>
  /// <param name="isOriginal">
  /// <strong>true</strong> if the parameters to be added represent original
  /// data (for handling concurrent programming issues); otherwise, <strong>false</strong>.</param>
  /// <returns>The <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see>
  /// object with the relevant parameters added.</returns>
  private static void AddParameters(Project project, SqlCommand objCommand, bool isOriginal)
  {
    string paramName = "";
    SqlParameter objParam;

    //Project_ID
    paramName = "Project_ID";
    objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

    if (isOriginal)
      objParam.ParameterName = "@Original_" + paramName;

    objParam.Value = project.Project_ID.Value;

    //Number
    paramName = "Number";
    objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.VarChar, 20);

    if (isOriginal)
```

```
        objParam.ParameterName = "@Original_" + paramName;

      objParam.Value = project.Number.Value;

      //Name
      paramName = "Name";
      objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.Text);

      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      objParam.Value = project.Name.Value;

      //ProjectType_ID
      paramName = "ProjectType_ID";
      objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.Int, 4);

      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      objParam.Value = project.ProjectType_ID.Value;

      //AssessmentType_ID
      paramName = "AssessmentType_ID";
      objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.Int, 4);

      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      objParam.Value = project.AssessmentType_ID.Value;

      //Period_ID
      paramName = "Period_ID";
      objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      if (project.Period_ID.IsNull)
        objParam.Value = DBNull.Value;
      else
        objParam.Value = project.Period_ID.Value;

      //Point_ID
      paramName = "Point_ID";
      objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      objParam.Value = project.Point_ID.Value;

      //Log
      objCommand = project.Log.AddParameters(objCommand, isOriginal);
    }

    /// <summary>
    /// Validates the current <see cref="StudyPlanning.DAL.Project"/> object. If the
    /// value of either of the private properties is null a
    /// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
    /// </summary>
    private void Validate()
    {
      try
      {
        _Project_ID.Validate("Project_ID");
        _Number.Validate("Number");
        _Name.Validate("Name");
        _ProjectType_ID.Validate("ProjectType_ID");
        _AssessmentType_ID.Validate("AssessmentType_ID");
        _Period_ID.Validate("Period_ID");
        _Point_ID.Validate("Point_ID");
        _Log.Validate();
      }
      catch (DalException excp)
      {
        throw excp;
      }
    }

    #endregion //Methods

  }
}
```

1.18 ProjectType

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL
{
```

```
/// <summary>
/// Represents a project type − every project is of some type.
/// </summary>
public class ProjectType : StudyPlanning.DAL.DbObject
{
  #region Private Properties

  private DalInt _ProjectType_ID = new DalInt(false);
  private DalStringLocalizable _Name = new DalStringLocalizable(false);
  private DataLog _Log;

  #endregion //Private Properties

  #region Constructors

  /// <summary>
  /// Creates a new instance of the <see cref="StudyPlanning.DAL.ProjectType" />
  /// class.
  /// </summary>
  public ProjectType()
  {
    _Log = new DataLog();
  }

  #endregion

  #region Public Properties

  /// <summary>
  /// Gets the ID of the project type.
  /// </summary>
  public DalInt ProjectType_ID
  {
    get { return _ProjectType_ID; }
  }

  /// <summary>
  /// Gets the name of the project.
  /// </summary>
  public DalStringLocalizable Name
  {
    get { return _Name; }
  }

  /// <summary>
  /// Gets the data log of the project.
  /// </summary>
  public DataLog Log
  {
    get { return _Log; }
  }

  #endregion //Public Properties

  #region Methods

  #region Retrieve Methods

  /// <summary>
  /// Retrieves the project type from the database using the value of the
  /// ProjectType_ID property of the current instance.
  /// </summary>
  public void Retrieve()
  {
    try
    {
      this._ProjectType_ID.Validate("ProjectType_ID");
    }
    catch (DalException excp)
    {
      throw excp;
    }

    try
    {
      RetrieveExecute(_ProjectType_ID.Value, this, objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }
  }

  /// <summary>
  /// Retrieves the project type with the specified identifier from the database.
  /// </summary>
  /// <param name="projectType_ID">The globally unique identifier of the project.</param>
  public static ProjectType Retrieve(int projectType_ID)
  {
    string connString = System.Con|guration.Con|gurationSettings.AppSettings["sqlConnectionString"];
    SqlConnection objConnection = new SqlConnection(connString);

    ProjectType objProjectType = new ProjectType();

    try
    {
      RetrieveExecute(projectType_ID, objProjectType, objConnection);
    }
    catch (DalException objExc)
```

```
      {
        throw objExc;
      }

      return objProjectType;
    }

    private static void RetrieveExecute(
      int projectType_ID,
      ProjectType objProjectType,
      SqlConnection sqlConnection)
    {
      SqlCommand objCommand = new SqlCommand("ProjectType_Select", sqlConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      SqlParameter objParam;

      objParam = objCommand.Parameters.Add("@ProjectType_ID", SqlDbType.Int, 4);
      objParam.Value = projectType_ID;

      DataSet objDataSet = new DataSet("ProjectType");
      SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

      try
      {
        sqlConnection.Open();
        objAdap.Fill(objDataSet, "ProjectType");
        sqlConnection.Close();
        DataRow objDataRow = objDataSet.Tables["ProjectType"].Rows[0];

        objProjectType.ProjectType_ID.Value = (int)objDataRow["ProjectType_ID"];
        objProjectType.Name.LoadXml(objDataRow["Name"].ToString());
        objProjectType.Log.SetValues(objDataRow);
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }
    }

    #endregion //Retrive Methods

    #endregion //Methods
  }
}
```

1.19 RecommendedPlacementConcepts

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.RecommendedPlacementConcepts
{
  /// <summary>
  /// Represents a recommended placement concept.
  /// </summary>
  public class RecommendedPlacementConcept : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalInt _RecommendedPlacementConcept_ID = new DalInt(false);
    private DalStringLocalizable _Name = new DalStringLocalizable(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Initializes a new instance of the <see cref="StudyPlanning.DAL.RecommendedPlacementConcepts.RecommendedPlacementConcept"/>
    /// class.
    /// </summary>
    public RecommendedPlacementConcept()
    {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the ID of recommended placement concept.
    /// </summary>
    public DalInt RecommendedPlacementConcept_ID
    {
      get { return _RecommendedPlacementConcept_ID; }
    }

    /// <summary>
    /// Gets the name of the recommended placement concept.
    /// </summary>
```

```
public DalStringLocalizable Name
{
  get { return _Name; }
}

/// <summary>
/// Gets the data log of the recommended placement concept.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Methods

#region Retrieve Methods

/// <summary>
/// Retrieves the recommended placement concept from the database using the value of the
/// RecommendedPlacementConcept_ID property of the current instance.
/// </summary>
public void Retrieve()
{
  try
  {
    this._RecommendedPlacementConcept_ID.Validate("RecommendedPlacementConcept_ID");
  }
  catch (DalException excp)
  {
    throw excp;
  }

  try
  {
      RetrieveExecute(_RecommendedPlacementConcept_ID.Value, this, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
}

/// <summary>
/// Retrieves the recommended placement concept with the specified identifier from the database.
/// </summary>
/// <param name="recommendedPlacementConcept_ID">The globally unique identifier of the recommended placement concept.</param>
public static RecommendedPlacementConcept Retrieve(int recommendedPlacementConcept_ID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  RecommendedPlacementConcept objRpc = new RecommendedPlacementConcept();

  try
  {
    RetrieveExecute(recommendedPlacementConcept_ID, objRpc, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }

  return objRpc;
}

private static void RetrieveExecute(
  int recommendedPlacementConcept_ID,
  RecommendedPlacementConcept objRecommendedPlacementConcept,
  SqlConnection sqlConnection)
{
  SqlCommand objCommand = new SqlCommand("RecommendedPlacementConcept_Select", sqlConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  SqlParameter objParam;

  objParam = objCommand.Parameters.Add("@RecommendedPlacementConcept_ID", SqlDbType.Int, 4);
  objParam.Value = recommendedPlacementConcept_ID;

  DataSet objDataSet = new DataSet("RecommendedPlacementConcept");
  SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

  try
  {
    sqlConnection.Open();
    objAdap.Fill(objDataSet, "RecommendedPlacementConcept");
    sqlConnection.Close();
    DataRow objDataRow = objDataSet.Tables["RecommendedPlacementConcept"].Rows[0];

    objRecommendedPlacementConcept.RecommendedPlacementConcept_ID.Value = (int)objDataRow["
      RecommendedPlacementConcept_ID"];
    objRecommendedPlacementConcept.Name.LoadXml(objDataRow["Name"].ToString());
    objRecommendedPlacementConcept.Log.SetValues(objDataRow);
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
```

```
    }

    #endregion //Retrive Methods

    #endregion //Methods
  }
}
```

1.19.1 StudyType

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.RecommendedPlacementConcepts
{
  /// <summary>
  /// Represents a study type of a recommended placement concept.
  /// </summary>
  public class StudyType : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalInt _RecommendedPlacementConcept_StudyType_ID = new DalInt(false);
    private DalInt _RecommendedPlacementConcept_ID = new DalInt(false);
    private DalInt _StudyType_ID = new DalInt(false);
    private DalGuid _Point_ID = new DalGuid(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Initializes a new instance of the <see cref="StudyPlanning.DAL.RecommendedPlacementConcepts.StudyType"/>
    /// class.
    /// </summary>
    public StudyType()
    {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the ID of the recommended placement concept study type.
    /// </summary>
    public DalInt RecommendedPlacementConcept_StudyType_ID
    {
      get { return _RecommendedPlacementConcept_StudyType_ID; }
    }

    /// <summary>
    /// Gets the ID of the recommended placement concept.
    /// </summary>
    public DalInt RecommendedPlacementConcept_ID
    {
      get { return _RecommendedPlacementConcept_ID; }
    }

    /// <summary>
    /// Gets the ID of the study type.
    /// </summary>
    public DalInt StudyType_ID
    {
      get { return _StudyType_ID; }
    }

    /// <summary>
    /// Gets the ID of the point.
    /// </summary>
    public DalGuid Point_ID
    {
      get { return _Point_ID; }
    }

    /// <summary>
    /// Gets the data log of the recommended placement concept study type.
    /// </summary>
    public DataLog Log
    {
      get { return _Log; }
    }

    #endregion //Public Properties

    #region Methods

    #region Retrieve Methods

    /// <summary>
    /// Retrieves the recommended placement concept study type from the database using the value of the
```

```
/// RecommendedPlacementConcept_StudyType_ID property of the current instance.
/// </summary>
public void Retrieve()
{
  try
  {
    this._RecommendedPlacementConcept_StudyType_ID.Validate("RecommendedPlacementConcept_StudyType_ID");
  }
  catch (DalException excp)
  {
    throw excp;
  }

  try
  {
      RetrieveExecute(_RecommendedPlacementConcept_StudyType_ID.Value, this, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
}

/// <summary>
/// Retrieves the recommended placement concept study type with the specified identifier from the database.
/// </summary>
/// <param name="recommendedPlacementConcept_StudyType_ID">The globally unique identifier of the recommended placement concept study type
  .</param>
public static StudyType Retrieve(int recommendedPlacementConcept_StudyType_ID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  StudyType objRpcSt = new StudyType();

  try
  {
    RetrieveExecute(recommendedPlacementConcept_StudyType_ID, objRpcSt, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }

  return objRpcSt;
}

private static void RetrieveExecute(
  int recommendedPlacementConcept_StudyType_ID,
  StudyType objRpcSt,
  SqlConnection sqlConnection)
{
  SqlCommand objCommand = new SqlCommand("RecommendedPlacementConcept_StudyType_Select", sqlConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  SqlParameter objParam;

  objParam = objCommand.Parameters.Add("@RecommendedPlacementConcept_StudyType_ID", SqlDbType.Int, 4);
  objParam.Value = recommendedPlacementConcept_StudyType_ID;

  DataSet objDataSet = new DataSet("RecommendedPlacementConcept_StudyType");
  SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

  try
  {
    sqlConnection.Open();
    objAdap.Fill(objDataSet, "RecommendedPlacementConcept_StudyType");
    sqlConnection.Close();
    DataRow objDataRow = objDataSet.Tables["RecommendedPlacementConcept_StudyType"].Rows[0];

    objRpcSt.RecommendedPlacementConcept_StudyType_ID.Value = (int)objDataRow["RecommendedPlacementConcept_StudyType_ID"];
    objRpcSt.RecommendedPlacementConcept_ID.Value = (int)objDataRow["RecommendedPlacementConcept_ID"];
    objRpcSt.StudyType_ID.Value = (int)objDataRow["StudyType_ID"];
    objRpcSt.Point_ID.Value = (Guid)objDataRow["Point_ID"];
    objRpcSt.Log.SetValues(objDataRow);
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
}

#endregion //Retrive Methods

/// <summary>
/// Retrieves the Course_RecommendedPlacementConcept_StudyType_IDs
/// for a given RecommedPlacementConcept_ID and StudyType_ID.
/// </summary>
/// <param name="recommendedPlacementConcept_ID">Identification of the
/// RecommendedPlacementConcept.</param>
/// <param name="studyType_ID">Identification of the StudyType.</param>
/// <returns>A RecommendedPlacementConcept_StudyType_ID.</returns>
public static int GetID(int recommendedPlacementConcept_ID, int studyType_ID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  SqlCommand objCommand = new SqlCommand("RecommendedPlacementConcept_StudyType_GetID", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;
```

```
      SqlParameter objParam;

      objParam = objCommand.Parameters.Add("@RecommendedPlacementConcept_ID", SqlDbType.Int);
      objParam.Value = recommendedPlacementConcept_ID;

      objParam = objCommand.Parameters.Add("@StudyType_ID", SqlDbType.Int);
      objParam.Value = studyType_ID;

      DataSet objDataSet = new DataSet("Result");
      SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

      try
      {
        objConnection.Open();
        objAdap.Fill(objDataSet, "Result");
        objConnection.Close();
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }
      return (int)objDataSet.Tables["Result"].Rows[0]["RecommendedPlacementConcept_StudyType_ID"];
    }

    #endregion //Methods
  }
}
```

1.20 SecurityRoles

1.20.1 SecurityPermission

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.SecurityRoles
{
  /// <summary>
  /// Represents a security permission of a security role.
  /// </summary>
  public class SecurityPermission : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalInt _SecurityRole_SecurityPermission_ID = new DalInt(false);
    private DalInt _SecurityRole_ID = new DalInt(false);
    private DalInt _SecurityPermission_ID = new DalInt(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.SecurityRoles.SecurityPermission"/>
    /// class.
    /// </summary>
    public SecurityPermission() {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the SecurityRole_SecurityPermission_ID.
    /// </summary>
    /// <value></value>
    public DalInt SecurityRole_SecurityPermission_ID
    {
      get { return _SecurityRole_SecurityPermission_ID; }
    }

    /// <summary>
    /// Gets the SecurityRole_ID.
    /// </summary>
    /// <value></value>
    public DalInt SecurityRole_ID
    {
      get { return _SecurityRole_ID; }
    }

    /// <summary>
    /// Gets the SecurityPermission_ID.
    /// </summary>
    public DalInt SecurityPermission_ID
    {
      get { return _SecurityPermission_ID; }
```

```
}

/// <summary>
/// Gets the data log of the user.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Public Methods

/// <summary>
/// Creates a new security permission of a security role in the database using the values of the properties
/// of the current instance.
/// </summary>
public void Create()
{
  try
  {
    this.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }

  SqlCommand objCommand = new SqlCommand("SecurityRole_SecurityPermission_Insert", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, false);

  try
  {
    objConnection.Open();
    objCommand.ExecuteNonQuery();
    objConnection.Close();
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
}

#region Retrieve Methods

/// <summary>
/// Retrieves the security permission of a security role from the database using the value of the
/// SecurityRole_SecurityPermission_ID property of the current instance
/// </summary>
public void Retrieve()
{
  try
  {
    this._SecurityRole_SecurityPermission_ID.Validate("SecurityRole_SecurityPermission_ID");
  }
  catch (DalException e)
  {
    throw e;
  }

  try
  {
    RetrieveExecute(_SecurityRole_SecurityPermission_ID.Value, this, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
}

/// <summary>
/// Retrieves the security permission of a security role with the specified identifier from the database.
/// </summary>
/// <param name="securityPermission_ID">The ID of the user security role.</param>
public static StudyPlanning.DAL.SecurityRoles.SecurityPermission Retrieve(int securityPermission_ID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  StudyPlanning.DAL.SecurityRoles.SecurityPermission objSecurityPermission =
    new StudyPlanning.DAL.SecurityRoles.SecurityPermission();

  try
  {
    RetrieveExecute(securityPermission_ID, objSecurityPermission, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
  return objSecurityPermission;
}

private static void RetrieveExecute(
  int securityPermission_ID,
  StudyPlanning.DAL.SecurityRoles.SecurityPermission objSecurityPermission,
```

```
    SqlConnection sqlConnection)
{
    DataRow objDataRow;

    SqlCommand objCommand = new SqlCommand("SecurityRole_SecurityPermission_Select", sqlConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam;

    objParam = objCommand.Parameters.Add("@SecurityRole_SecurityPermission_ID", SqlDbType.Int);
    objParam.Value = securityPermission_ID;

    DataSet objDataSet = new DataSet("SecurityRole_SecurityPermission");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
      sqlConnection.Open();
      objAdap.Fill(objDataSet, "SecurityRole_SecurityPermission");
      sqlConnection.Close();
      objDataRow = objDataSet.Tables["SecurityRole_SecurityPermission"].Rows[0];
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }

    objSecurityPermission._SecurityRole_SecurityPermission_ID.Value = (int)objDataRow["SecurityRole_SecurityPermission_ID"];
    objSecurityPermission._SecurityRole_ID.Value = (int)objDataRow["SecurityRole_ID"];
    objSecurityPermission._SecurityPermission_ID.Value = (int)objDataRow["SecurityPermission_ID"];
    objSecurityPermission._Log.SetValues(objDataRow);
}

#endregion //Retrieve Methods

/// <summary>
/// Updates the security permission of the security role in the database using the original
/// security permission of the security role to resolve possible concurrency issues.
/// </summary>
/// <param name="originalSecurityPermission">The original <see cref="StudyPlanning.DAL.SecurityRoles.SecurityPermission"/> object.</param
    >
/// <returns><strong>true</strong> if the update was succesfully executed; otherwise, <strong>false</strong>.</returns>
public bool Update(StudyPlanning.DAL.SecurityRoles.SecurityPermission originalSecurityPermission)
{
    try
    {
      this.Validate();
    }
    catch (DalException excp)
    {
      throw excp;
    }

    bool blnResult = false;

    SqlCommand objCommand = new SqlCommand("SecurityRole_SecurityPermission_Update", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, false);
    AddParameters(originalSecurityPermission, objCommand, true);

    int rowsA↑ected;

    try
    {
      objConnection.Open();
      rowsA↑ected = objCommand.ExecuteNonQuery();
      objConnection.Close();

      if (rowsA↑ected > 0)
        blnResult = true;
    }
    catch(SqlException excp)
    {
      throw new DalException(excp);
    }

    return blnResult;
}

/// <summary>
/// Deletes the current security role from the database.
/// </summary>
/// <returns><strong>true</strong> if the deletion is succesfully executed; otherwise, <strong>false</strong>.</returns>
public bool Delete()
{
    bool blnResult = false;

    try
    {
      this.Validate();
    }
    catch (DalException excp)
    {
      throw excp;
    }

    SqlCommand objCommand = new SqlCommand("SecurityRole_SecurityPermission_Delete", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;
```

```
    AddParameters(this, objCommand, true);

    try
    {
      objConnection.Open();
      int rowsA↑ected = objCommand.ExecuteNonQuery();
      objConnection.Close();

      if (rowsA↑ected > 0)
        blnResult = true;
    }
    catch(SqlException objEx)
    {
      throw new DalException(objEx);
    }

    return blnResult;
  }

  /// <summary>
  /// Adds the relevant parameters to the specified <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see> object
  /// setting the values to ones of the specified <see cref="StudyPlanning.DAL.SecurityRoles.SecurityPermission"/> object.
  /// </summary>
  /// <param name="securityPermission">
  /// The <see cref="StudyPlanning.DAL.SecurityRoles.SecurityPermission"/> object containing the values that the parameters have to be
  /// initialized to.
  /// </param>
  /// <param name="objCommand">
  /// The <see cref="System.Data.SqlClient.SqlCommand"/> object to which parameters should be added.
  /// </param>
  /// <param name="isOriginal">
  /// <strong>true</strong> if the parameters to be added represent original
  /// data (for handling concurrent programming issues); otherwise, <strong>false</strong>.</param>
  /// <returns>The <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see>
  /// object with the relevant parameters added.</returns>
  private static void AddParameters(StudyPlanning.DAL.SecurityRoles.SecurityPermission securityPermission, SqlCommand
    objCommand, bool isOriginal)
  {
    string paramName = "";
    SqlParameter objParam;

    //SecurityRole_SecurityPermission_ID
    paramName = "SecurityRole_SecurityPermission_ID";
    objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.Int, 16);

    if (isOriginal)
      objParam.ParameterName = "@Original_" + paramName;

    objParam.Value = securityPermission.SecurityRole_SecurityPermission_ID.Value;

    //SecurityRole_ID
    paramName = "SecurityRole_ID";
    objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.Int, 16);

    if (isOriginal)
      objParam.ParameterName = "@Original_" + paramName;

    objParam.Value = securityPermission.SecurityRole_ID.Value;

    //SecurityPermission_ID
    paramName = "SecurityPermission_ID";
    objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.Int, 4);

    if (isOriginal)
      objParam.ParameterName = "@Original_" + paramName;

    objParam.Value = securityPermission.SecurityPermission_ID.Value;

    //Log
    objCommand = securityPermission.Log.AddParameters(objCommand, isOriginal);
  }

  /// <summary>
  /// Validates the current <see cref="StudyPlanning.DAL.SecurityRoles.SecurityPermission"/> object. If the
  /// value of either of the private properties is null a
  /// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
  /// </summary>
  private void Validate()
  {
    try
    {
      _SecurityRole_SecurityPermission_ID.Validate("SecurityRole_SecurityPermission_ID");
      _SecurityRole_ID.Validate("SecurityRole_ID");
      _SecurityPermission_ID.Validate("SecurityPermission_ID");
      _Log.Validate();
    }
    catch (DalException excp)
    {
      throw excp;
    }
  }

  /// <summary>
  /// Retrieves the security permissions of the specified security role.
  /// </summary>
  /// <param name="securityRole_ID">The ID of the security role for which to retrieve security permissions.</param>
  /// <returns>An array containing IDs of the security permissions of the specified role.</returns>
  public static int[] GetPermissionsFromRoleID(int securityRole_ID)
```

```
    {
      int[] permissions;

      string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
      SqlConnection objConnection = new SqlConnection(connString);

      SqlCommand objCommand = new SqlCommand("SecurityRole_SecurityPermission_GetPermissionsFromRoleID", objConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      SqlParameter objParam;

      objParam = objCommand.Parameters.Add("@SecurityRole_ID", SqlDbType.Int, 4);
      objParam.Value = securityRole_ID;

      DataSet objDataSet = new DataSet("Permissions");
      SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

      try
      {
        objConnection.Open();
        objAdap.Fill(objDataSet, "Permissions");
        objConnection.Close();

        int numRows = objDataSet.Tables["Permissions"].Rows.Count;
        permissions = new int[numRows];

        for(int i=0; i<numRows; i++)
        {
          permissions[i] = (int)objDataSet.Tables["Permissions"].Rows[i]["SecurityPermission_ID"];
        }
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }

      return permissions;
    }

    #endregion //Public Methods

  }
}
```

1.21 Specializations

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.Specializations
{
  /// <summary>
  /// Represents a Specialization.
  /// </summary>
  public class Specialization : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalInt _Specialization_ID = new DalInt(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.Specializations.Specialization"/>
    /// class.
    /// </summary>
    public Specialization() {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the Specialization_ID.
    /// </summary>
    public DalInt Specialization_ID
    {
      get { return _Specialization_ID; }
    }

    /// <summary>
    /// Gets the data log of the Specialization.
    /// </summary>
    public DataLog Log
    {
      get { return _Log; }
    }
```

```
#endregion //Public Properties

#region Public Methods

#region Retrieve Methods

/// <summary>
/// Retrieves the Specialization from the database using the value of the
/// Specialization_ID property of the current instance
/// </summary>
public void Retrieve()
{
  try
  {
    this._Specialization_ID.Validate("Specialization_ID");
  }
  catch (DalException e)
  {
    throw e;
  }

  try
  {
    RetrieveExecute(_Specialization_ID.Value, this, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
}

/// <summary>
/// Retrieves the Specialization with the specified identifier from the database.
/// </summary>
/// <param name="specialization_ID"></param>
public static Specialization Retrieve(int specialization_ID)
{
  string connString = System.Con|guration.Con|gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  Specialization objSpecialization = new Specialization();

  try
  {
    RetrieveExecute(specialization_ID, objSpecialization, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
  return objSpecialization;
}

private static void RetrieveExecute(
  int specialization_ID,
  Specialization specialization,
  SqlConnection sqlConnection)
{
  DataRow objDataRow;

  SqlCommand objCommand = new SqlCommand("Specialization_Select", sqlConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  SqlParameter objParam;

  objParam = objCommand.Parameters.Add("@Specialization_ID", SqlDbType.Int);
  objParam.Value = specialization._Specialization_ID.Value;

  DataSet objDataSet = new DataSet("Specialization");
  SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

  try
  {
    sqlConnection.Open();
    objAdap.Fill(objDataSet, "Specialization");
    sqlConnection.Close();
    objDataRow = objDataSet.Tables["Specialization"].Rows[0];
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
  specialization._Specialization_ID.Value = (int)objDataRow["Specialization_ID"];
  specialization._Log.SetValues(objDataRow);
}

#endregion //Retrieve Methods

  #endregion //Public Methods

#region Private Methods

/// <summary>
/// Validates the current <see cref="StudyPlanning.DAL.Specializations.Specialization"/> object. If the
/// value of either of the private properties is null a
/// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
/// </summary>
private void Validate()
```

```
    {
      try
      {
        _Specialization_ID.Validate("Specialization_ID");
        _Log.Validate();
      }
      catch (DalException excp)
      {
        throw excp;
      }
    }
    #endregion //Private Methods
  }
}
```

1.21.1 Course

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.Specializations
{
  /// <summary>
  /// Represents a studyPlanCriterion.
  /// </summary>
  public class Course : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalInt _Specialization_Course_ID = new DalInt(false);
    private DalInt _SpecializationVersion_ID = new DalInt(false);
    private DalGuid _Course_ID = new DalGuid(true);
    private DalString _Number = new DalString(true);
    private DalBool _PointGiving = new DalBool(false);
    private DalBool _Supplementary = new DalBool(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.Specializations.Course"/>
    /// class.
    /// </summary>
    public Course() {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the Specialization_Course_ID.
    /// </summary>
    public DalInt Specialization_Course_ID
    {
      get { return _Specialization_Course_ID; }
    }

    /// <summary>
    /// Gets the SpecializationVersion_ID.
    /// </summary>
    public DalInt SpecializationVersion_ID
    {
      get { return _SpecializationVersion_ID; }
    }

    /// <summary>
    /// Gets the Course_ID.
    /// </summary>
    public DalGuid Course_ID
    {
      get { return _Course_ID; }
    }

    /// <summary>
    /// Gets the course number.
    /// </summary>
    public DalString Number
    {
      get { return _Number; }
    }

    /// <summary>
    /// Gets the PointGiving property.
    /// </summary>
    public DalBool PointGiving
    {
      get { return _PointGiving; }
    }

    /// <summary>
```

```
/// Gets the Supplementary property.
/// </summary>
public DalBool Supplementary
{
  get { return _Supplementary; }
}

/// <summary>
/// Gets the data log of the Specialization_Course.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Public Methods

#region Retrieve Methods

/// <summary>
/// Retrieves the Specialization_Course from the database using the value of the
/// Specialization_Course_ID property of the current instance
/// </summary>
public void Retrieve()
{
  try
  {
    this._Specialization_Course_ID.Validate("Specialization_Course_ID");
  }
  catch (DalException e)
  {
    throw e;
  }

  try
  {
    RetrieveExecute(_Specialization_Course_ID.Value, this, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
}

/// <summary>
/// Retrieves the Specialization_Course with the specified identifier from the database.
/// </summary>
/// <param name="specialization_Course_ID"></param>
public static Course Retrieve(int specialization_Course_ID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  Course objSpecialization_Course = new Course();

  try
  {
    RetrieveExecute(specialization_Course_ID, objSpecialization_Course, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
  return objSpecialization_Course;
}

private static void RetrieveExecute(
  int specialization_Course_ID,
  Course specialization_Course,
  SqlConnection sqlConnection)
{
  DataRow objDataRow;

  SqlCommand objCommand = new SqlCommand("Specialization_Course_Select", sqlConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  SqlParameter objParam;

  objParam = objCommand.Parameters.Add("@Specialization_Course_ID", SqlDbType.Int);
  objParam.Value = specialization_Course._Specialization_Course_ID.Value;

  DataSet objDataSet = new DataSet("Specialization_Course");
  SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

  try
  {
    sqlConnection.Open();
    objAdap.Fill(objDataSet, "Specialization_Course");
    sqlConnection.Close();
    objDataRow = objDataSet.Tables["Specialization_Course"].Rows[0];
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
  specialization_Course._Specialization_Course_ID.Value = (int)objDataRow["Specialization_Course_ID"];
  specialization_Course._SpecializationVersion_ID.Value = (int)objDataRow["SpecializationVersion_ID"];
```

```
      if (objDataRow["Course_ID"].Equals(DBNull.Value))
        specialization_Course._Course_ID.IsNull = true;
      else
        specialization_Course._Course_ID.Value = (Guid)objDataRow["Course_ID"];

      if (objDataRow["Number"].Equals(DBNull.Value))
        specialization_Course._Number.IsNull = true;
      else
        specialization_Course._Number.Value = (string)objDataRow["Number"];

      specialization_Course._PointGiving.Value  = (bool)objDataRow["PointGiving"];
      specialization_Course._Supplementary.Value = (bool)objDataRow["Supplementary"];
      specialization_Course._Log.SetValues(objDataRow);
    }

    #endregion //Retrieve Methods

    /// <summary>
    /// Retrieves a list of courses which constitutes the specified specialization version.
    /// </summary>
    /// <param name="specializationVersion_ID">The ID of the specialization version for which to retrieve a list of constituting courses.</param>
    /// <param name="supplementary">An indication of whether the list should be constituted by supplementary courses or not.</param>
    /// <returns>An array containing IDs of the constituting courses of the specified specialization version.</returns>
    public static Guid[] GetCourses(int specializationVersion_ID, bool supplementary)
    {
      Guid[] courses;

      string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
      SqlConnection objConnection = new SqlConnection(connString);

      SqlCommand objCommand = new SqlCommand("Specialization_Course_GetCourses", objConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      SqlParameter objParam;

      objParam = objCommand.Parameters.Add("@SpecializationVersion_ID", SqlDbType.Int, 4);
      objParam.Value = specializationVersion_ID;

      objParam = objCommand.Parameters.Add("@Supplementary", SqlDbType.Bit, 1);
      objParam.Value = supplementary;

      DataSet objDataSet = new DataSet("Courses");
      SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

      try
      {
        objConnection.Open();
        objAdap.Fill(objDataSet, "Courses");
        objConnection.Close();

        int numRows = objDataSet.Tables["Courses"].Rows.Count;
        courses = new Guid[numRows];

        for(int i=0; i<numRows; i++)
        {
          courses[i] = (Guid)objDataSet.Tables["Courses"].Rows[i]["Course_ID"];
        }
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }

      return courses;
    }

    #endregion //Public Methods

    #region Private Methods

    /// <summary>
    /// Validates the current <see cref="StudyPlanning.DAL.Specializations.Course"/> object. If the
    /// value of either of the private properties is null a
    /// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
    /// </summary>
    private void Validate()
    {
      try
      {
        _Specialization_Course_ID.Validate("Specialization_Course_ID");
        _Log.Validate();
      }
      catch (DalException excp)
      {
        throw excp;
      }
    }
    #endregion //Private Methods
  }
}
```

1.21.2 SpecializationVersion

```
using System;
```

```
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.Specializations
{
  /// <summary>
  /// Represents a studyPlanCriterion.
  /// </summary>
  public class SpecializationVersion : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalInt _SpecializationVersion_ID = new DalInt(false);
    private DalInt _Specialization_ID = new DalInt(false);
    private DalInt _Version = new DalInt(false);
    private DalStringLocalizable _Name = new DalStringLocalizable(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.Specializations.SpecializationVersion"/>
    /// class.
    /// </summary>
    public SpecializationVersion() {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the SpecializationVersion_ID.
    /// </summary>
    public DalInt SpecializationVersion_ID
    {
      get { return _SpecializationVersion_ID; }
    }

    /// <summary>
    /// Gets the Specialization_ID.
    /// </summary>
    public DalInt Specialization_ID
    {
      get { return _Specialization_ID; }
    }

    /// <summary>
    /// Gets the Version number.
    /// </summary>
    public DalInt Version
    {
      get { return _Version; }
    }

    /// <summary>
    /// Gets the name of the SpecializationVersion.
    /// </summary>
    public DalStringLocalizable Name
    {
      get { return _Name; }
    }

    /// <summary>
    /// Gets the data log of the SpecializationVersion.
    /// </summary>
    public DataLog Log
    {
      get { return _Log; }
    }

    #endregion //Public Properties

    #region Public Methods

    #region Retrieve Methods

    /// <summary>
    /// Retrieves the SpecializationVersion from the database using the value of the
    /// SpecializationVersion_ID property of the current instance
    /// </summary>
    public void Retrieve()
    {
      try
      {
        this._SpecializationVersion_ID.Validate("SpecializationVersion_ID");
      }
      catch (DalException e)
      {
        throw e;
      }

      try
      {
        RetrieveExecute(_SpecializationVersion_ID.Value, this, objConnection);
```

```
      }
      catch (DalException objExc)
      {
        throw objExc;
      }
    }

    /// <summary>
    /// Retrieves the SpecializationVersion with the specified identifier from the database.
    /// </summary>
    /// <param name="specializationVersion_ID"></param>
    public static SpecializationVersion Retrieve(int specializationVersion_ID)
    {
      string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
      SqlConnection objConnection = new SqlConnection(connString);

      SpecializationVersion objSpecializationVersion = new SpecializationVersion();

      try
      {
        RetrieveExecute(specializationVersion_ID, objSpecializationVersion, objConnection);
      }
      catch (DalException objExc)
      {
        throw objExc;
      }
      return objSpecializationVersion;
    }

    private static void RetrieveExecute(
      int specializationVersion_ID,
      SpecializationVersion specializationVersion,
      SqlConnection sqlConnection)
    {
      DataRow objDataRow;

      SqlCommand objCommand = new SqlCommand("SpecializationVersion_Select", sqlConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      SqlParameter objParam;

      objParam = objCommand.Parameters.Add("@SpecializationVersion_ID", SqlDbType.Int);
      objParam.Value = specializationVersion._SpecializationVersion_ID.Value;

      DataSet objDataSet = new DataSet("SpecializationVersion");
      SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

      try
      {
        sqlConnection.Open();
        objAdap.Fill(objDataSet, "SpecializationVersion");
        sqlConnection.Close();
        objDataRow = objDataSet.Tables["SpecializationVersion"].Rows[0];
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }
      specializationVersion._SpecializationVersion_ID.Value = (int)objDataRow["SpecializationVersion_ID"];
      specializationVersion._Specialization_ID.Value    = (int)objDataRow["Specialization_ID"];
      specializationVersion._Version.Value         = (int)objDataRow["Version"];
      specializationVersion._Name.LoadXml((string)objDataRow["Name"]);
      specializationVersion._Log.SetValues(objDataRow);
    }

    #endregion //Retrieve Methods

      #endregion //Public Methods

    #region Private Methods

    /// <summary>
    /// Validates the current <see cref="StudyPlanning.DAL.Specializations.SpecializationVersion"/> object. If the
    /// value of either of the private properties is null a
    /// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
    /// </summary>
    private void Validate()
    {
      try
      {
        _SpecializationVersion_ID.Validate("SpecializationVersion_ID");
        _Log.Validate();
      }
      catch (DalException excp)
      {
        throw excp;
      }
    }
    #endregion //Private Methods
  }
}
```

1.22 Students

```
using System;
```

```
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.Students
{
  /// <summary>
  /// Represents a student.
  /// </summary>
  public class Student : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalGuid _Student_ID = new DalGuid(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Initializes a new instance of the <see cref="StudyPlanning.DAL.Students.Student"/>
    /// class.
    /// </summary>
    public Student()
    {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the ID of student.
    /// </summary>
    public DalGuid Student_ID
    {
      get { return _Student_ID; }
    }

    /// <summary>
    /// Gets the data log of the student.
    /// </summary>
    public DataLog Log
    {
      get { return _Log; }
    }

    #endregion //Public Properties

    #region Methods

    /// <summary>
    /// Creates a new student in the database using the values of the properties
    /// of the current instance.
    /// </summary>
    public void Create()
    {
      try
      {
        this.Validate();
      }
      catch (DalException excp)
      {
        throw excp;
      }

      SqlCommand objCommand = new SqlCommand("Student_Insert", objConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      AddParameters(this, objCommand, false);

      try
      {
        objConnection.Open();
        objCommand.ExecuteNonQuery();
        objConnection.Close();
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }
    }

    #region Retrieve Methods

    /// <summary>
    /// Retrieves the student from the database using the value of the
    /// Student_ID property of the current instance.
    /// </summary>
    public void Retrieve()
    {
      try
      {
        this._Student_ID.Validate("Student_ID");
      }
      catch (DalException excp)
      {
```

```
    throw excp;
  }

  try
  {
      RetrieveExecute(_Student_ID.Value, this, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
}

/// <summary>
/// Retrieves the student with the specified identifier from the database.
/// </summary>
/// <param name="student_ID">The globally unique identifier of the student.</param>
public static Student Retrieve(Guid student_ID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  Student objSp = new Student();

  try
  {
    RetrieveExecute(student_ID, objSp, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }

  return objSp;
}

private static void RetrieveExecute(
  Guid student_ID,
  Student objStudent,
  SqlConnection sqlConnection)
{
  SqlCommand objCommand = new SqlCommand("Student_Select", sqlConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  SqlParameter objParam;

  objParam = objCommand.Parameters.Add("@Student_ID", SqlDbType.UniqueIdenti↓er, 16);
  objParam.Value = student_ID;

  DataSet objDataSet = new DataSet("Student");
  SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

  try
  {
    sqlConnection.Open();
    objAdap.Fill(objDataSet, "Student");
    sqlConnection.Close();
    DataRow objDataRow = objDataSet.Tables["Student"].Rows[0];

    objStudent.Student_ID.Value = (Guid)objDataRow["Student_ID"];
    objStudent.Log.SetValues(objDataRow);
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
}

#endregion //Retrive Methods

/// <summary>
/// Updates the current student in the database using the original
/// student to resolve possible concurrency issues.
/// </summary>
/// <param name="originalStudent">The original <see cref="StudyPlanning.DAL.Students.Student"/> object.</param>
/// <returns><strong>true</strong> if the update was succesfully executed; otherwise, <strong>false</strong>.</returns>
public bool Update(Student originalStudent)
{
  try
  {
    this.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }

  bool blnResult = false;

  SqlCommand objCommand = new SqlCommand("Student_Update", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, false);
  AddParameters(originalStudent, objCommand, true);

  int rowsA↑ected;

  try
  {
```

```
      objConnection.Open();
      rowsA↑ected = objCommand.ExecuteNonQuery();
      objConnection.Close();

      if (rowsA↑ected > 0)
        blnResult = true;
    }
    catch(SqlException excp)
    {
      throw new DalException(excp);
    }

    return blnResult;
  }

  /// <summary>
  /// Deletes the current student from the database.
  /// </summary>
  /// <returns><strong>true</strong> if the deletion is succesfully executed; otherwise, <strong>false</strong>.</returns>
  public bool Delete()
  {
    bool blnResult = false;

    try
    {
      this.Validate();
    }
    catch (DalException excp)
    {
      throw excp;
    }

    SqlCommand objCommand = new SqlCommand("Student_Delete", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, true);

    try
    {
      objConnection.Open();
      int rowsA↑ected = objCommand.ExecuteNonQuery();
      objConnection.Close();

      if (rowsA↑ected > 0)
        blnResult = true;
    }
    catch(SqlException objEx)
    {
      throw new DalException(objEx);
    }

    return blnResult;
  }

  /// <summary>
  /// Adds the relevant parameters to the specified <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see> object
  /// setting the values to ones of the specified <see cref="StudyPlanning.DAL.Students.Student"/> object.
  /// </summary>
  /// <param name="student">
  /// The <see cref="StudyPlanning.DAL.Students.Student"/> object containing the values that the parameters have to be
  /// initialized to.
  /// </param>
  /// <param name="objCommand">
  /// The <see cref="System.Data.SqlClient.SqlCommand"/> object to which parameters should be added.
  /// </param>
  /// <param name="isOriginal">
  /// <strong>true</strong> if the parameters to be added represent original
  /// data (for handling concurrent programming issues); otherwise, <strong>false</strong>.</param>
  /// <returns>The <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see>
  /// object with the relevant parameters added.</returns>
  private static void AddParameters(Student student, SqlCommand objCommand, bool isOriginal)
  {
    string paramName = "";
    SqlParameter objParam;

    //Student_ID
    paramName = "Student_ID";
    objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

    if (isOriginal)
      objParam.ParameterName = "@Original_" + paramName;

    objParam.Value = student.Student_ID.Value;

    //Log
    objCommand = student.Log.AddParameters(objCommand, isOriginal);
  }

  /// <summary>
  /// Validates the current <see cref="StudyPlanning.DAL.Students.Student"/> object. If the
  /// value of either of the private properties is null a
  /// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
  /// </summary>
  private void Validate()
  {
    try
    {
      _Student_ID.Validate("Student_ID");
      _Log.Validate();
```

```
      }
      catch (DalException excp)
      {
        throw excp;
      }
    }

    #endregion //Methods
  }
}
```

1.22.1 Course

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.Students
{
  /// <summary>
  /// Represents a course of a student.
  /// </summary>
  public class Course : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalGuid _Student_Course_ID = new DalGuid(false);
    private DalGuid _StudentVersion_ID = new DalGuid(false);
    private DalGuid _CourseVersion_ID = new DalGuid(false);
    private DalInt _Part = new DalInt(false);
    private DalDateTime _DateOfSignUp = new DalDateTime(false);
    private DalDateTime _DateOfCancel = new DalDateTime(true);
    private DalDateTime _DateOfExaminationSignUp = new DalDateTime(true);
    private DalDateTime _DateOfExaminationCancel = new DalDateTime(true);
    private DalGuid _Assessment_ID = new DalGuid(true);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Initializes a new instance of the <see cref="StudyPlanning.DAL.Students.Course"/>
    /// class.
    /// </summary>
    public Course()
    {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the ID of the student course.
    /// </summary>
    public DalGuid Student_Course_ID
    {
      get { return _Student_Course_ID; }
    }

    /// <summary>
    /// Gets the ID of the student version.
    /// </summary>
    public DalGuid StudentVersion_ID
    {
      get { return _StudentVersion_ID; }
    }

    /// <summary>
    /// Gets the ID of the course version.
    /// </summary>
    public DalGuid CourseVersion_ID
    {
      get { return _CourseVersion_ID; }
    }

    /// <summary>
    /// Gets the part of the course version.
    /// </summary>
    public DalInt Part
    {
      get { return _Part; }
    }

    /// <summary>
    /// Gets the date at which the student signed up
    /// for the course.
    /// </summary>
    public DalDateTime DateOfSignUp
    {
      get { return _DateOfSignUp; }
    }
```

```
/// <summary>
/// Gets the date at which the student has cancelled
/// his previous sign up for the course.
/// </summary>
public DalDateTime DateOfCancel
{
  get { return _DateOfCancel; }
}

/// <summary>
/// Gets the date at which the student signed up
/// for examination in the course.
/// </summary>
public DalDateTime DateOfExaminationSignUp
{
  get { return _DateOfExaminationSignUp; }
}

/// <summary>
/// Gets the date at which the student has cancelled his
/// previous sign up for examination in the course.
/// </summary>
public DalDateTime DateOfExaminationCancel
{
  get { return _DateOfExaminationCancel; }
}

/// <summary>
/// Gets the ID of the assessment.
/// </summary>
public DalGuid Assessment_ID
{
  get { return _Assessment_ID; }
}

/// <summary>
/// Gets the data log of the course.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Methods

#region Retrieve Methods

/// <summary>
/// Retrieves the student course from the database using the value of the
/// Student_Course_ID property of the current instance.
/// </summary>
public void Retrieve()
{
  try
  {
    this._Student_Course_ID.Validate("Student_Course_ID");
  }
  catch (DalException excp)
  {
    throw excp;
  }

  try
  {
      RetrieveExecute(_Student_Course_ID.Value, this, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
}

/// <summary>
/// Retrieves the student course with the specified identifier from the database.
/// </summary>
/// <param name="student_Course_ID">The globally unique identifier of the student course.</param>
public static Course Retrieve(Guid student_Course_ID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  Course objCourse = new Course();

  try
  {
    RetrieveExecute(student_Course_ID, objCourse, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }

  return objCourse;
}

private static void RetrieveExecute(
  Guid student_Course_ID,
```

```
    Course objCourse,
    SqlConnection sqlConnection)
{
    SqlCommand objCommand = new SqlCommand("Student_Course_Select", sqlConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam;

    objParam = objCommand.Parameters.Add("@Student_Course_ID", SqlDbType.UniqueIdenti↓er, 16);
    objParam.Value = student_Course_ID;

    DataSet objDataSet = new DataSet("Student_Course");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
      sqlConnection.Open();
      objAdap.Fill(objDataSet, "Student_Course");
      sqlConnection.Close();
      DataRow objDataRow = objDataSet.Tables["Student_Course"].Rows[0];

      objCourse.Student_Course_ID.Value = (Guid)objDataRow["Student_Course_ID"];
      objCourse.StudentVersion_ID.Value = (Guid)objDataRow["StudentVersion_ID"];
      objCourse.CourseVersion_ID.Value = (Guid)objDataRow["CourseVersion_ID"];
      objCourse.Part.Value = (int)objDataRow["Part"];
      objCourse.DateOfSignUp.Value = (DateTime)objDataRow["DateOfSignUp"];

      if (objDataRow["DateOfCancel"].Equals(System.DBNull.Value))
        objCourse.DateOfCancel.IsNull = true;
      else
        objCourse.DateOfCancel.Value = (DateTime)objDataRow["DateOfCancel"];

      if (objDataRow["DateOfExaminationSignUp"].Equals(System.DBNull.Value))
        objCourse.DateOfExaminationSignUp.IsNull = true;
      else
        objCourse.DateOfExaminationSignUp.Value = (DateTime)objDataRow["DateOfExaminationSignUp"];

      if (objDataRow["DateOfExaminationCancel"].Equals(System.DBNull.Value))
        objCourse.DateOfExaminationCancel.IsNull = true;
      else
        objCourse.DateOfExaminationCancel.Value = (DateTime)objDataRow["DateOfExaminationCancel"];

      if (objDataRow["Assessment_ID"].Equals(System.DBNull.Value))
        objCourse.Assessment_ID.IsNull = true;
      else
        objCourse.Assessment_ID.Value = (Guid)objDataRow["Assessment_ID"];

      objCourse.Log.SetValues(objDataRow);
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
}

#endregion //Retrive Methods

/// <summary>
/// Retrieves the list of courses signed up by the specified student version.
/// </summary>
/// <param name="studentVersion_ID">The ID of the student version for which to retrieve the
/// list of signed up courses.</param>
/// <returns>An array containing Student_Course_IDs.</returns>
public static Guid[] GetCourses(Guid studentVersion_ID)
{
    Guid[] courses;

    string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
    SqlConnection objConnection = new SqlConnection(connString);

    SqlCommand objCommand = new SqlCommand("Student_Course_GetCourses", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam;

    objParam = objCommand.Parameters.Add("@StudentVersion_ID", SqlDbType.UniqueIdenti↓er, 16);
    objParam.Value = studentVersion_ID;

    DataSet objDataSet = new DataSet("Courses");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
      objConnection.Open();
      objAdap.Fill(objDataSet, "Courses");
      objConnection.Close();

      int numRows = objDataSet.Tables["Courses"].Rows.Count;
      courses = new Guid[numRows];

      for(int i=0; i<numRows; i++)
      {
        courses[i] = (Guid)objDataSet.Tables["Courses"].Rows[i]["Student_Course_ID"];
      }
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
```

```
      return courses;
    }

    /// <summary>
    /// Retrieves the list of courses signed up by the specified student version.
    /// </summary>
    /// <param name="studentVersion_ID">The ID of the student version for which to retrieve the
    /// list of signed up courses.</param>
    /// <param name="courseVersion_ID">???</param>
    /// <returns>An array containing IDs of courses.</returns>
    public static Guid[] GetCourses(Guid studentVersion_ID, Guid courseVersion_ID)
    {
      Guid[] courses;

      string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
      SqlConnection objConnection = new SqlConnection(connString);

      SqlCommand objCommand = new SqlCommand("Student_Course_GetIDsFromStudentAndCourse", objConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      SqlParameter objParam;

      objParam = objCommand.Parameters.Add("@StudentVersion_ID", SqlDbType.UniqueIdenti↓er, 16);
      objParam.Value = studentVersion_ID;

      objParam = objCommand.Parameters.Add("@CourseVersion_ID", SqlDbType.UniqueIdenti↓er, 16);
      objParam.Value = courseVersion_ID;

      DataSet objDataSet = new DataSet("Courses");
      SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

      try
      {
        objConnection.Open();
        objAdap.Fill(objDataSet, "Courses");
        objConnection.Close();

        int numRows = objDataSet.Tables["Courses"].Rows.Count;
        courses = new Guid[numRows];

        for(int i=0; i<numRows; i++)
        {
          courses[i] = (Guid)objDataSet.Tables["Courses"].Rows[i]["Student_Course_ID"];
        }
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }

      return courses;
    }

    #endregion //Methods
  }
}
```

1.22.2 Department

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.Students
{
  /// <summary>
  /// Represents a department of a student.
  /// </summary>
  public class Department : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalGuid _Student_Department_ID = new DalGuid(false);
    private DalGuid _StudentVersion_ID = new DalGuid(false);
    private DalInt _Department_ID = new DalInt(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Initializes a new instance of the <see cref="StudyPlanning.DAL.Students.Department"/>
    /// class.
    /// </summary>
    public Department()
    {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties
```

```
/// <summary>
/// Gets the ID of the student's department.
/// </summary>
public DalGuid Student_Department_ID
{
  get { return _Student_Department_ID; }
}

/// <summary>
/// Gets the ID of the student.
/// </summary>
public DalGuid StudentVersion_ID
{
  get { return _StudentVersion_ID; }
}

/// <summary>
/// Gets the ID of the department.
/// </summary>
public DalInt Department_ID
{
  get { return _Department_ID; }
}

/// <summary>
/// Gets the data log of the student department.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Methods

#region Retrieve Methods

/// <summary>
/// Retrieves the student study plan from the database using the value of the
/// Student_Department_ID property of the current instance.
/// </summary>
public void Retrieve()
{
  try
  {
    this._Student_Department_ID.Validate("Student_Department_ID");
  }
  catch (DalException excp)
  {
    throw excp;
  }

  try
  {
      RetrieveExecute(_Student_Department_ID.Value, this, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
}

/// <summary>
/// Retrieves the student department with the specified identifier from the database.
/// </summary>
/// <param name="student_Department_ID">The globally unique identifier of the department period.</param>
public static Department Retrieve(Guid student_Department_ID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  Department objDepartment = new Department();

  try
  {
    RetrieveExecute(student_Department_ID, objDepartment, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }

  return objDepartment;
}

private static void RetrieveExecute(
  Guid student_Department_ID,
  Department objDepartment,
  SqlConnection sqlConnection)
{
  SqlCommand objCommand = new SqlCommand("Student_Department_Select", sqlConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  SqlParameter objParam;

  objParam = objCommand.Parameters.Add("@Student_Department_ID", SqlDbType.UniqueIdenti↓er, 16);
  objParam.Value = student_Department_ID;
```

```
      DataSet objDataSet = new DataSet("Student_Department");
      SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

      try
      {
        sqlConnection.Open();
        objAdap.Fill(objDataSet, "Student_Department");
        sqlConnection.Close();
        DataRow objDataRow = objDataSet.Tables["Student_Department"].Rows[0];

        objDepartment.Student_Department_ID.Value = (Guid)objDataRow["Student_Department_ID"];
        objDepartment.StudentVersion_ID.Value = (Guid)objDataRow["StudentVersion_ID"];
        objDepartment.Department_ID.Value = (int)objDataRow["Department_ID"];

        objDepartment.Log.SetValues(objDataRow);
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }
    }

    #endregion //Retrive Methods

    /// <summary>
    /// Adds the relevant parameters to the specified <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see> object
    /// setting the values to ones of the specified <see cref="StudyPlanning.DAL.Students.Department"/> object.
    /// </summary>
    /// <param name="department">
    /// The <see cref="StudyPlanning.DAL.Students.Department"/> object containing the values that the parameters have to be
    /// initialized to.
    /// </param>
    /// <param name="objCommand">
    /// The <see cref="System.Data.SqlClient.SqlCommand"/> object to which parameters should be added.
    /// </param>
    /// <param name="isOriginal">
    /// <strong>true</strong> if the parameters to be added represent original
    /// data (for handling concurrent programming issues); otherwise, <strong>false</strong>.</param>
    /// <returns>The <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see>
    /// object with the relevant parameters added.</returns>
    private static void AddParameters(Department department, SqlCommand objCommand, bool isOriginal)
    {
      string paramName = "";
      SqlParameter objParam;

      //Student_Department_ID
      paramName = "Student_Department_ID";
      objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      objParam.Value = department.Student_Department_ID.Value;

      //StudentVersion_ID
      paramName = "StudentVersion_ID";
      objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      objParam.Value = department.StudentVersion_ID.Value;

      //Department_ID
      paramName = "Department_ID";
      objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.Int, 4);

      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      objParam.Value = department.Department_ID.Value;

      //Log
      objCommand = department.Log.AddParameters(objCommand, isOriginal);
    }

    /// <summary>
    /// Validates the current <see cref="StudyPlanning.DAL.Students.Department"/> object. If the
    /// value of either of the private properties is null a
    /// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
    /// </summary>
    private void Validate()
    {
      try
      {
        _Student_Department_ID.Validate("Student_Department_ID");
        _StudentVersion_ID.Validate("StudentVersion_ID");
        _Department_ID.Validate("Department_ID");
        _Log.Validate();
      }
      catch (DalException excp)
      {
        throw excp;
      }
    }

    #endregion //Methods
  }
}
```

1.22.3 Project

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.Students
{
  /// <summary>
  /// Represents a project of a student.
  /// </summary>
  public class Project : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalGuid _Student_Project_ID = new DalGuid(false);
    private DalGuid _StudentVersion_ID = new DalGuid(false);
    private DalGuid _Project_ID = new DalGuid(false);
    private DalGuid _Assessment_ID = new DalGuid(true);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Initializes a new instance of the <see cref="StudyPlanning.DAL.Students.Project"/>
    /// class.
    /// </summary>
    public Project()
    {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the ID of the student project.
    /// </summary>
    public DalGuid Student_Project_ID
    {
      get { return _Student_Project_ID; }
    }

    /// <summary>
    /// Gets the ID of the student version.
    /// </summary>
    public DalGuid StudentVersion_ID
    {
      get { return _StudentVersion_ID; }
    }

    /// <summary>
    /// Gets the ID of the project.
    /// </summary>
    public DalGuid Project_ID
    {
      get { return _Project_ID; }
    }

    /// <summary>
    /// Gets the ID of the assessment.
    /// </summary>
    public DalGuid Assessment_ID
    {
      get { return _Assessment_ID; }
    }

    /// <summary>
    /// Gets the data log of the course.
    /// </summary>
    public DataLog Log
    {
      get { return _Log; }
    }

    #endregion //Public Properties

    #region Methods

    #region Retrieve Methods

    /// <summary>
    /// Retrieves the student project from the database using the value of the
    /// Student_Project_ID property of the current instance.
    /// </summary>
    public void Retrieve()
    {
      try
      {
        this._Student_Project_ID.Validate("Student_Project_ID");
      }
      catch (DalException excp)
      {
        throw excp;
```

```
      }

      try
      {
          RetrieveExecute(_Student_Project_ID.Value, this, objConnection);
      }
      catch (DalException objExc)
      {
        throw objExc;
      }
    }

    /// <summary>
    /// Retrieves the student project with the specified identifier from the database.
    /// </summary>
    /// <param name="student_Project_ID">The globally unique identifier of the student project.</param>
    public static Project Retrieve(Guid student_Project_ID)
    {
      string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
      SqlConnection objConnection = new SqlConnection(connString);

      Project objProject = new Project();

      try
      {
        RetrieveExecute(student_Project_ID, objProject, objConnection);
      }
      catch (DalException objExc)
      {
        throw objExc;
      }

      return objProject;
    }

    private static void RetrieveExecute(
      Guid student_Project_ID,
      Project objProject,
      SqlConnection sqlConnection)
    {
      SqlCommand objCommand = new SqlCommand("Student_Project_Select", sqlConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      SqlParameter objParam;

      objParam = objCommand.Parameters.Add("@Student_Project_ID", SqlDbType.UniqueIdenti↓er, 16);
      objParam.Value = student_Project_ID;

      DataSet objDataSet = new DataSet("Student_Project");
      SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

      try
      {
        sqlConnection.Open();
        objAdap.Fill(objDataSet, "Student_Project");
        sqlConnection.Close();
        DataRow objDataRow = objDataSet.Tables["Student_Project"].Rows[0];

        objProject.Student_Project_ID.Value = (Guid)objDataRow["Student_Project_ID"];
        objProject.StudentVersion_ID.Value = (Guid)objDataRow["StudentVersion_ID"];
        objProject.Project_ID.Value = (Guid)objDataRow["Project_ID"];

        if (objDataRow["Assessment_ID"].Equals(System.DBNull.Value))
          objProject.Assessment_ID.IsNull = true;
        else
          objProject.Assessment_ID.Value = (Guid)objDataRow["Assessment_ID"];

        objProject.Log.SetValues(objDataRow);
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }
    }

    #endregion //Retrive Methods

    #endregion //Methods
  }
}
```

1.22.4 StudentVersion

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.Students
{
  /// <summary>
  /// Represents a version of a student.
  /// </summary>
  public class StudentVersion : StudyPlanning.DAL.DbObject
  {
    #region Private Properties
```

```
private DalGuid _StudentVersion_ID = new DalGuid(false);
private DalGuid _Student_ID = new DalGuid(false);
private DalInt _Version = new DalInt(false);
private DalString _StudyNumber = new DalString(false);
private DalDateTime _DateOfSignUp = new DalDateTime(false);
private DalGuid _Person_ID = new DalGuid(false);
private DalInt _StudyTypeVersion_ID = new DalInt(false);
private DalInt _TechnicalPackageVersion_ID = new DalInt(true);
private DalInt _TechnicalLineVersion_ID = new DalInt(false);
private DataLog _Log;

#endregion //Private Properties

#region Constructors

/// <summary>
/// Initializes a new instance of the <see cref="StudyPlanning.DAL.Students.StudentVersion"/>
/// class.
/// </summary>
public StudentVersion()
{
  _Log = new DataLog();
}

#endregion

#region Public Properties

/// <summary>
/// Gets the ID of the student version.
/// </summary>
public DalGuid StudentVersion_ID
{
  get { return _StudentVersion_ID; }
}

/// <summary>
/// Gets the ID of the student.
/// </summary>
public DalGuid Student_ID
{
  get { return _Student_ID; }
}

/// <summary>
/// Gets the ID of the version.
/// </summary>
public DalInt Version
{
  get { return _Version; }
}

/// <summary>
/// Gets the student's study number.
/// </summary>
public DalString StudyNumber
{
  get { return _StudyNumber; }
}

/// <summary>
/// Gets the date at which the student signed up at the university.
/// </summary>
public DalDateTime DateOfSignUp
{
  get { return _DateOfSignUp; }
}

/// <summary>
/// Gets the ID of the person.
/// </summary>
public DalGuid Person_ID
{
  get { return _Person_ID; }
}

/// <summary>
/// Gets the ID of the study type version of the student version.
/// </summary>
public DalInt StudyTypeVersion_ID
{
  get { return _StudyTypeVersion_ID; }
}

/// <summary>
/// Gets the ID of the technical package version of the student version.
/// </summary>
public DalInt TechnicalPackageVersion_ID
{
  get { return _TechnicalPackageVersion_ID; }
}

/// <summary>
/// Gets the ID of the technical line version of the student version.
/// </summary>
public DalInt TechnicalLineVersion_ID
{
  get { return _TechnicalLineVersion_ID; }
```

```
}

/// <summary>
/// Gets the data log of the student version.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Methods

#region Retrieve Methods

/// <summary>
/// Retrieves the student study plan from the database using the value of the
/// StudentVersion_ID property of the current instance.
/// </summary>
public void Retrieve()
{
  try
  {
    this._StudentVersion_ID.Validate("StudentVersion_ID");
  }
  catch (DalException excp)
  {
    throw excp;
  }

  try
  {
      RetrieveExecute(_StudentVersion_ID.Value, this, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
}

/// <summary>
/// Retrieves the student version with the specified identifier from the database.
/// </summary>
/// <param name="studentVersion_ID">The globally unique identifier of the version period.</param>
public static StudentVersion Retrieve(Guid studentVersion_ID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  StudentVersion objStudentVersion = new StudentVersion();

  try
  {
    RetrieveExecute(studentVersion_ID, objStudentVersion, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }

  return objStudentVersion;
}

private static void RetrieveExecute(
  Guid studentVersion_ID,
  StudentVersion objStudentVersion,
  SqlConnection sqlConnection)
{
  SqlCommand objCommand = new SqlCommand("StudentVersion_Select", sqlConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  SqlParameter objParam;

  objParam = objCommand.Parameters.Add("@StudentVersion_ID", SqlDbType.UniqueIdenti↓er, 16);
  objParam.Value = studentVersion_ID;

  DataSet objDataSet = new DataSet("Student_StudentVersion");
  SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

  try
  {
    sqlConnection.Open();
    objAdap.Fill(objDataSet, "Student_StudentVersion");
    sqlConnection.Close();
    DataRow objDataRow = objDataSet.Tables["Student_StudentVersion"].Rows[0];

    objStudentVersion.StudentVersion_ID.Value = (Guid)objDataRow["StudentVersion_ID"];
    objStudentVersion.Student_ID.Value = (Guid)objDataRow["Student_ID"];
    objStudentVersion.Version.Value = (int)objDataRow["Version"];
    objStudentVersion.StudyNumber.Value = (string)objDataRow["StudyNumber"];
    objStudentVersion.DateOfSignUp.Value = (DateTime)objDataRow["DateOfSignUp"];
    objStudentVersion.Person_ID.Value = (Guid)objDataRow["Person_ID"];
    objStudentVersion.StudyTypeVersion_ID.Value = (int)objDataRow["StudyTypeVersion_ID"];

    if (objDataRow["TechnicalPackageVersion_ID"].Equals(System.DBNull.Value))
      objStudentVersion.TechnicalPackageVersion_ID.IsNull = true;
    else
      objStudentVersion.TechnicalPackageVersion_ID.Value = (int)objDataRow["TechnicalPackageVersion_ID"];
```

```
        if (objDataRow["TechnicalLineVersion_ID"].Equals(System.DBNull.Value))
          objStudentVersion.TechnicalLineVersion_ID.IsNull = true;
        else
          objStudentVersion.TechnicalLineVersion_ID.Value = (int)objDataRow["TechnicalLineVersion_ID"];

        objStudentVersion.Log.SetValues(objDataRow);
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }
    }

    #endregion //Retrive Methods

    #endregion //Methods
  }
}
```

1.22.5 StudyPlan

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.Students
{
  /// <summary>
  /// Represents a study plan of a student.
  /// </summary>
  public class StudyPlan : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalGuid _Student_StudyPlan_ID = new DalGuid(false);
    private DalGuid _Student_ID = new DalGuid(false);
    private DalGuid _StudyPlan_ID = new DalGuid(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Initializes a new instance of the <see cref="StudyPlanning.DAL.Students.StudyPlan"/>
    /// class.
    /// </summary>
    public StudyPlan()
    {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the ID of the student's study plan.
    /// </summary>
    public DalGuid Student_StudyPlan_ID
    {
      get { return _Student_StudyPlan_ID; }
    }

    /// <summary>
    /// Gets the ID of the student.
    /// </summary>
    public DalGuid Student_ID
    {
      get { return _Student_ID; }
    }

    /// <summary>
    /// Gets the ID of the study plan.
    /// </summary>
    public DalGuid StudyPlan_ID
    {
      get { return _StudyPlan_ID; }
    }

    /// <summary>
    /// Gets the data log of the course.
    /// </summary>
    public DataLog Log
    {
      get { return _Log; }
    }

    #endregion //Public Properties

    #region Methods

    /// <summary>
```

```
/// Creates a new student study plan in the database using the values of the properties
/// of the current instance.
/// </summary>
public void Create()
{
  try
  {
    this.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }

  SqlCommand objCommand = new SqlCommand("Student_StudyPlan_Insert", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, false);

  try
  {
    objConnection.Open();
    objCommand.ExecuteNonQuery();
    objConnection.Close();
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
}

#region Retrieve Methods

/// <summary>
/// Retrieves the student study plan from the database using the value of the
/// Student_StudyPlan_ID property of the current instance.
/// </summary>
public void Retrieve()
{
  try
  {
    this._Student_StudyPlan_ID.Validate("Student_StudyPlan_ID");
  }
  catch (DalException excp)
  {
    throw excp;
  }

  try
  {
      RetrieveExecute(_Student_StudyPlan_ID.Value, this, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
}

/// <summary>
/// Retrieves the student study plan with the specified identifier from the database.
/// </summary>
/// <param name="student_StudyPlan_ID">The globally unique identifier of the study plan period.</param>
public static StudyPlan Retrieve(Guid student_StudyPlan_ID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  StudyPlan objStudyPlan = new StudyPlan();

  try
  {
    RetrieveExecute(student_StudyPlan_ID, objStudyPlan, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }

  return objStudyPlan;
}

private static void RetrieveExecute(
  Guid student_StudyPlan_ID,
  StudyPlan objStudyPlan,
  SqlConnection sqlConnection)
{
  SqlCommand objCommand = new SqlCommand("Student_StudyPlan_Select", sqlConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  SqlParameter objParam;

  objParam = objCommand.Parameters.Add("@Student_StudyPlan_ID", SqlDbType.UniqueIdenti↓er, 16);
  objParam.Value = student_StudyPlan_ID;

  DataSet objDataSet = new DataSet("Student_StudyPlan");
  SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

  try
  {
```

```
      sqlConnection.Open();
      objAdap.Fill(objDataSet, "Student_StudyPlan");
      sqlConnection.Close();
      DataRow objDataRow = objDataSet.Tables["Student_StudyPlan"].Rows[0];

      objStudyPlan.Student_StudyPlan_ID.Value = (Guid)objDataRow["Student_StudyPlan_ID"];
      objStudyPlan.Student_ID.Value = (Guid)objDataRow["Student_ID"];
      objStudyPlan.StudyPlan_ID.Value = (Guid)objDataRow["StudyPlan_ID"];

      objStudyPlan.Log.SetValues(objDataRow);
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
  }

  #endregion //Retrive Methods

  /// <summary>
  /// Updates the current student study plan in the database using the original
  /// student study plan to resolve possible concurrency issues.
  /// </summary>
  /// <param name="originalStudent_StudyPlan">The original <see cref="StudyPlanning.DAL.Students.StudyPlan"/> object.</param>
  /// <returns><strong>true</strong> if the update was succesfully executed; otherwise, <strong>false</strong>.</returns>
  public bool Update(StudyPlan originalStudent_StudyPlan)
  {
    try
    {
      this.Validate();
    }
    catch (DalException excp)
    {
      throw excp;
    }

    bool blnResult = false;

    SqlCommand objCommand = new SqlCommand("Student_StudyPlan_Update", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, false);
    AddParameters(originalStudent_StudyPlan, objCommand, true);

    int rowsA↑ected;

    try
    {
      objConnection.Open();
      rowsA↑ected = objCommand.ExecuteNonQuery();
      objConnection.Close();

      if (rowsA↑ected > 0)
        blnResult = true;
    }
    catch(SqlException excp)
    {
      throw new DalException(excp);
    }

    return blnResult;
  }

  /// <summary>
  /// Deletes the current student study plan from the database.
  /// </summary>
  /// <returns><strong>true</strong> if the deletion is succesfully executed; otherwise, <strong>false</strong>.</returns>
  public bool Delete()
  {
    bool blnResult = false;

    try
    {
      this.Validate();
    }
    catch (DalException excp)
    {
      throw excp;
    }

    SqlCommand objCommand = new SqlCommand("Student_StudyPlan_Delete", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, true);

    try
    {
      objConnection.Open();
      int rowsA↑ected = objCommand.ExecuteNonQuery();
      objConnection.Close();

      if (rowsA↑ected > 0)
        blnResult = true;
    }
    catch(SqlException objEx)
    {
      throw new DalException(objEx);
    }
```

```
  return blnResult;
}

/// <summary>
/// Adds the relevant parameters to the specified <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see> object
/// setting the values to ones of the specified <see cref="StudyPlanning.DAL.Students.StudyPlan"/> object.
/// </summary>
/// <param name="studyPlan">
/// The <see cref="StudyPlanning.DAL.Students.StudyPlan"/> object containing the values that the parameters have to be
/// initialized to.
/// </param>
/// <param name="objCommand">
/// The <see cref="System.Data.SqlClient.SqlCommand"/> object to which parameters should be added.
/// </param>
/// <param name="isOriginal">
/// <strong>true</strong> if the parameters to be added represent original
/// data (for handling concurrent programming issues); otherwise, <strong>false</strong>.</param>
/// <returns>The <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see>
/// object with the relevant parameters added.</returns>
private static void AddParameters(StudyPlan studyPlan, SqlCommand objCommand, bool isOriginal)
{
  string paramName = "";
  SqlParameter objParam;

  //Student_StudyPlan_ID
  paramName = "Student_StudyPlan_ID";
  objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

  if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

  objParam.Value = studyPlan.Student_StudyPlan_ID.Value;

  //Student_ID
  paramName = "Student_ID";
  objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

  if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

  objParam.Value = studyPlan.Student_ID.Value;

  //StudyPlan_ID
  paramName = "StudyPlan_ID";
  objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

  if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

  objParam.Value = studyPlan.StudyPlan_ID.Value;

  //Log
  objCommand = studyPlan.Log.AddParameters(objCommand, isOriginal);
}

/// <summary>
/// Validates the current <see cref="StudyPlanning.DAL.Students.StudyPlan"/> object. If the
/// value of either of the private properties is null a
/// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
/// </summary>
private void Validate()
{
  try
  {
    _Student_StudyPlan_ID.Validate("Student_StudyPlan_ID");
    _Student_ID.Validate("Student_ID");
    _StudyPlan_ID.Validate("StudyPlan_ID");
    _Log.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }
}

/// <summary>
/// Retrieves a list of study plan identifiers representing the study plans of the student
/// with the specified ID.
/// </summary>
/// <param name="student_ID">Identification of the student for which to retrieve a list of study plans.</param>
/// <returns>An array (possibly empty) of study plan identifiers.</returns>
public static Guid[] GetStudyPlans(Guid student_ID)
{
  Guid[] studyplans = null;

  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  SqlCommand objCommand = new SqlCommand("Student_StudyPlan_GetStudyPlans", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  SqlParameter objParam;

  objParam = objCommand.Parameters.Add("@Student_ID", SqlDbType.UniqueIdenti↓er, 16);
  objParam.Value = student_ID;

  DataSet objDataSet = new DataSet("Result");
  SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);
```

```
      try
      {
        objConnection.Open();
        objAdap.Fill(objDataSet, "Result");
        objConnection.Close();

        int numRows = objDataSet.Tables["Result"].Rows.Count;
        studyplans = new Guid[numRows];

        for(int i=0; i<numRows; i++)
        {
          studyplans[i] = (Guid)objDataSet.Tables["Result"].Rows[i]["StudyPlan_ID"];
        }
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }
      return studyplans;
    }

    /// <summary>
    /// Retrives the ID of the table row containing the specified study plan ID.
    /// </summary>
    /// <param name="studyPlan_ID">The ID of the study plan for which to get the row ID.</param>
    /// <returns>A <see cref="System.Guid"/> being the ID of the row.</returns>
    public static Guid GetID(Guid studyPlan_ID)
    {
      Guid studentStudyPlan_ID = Guid.Empty;

      string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
      SqlConnection objConnection = new SqlConnection(connString);

      SqlCommand objCommand = new SqlCommand("Student_StudyPlan_GetID", objConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      SqlParameter objParam;

      objParam = objCommand.Parameters.Add("@StudyPlan_ID", SqlDbType.UniqueIdenti↓er, 16);
      objParam.Value = studyPlan_ID;

      objParam = objCommand.Parameters.Add("@Student_StudyPlan_ID", SqlDbType.UniqueIdenti↓er, 16);
      objParam.Direction = ParameterDirection.Output;

      try
      {
        objConnection.Open();
        objCommand.ExecuteNonQuery();
        studentStudyPlan_ID = (Guid)objCommand.Parameters["@Student_StudyPlan_ID"].Value;
        objConnection.Close();
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }

      return studentStudyPlan_ID;
    }

    #endregion //Methods
  }
}
```

1.22.6 StudyPlanCriterion

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.Students
{
  /// <summary>
  /// Represents a study plan criterion of a student.
  /// </summary>
  public class StudyPlanCriterion : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalGuid _Student_StudyPlanCriterion_ID = new DalGuid(false);
    private DalGuid _Student_ID = new DalGuid(false);
    private DalGuid _StudyPlanCriterion_ID = new DalGuid(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Initializes a new instance of the <see cref="StudyPlanning.DAL.Students.StudyPlanCriterion"/>
    /// class.
    /// </summary>
    public StudyPlanCriterion()
    {
      _Log = new DataLog();
```

```
}

#endregion

#region Public Properties

/// <summary>
/// Gets the ID of the student's study plan criterion.
/// </summary>
public DalGuid Student_StudyPlanCriterion_ID
{
  get { return _Student_StudyPlanCriterion_ID; }
}

/// <summary>
/// Gets the ID of the student.
/// </summary>
public DalGuid Student_ID
{
  get { return _Student_ID; }
}

/// <summary>
/// Gets the ID of the study plan criterion.
/// </summary>
public DalGuid StudyPlanCriterion_ID
{
  get { return _StudyPlanCriterion_ID; }
}

/// <summary>
/// Gets the data log of the course.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Methods

/// <summary>
/// Creates a new student study plan criterion in the database using the values of the properties
/// of the current instance.
/// </summary>
public void Create()
{
  try
  {
    this.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }

  SqlCommand objCommand = new SqlCommand("Student_StudyPlanCriterion_Insert", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, false);

  try
  {
    objConnection.Open();
    objCommand.ExecuteNonQuery();
    objConnection.Close();
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
}

#region Retrieve Methods

/// <summary>
/// Retrieves the student study plan criterion from the database using the value of the
/// Student_StudyPlanCriterion_ID property of the current instance.
/// </summary>
public void Retrieve()
{
  try
  {
    this._Student_StudyPlanCriterion_ID.Validate("Student_StudyPlanCriterion_ID");
  }
  catch (DalException excp)
  {
    throw excp;
  }

  try
  {
      RetrieveExecute(_Student_StudyPlanCriterion_ID.Value, this, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
```

```
}

/// <summary>
/// Retrieves the student study plan criterion with the specified identifier from the database.
/// </summary>
/// <param name="student_StudyPlanCriterion_ID">The globally unique identifier of the study plan criterion period.</param>
public static StudyPlanCriterion Retrieve(Guid student_StudyPlanCriterion_ID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  StudyPlanCriterion objStudyPlanCriterion = new StudyPlanCriterion();

  try
  {
    RetrieveExecute(student_StudyPlanCriterion_ID, objStudyPlanCriterion, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }

  return objStudyPlanCriterion;
}

private static void RetrieveExecute(
  Guid student_StudyPlanCriterion_ID,
  StudyPlanCriterion objStudyPlanCriterion,
  SqlConnection sqlConnection)
{
  SqlCommand objCommand = new SqlCommand("Student_StudyPlanCriterion_Select", sqlConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  SqlParameter objParam;

  objParam = objCommand.Parameters.Add("@Student_StudyPlanCriterion_ID", SqlDbType.UniqueIdenti↓er, 16);
  objParam.Value = student_StudyPlanCriterion_ID;

  DataSet objDataSet = new DataSet("Student_StudyPlanCriterion");
  SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

  try
  {
    sqlConnection.Open();
    objAdap.Fill(objDataSet, "Student_StudyPlanCriterion");
    sqlConnection.Close();
    DataRow objDataRow = objDataSet.Tables["Student_StudyPlanCriterion"].Rows[0];

    objStudyPlanCriterion.Student_StudyPlanCriterion_ID.Value = (Guid)objDataRow["Student_StudyPlanCriterion_ID"];
    objStudyPlanCriterion.Student_ID.Value = (Guid)objDataRow["Student_ID"];
    objStudyPlanCriterion.StudyPlanCriterion_ID.Value = (Guid)objDataRow["StudyPlanCriterion_ID"];

    objStudyPlanCriterion.Log.SetValues(objDataRow);
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
}

#endregion //Retrive Methods

/// <summary>
/// Updates the current student study plan criterion in the database using the original
/// student study plan criterion to resolve possible concurrency issues.
/// </summary>
/// <param name="originalStudent_StudyPlanCriterion">The original <see cref="StudyPlanning.DAL.Students.StudyPlanCriterion"/> object.</
  param>
/// <returns><strong>true</strong> if the update was succesfully executed; otherwise, <strong>false</strong>.</returns>
public bool Update(StudyPlanCriterion originalStudent_StudyPlanCriterion)
{
  try
  {
    this.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }

  bool blnResult = false;

  SqlCommand objCommand = new SqlCommand("Student_StudyPlanCriterion_Update", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, false);
  AddParameters(originalStudent_StudyPlanCriterion, objCommand, true);

  int rowsA↑ected;

  try
  {
    objConnection.Open();
    rowsA↑ected = objCommand.ExecuteNonQuery();
    objConnection.Close();

    if (rowsA↑ected > 0)
      blnResult = true;
  }
```

```
    catch(SqlException excp)
    {
      throw new DalException(excp);
    }

    return blnResult;
  }

  /// <summary>
  /// Deletes the current student study plan criterion from the database.
  /// </summary>
  /// <returns><strong>true</strong> if the deletion is succesfully executed; otherwise, <strong>false</strong>.</returns>
  public bool Delete()
  {
    bool blnResult = false;

    try
    {
      this.Validate();
    }
    catch (DalException excp)
    {
      throw excp;
    }

    SqlCommand objCommand = new SqlCommand("Student_StudyPlanCriterion_Delete", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, true);

    try
    {
      objConnection.Open();
      int rowsA↑ected = objCommand.ExecuteNonQuery();
      objConnection.Close();

      if (rowsA↑ected > 0)
        blnResult = true;
    }
    catch(SqlException objEx)
    {
      throw new DalException(objEx);
    }

    return blnResult;
  }

  /// <summary>
  /// Adds the relevant parameters to the specified <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see> object
  /// setting the values to ones of the specified <see cref="StudyPlanning.DAL.Students.StudyPlanCriterion"/> object.
  /// </summary>
  /// <param name="studyPlanCriterion">
  /// The <see cref="StudyPlanning.DAL.Students.StudyPlanCriterion"/> object containing the values that the parameters have to be
  /// initialized to.
  /// </param>
  /// <param name="objCommand">
  /// The <see cref="System.Data.SqlClient.SqlCommand"/> object to which parameters should be added.
  /// </param>
  /// <param name="isOriginal">
  /// <strong>true</strong> if the parameters to be added represent original
  /// data (for handling concurrent programming issues); otherwise, <strong>false</strong>.</param>
  /// <returns>The <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see>
  /// object with the relevant parameters added.</returns>
  private static void AddParameters(StudyPlanCriterion studyPlanCriterion, SqlCommand objCommand, bool isOriginal)
  {
    string paramName = "";
    SqlParameter objParam;

    //Student_StudyPlanCriterion_ID
    paramName = "Student_StudyPlanCriterion_ID";
    objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

    if (isOriginal)
      objParam.ParameterName = "@Original_" + paramName;

    objParam.Value = studyPlanCriterion.Student_StudyPlanCriterion_ID.Value;

    //Student_ID
    paramName = "Student_ID";
    objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

    if (isOriginal)
      objParam.ParameterName = "@Original_" + paramName;

    objParam.Value = studyPlanCriterion.Student_ID.Value;

    //StudyPlanCriterion_ID
    paramName = "StudyPlanCriterion_ID";
    objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

    if (isOriginal)
      objParam.ParameterName = "@Original_" + paramName;

    objParam.Value = studyPlanCriterion.StudyPlanCriterion_ID.Value;

    //Log
    objCommand = studyPlanCriterion.Log.AddParameters(objCommand, isOriginal);
  }
```

```
/// <summary>
/// Validates the current <see cref="StudyPlanning.DAL.Students.StudyPlanCriterion"/> object. If the
/// value of either of the private properties is null a
/// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
/// </summary>
private void Validate()
{
  try
  {
    _Student_StudyPlanCriterion_ID.Validate("Student_StudyPlanCriterion_ID");
    _Student_ID.Validate("Student_ID");
    _StudyPlanCriterion_ID.Validate("StudyPlanCriterion_ID");
    _Log.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }
}

/// <summary>
/// Retrieves a list of study plan identifiers representing the study plans of the student
/// with the specified ID.
/// </summary>
/// <param name="student_ID">Identification of the student for which to retrieve a list of study plans.</param>
/// <returns>An array (possibly empty) of study plan identifiers.</returns>
public static Guid[] GetStudyPlanCriteria(Guid student_ID)
{
  Guid[] criteria = null;

  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  SqlCommand objCommand = new SqlCommand("Student_StudyPlanCriterion_GetStudyPlanCriteria", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  SqlParameter objParam;

  objParam = objCommand.Parameters.Add("@Student_ID", SqlDbType.UniqueIdenti↓er, 16);
  objParam.Value = student_ID;

  DataSet objDataSet = new DataSet("Result");
  SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

  try
  {
    objConnection.Open();
    objAdap.Fill(objDataSet, "Result");
    objConnection.Close();

    int numRows = objDataSet.Tables["Result"].Rows.Count;
    criteria = new Guid[numRows];

    for(int i=0; i<numRows; i++)
    {
      criteria[i] = (Guid)objDataSet.Tables["Result"].Rows[i]["StudyPlanCriterion_ID"];
    }
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
  return criteria;
}

/// <summary>
/// Retrives the ID of the table row containing the specified study plan criterion ID.
/// </summary>
/// <param name="studyPlanCriterion_ID">The ID of the study plan criterion for which to get the row ID.</param>
/// <returns>A <see cref="System.Guid"/> being the ID of the row.</returns>
public static Guid GetID(Guid studyPlanCriterion_ID)
{
  Guid studentStudyPlanCriterion_ID = Guid.Empty;

  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  SqlCommand objCommand = new SqlCommand("Student_StudyPlanCriterion_GetID", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  SqlParameter objParam;

  objParam = objCommand.Parameters.Add("@StudyPlanCriterion_ID", SqlDbType.UniqueIdenti↓er, 16);
  objParam.Value = studyPlanCriterion_ID;

  objParam = objCommand.Parameters.Add("@Student_StudyPlanCriterion_ID", SqlDbType.UniqueIdenti↓er, 16);
  objParam.Direction = ParameterDirection.Output;

  try
  {
    objConnection.Open();
    objCommand.ExecuteNonQuery();
    studentStudyPlanCriterion_ID = (Guid)objCommand.Parameters["@Student_StudyPlanCriterion_ID"].Value;
    objConnection.Close();
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
```

```
      return studentStudyPlanCriterion_ID;
    }

    #endregion //Methods
  }
}
```

1.23 StudyPlanCriteria

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.StudyPlanCriteria
{
  /// <summary>
  /// Represents a studyPlanCriterion.
  /// </summary>
  public class StudyPlanCriterion : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalGuid _StudyPlanCriterion_ID = new DalGuid(false);
    private DalString _Name = new DalString(false);
    private DalBool _AllowPointBlockCourses = new DalBool(false);
    private DalBool _RecommendedPlacementCogent = new DalBool(false);
    private DalBool _AllowForTechnicalPrerequisites = new DalBool(false);
    private DalInt _TechnicalPackageVersion_ID = new DalInt(true);
    private DalInt _TechnicalLineVersion_ID = new DalInt(true);
    private DalInt _TechnicalFieldVersion_ID = new DalInt(true);
    private DalInt _SpecializationVersion_ID = new DalInt(true);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.StudyPlanCriteria.StudyPlanCriterion"/>
    /// class.
    /// </summary>
    public StudyPlanCriterion() {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the ID of the studyPlanCriterion.
    /// </summary>
    /// <value></value>
    public DalGuid StudyPlanCriterion_ID
    {
      get { return _StudyPlanCriterion_ID; }
    }

    /// <summary>
    /// Gets the name of the studyPlanCriterion.
    /// </summary>
    public DalString Name
    {
      get { return _Name; }
    }

    /// <summary>
    /// Gets the allowpointblockcourses property of the studyPlanCriterion.
    /// </summary>
    public DalBool AllowPointBlockCourses
    {
      get { return _AllowPointBlockCourses; }
    }

    /// <summary>
    /// Gets the recommendedplacementcogent property of the studyPlanCriterion.
    /// </summary>
    public DalBool RecommendedPlacementCogent
    {
      get { return _RecommendedPlacementCogent; }
    }

    /// <summary>
    /// Gets the allowfortechnicalprerequisites property of the studyPlanCriterion.
    /// </summary>
    public DalBool AllowForTechnicalPrerequisites
    {
      get { return _AllowForTechnicalPrerequisites; }
    }

    /// <summary>
    /// Gets the technicalpackageversion_id of the studyPlanCriterion.
```

```
/// </summary>
public DalInt TechnicalPackageVersion_ID
{
  get { return _TechnicalPackageVersion_ID; }
}

/// <summary>
/// Gets the technicallineversion_id of the studyPlanCriterion.
/// </summary>
public DalInt TechnicalLineVersion_ID
{
  get { return _TechnicalLineVersion_ID; }
}

/// <summary>
/// Gets the technicalfieldversion_id of the studyPlanCriterion.
/// </summary>
public DalInt TechnicalFieldVersion_ID
{
  get { return _TechnicalFieldVersion_ID; }
}

/// <summary>
/// Gets the specializationversion_id of the studyPlanCriterion.
/// </summary>
public DalInt SpecializationVersion_ID
{
  get { return _SpecializationVersion_ID; }
}

/// <summary>
/// Gets the data log of the course.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Public Methods

#region Create Methods

/// <summary>
/// Creates a studyPlanCriterion in the database using the values of the properties
/// of the current instance.
/// </summary>
public void Create()
{
  try
  {
    _StudyPlanCriterion_ID.Validate("StudyPlanCriterion_ID");
    _AllowPointBlockCourses.Validate("AllowPointBlockCourses");
    _AllowForTechnicalPrerequisites.Validate("AllowForTechnicalPrerequisites");
    _RecommendedPlacementCogent.Validate("RecommendedPlacementCogent");
    _Name.Validate("Name");
    _TechnicalPackageVersion_ID.Validate("TechnicalPackageVersion_ID");
    _TechnicalLineVersion_ID.Validate("TechnicalLineVersion_ID");
    _TechnicalFieldVersion_ID.Validate("TechnicalFieldVersion_ID");
    _SpecializationVersion_ID.Validate("SpecializationVersion_ID");
    _Log.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }

  SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_Insert", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  SqlParameter objParam;

  objParam = objCommand.Parameters.Add("@StudyPlanCriterion_ID", SqlDbType.UniqueIdenti↓er);
  objParam.Value = _StudyPlanCriterion_ID.Value;

  objParam = objCommand.Parameters.Add("@Name", SqlDbType.VarChar);
  objParam.Value = _Name.Value;

  objParam = objCommand.Parameters.Add("@AllowPointBlockCourses", SqlDbType.Bit);
  objParam.Value = _AllowPointBlockCourses.Value;

  objParam = objCommand.Parameters.Add("@AllowForTechnicalPrerequisites", SqlDbType.Bit);
  objParam.Value = _AllowForTechnicalPrerequisites.Value;

  objParam = objCommand.Parameters.Add("@RecommendedPlacementCogent", SqlDbType.Bit);
  objParam.Value = _RecommendedPlacementCogent.Value;

  objParam = objCommand.Parameters.Add("@TechnicalPackageVersion_ID", SqlDbType.Int);
  if (_TechnicalPackageVersion_ID.IsNull)
    objParam.Value = System.DBNull.Value;
  else
    objParam.Value = _TechnicalPackageVersion_ID.Value;

  objParam = objCommand.Parameters.Add("@TechnicalLineVersion_ID", SqlDbType.Int);
  if (_TechnicalLineVersion_ID.IsNull)
    objParam.Value = System.DBNull.Value;
  else
    objParam.Value = _TechnicalLineVersion_ID.Value;
```

```
    objParam = objCommand.Parameters.Add("@TechnicalFieldVersion_ID", SqlDbType.Int);
    if (_TechnicalFieldVersion_ID.IsNull)
      objParam.Value = System.DBNull.Value;
    else
      objParam.Value = _TechnicalFieldVersion_ID.Value;

    objParam = objCommand.Parameters.Add("@SpecializationVersion_ID", SqlDbType.Int);
    if (_SpecializationVersion_ID.IsNull)
      objParam.Value = System.DBNull.Value;
    else
      objParam.Value = _SpecializationVersion_ID.Value;

    objCommand = _Log.AddParameters(objCommand, false);

    DataSet objDataSet = new DataSet("StudyPlanCriterion");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
      objConnection.Open();
      objAdap.Fill(objDataSet, "StudyPlanCriterion");
      objConnection.Close();
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
  }

  #endregion

  #region Retrieve Methods

  /// <summary>
  /// Retrieves the studyPlanCriterion from the database using the value of the
  /// StudyPlanCriterion_ID property of the current instance
  /// </summary>
  public void Retrieve()
  {
    DataRow objDataRow;

    try
    {
      _StudyPlanCriterion_ID.Validate("StudyPlanCriterion_ID");
    }
    catch (DalException e)
    {
      throw e;
    }

    try
    {
      objDataRow = RetrieveExecute(_StudyPlanCriterion_ID.Value, objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }
    this._StudyPlanCriterion_ID.Value      = (Guid)objDataRow["StudyPlanCriterion_ID"];
    this._Name.Value               = (string)objDataRow["Name"];
    this._AllowPointBlockCourses.Value     = (bool)objDataRow["AllowPointBlockCourses"];
    this._AllowForTechnicalPrerequisites.Value = (bool)objDataRow["AllowForTechnicalPrerequisites"];
    this._RecommendedPlacementCogent.Value   = (bool)objDataRow["RecommendedPlacementCogent"];

    if (objDataRow["TechnicalPackageVersion_ID"].Equals(System.DBNull.Value))
      this._TechnicalPackageVersion_ID.IsNull = true;
    else
      this._TechnicalPackageVersion_ID.Value = (int)objDataRow["TechnicalPackageVersion_ID"];

    if (objDataRow["TechnicalLineVersion_ID"].Equals(System.DBNull.Value))
      this._TechnicalLineVersion_ID.IsNull = true;
    else
      this._TechnicalLineVersion_ID.Value = (int)objDataRow["TechnicalLineVersion_ID"];

    if (objDataRow["TechnicalFieldVersion_ID"].Equals(System.DBNull.Value))
      this._TechnicalFieldVersion_ID.IsNull = true;
    else
      this._TechnicalFieldVersion_ID.Value = (int)objDataRow["TechnicalFieldVersion_ID"];

    if (objDataRow["SpecializationVersion_ID"].Equals(System.DBNull.Value))
      this._SpecializationVersion_ID.IsNull = true;
    else
      this._SpecializationVersion_ID.Value = (int)objDataRow["SpecializationVersion_ID"];

    this._Log.SetValues(objDataRow);
  }

  /// <summary>
  /// Retrieves the studyPlanCriterion with the specified identifier from the database.
  /// </summary>
  /// <param name="studyPlanCriterion_ID">The globally unique identifier of the studyPlanCriterion.</param>
  public static StudyPlanCriterion Retrieve(Guid studyPlanCriterion_ID)
  {
    DataRow objDataRow;
    string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
    SqlConnection objConnection = new SqlConnection(connString);

    StudyPlanCriterion objStudyPlanCriterion = new StudyPlanCriterion();
```

```
    try
    {
      objDataRow = RetrieveExecute(studyPlanCriterion_ID, objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }
    objStudyPlanCriterion._StudyPlanCriterion_ID.Value = (Guid)objDataRow["StudyPlanCriterion_ID"];
    objStudyPlanCriterion._Name.Value            = (string)objDataRow["Name"];
    objStudyPlanCriterion._AllowPointBlockCourses.Value      = (bool)objDataRow["AllowPointBlockCourses"];
    objStudyPlanCriterion._AllowForTechnicalPrerequisites.Value = (bool)objDataRow["AllowForTechnicalPrerequisites"];
    objStudyPlanCriterion._RecommendedPlacementCogent.Value    = (bool)objDataRow["RecommendedPlacementCogent"];

    if (objDataRow["TechnicalPackageVersion_ID"].Equals(System.DBNull.Value))
      objStudyPlanCriterion._TechnicalPackageVersion_ID.IsNull = true;
    else
      objStudyPlanCriterion._TechnicalPackageVersion_ID.Value = (int)objDataRow["TechnicalPackageVersion_ID"];

    if (objDataRow["TechnicalLineVersion_ID"].Equals(System.DBNull.Value))
      objStudyPlanCriterion._TechnicalLineVersion_ID.IsNull = true;
    else
      objStudyPlanCriterion._TechnicalLineVersion_ID.Value = (int)objDataRow["TechnicalLineVersion_ID"];

    if (objDataRow["TechnicalFieldVersion_ID"].Equals(System.DBNull.Value))
      objStudyPlanCriterion._TechnicalFieldVersion_ID.IsNull = true;
    else
      objStudyPlanCriterion._TechnicalFieldVersion_ID.Value = (int)objDataRow["TechnicalFieldVersion_ID"];

    if (objDataRow["SpecializationVersion_ID"].Equals(System.DBNull.Value))
      objStudyPlanCriterion._SpecializationVersion_ID.IsNull = true;
    else
      objStudyPlanCriterion._SpecializationVersion_ID.Value = (int)objDataRow["SpecializationVersion_ID"];

    objStudyPlanCriterion._Log.SetValues(objDataRow);
    return objStudyPlanCriterion;
  }

  private static DataRow RetrieveExecute(
    Guid studyPlanCriterion_ID,
    SqlConnection sqlConnection)
  {
    SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_Select", sqlConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam;

    objParam = objCommand.Parameters.Add("@StudyPlanCriterion_ID", SqlDbType.UniqueIdenti↓er);
    objParam.Value = studyPlanCriterion_ID;

    DataSet objDataSet = new DataSet("StudyPlanCriterion");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
      sqlConnection.Open();
      objAdap.Fill(objDataSet, "StudyPlanCriterion");
      sqlConnection.Close();
      DataRow objDataRow = objDataSet.Tables["StudyPlanCriterion"].Rows[0];
      return objDataRow;
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
  }

  #endregion //Retrieve Methods

  #region Update Methods

  /// <summary>
  /// Updates the current studyPlanCriterion in the database using the original
  /// studyPlanCriterion to resolve possible concurrency issues.
  /// </summary>
  /// <param name="originalStudyPlanCriterion">The studyPlanCriterion <see cref="StudyPlanning.DAL.StudyPlanCriteria.StudyPlanCriterion">
    StudyPlanCriterion</see> object.</param>
  /// <returns><strong>true</strong> if the update was succesfully executed; otherwise, <strong>false</strong>.</returns>
  public bool Update(StudyPlanCriterion originalStudyPlanCriterion)
  {
    bool blnResult = false;

    try
    {
      _StudyPlanCriterion_ID.Validate("StudyPlanCriterion_ID");
      _AllowPointBlockCourses.Validate("AllowPointBlockCourses");
      _AllowForTechnicalPrerequisites.Validate("AllowForTechnicalPrerequisites");
      _RecommendedPlacementCogent.Validate("RecommendedPlacementCogent");
      _Name.Validate("Name");
      _TechnicalPackageVersion_ID.Validate("TechnicalPackageVersion_ID");
      _TechnicalLineVersion_ID.Validate("TechnicalLineVersion_ID");
      _TechnicalFieldVersion_ID.Validate("TechnicalFieldVersion_ID");
      _SpecializationVersion_ID.Validate("SpecializationVersion_ID");
      _Log.Validate();
    }
    catch (DalException excp)
    {
      throw excp;
```

```
}

SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_Update", objConnection);
objCommand.CommandType = CommandType.StoredProcedure;

SqlParameter objParam;

objParam = objCommand.Parameters.Add("@StudyPlanCriterion_ID", SqlDbType.UniqueIdenti↓er);
objParam.Value = _StudyPlanCriterion_ID.Value;

objParam = objCommand.Parameters.Add("@Name", SqlDbType.VarChar);
objParam.Value = _Name.Value;

objParam = objCommand.Parameters.Add("@AllowPointBlockCourses", SqlDbType.Bit);
objParam.Value = _AllowPointBlockCourses.Value;

objParam = objCommand.Parameters.Add("@AllowForTechnicalPrerequisites", SqlDbType.Bit);
objParam.Value = _AllowForTechnicalPrerequisites.Value;

objParam = objCommand.Parameters.Add("@RecommendedPlacementCogent", SqlDbType.Bit);
objParam.Value = _RecommendedPlacementCogent.Value;

objParam = objCommand.Parameters.Add("@TechnicalPackageVersion_ID", SqlDbType.Int);
if (_TechnicalPackageVersion_ID.IsNull)
  objParam.Value = System.DBNull.Value;
else
  objParam.Value = _TechnicalPackageVersion_ID.Value;

objParam = objCommand.Parameters.Add("@TechnicalLineVersion_ID", SqlDbType.Int);
if (_TechnicalLineVersion_ID.IsNull)
  objParam.Value = System.DBNull.Value;
else
  objParam.Value = _TechnicalLineVersion_ID.Value;

objParam = objCommand.Parameters.Add("@TechnicalFieldVersion_ID", SqlDbType.Int);
if (_TechnicalFieldVersion_ID.IsNull)
  objParam.Value = System.DBNull.Value;
else
  objParam.Value = _TechnicalFieldVersion_ID.Value;

objParam = objCommand.Parameters.Add("@SpecializationVersion_ID", SqlDbType.Int);
if (_SpecializationVersion_ID.IsNull)
  objParam.Value = System.DBNull.Value;
else
  objParam.Value = _SpecializationVersion_ID.Value;

objCommand = _Log.AddParameters(objCommand, false);

objParam = objCommand.Parameters.Add("@Original_StudyPlanCriterion_ID", SqlDbType.UniqueIdenti↓er);
objParam.Value = originalStudyPlanCriterion._StudyPlanCriterion_ID.Value;

objParam = objCommand.Parameters.Add("@Original_Name", SqlDbType.VarChar);
objParam.Value = originalStudyPlanCriterion._Name.Value;

objParam = objCommand.Parameters.Add("@Original_AllowPointBlockCourses", SqlDbType.Bit);
objParam.Value = originalStudyPlanCriterion._AllowPointBlockCourses.Value;

objParam = objCommand.Parameters.Add("@Original_AllowForTechnicalPrerequisites", SqlDbType.Bit);
objParam.Value = originalStudyPlanCriterion._AllowForTechnicalPrerequisites.Value;

objParam = objCommand.Parameters.Add("@Original_RecommendedPlacementCogent", SqlDbType.Bit);
objParam.Value = originalStudyPlanCriterion._RecommendedPlacementCogent.Value;

objParam = objCommand.Parameters.Add("@Original_TechnicalPackageVersion_ID", SqlDbType.Int);
if (originalStudyPlanCriterion._TechnicalPackageVersion_ID.IsNull)
  objParam.Value = System.DBNull.Value;
else
  objParam.Value = originalStudyPlanCriterion._TechnicalPackageVersion_ID.Value;

objParam = objCommand.Parameters.Add("@Original_TechnicalLineVersion_ID", SqlDbType.Int);
if (originalStudyPlanCriterion._TechnicalLineVersion_ID.IsNull)
  objParam.Value = System.DBNull.Value;
else
  objParam.Value = originalStudyPlanCriterion._TechnicalLineVersion_ID.Value;

objParam = objCommand.Parameters.Add("@Original_TechnicalFieldVersion_ID", SqlDbType.Int);
if (originalStudyPlanCriterion._TechnicalFieldVersion_ID.IsNull)
  objParam.Value = System.DBNull.Value;
else
  objParam.Value = originalStudyPlanCriterion._TechnicalFieldVersion_ID.Value;

objParam = objCommand.Parameters.Add("@Original_SpecializationVersion_ID", SqlDbType.Int);
if (originalStudyPlanCriterion._SpecializationVersion_ID.IsNull)
  objParam.Value = System.DBNull.Value;
else
  objParam.Value = originalStudyPlanCriterion._SpecializationVersion_ID.Value;

objCommand = originalStudyPlanCriterion._Log.AddParameters(objCommand, true);

int rowsA↑ected;

try
{
  objConnection.Open();
  rowsA↑ected = objCommand.ExecuteNonQuery();
  objConnection.Close();

  if (rowsA↑ected > 0)
    blnResult = true;
```

```
      }
      catch(SqlException excp)
      {
        throw new DalException(excp);
      }
      return blnResult;
    }

    #endregion //Update Methods

    #region Delete Methods

    /// <summary>
    /// Deletes the current studyPlanCriterion from the database.
    /// </summary>
    /// <returns><strong>true</strong> if the deletion is succesfully executed; otherwise, <strong>false</strong>.</returns>
    public bool Delete()
    {
      bool blnResult = false;

      try
      {
        _StudyPlanCriterion_ID.Validate("StudyPlanCriterion_ID");
      }
      catch (DalException excp)
      {
        throw excp;
      }

      SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_Delete", objConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      SqlParameter objParam;

      objParam = objCommand.Parameters.Add("@Original_StudyPlanCriterion_ID", SqlDbType.UniqueIdenti↓er);
      objParam.Value = _StudyPlanCriterion_ID.Value;

      objParam = objCommand.Parameters.Add("@Original_Name", SqlDbType.VarChar);
      objParam.Value = _Name.Value;

      objParam = objCommand.Parameters.Add("@Original_AllowPointBlockCourses", SqlDbType.Bit);
      objParam.Value = _AllowPointBlockCourses.Value;

      objParam = objCommand.Parameters.Add("@Original_AllowForTechnicalPrerequisites", SqlDbType.Bit);
      objParam.Value = _AllowForTechnicalPrerequisites.Value;

      objParam = objCommand.Parameters.Add("@Original_RecommendedPlacementCogent", SqlDbType.Bit);
      objParam.Value = _RecommendedPlacementCogent.Value;

      objParam = objCommand.Parameters.Add("@Original_TechnicalPackageVersion_ID", SqlDbType.Int);
      if (_TechnicalPackageVersion_ID.IsNull)
        objParam.Value = System.DBNull.Value;
      else
        objParam.Value = _TechnicalPackageVersion_ID.Value;

      objParam = objCommand.Parameters.Add("@Original_TechnicalLineVersion_ID", SqlDbType.Int);
      if (_TechnicalLineVersion_ID.IsNull)
        objParam.Value = System.DBNull.Value;
      else
        objParam.Value = _TechnicalLineVersion_ID.Value;

      objParam = objCommand.Parameters.Add("@Original_TechnicalFieldVersion_ID", SqlDbType.Int);
      if (_TechnicalFieldVersion_ID.IsNull)
        objParam.Value = System.DBNull.Value;
      else
        objParam.Value = _TechnicalFieldVersion_ID.Value;

      objParam = objCommand.Parameters.Add("@Original_SpecializationVersion_ID", SqlDbType.Int);
      if (_SpecializationVersion_ID.IsNull)
        objParam.Value = System.DBNull.Value;
      else
        objParam.Value = _SpecializationVersion_ID.Value;

      objCommand = _Log.AddParameters(objCommand, true);

      int rowsA↑ected;
      try
      {
        objConnection.Open();
        rowsA↑ected = objCommand.ExecuteNonQuery();
        objConnection.Close();

        if (rowsA↑ected > 0)
          blnResult = true;
      }
      catch(SqlException e)
      {
        throw new DalException(e);
      }

      return blnResult;
    }

    #endregion //Delete Methods

    #endregion //Public Methods

  }
}
```

1.23.1 Course

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.StudyPlanCriteria
{
  /// <summary>
  /// Represents a studyPlanCriterion.
  /// </summary>
  public class Course : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalGuid _StudyPlanCriterion_Course_ID = new DalGuid(false);
    private DalGuid _StudyPlanCriterion_ID = new DalGuid(false);
    private DalGuid _Course_ID = new DalGuid(false);
    private DalBool _AdditionalChoice = new DalBool(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.StudyPlanCriteria.Course"/>
    /// class.
    /// </summary>
    public Course() {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the StudyPlanCriterion_Course_ID.
    /// </summary>
    /// <value></value>
    public DalGuid StudyPlanCriterion_Course_ID
    {
      get { return _StudyPlanCriterion_Course_ID; }
    }

    /// <summary>
    /// Gets the ID of the study plan criterion.
    /// </summary>
    public DalGuid StudyPlanCriterion_ID
    {
      get { return _StudyPlanCriterion_ID; }
    }

    /// <summary>
    /// Gets the course ID.
    /// </summary>
    public DalGuid Course_ID
    {
      get { return _Course_ID; }
    }

    /// <summary>
    /// Gets the additional choice property.
    /// </summary>
    public DalBool AdditionalChoice
    {
      get { return _AdditionalChoice; }
    }

    /// <summary>
    /// Gets the data log of the course.
    /// </summary>
    public DataLog Log
    {
      get { return _Log; }
    }

    #endregion //Public Properties

    #region Public Methods

    #region Create Methods

    /// <summary>
    /// Creates a StudyPlanCriterion_Course object in the database using the values of the properties
    /// of the current instance.
    /// </summary>
    public void Create()
    {
      try
      {
        this.Validate();
      }
      catch (DalException excp)
      {
        throw excp;
```

```
    }

    SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_Course_Insert", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, false);

    try
    {
      objConnection.Open();
      objCommand.ExecuteNonQuery();
      objConnection.Close();
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
  }

  #endregion

  #region Retrieve Methods

  /// <summary>
  /// Retrieves the studyPlanCriterion from the database using the value of the
  /// Course_ID property of the current instance
  /// </summary>
  public void Retrieve()
  {
    try
    {
      this._StudyPlanCriterion_Course_ID.Validate("StudyPlanCriterion_Course_ID");
    }
    catch (DalException e)
    {
      throw e;
    }

    try
    {
      RetrieveExecute(_StudyPlanCriterion_Course_ID.Value, this, objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }
  }

  /// <summary>
  /// Retrieves the StudyPlanCriterion_Course with the specified identifier from the database.
  /// </summary>
  /// <param name="studyPlanCriterion_Course_ID"></param>
  public static Course Retrieve(Guid studyPlanCriterion_Course_ID)
  {
    string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
    SqlConnection objConnection = new SqlConnection(connString);

    Course objCourse = new Course();

    try
    {
      RetrieveExecute(studyPlanCriterion_Course_ID, objCourse, objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }
    return objCourse;
  }

  private static void RetrieveExecute(
    Guid studyPlanCriterion_Course_ID,
    Course course,
    SqlConnection sqlConnection)
  {
    DataRow objDataRow;

    SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_Course_Select", sqlConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam;

    objParam = objCommand.Parameters.Add("@StudyPlanCriterion_Course_ID", SqlDbType.UniqueIdenti↓er);
    objParam.Value = studyPlanCriterion_Course_ID;

    DataSet objDataSet = new DataSet("StudyPlanCriterion_Course");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
      sqlConnection.Open();
      objAdap.Fill(objDataSet, "Course");
      sqlConnection.Close();
      objDataRow = objDataSet.Tables["Course"].Rows[0];
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
```

```
    course._StudyPlanCriterion_Course_ID.Value = (Guid)objDataRow["StudyPlanCriterion_Course_ID"];
    course._StudyPlanCriterion_ID.Value     = (Guid)objDataRow["StudyPlanCriterion_ID"];
    course._Course_ID.Value          = (Guid)objDataRow["Course_ID"];
    course._AdditionalChoice.Value       = (bool)objDataRow["AdditionalChoice"];
    course._Log.SetValues(objDataRow);
  }

  /// <summary>
  /// Retrieves courses that have either been deselected or selected in a study plan criterion.
  /// </summary>
  /// <param name="studyPlanCriterion_ID">Identification of the
  /// <see cref="StudyPlanning.DAL.StudyPlanCriteria.StudyPlanCriterion"/>.</param>
  /// <param name="additionalChoice">When <strong>true</strong> selected courses are retrieved.
  /// When <strong>false</strong> deselected courses are retrieved.</param>
  /// <returns>An array of <see cref="StudyPlanning.DAL.Courses.Course"/> identifications.</returns>
  public static Guid[] GetCourses(Guid studyPlanCriterion_ID, bool additionalChoice)
  {
    string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
    SqlConnection objConnection = new SqlConnection(connString);

    SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_Course_GetCourses", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam = objCommand.Parameters.Add("@StudyPlanCriterion_ID", SqlDbType.UniqueIdenti↓er);
    objParam.Value = studyPlanCriterion_ID;

    objParam = objCommand.Parameters.Add("@AdditionalChoice", SqlDbType.Bit);
    objParam.Value = additionalChoice;

    DataSet objDataSet = new DataSet("Result");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
      objConnection.Open();
      objAdap.Fill(objDataSet, "Result");
      objConnection.Close();
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }

    int intRows = objDataSet.Tables["Result"].Rows.Count;
    Guid[] courses = new Guid[intRows];

    for (int i=0; i < intRows; i++)
    {
      courses[i] = (Guid)objDataSet.Tables["Result"].Rows[i]["Course_ID"];
    }
    return courses;
  }

  #endregion //Retrieve Methods

  #region Update Methods

  /// <summary>
  /// Updates the current StudyPlanCriterion_Course in the database using the original
  /// StudyPlanCriterion_Course to resolve possible concurrency issues.
  /// </summary>
  /// <param name="originalCourse">The <see cref="StudyPlanning.DAL.StudyPlanCriteria.Course"/> object.</param>
  /// <returns><strong>true</strong> if the update was succesfully executed; otherwise, <strong>false</strong>.</returns>
  public bool Update(Course originalCourse)
  {
    bool blnResult = false;

    SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_Course_Update", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, false);
    AddParameters(originalCourse, objCommand, true);

    try
    {
      objConnection.Open();
      int rowsA↑ected = objCommand.ExecuteNonQuery();
      objConnection.Close();

      if (rowsA↑ected > 0)
        blnResult = true;
    }
    catch(SqlException excp)
    {
      throw new DalException(excp);
    }
    return blnResult;
  }

  #endregion //Update Methods

  #region Delete Methods

  /// <summary>
  /// Deletes the current studyPlanCriterion from the database.
  /// </summary>
  /// <returns><strong>true</strong> if the deletion is succesfully executed; otherwise, <strong>false</strong>.</returns>
  public bool Delete()
  {
```

```
  bool blnResult = false;

  try
  {
    _StudyPlanCriterion_Course_ID.Validate("StudyPlanCriterion_Course_ID");
  }
  catch (DalException excp)
  {
    throw excp;
  }

  SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_Course_Delete", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, true);

  try
  {
    objConnection.Open();
    int rowsA↑ected = objCommand.ExecuteNonQuery();
    objConnection.Close();

    if (rowsA↑ected > 0)
      blnResult = true;
  }
  catch(SqlException e)
  {
    throw new DalException(e);
  }
  return blnResult;
}

#endregion //Delete Methods

#endregion //Public Methods

#region Private Methods

/// <summary>
/// Adds the relevant parameters to the specified <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see> object
/// setting the values to ones of the specified <see cref="StudyPlanning.DAL.StudyPlanCriteria.Course"/> object.
/// </summary>
/// <param name="course">
/// The <see cref="StudyPlanning.DAL.StudyPlanCriteria.Course"/> object containing the values that the parameters have to be
/// initialized to.
/// </param>
/// <param name="objCommand">
/// The <see cref="System.Data.SqlClient.SqlCommand"/> object to which parameters should be added.
/// </param>
/// <param name="isOriginal">
/// <strong>true</strong> if the parameters to be added represent original
/// data (for handling concurrent programming issues); otherwise, <strong>false</strong>.</param>
/// <returns>The <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see>
/// object with the relevant parameters added.</returns>
private static void AddParameters(Course course, SqlCommand objCommand, bool isOriginal)
{
  string paramName = "";
  SqlParameter objParam;

  paramName = "StudyPlanCriterion_Course_ID";
  objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er);
  objParam.Value = course._StudyPlanCriterion_Course_ID.Value;
  if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

  paramName = "StudyPlanCriterion_ID";
  objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er);
  objParam.Value = course._StudyPlanCriterion_ID.Value;
  if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

  paramName = "Course_ID";
  objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er);
  objParam.Value = course._Course_ID.Value;
  if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

  paramName = "AdditionalChoice";
  objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.Bit);
  objParam.Value = course._AdditionalChoice.Value;
  if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

  course._Log.AddParameters(objCommand, isOriginal);
}

/// <summary>
/// Validates the current <see cref="StudyPlanning.DAL.StudyPlanCriteria.Course"/> object. If the
/// value of either of the private properties is null a
/// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
/// </summary>
private void Validate()
{
  try
  {
    _StudyPlanCriterion_Course_ID.Validate("StudyPlanCriterion_Course_ID");
    _StudyPlanCriterion_ID.Validate("StudyPlanCriterion_ID");
    _Course_ID.Validate("Course_ID");
    _AdditionalChoice.Validate("AdditionalChoice");
```

```
        _Log.Validate();
      }
      catch (DalException excp)
      {
        throw excp;
      }
    }
    #endregion //Private Methods
  }
}
```

1.23.2 CoursePeriod

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.StudyPlanCriteria
{
  /// <summary>
  /// Represents a studyPlanCriterion.
  /// </summary>
  public class CoursePeriod : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalGuid _StudyPlanCriterion_CoursePeriod_ID = new DalGuid(false);
    private DalGuid _StudyPlanCriterion_ID = new DalGuid(false);
    private DalGuid _Period_ID = new DalGuid(false);
    private DalGuid _Course_ID = new DalGuid(false);
    private DalBool _AdditionalChoice = new DalBool(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.StudyPlanCriteria.CoursePeriod"/>
    /// class.
    /// </summary>
    public CoursePeriod() {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the StudyPlanCriterion_CoursePeriod_ID.
    /// </summary>
    /// <value></value>
    public DalGuid StudyPlanCriterion_CoursePeriod_ID
    {
      get { return _StudyPlanCriterion_CoursePeriod_ID; }
    }

    /// <summary>
    /// Gets the ID of the study plan criterion.
    /// </summary>
    public DalGuid StudyPlanCriterion_ID
    {
      get { return _StudyPlanCriterion_ID; }
    }

    /// <summary>
    /// Gets the period ID.
    /// </summary>
    public DalGuid Period_ID
    {
      get { return _Period_ID; }
    }

    /// <summary>
    /// Gets the course ID.
    /// </summary>
    public DalGuid Course_ID
    {
      get { return _Course_ID; }
    }

    /// <summary>
    /// Gets the additional choice property.
    /// </summary>
    public DalBool AdditionalChoice
    {
      get { return _AdditionalChoice; }
    }

    /// <summary>
    /// Gets the data log of the course.
    /// </summary>
    public DataLog Log
    {
```

```
    get { return _Log; }
  }

  #endregion //Public Properties

  #region Public Methods

  #region Create Methods

  /// <summary>
  /// Creates a StudyPlanCriterion_CoursePeriod object in the database using the values of the properties
  /// of the current instance.
  /// </summary>
  public void Create()
  {
    try
    {
      this.Validate();
    }
    catch (DalException excp)
    {
      throw excp;
    }

    SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_CoursePeriod_Insert", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, false);

    try
    {
      objConnection.Open();
      objCommand.ExecuteNonQuery();
      objConnection.Close();
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
  }

  #endregion

  #region Retrieve Methods

  /// <summary>
  /// Retrieves the studyPlanCriterion from the database using the value of the
  /// CoursePeriod_ID property of the current instance
  /// </summary>
  public void Retrieve()
  {
    try
    {
      this._StudyPlanCriterion_CoursePeriod_ID.Validate("StudyPlanCriterion_CoursePeriod_ID");
    }
    catch (DalException e)
    {
      throw e;
    }

    try
    {
      RetrieveExecute(_StudyPlanCriterion_CoursePeriod_ID.Value, this, objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }
  }

  /// <summary>
  /// Retrieves the StudyPlanCriterion_CoursePeriod with the specified identifier from the database.
  /// </summary>
  /// <param name="studyPlanCriterion_CoursePeriod_ID"></param>
  public static CoursePeriod Retrieve(Guid studyPlanCriterion_CoursePeriod_ID)
  {
    string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
    SqlConnection objConnection = new SqlConnection(connString);

    CoursePeriod objCoursePeriod = new CoursePeriod();

    try
    {
      RetrieveExecute(studyPlanCriterion_CoursePeriod_ID, objCoursePeriod, objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }
    return objCoursePeriod;
  }

  private static void RetrieveExecute(
    Guid studyPlanCriterion_CoursePeriod_ID,
    CoursePeriod course,
    SqlConnection sqlConnection)
  {
    DataRow objDataRow;
```

```
    SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_CoursePeriod_Select", sqlConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam;

    objParam = objCommand.Parameters.Add("@StudyPlanCriterion_CoursePeriod_ID", SqlDbType.UniqueIdenti↓er);
    objParam.Value = studyPlanCriterion_CoursePeriod_ID;

    DataSet objDataSet = new DataSet("StudyPlanCriterion_CoursePeriod");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
      sqlConnection.Open();
      objAdap.Fill(objDataSet, "CoursePeriod");
      sqlConnection.Close();
      objDataRow = objDataSet.Tables["CoursePeriod"].Rows[0];
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
    course._StudyPlanCriterion_CoursePeriod_ID.Value = (Guid)objDataRow["StudyPlanCriterion_CoursePeriod_ID"];
    course._StudyPlanCriterion_ID.Value        = (Guid)objDataRow["StudyPlanCriterion_ID"];
    course._Period_ID.Value                = (Guid)objDataRow["Period_ID"];
    course._Course_ID.Value                = (Guid)objDataRow["Course_ID"];
    course._AdditionalChoice.Value          = (bool)objDataRow["AdditionalChoice"];
    course._Log.SetValues(objDataRow);
  }

  /// <summary>
  /// Retrieves courses that have either been deselected or selected
  /// for a given period in a study plan criterion.
  /// </summary>
  /// <param name="studyPlanCriterion_ID">Identification of the
  /// <see cref="StudyPlanning.DAL.StudyPlanCriteria.StudyPlanCriterion"/>.</param>
  /// <param name="period_ID">Identification of the
  /// <see cref="StudyPlanning.DAL.Period"/>.</param>
  /// <param name="additionalChoice">When <strong>true</strong> selected courses are retrieved.
  /// When <strong>false</strong> deselected courses are retrieved.</param>
  /// <returns>An array of <see cref="StudyPlanning.DAL.Courses.Course"/> identifications.</returns>
  public static Guid[] GetCourses(Guid studyPlanCriterion_ID, Guid period_ID, bool additionalChoice)
  {
    string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
    SqlConnection objConnection = new SqlConnection(connString);

    SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_CoursePeriod_GetCourses", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam = objCommand.Parameters.Add("@StudyPlanCriterion_ID", SqlDbType.UniqueIdenti↓er);
    objParam.Value = studyPlanCriterion_ID;

    objParam = objCommand.Parameters.Add("@Period_ID", SqlDbType.UniqueIdenti↓er);
    objParam.Value = period_ID;

    objParam = objCommand.Parameters.Add("@AdditionalChoice", SqlDbType.Bit);
    objParam.Value = additionalChoice;

    DataSet objDataSet = new DataSet("Result");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
      objConnection.Open();
      objAdap.Fill(objDataSet, "Result");
      objConnection.Close();
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }

    int intRows = objDataSet.Tables["Result"].Rows.Count;
    Guid[] courses = new Guid[intRows];

    for (int i=0; i < intRows; i++)
    {
      courses[i] = (Guid)objDataSet.Tables["Result"].Rows[i]["Course_ID"];
    }
    return courses;
  }

  #endregion //Retrieve Methods

  #region Update Methods

  /// <summary>
  /// Updates the current StudyPlanCriterion_CoursePeriod in the database using the original
  /// StudyPlanCriterion_CoursePeriod to resolve possible concurrency issues.
  /// </summary>
  /// <param name="originalCoursePeriod">The <see cref="StudyPlanning.DAL.StudyPlanCriteria.CoursePeriod"/> object.</param>
  /// <returns><strong>true</strong> if the update was succesfully executed; otherwise, <strong>false</strong>.</returns>
  public bool Update(CoursePeriod originalCoursePeriod)
  {
    bool blnResult = false;

    try
    {
      this.Validate();
```

```
    }
    catch (DalException excp)
    {
      throw excp;
    }

    SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_CoursePeriod_Update", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, false);
    AddParameters(originalCoursePeriod, objCommand, true);

    try
    {
      objConnection.Open();
      int rowsA↑ected = objCommand.ExecuteNonQuery();
      objConnection.Close();

      if (rowsA↑ected > 0)
        blnResult = true;
    }
    catch(SqlException excp)
    {
      throw new DalException(excp);
    }
    return blnResult;
  }

  #endregion //Update Methods

  #region Delete Methods

  /// <summary>
  /// Deletes the current studyPlanCriterion from the database.
  /// </summary>
  /// <returns><strong>true</strong> if the deletion is succesfully executed; otherwise, <strong>false</strong>.</returns>
  public bool Delete()
  {
    bool blnResult = false;

    try
    {
      _StudyPlanCriterion_CoursePeriod_ID.Validate("StudyPlanCriterion_CoursePeriod_ID");
    }
    catch (DalException excp)
    {
      throw excp;
    }

    SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_CoursePeriod_Delete", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, true);

    try
    {
      objConnection.Open();
      int rowsA↑ected = objCommand.ExecuteNonQuery();
      objConnection.Close();

      if (rowsA↑ected > 0)
        blnResult = true;
    }
    catch(SqlException e)
    {
      throw new DalException(e);
    }
    return blnResult;
  }

  #endregion //Delete Methods

  #endregion //Public Methods

  #region Private Methods

  /// <summary>
  /// Adds the relevant parameters to the specified <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see> object
  /// setting the values to ones of the specified <see cref="StudyPlanning.DAL.StudyPlanCriteria.CoursePeriod"/> object.
  /// </summary>
  /// <param name="coursePeriod">
  /// The <see cref="StudyPlanning.DAL.StudyPlanCriteria.CoursePeriod"/> object containing the values that the parameters have to be
  /// initialized to.
  /// </param>
  /// <param name="objCommand">
  /// The <see cref="System.Data.SqlClient.SqlCommand"/> object to which parameters should be added.
  /// </param>
  /// <param name="isOriginal">
  /// <strong>true</strong> if the parameters to be added represent original
  /// data (for handling concurrent programming issues); otherwise, <strong>false</strong>.</param>
  /// <returns>The <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see>
  /// object with the relevant parameters added.</returns>
  private static void AddParameters(CoursePeriod coursePeriod, SqlCommand objCommand, bool isOriginal)
  {
    string paramName = "";
    SqlParameter objParam;

    paramName = "StudyPlanCriterion_CoursePeriod_ID";
    objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er);
```

```
      objParam.Value = coursePeriod._StudyPlanCriterion_CoursePeriod_ID.Value;
      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      paramName = "StudyPlanCriterion_ID";
      objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er);
      objParam.Value = coursePeriod._StudyPlanCriterion_ID.Value;
      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      paramName = "Period_ID";
      objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er);
      objParam.Value = coursePeriod._Period_ID.Value;
      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      paramName = "Course_ID";
      objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er);
      objParam.Value = coursePeriod._Course_ID.Value;
      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      paramName = "AdditionalChoice";
      objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.Bit);
      objParam.Value = coursePeriod._AdditionalChoice.Value;
      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      coursePeriod._Log.AddParameters(objCommand, isOriginal);
    }

    /// <summary>
    /// Validates the current <see cref="StudyPlanning.DAL.StudyPlanCriteria.CoursePeriod"/> object. If the
    /// value of either of the private properties is null a
    /// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
    /// </summary>
    private void Validate()
    {
      try
      {
        _StudyPlanCriterion_CoursePeriod_ID.Validate("StudyPlanCriterion_CoursePeriod_ID");
        _StudyPlanCriterion_ID.Validate("StudyPlanCriterion_ID");
        _Period_ID.Validate("Period_ID");
        _Course_ID.Validate("Course_ID");
        _AdditionalChoice.Validate("AdditionalChoice");
        _Log.Validate();
      }
      catch (DalException excp)
      {
        throw excp;
      }
    }
    #endregion //Private Methods
  }
}
```

1.23.3 Keyword

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.StudyPlanCriteria
{
  /// <summary>
  /// Represents a studyPlanCriterion.
  /// </summary>
  public class Keyword : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalGuid _StudyPlanCriterion_Keyword_ID = new DalGuid(false);
    private DalGuid _StudyPlanCriterion_ID = new DalGuid(false);
    private DalGuid _Keyword_ID = new DalGuid(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.StudyPlanCriteria.Keyword"/>
    /// class.
    /// </summary>
    public Keyword() {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the StudyPlanCriterion_Keyword_ID.
```

```
/// </summary>
/// <value></value>
public DalGuid StudyPlanCriterion_Keyword_ID
{
  get { return _StudyPlanCriterion_Keyword_ID; }
}

/// <summary>
/// Gets the ID of the study plan criterion.
/// </summary>
public DalGuid StudyPlanCriterion_ID
{
  get { return _StudyPlanCriterion_ID; }
}

/// <summary>
/// Gets the keyword ID.
/// </summary>
public DalGuid Keyword_ID
{
  get { return _Keyword_ID; }
}

/// <summary>
/// Gets the data log of the course.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Public Methods

#region Create Methods

/// <summary>
/// Creates a StudyPlanCriterion_Keyword object in the database using the values of the properties
/// of the current instance.
/// </summary>
public void Create()
{
  try
  {
    this.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }

  SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_Keyword_Insert", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, false);

  try
  {
    objConnection.Open();
    objCommand.ExecuteNonQuery();
    objConnection.Close();
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
}

#endregion

#region Retrieve Methods

/// <summary>
/// Retrieves the StudyPlanCriterion_Keyword from the database using the value of the
/// StudyPlanCriterion_Keyword_ID property of the current instance
/// </summary>
public void Retrieve()
{
  try
  {
    this._StudyPlanCriterion_Keyword_ID.Validate("StudyPlanCriterion_Keyword_ID");
  }
  catch (DalException e)
  {
    throw e;
  }

  try
  {
    RetrieveExecute(_StudyPlanCriterion_Keyword_ID.Value, this, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
}

/// <summary>
```

```
/// Retrieves the StudyPlanCriterion_Keyword with the specified identifier from the database.
/// </summary>
/// <param name="studyPlanCriterion_Keyword_ID"></param>
public static Keyword Retrieve(Guid studyPlanCriterion_Keyword_ID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  Keyword objKeyword = new Keyword();

  try
  {
    RetrieveExecute(studyPlanCriterion_Keyword_ID, objKeyword, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
  return objKeyword;
}

private static void RetrieveExecute(
  Guid studyPlanCriterion_Keyword_ID,
  Keyword keyword,
  SqlConnection sqlConnection)
{
  DataRow objDataRow;

  SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_Keyword_Select", sqlConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  SqlParameter objParam;

  objParam = objCommand.Parameters.Add("@StudyPlanCriterion_Keyword_ID", SqlDbType.UniqueIdenti↓er);
  objParam.Value = studyPlanCriterion_Keyword_ID;

  DataSet objDataSet = new DataSet("StudyPlanCriterion_Keyword");
  SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

  try
  {
    sqlConnection.Open();
    objAdap.Fill(objDataSet, "Keyword");
    sqlConnection.Close();
    objDataRow = objDataSet.Tables["Keyword"].Rows[0];
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
  keyword._StudyPlanCriterion_Keyword_ID.Value = (Guid)objDataRow["StudyPlanCriterion_Keyword_ID"];
  keyword._StudyPlanCriterion_ID.Value     = (Guid)objDataRow["StudyPlanCriterion_ID"];
  keyword._Keyword_ID.Value          = (Guid)objDataRow["Keyword_ID"];
  keyword._Log.SetValues(objDataRow);
}

/// <summary>
/// Retrieves a list of the keywords associated with a study plan criterion.
/// </summary>
/// <param name="studyPlanCriterion_ID">Identification of the
/// <see cref="StudyPlanning.DAL.StudyPlanCriteria.StudyPlanCriterion"/>.</param>
/// <returns>An array of <see cref="StudyPlanning.DAL.Keyword"/> identifications.</returns>
public static Guid[] GetKeywords(Guid studyPlanCriterion_ID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_Keyword_GetKeywords", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  SqlParameter objParam = objCommand.Parameters.Add("@StudyPlanCriterion_ID", SqlDbType.UniqueIdenti↓er);
  objParam.Value = studyPlanCriterion_ID;

  DataSet objDataSet = new DataSet("Keywords");
  SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

  try
  {
    objConnection.Open();
    objAdap.Fill(objDataSet, "Keywords");
    objConnection.Close();
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
  int intRows = objDataSet.Tables["Keywords"].Rows.Count;
  Guid[] keywords = new Guid[intRows];

  for(int i=0; i < intRows; i++)
  {
    keywords[i] = (Guid)objDataSet.Tables["Keywords"].Rows[i]["Keyword_ID"];
  }
  return keywords;
}

#endregion //Retrieve Methods

#region Update Methods
```

```
/// <summary>
/// Updates the current StudyPlanCriterion_Keyword in the database using the original
/// StudyPlanCriterion_Keyword to resolve possible concurrency issues.
/// </summary>
/// <param name="originalKeyword">The <see cref="StudyPlanning.DAL.StudyPlanCriteria.Keyword"/> object.</param>
/// <returns><strong>true</strong> if the update was succesfully executed; otherwise, <strong>false</strong>.</returns>
public bool Update(Keyword originalKeyword)
{
  bool blnResult = false;

  try
  {
    this.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }

  SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_Keyword_Update", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, false);
  AddParameters(originalKeyword, objCommand, true);

  try
  {
    objConnection.Open();
    int rowsA↑ected = objCommand.ExecuteNonQuery();
    objConnection.Close();

    if (rowsA↑ected > 0)
      blnResult = true;
  }
  catch(SqlException excp)
  {
    throw new DalException(excp);
  }
  return blnResult;
}

#endregion //Update Methods

#region Delete Methods

/// <summary>
/// Deletes the current StudyPlanCriterion_Keyword from the database.
/// </summary>
/// <returns><strong>true</strong> if the deletion is succesfully executed; otherwise, <strong>false</strong>.</returns>
public bool Delete()
{
  bool blnResult = false;

  try
  {
    _StudyPlanCriterion_Keyword_ID.Validate("StudyPlanCriterion_Keyword_ID");
  }
  catch (DalException excp)
  {
    throw excp;
  }

  SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_Keyword_Delete", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, true);

  try
  {
    objConnection.Open();
    int rowsA↑ected = objCommand.ExecuteNonQuery();
    objConnection.Close();

    if (rowsA↑ected > 0)
      blnResult = true;
  }
  catch(SqlException e)
  {
    throw new DalException(e);
  }
  return blnResult;
}

#endregion //Delete Methods

#endregion //Public Methods

#region Private Methods

/// <summary>
/// Adds the relevant parameters to the specified <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see> object
/// setting the values to ones of the specified <see cref="StudyPlanning.DAL.StudyPlanCriteria.Keyword"/> object.
/// </summary>
/// <param name="keyword">
/// The <see cref="StudyPlanning.DAL.StudyPlanCriteria.Keyword"/> object containing the values that the parameters have to be
/// initialized to.
/// </param>
/// <param name="objCommand">
```

```
    /// The <see cref="System.Data.SqlClient.SqlCommand"/> object to which parameters should be added.
    /// </param>
    /// <param name="isOriginal">
    /// <strong>true</strong> if the parameters to be added represent original
    /// data (for handling concurrent programming issues); otherwise, <strong>false</strong>.</param>
    /// <returns>The <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see>
    /// object with the relevant parameters added.</returns>
    private static void AddParameters(Keyword keyword, SqlCommand objCommand, bool isOriginal)
    {
      string paramName = "";
      SqlParameter objParam;

      paramName = "StudyPlanCriterion_Keyword_ID";
      objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er);
      objParam.Value = keyword._StudyPlanCriterion_Keyword_ID.Value;
      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      paramName = "StudyPlanCriterion_ID";
      objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er);
      objParam.Value = keyword._StudyPlanCriterion_ID.Value;
      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      paramName = "Keyword_ID";
      objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er);
      objParam.Value = keyword._Keyword_ID.Value;
      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      keyword._Log.AddParameters(objCommand, isOriginal);
    }

    /// <summary>
    /// Validates the current <see cref="StudyPlanning.DAL.StudyPlanCriteria.Keyword"/> object. If the
    /// value of either of the private properties is null a
    /// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
    /// </summary>
    private void Validate()
    {
      try
      {
        _StudyPlanCriterion_Keyword_ID.Validate("StudyPlanCriterion_Keyword_ID");
        _StudyPlanCriterion_ID.Validate("StudyPlanCriterion_ID");
        _Keyword_ID.Validate("Keyword_ID");
        _Log.Validate();
      }
      catch (DalException excp)
      {
        throw excp;
      }
    }
    #endregion //Private Methods
  }
}
```

1.23.4 Language

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.StudyPlanCriteria
{
  /// <summary>
  /// Represents a studyPlanCriterion.
  /// </summary>
  public class Language : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalGuid _StudyPlanCriterion_Language_ID = new DalGuid(false);
    private DalGuid _StudyPlanCriterion_ID = new DalGuid(false);
    private DalString _Language_ID = new DalString(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.StudyPlanCriteria.Language"/>
    /// class.
    /// </summary>
    public Language() {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the StudyPlanCriterion_Language_ID.
```

```
/// </summary>
/// <value></value>
public DalGuid StudyPlanCriterion_Language_ID
{
  get { return _StudyPlanCriterion_Language_ID; }
}

/// <summary>
/// Gets the ID of the study plan criterion.
/// </summary>
public DalGuid StudyPlanCriterion_ID
{
  get { return _StudyPlanCriterion_ID; }
}

/// <summary>
/// Gets the language ID.
/// </summary>
public DalString Language_ID
{
  get { return _Language_ID; }
}

/// <summary>
/// Gets the data log of the course.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Public Methods

#region Create Methods

/// <summary>
/// Creates a StudyPlanCriterion_Language object in the database using the values of the properties
/// of the current instance.
/// </summary>
public void Create()
{
  try
  {
    this.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }

  SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_Language_Insert", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, false);

  try
  {
    objConnection.Open();
    objCommand.ExecuteNonQuery();
    objConnection.Close();
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
}

#endregion

#region Retrieve Methods

/// <summary>
/// Retrieves the StudyPlanCriterion_Language from the database using the value of the
/// StudyPlanCriterion_Language_ID property of the current instance
/// </summary>
public void Retrieve()
{
  try
  {
    this._StudyPlanCriterion_Language_ID.Validate("StudyPlanCriterion_Language_ID");
  }
  catch (DalException e)
  {
    throw e;
  }

  try
  {
    RetrieveExecute(_StudyPlanCriterion_Language_ID.Value, this, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
}

/// <summary>
```

```
/// Retrieves the StudyPlanCriterion_Language with the specified identifier from the database.
/// </summary>
/// <param name="studyPlanCriterion_Language_ID"></param>
public static Language Retrieve(Guid studyPlanCriterion_Language_ID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  Language objLanguage = new Language();

  try
  {
    RetrieveExecute(studyPlanCriterion_Language_ID, objLanguage, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
  return objLanguage;
}

private static void RetrieveExecute(
  Guid studyPlanCriterion_Language_ID,
  Language language,
  SqlConnection sqlConnection)
{
  DataRow objDataRow;

  SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_Language_Select", sqlConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  SqlParameter objParam;

  objParam = objCommand.Parameters.Add("@StudyPlanCriterion_Language_ID", SqlDbType.UniqueIdenti↓er);
  objParam.Value = studyPlanCriterion_Language_ID;

  DataSet objDataSet = new DataSet("StudyPlanCriterion_Language");
  SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

  try
  {
    sqlConnection.Open();
    objAdap.Fill(objDataSet, "Language");
    sqlConnection.Close();
    objDataRow = objDataSet.Tables["Language"].Rows[0];
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
  language._StudyPlanCriterion_Language_ID.Value = (Guid)objDataRow["StudyPlanCriterion_Language_ID"];
  language._StudyPlanCriterion_ID.Value      = (Guid)objDataRow["StudyPlanCriterion_ID"];
  language._Language_ID.Value           = (string)objDataRow["Language_ID"];
  language._Log.SetValues(objDataRow);
}

/// <summary>
/// Retrieves a list of the languages associated with a study plan criterion.
/// </summary>
/// <param name="studyPlanCriterion_ID">Identification of the
/// <see cref="StudyPlanning.DAL.StudyPlanCriteria.StudyPlanCriterion"/>.</param>
/// <returns>An array of <see cref="StudyPlanning.DAL.Language"/> identifications.</returns>
public static string[] GetLanguages(Guid studyPlanCriterion_ID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_Language_GetLanguages", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  SqlParameter objParam = objCommand.Parameters.Add("@StudyPlanCriterion_ID", SqlDbType.UniqueIdenti↓er);
  objParam.Value = studyPlanCriterion_ID;

  DataSet objDataSet = new DataSet("Languages");
  SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

  try
  {
    objConnection.Open();
    objAdap.Fill(objDataSet, "Languages");
    objConnection.Close();
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
  int intRows = objDataSet.Tables["Languages"].Rows.Count;
  string[] languages = new string[intRows];

  for(int i=0; i < intRows; i++)
  {
    languages[i] = (string)objDataSet.Tables["Languages"].Rows[i]["Language_ID"];
  }
  return languages;
}

#endregion //Retrieve Methods

#region Update Methods
```

```
/// <summary>
/// Updates the current StudyPlanCriterion_Language in the database using the original
/// StudyPlanCriterion_Language to resolve possible concurrency issues.
/// </summary>
/// <param name="originalLanguage">The <see cref="StudyPlanning.DAL.StudyPlanCriteria.Language"/> object.</param>
/// <returns><strong>true</strong> if the update was succesfully executed; otherwise, <strong>false</strong>.</returns>
public bool Update(Language originalLanguage)
{
  bool blnResult = false;

  try
  {
    this.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }

  SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_Language_Update", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, false);
  AddParameters(originalLanguage, objCommand, true);

  try
  {
    objConnection.Open();
    int rowsA↑ected = objCommand.ExecuteNonQuery();
    objConnection.Close();

    if (rowsA↑ected > 0)
      blnResult = true;
  }
  catch(SqlException excp)
  {
    throw new DalException(excp);
  }
  return blnResult;
}

#endregion //Update Methods

#region Delete Methods

/// <summary>
/// Deletes the current StudyPlanCriterion_Language from the database.
/// </summary>
/// <returns><strong>true</strong> if the deletion is succesfully executed; otherwise, <strong>false</strong>.</returns>
public bool Delete()
{
  bool blnResult = false;

  try
  {
    _StudyPlanCriterion_Language_ID.Validate("StudyPlanCriterion_Language_ID");
  }
  catch (DalException excp)
  {
    throw excp;
  }

  SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_Language_Delete", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, true);

  try
  {
    objConnection.Open();
    int rowsA↑ected = objCommand.ExecuteNonQuery();
    objConnection.Close();

    if (rowsA↑ected > 0)
      blnResult = true;
  }
  catch(SqlException e)
  {
    throw new DalException(e);
  }
  return blnResult;
}

#endregion //Delete Methods

#endregion //Public Methods

#region Private Methods

/// <summary>
/// Adds the relevant parameters to the specified <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see> object
/// setting the values to ones of the specified <see cref="StudyPlanning.DAL.StudyPlanCriteria.Language"/> object.
/// </summary>
/// <param name="language">
/// The <see cref="StudyPlanning.DAL.StudyPlanCriteria.Language"/> object containing the values that the parameters have to be
/// initialized to.
/// </param>
/// <param name="objCommand">
```

```
    /// The <see cref="System.Data.SqlClient.SqlCommand"/> object to which parameters should be added.
    /// </param>
    /// <param name="isOriginal">
    /// <strong>true</strong> if the parameters to be added represent original
    /// data (for handling concurrent programming issues); otherwise, <strong>false</strong>.</param>
    /// <returns>The <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see>
    /// object with the relevant parameters added.</returns>
    private static void AddParameters(Language language, SqlCommand objCommand, bool isOriginal)
    {
      string paramName = "";
      SqlParameter objParam;

      paramName = "StudyPlanCriterion_Language_ID";
      objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er);
      objParam.Value = language._StudyPlanCriterion_Language_ID.Value;
      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      paramName = "StudyPlanCriterion_ID";
      objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er);
      objParam.Value = language._StudyPlanCriterion_ID.Value;
      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      paramName = "Language_ID";
      objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.Char);
      objParam.Value = language._Language_ID.Value;
      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      language._Log.AddParameters(objCommand, isOriginal);
    }

    /// <summary>
    /// Validates the current <see cref="StudyPlanning.DAL.StudyPlanCriteria.Language"/> object. If the
    /// value of either of the private properties is null a
    /// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
    /// </summary>
    private void Validate()
    {
      try
      {
        _StudyPlanCriterion_Language_ID.Validate("StudyPlanCriterion_Language_ID");
        _StudyPlanCriterion_ID.Validate("StudyPlanCriterion_ID");
        _Language_ID.Validate("Language_ID");
        _Log.Validate();
      }
      catch (DalException excp)
      {
        throw excp;
      }
    }
    #endregion //Private Methods
  }
}
```

1.23.5 Lecturer

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.StudyPlanCriteria
{
  /// <summary>
  /// Represents a studyPlanCriterion.
  /// </summary>
  public class Lecturer : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalGuid _StudyPlanCriterion_Lecturer_ID = new DalGuid(false);
    private DalGuid _StudyPlanCriterion_ID = new DalGuid(false);
    private DalGuid _Lecturer_ID = new DalGuid(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.StudyPlanCriteria.Lecturer"/>
    /// class.
    /// </summary>
    public Lecturer() {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the StudyPlanCriterion_Lecturer_ID.
    /// </summary>
```

```
/// <value></value>
public DalGuid StudyPlanCriterion_Lecturer_ID
{
  get { return _StudyPlanCriterion_Lecturer_ID; }
}

/// <summary>
/// Gets the ID of the study plan criterion.
/// </summary>
public DalGuid StudyPlanCriterion_ID
{
  get { return _StudyPlanCriterion_ID; }
}

/// <summary>
/// Gets the lecturer ID.
/// </summary>
public DalGuid Lecturer_ID
{
  get { return _Lecturer_ID; }
}

/// <summary>
/// Gets the data log of the course.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Public Methods

#region Create Methods

/// <summary>
/// Creates a StudyPlanCriterion_Lecturer object in the database using the values of the properties
/// of the current instance.
/// </summary>
public void Create()
{
  try
  {
    this.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }

  SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_Lecturer_Insert", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, false);

  try
  {
    objConnection.Open();
    objCommand.ExecuteNonQuery();
    objConnection.Close();
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
}

#endregion

#region Retrieve Methods

/// <summary>
/// Retrieves the StudyPlanCriterion_Lecturer from the database using the value of the
/// StudyPlanCriterion_Lecturer_ID property of the current instance
/// </summary>
public void Retrieve()
{
  try
  {
    this._StudyPlanCriterion_Lecturer_ID.Validate("StudyPlanCriterion_Lecturer_ID");
  }
  catch (DalException e)
  {
    throw e;
  }

  try
  {
    RetrieveExecute(_StudyPlanCriterion_Lecturer_ID.Value, this, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
}

/// <summary>
/// Retrieves the StudyPlanCriterion_Lecturer with the specified identifier from the database.
```

```
/// </summary>
/// <param name="studyPlanCriterion_Lecturer_ID"></param>
public static Lecturer Retrieve(Guid studyPlanCriterion_Lecturer_ID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  Lecturer objLecturer = new Lecturer();

  try
  {
    RetrieveExecute(studyPlanCriterion_Lecturer_ID, objLecturer, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
  return objLecturer;
}

private static void RetrieveExecute(
  Guid studyPlanCriterion_Lecturer_ID,
  Lecturer lecturer,
  SqlConnection sqlConnection)
{
  DataRow objDataRow;

  SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_Lecturer_Select", sqlConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  SqlParameter objParam;

  objParam = objCommand.Parameters.Add("@StudyPlanCriterion_Lecturer_ID", SqlDbType.UniqueIdenti↓er);
  objParam.Value = studyPlanCriterion_Lecturer_ID;

  DataSet objDataSet = new DataSet("StudyPlanCriterion_Lecturer");
  SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

  try
  {
    sqlConnection.Open();
    objAdap.Fill(objDataSet, "Lecturer");
    sqlConnection.Close();
    objDataRow = objDataSet.Tables["Lecturer"].Rows[0];
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
  lecturer._StudyPlanCriterion_Lecturer_ID.Value = (Guid)objDataRow["StudyPlanCriterion_Lecturer_ID"];
  lecturer._StudyPlanCriterion_ID.Value      = (Guid)objDataRow["StudyPlanCriterion_ID"];
  lecturer._Lecturer_ID.Value          = (Guid)objDataRow["Lecturer_ID"];
  lecturer._Log.SetValues(objDataRow);
}

/// <summary>
/// Retrieves a list of the lecturers associated with a study plan criterion.
/// </summary>
/// <param name="studyPlanCriterion_ID">Identification of the
/// <see cref="StudyPlanning.DAL.StudyPlanCriteria.StudyPlanCriterion"/>.</param>
/// <returns>An array of <see cref="StudyPlanning.DAL.Lecturer"/> identifications.</returns>
public static Guid[] GetLecturers(Guid studyPlanCriterion_ID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_Lecturer_GetLecturers", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  SqlParameter objParam = objCommand.Parameters.Add("@StudyPlanCriterion_ID", SqlDbType.UniqueIdenti↓er);
  objParam.Value = studyPlanCriterion_ID;

  DataSet objDataSet = new DataSet("Lecturers");
  SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

  try
  {
    objConnection.Open();
    objAdap.Fill(objDataSet, "Lecturers");
    objConnection.Close();
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
  int intRows = objDataSet.Tables["Lecturers"].Rows.Count;
  Guid[] lecturers = new Guid[intRows];

  for(int i=0; i < intRows; i++)
  {
    lecturers[i] = (Guid)objDataSet.Tables["Lecturers"].Rows[i]["Lecturer_ID"];
  }
  return lecturers;
}

#endregion //Retrieve Methods

#region Update Methods
```

```
/// <summary>
/// Updates the current StudyPlanCriterion_Lecturer in the database using the original
/// StudyPlanCriterion_Lecturer to resolve possible concurrency issues.
/// </summary>
/// <param name="originalLecturer">The <see cref="StudyPlanning.DAL.StudyPlanCriteria.Lecturer"/> object.</param>
/// <returns><strong>true</strong> if the update was succesfully executed; otherwise, <strong>false</strong>.</returns>
public bool Update(Lecturer originalLecturer)
{
  bool blnResult = false;

  try
  {
    this.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }

  SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_Lecturer_Update", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, false);
  AddParameters(originalLecturer, objCommand, true);

  try
  {
    objConnection.Open();
    int rowsA↑ected = objCommand.ExecuteNonQuery();
    objConnection.Close();

    if (rowsA↑ected > 0)
      blnResult = true;
  }
  catch(SqlException excp)
  {
    throw new DalException(excp);
  }
  return blnResult;
}

#endregion //Update Methods

#region Delete Methods

/// <summary>
/// Deletes the current StudyPlanCriterion_Lecturer from the database.
/// </summary>
/// <returns><strong>true</strong> if the deletion is succesfully executed; otherwise, <strong>false</strong>.</returns>
public bool Delete()
{
  bool blnResult = false;

  try
  {
    _StudyPlanCriterion_Lecturer_ID.Validate("StudyPlanCriterion_Lecturer_ID");
  }
  catch (DalException excp)
  {
    throw excp;
  }

  SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_Lecturer_Delete", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, true);

  try
  {
    objConnection.Open();
    int rowsA↑ected = objCommand.ExecuteNonQuery();
    objConnection.Close();

    if (rowsA↑ected > 0)
      blnResult = true;
  }
  catch(SqlException e)
  {
    throw new DalException(e);
  }
  return blnResult;
}

#endregion //Delete Methods

#endregion //Public Methods

#region Private Methods

/// <summary>
/// Adds the relevant parameters to the specified <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see> object
/// setting the values to ones of the specified <see cref="StudyPlanning.DAL.StudyPlanCriteria.Lecturer"/> object.
/// </summary>
/// <param name="lecturer">
/// The <see cref="StudyPlanning.DAL.StudyPlanCriteria.Lecturer"/> object containing the values that the parameters have to be
/// initialized to.
/// </param>
/// <param name="objCommand">
/// The <see cref="System.Data.SqlClient.SqlCommand"/> object to which parameters should be added.
```

```
    /// </param>
    /// <param name="isOriginal">
    /// <strong>true</strong> if the parameters to be added represent original
    /// data (for handling concurrent programming issues); otherwise, <strong>false</strong>.</param>
    /// <returns>The <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see>
    /// object with the relevant parameters added.</returns>
    private static void AddParameters(Lecturer lecturer, SqlCommand objCommand, bool isOriginal)
    {
      string paramName = "";
      SqlParameter objParam;

      paramName = "StudyPlanCriterion_Lecturer_ID";
      objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er);
      objParam.Value = lecturer._StudyPlanCriterion_Lecturer_ID.Value;
      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      paramName = "StudyPlanCriterion_ID";
      objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er);
      objParam.Value = lecturer._StudyPlanCriterion_ID.Value;
      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      paramName = "Lecturer_ID";
      objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er);
      objParam.Value = lecturer._Lecturer_ID.Value;
      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      lecturer._Log.AddParameters(objCommand, isOriginal);
    }

    /// <summary>
    /// Validates the current <see cref="StudyPlanning.DAL.StudyPlanCriteria.Lecturer"/> object. If the
    /// value of either of the private properties is null a
    /// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
    /// </summary>
    private void Validate()
    {
      try
      {
        _StudyPlanCriterion_Lecturer_ID.Validate("StudyPlanCriterion_Lecturer_ID");
        _StudyPlanCriterion_ID.Validate("StudyPlanCriterion_ID");
        _Lecturer_ID.Validate("Lecturer_ID");
        _Log.Validate();
      }
      catch (DalException excp)
      {
        throw excp;
      }
    }
    #endregion //Private Methods
  }
}
```

1.23.6 ProjectWorkload

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.StudyPlanCriteria
{
  /// <summary>
  /// Represents the StudyPlanCriterion_ProjectWorkload table.
  /// </summary>
  public class ProjectWorkload : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalGuid _StudyPlanCriterion_ProjectWorkload_ID = new DalGuid(false);
    private DalGuid _StudyPlanCriterion_ID = new DalGuid(false);
    private DalInt _TechnicalPackage_Project_ID = new DalInt(true);
    private DalInt _StudyType_ProjectType_ID = new DalInt(true);
    private DalInt _FromPercent = new DalInt(false);
    private DalInt _ToPercent = new DalInt(false);
    private DalGuid _Point_ID = new DalGuid(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.StudyPlanCriteria.ProjectWorkload"/>
    /// class.
    /// </summary>
    public ProjectWorkload() {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties
```

```
/// <summary>
/// Gets the StudyPlanCriterion_ProjectWorkload_ID.
/// </summary>
/// <value></value>
public DalGuid StudyPlanCriterion_ProjectWorkload_ID
{
  get { return _StudyPlanCriterion_ProjectWorkload_ID; }
}

/// <summary>
/// Gets the ID of the study plan criterion.
/// </summary>
public DalGuid StudyPlanCriterion_ID
{
  get { return _StudyPlanCriterion_ID; }
}

/// <summary>
/// Gets the TechnicalPackage_Project_ID.
/// </summary>
public DalInt TechnicalPackage_Project_ID
{
  get { return _TechnicalPackage_Project_ID; }
}

/// <summary>
/// Gets the StudyType_ProjectType_ID.
/// </summary>
public DalInt StudyType_ProjectType_ID
{
  get { return _StudyType_ProjectType_ID; }
}

/// <summary>
/// Gets the lower percentage bound.
/// </summary>
public DalInt FromPercent
{
  get { return _FromPercent; }
}

/// <summary>
/// Gets the upper percentage bound.
/// </summary>
public DalInt ToPercent
{
  get { return _ToPercent; }
}

  /// <summary>
  /// Gets the point ID.
  /// </summary>
  public DalGuid Point_ID
{
  get { return _Point_ID; }
}

/// <summary>
/// Gets the data log of the course.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Public Methods

#region Create Methods

/// <summary>
/// Creates a StudyPlanCriterion_ProjectWorkload object in the database using the values of the properties
/// of the current instance.
/// </summary>
public void Create()
{
  try
  {
    this.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }

  SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_ProjectWorkload_Insert", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, false);

  try
  {
    objConnection.Open();
    objCommand.ExecuteNonQuery();
    objConnection.Close();
  }
  catch (SqlException objExc)
```

```
    {
      throw new DalException(objExc);
    }
  }

  #endregion

  #region Retrieve Methods

  /// <summary>
  /// Retrieves the StudyPlanCriterion_ProjectWorkload from the database using the value of the
  /// StudyPlanCriterion_ProjectWorkload_ID property of the current instance
  /// </summary>
  public void Retrieve()
  {
    try
    {
      this._StudyPlanCriterion_ProjectWorkload_ID.Validate("StudyPlanCriterion_ProjectWorkload_ID");
    }
    catch (DalException e)
    {
      throw e;
    }

    try
    {
      RetrieveExecute(_StudyPlanCriterion_ProjectWorkload_ID.Value, this, objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }
  }

  /// <summary>
  /// Retrieves the StudyPlanCriterion_ProjectWorkload with the specified identifier from the database.
  /// </summary>
  /// <param name="studyPlanCriterion_ProjectWorkload_ID"></param>
  public static ProjectWorkload Retrieve(Guid studyPlanCriterion_ProjectWorkload_ID)
  {
    string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
    SqlConnection objConnection = new SqlConnection(connString);

    ProjectWorkload objProjectWorkload = new ProjectWorkload();

    try
    {
      RetrieveExecute(studyPlanCriterion_ProjectWorkload_ID, objProjectWorkload, objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }
    return objProjectWorkload;
  }

  private static void RetrieveExecute(
    Guid studyPlanCriterion_ProjectWorkload_ID,
    ProjectWorkload ProjectWorkload,
    SqlConnection sqlConnection)
  {
    DataRow objDataRow;

    SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_ProjectWorkload_Select", sqlConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam;

    objParam = objCommand.Parameters.Add("@StudyPlanCriterion_ProjectWorkload_ID", SqlDbType.UniqueIdenti↓er);
    objParam.Value = studyPlanCriterion_ProjectWorkload_ID;

    DataSet objDataSet = new DataSet("StudyPlanCriterion_ProjectWorkload");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
      sqlConnection.Open();
      objAdap.Fill(objDataSet, "ProjectWorkload");
      sqlConnection.Close();
      objDataRow = objDataSet.Tables["ProjectWorkload"].Rows[0];
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
    ProjectWorkload._StudyPlanCriterion_ProjectWorkload_ID.Value = (Guid)objDataRow["StudyPlanCriterion_ProjectWorkload_ID"];
    ProjectWorkload._StudyPlanCriterion_ID.Value            = (Guid)objDataRow["StudyPlanCriterion_ID"];

    if (objDataRow["TechnicalPackage_Project_ID"].Equals(DBNull.Value))
      ProjectWorkload._TechnicalPackage_Project_ID.IsNull = true;
    else
      ProjectWorkload._TechnicalPackage_Project_ID.Value       = (int)objDataRow["TechnicalPackage_Project_ID"];

    if (objDataRow["StudyType_ProjectType_ID"].Equals(DBNull.Value))
      ProjectWorkload._StudyType_ProjectType_ID.IsNull = true;
    else
      ProjectWorkload._StudyType_ProjectType_ID.Value       = (int)objDataRow["StudyType_ProjectType_ID"];

    ProjectWorkload._FromPercent.Value               = (int)objDataRow["FromPercent"];
```

```
    ProjectWorkload._ToPercent.Value                = (int)objDataRow["ToPercent"];
    ProjectWorkload._Point_ID.Value                 = (Guid)objDataRow["Point_ID"];
    ProjectWorkload._Log.SetValues(objDataRow);
  }

  /// <summary>
  /// Retrieves a list of StudyPlanCriterion_ProjectWorkload_IDs from a given
  /// StudyPlanCriterion_ID.
  /// </summary>
  /// <param name="studyPlanCriterion_ID">Identification of a study plan criterion.</param>
  /// <returns>An array of StudyPlanCriterion_ProjectWorkload_IDs.</returns>
  public static Guid[] GetID(Guid studyPlanCriterion_ID)
  {
    string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
    SqlConnection objConnection = new SqlConnection(connString);

    SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_ProjectWorkload_GetID", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam = objCommand.Parameters.Add("@StudyPlanCriterion_ID", SqlDbType.UniqueIdenti↓er);
    objParam.Value = studyPlanCriterion_ID;

    DataSet objDataSet = new DataSet("Result");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    Guid[] ids = new Guid[0];

    try
    {
      objConnection.Open();
      objAdap.Fill(objDataSet, "Result");
      objConnection.Close();
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }

    int rows = objDataSet.Tables["Result"].Rows.Count;
    ids = new Guid[rows];
    for(int i=0; i < rows; i++)
    {
       ids[i] = (Guid)objDataSet.Tables["Result"].Rows[i]["StudyPlanCriterion_ProjectWorkload_ID"];
    }
    return ids;
  }

  #endregion //Retrieve Methods

  #region Update Methods

  /// <summary>
  /// Updates the current StudyPlanCriterion_ProjectWorkload in the database using the original
  /// StudyPlanCriterion_ProjectWorkload to resolve possible concurrency issues.
  /// </summary>
  /// <param name="originalProjectWorkload">The <see cref="StudyPlanning.DAL.StudyPlanCriteria.ProjectWorkload"/> object.</param>
  /// <returns><strong>true</strong> if the update was succesfully executed; otherwise, <strong>false</strong>.</returns>
  public bool Update(ProjectWorkload originalProjectWorkload)
  {
    bool blnResult = false;

    try
    {
      this.Validate();
    }
    catch (DalException excp)
    {
      throw excp;
    }

    SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_ProjectWorkload_Update", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, false);
    AddParameters(originalProjectWorkload, objCommand, true);

    try
    {
      objConnection.Open();
      int rowsA↑ected = objCommand.ExecuteNonQuery();
      objConnection.Close();

      if (rowsA↑ected > 0)
        blnResult = true;
    }
    catch(SqlException excp)
    {
      throw new DalException(excp);
    }
    return blnResult;
  }

  #endregion //Update Methods

  #region Delete Methods

  /// <summary>
  /// Deletes the current StudyPlanCriterion_ProjectWorkload from the database.
  /// </summary>
```

```
/// <returns><strong>true</strong> if the deletion is succesfully executed; otherwise, <strong>false</strong>.</returns>
public bool Delete()
{
  bool blnResult = false;

  try
  {
    _StudyPlanCriterion_ProjectWorkload_ID.Validate("StudyPlanCriterion_ProjectWorkload_ID");
  }
  catch (DalException excp)
  {
    throw excp;
  }

  SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_ProjectWorkload_Delete", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, true);

  try
  {
    objConnection.Open();
    int rowsA↑ected = objCommand.ExecuteNonQuery();
    objConnection.Close();

    if (rowsA↑ected > 0)
      blnResult = true;
  }
  catch(SqlException e)
  {
    throw new DalException(e);
  }
  return blnResult;
}

#endregion //Delete Methods

#endregion //Public Methods

#region Private Methods

/// <summary>
/// Adds the relevant parameters to the specified <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see> object
/// setting the values to ones of the specified <see cref="StudyPlanning.DAL.StudyPlanCriteria.ProjectWorkload"/> object.
/// </summary>
/// <param name="ProjectWorkload">
/// The <see cref="StudyPlanning.DAL.StudyPlanCriteria.ProjectWorkload"/> object containing the values that the parameters have to be
/// initialized to.
/// </param>
/// <param name="objCommand">
/// The <see cref="System.Data.SqlClient.SqlCommand"/> object to which parameters should be added.
/// </param>
/// <param name="isOriginal">
/// <strong>true</strong> if the parameters to be added represent original
/// data (for handling concurrent programming issues); otherwise, <strong>false</strong>.</param>
/// <returns>The <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see>
/// object with the relevant parameters added.</returns>
private static void AddParameters(ProjectWorkload ProjectWorkload, SqlCommand objCommand, bool isOriginal)
{
  string paramName = "";
  SqlParameter objParam;

  paramName = "StudyPlanCriterion_ProjectWorkload_ID";
  objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er);
  objParam.Value = ProjectWorkload._StudyPlanCriterion_ProjectWorkload_ID.Value;
  if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

  paramName = "StudyPlanCriterion_ID";
  objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er);
  objParam.Value = ProjectWorkload._StudyPlanCriterion_ID.Value;
  if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

  paramName = "TechnicalPackage_Project_ID";
  objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.Int);
  if (ProjectWorkload._TechnicalPackage_Project_ID.IsNull)
    objParam.Value = DBNull.Value;
  else
    objParam.Value = ProjectWorkload._TechnicalPackage_Project_ID.Value;
  if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

  paramName = "StudyType_ProjectType_ID";
  objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.Int);
  if (ProjectWorkload._StudyType_ProjectType_ID.IsNull)
    objParam.Value = DBNull.Value;
  else
    objParam.Value = ProjectWorkload._StudyType_ProjectType_ID.Value;
  if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

  paramName = "FromPercent";
  objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.Int);
  objParam.Value = ProjectWorkload._FromPercent.Value;
  if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

  paramName = "ToPercent";
```

```
      objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.Int);
      objParam.Value = ProjectWorkload._ToPercent.Value;
      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      paramName = "Point_ID";
      objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er);
      objParam.Value = ProjectWorkload._Point_ID.Value;
      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      ProjectWorkload._Log.AddParameters(objCommand, isOriginal);
    }

    /// <summary>
    /// Validates the current <see cref="StudyPlanning.DAL.StudyPlanCriteria.ProjectWorkload"/> object. If the
    /// value of either of the private properties is null a
    /// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
    /// </summary>
    private void Validate()
    {
      try
      {
        _StudyPlanCriterion_ProjectWorkload_ID.Validate("StudyPlanCriterion_ProjectWorkload_ID");
        _StudyPlanCriterion_ID.Validate("StudyPlanCriterion_ID");
        _TechnicalPackage_Project_ID.Validate("TechnicalPackage_Project_ID");
        _StudyType_ProjectType_ID.Validate("StudyType_ProjectType_ID");
        _FromPercent.Validate("FromPercent");
        _ToPercent.Validate("ToPercent");
        _Point_ID.Validate("Point_ID");
        _Log.Validate();
      }
      catch (DalException excp)
      {
        throw excp;
      }
    }
    #endregion //Private Methods
  }
}
```

1.23.7 StudyType

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.StudyPlanCriteria
{
  /// <summary>
  /// Represents a studyPlanCriterion.
  /// </summary>
  public class StudyType : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalGuid _StudyPlanCriterion_StudyType_ID = new DalGuid(false);
    private DalGuid _StudyPlanCriterion_ID = new DalGuid(false);
    private DalInt _StudyType_ID = new DalInt(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.StudyPlanCriteria.StudyType"/>
    /// class.
    /// </summary>
    public StudyType() {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the StudyPlanCriterion_StudyType_ID.
    /// </summary>
    /// <value></value>
    public DalGuid StudyPlanCriterion_StudyType_ID
    {
      get { return _StudyPlanCriterion_StudyType_ID; }
    }

    /// <summary>
    /// Gets the ID of the study plan criterion.
    /// </summary>
    public DalGuid StudyPlanCriterion_ID
    {
      get { return _StudyPlanCriterion_ID; }
    }
```

```
/// <summary>
/// Gets the study type ID.
/// </summary>
public DalInt StudyType_ID
{
  get { return _StudyType_ID; }
}

/// <summary>
/// Gets the data log of the course.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Public Methods

#region Create Methods

/// <summary>
/// Creates a StudyPlanCriterion_StudyType object in the database using the values of the properties
/// of the current instance.
/// </summary>
public void Create()
{
  try
  {
    this.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }

  SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_StudyType_Insert", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, false);

  try
  {
    objConnection.Open();
    objCommand.ExecuteNonQuery();
    objConnection.Close();
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
}

#endregion

#region Retrieve Methods

/// <summary>
/// Retrieves the StudyPlanCriterion_StudyType from the database using the value of the
/// StudyPlanCriterion_StudyType_ID property of the current instance
/// </summary>
public void Retrieve()
{
  try
  {
    this._StudyPlanCriterion_StudyType_ID.Validate("StudyPlanCriterion_StudyType_ID");
  }
  catch (DalException e)
  {
    throw e;
  }

  try
  {
    RetrieveExecute(_StudyPlanCriterion_StudyType_ID.Value, this, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
}

/// <summary>
/// Retrieves the StudyPlanCriterion_StudyType with the specified identifier from the database.
/// </summary>
/// <param name="studyPlanCriterion_StudyType_ID"></param>
public static StudyType Retrieve(Guid studyPlanCriterion_StudyType_ID)
{
  string connString = System.Con|guration.Con|gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  StudyType objStudyType = new StudyType();

  try
  {
    RetrieveExecute(studyPlanCriterion_StudyType_ID, objStudyType, objConnection);
  }
  catch (DalException objExc)
```

```
    {
      throw objExc;
    }
    return objStudyType;
  }

  private static void RetrieveExecute(
    Guid studyPlanCriterion_StudyType_ID,
    StudyType studyType,
    SqlConnection sqlConnection)
  {
    DataRow objDataRow;

    SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_StudyType_Select", sqlConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam;

    objParam = objCommand.Parameters.Add("@StudyPlanCriterion_StudyType_ID", SqlDbType.UniqueIdenti↓er);
    objParam.Value = studyPlanCriterion_StudyType_ID;

    DataSet objDataSet = new DataSet("StudyPlanCriterion_StudyType");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
      sqlConnection.Open();
      objAdap.Fill(objDataSet, "StudyType");
      sqlConnection.Close();
      objDataRow = objDataSet.Tables["StudyType"].Rows[0];
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
    studyType._StudyPlanCriterion_StudyType_ID.Value = (Guid)objDataRow["StudyPlanCriterion_StudyType_ID"];
    studyType._StudyPlanCriterion_ID.Value      = (Guid)objDataRow["StudyPlanCriterion_ID"];
    studyType._StudyType_ID.Value           = (int)objDataRow["StudyType_ID"];
    studyType._Log.SetValues(objDataRow);
  }

  /// <summary>
  /// Retrieves a list of the study types associated with a study plan criterion.
  /// </summary>
  /// <param name="studyPlanCriterion_ID">Identification of the
  /// <see cref="StudyPlanning.DAL.StudyPlanCriteria.StudyPlanCriterion"/>.</param>
  /// <returns>An array of <see cref="StudyPlanning.DAL.StudyTypes.StudyType"/> identifications.</returns>
  public static int[] GetStudyTypes(Guid studyPlanCriterion_ID)
  {
    string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
    SqlConnection objConnection = new SqlConnection(connString);

    SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_StudyType_GetStudyTypes", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam = objCommand.Parameters.Add("@StudyPlanCriterion_ID", SqlDbType.UniqueIdenti↓er);
    objParam.Value = studyPlanCriterion_ID;

    DataSet objDataSet = new DataSet("StudyTypes");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
      objConnection.Open();
      objAdap.Fill(objDataSet, "StudyTypes");
      objConnection.Close();
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
    int intRows = objDataSet.Tables["StudyTypes"].Rows.Count;
    int[] studytypes = new int[intRows];

    for(int i=0; i < intRows; i++)
    {
      studytypes[i] = (int)objDataSet.Tables["StudyTypes"].Rows[i]["StudyType_ID"];
    }
    return studytypes;
  }

  #endregion //Retrieve Methods

  #region Update Methods

  /// <summary>
  /// Updates the current StudyPlanCriterion_StudyType in the database using the original
  /// StudyPlanCriterion_StudyType to resolve possible concurrency issues.
  /// </summary>
  /// <param name="originalStudyType">The <see cref="StudyPlanning.DAL.StudyPlanCriteria.StudyType"/> object.</param>
  /// <returns><strong>true</strong> if the update was succesfully executed; otherwise, <strong>false</strong>.</returns>
  public bool Update(StudyType originalStudyType)
  {
    bool blnResult = false;

    try
    {
      this.Validate();
    }
```

```
  catch (DalException excp)
  {
    throw excp;
  }

  SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_StudyType_Update", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, false);
  AddParameters(originalStudyType, objCommand, true);

  try
  {
    objConnection.Open();
    int rowsA↑ected = objCommand.ExecuteNonQuery();
    objConnection.Close();

    if (rowsA↑ected > 0)
      blnResult = true;
  }
  catch(SqlException excp)
  {
    throw new DalException(excp);
  }
  return blnResult;
}

#endregion //Update Methods

#region Delete Methods

/// <summary>
/// Deletes the current StudyPlanCriterion_StudyType from the database.
/// </summary>
/// <returns><strong>true</strong> if the deletion is succesfully executed; otherwise, <strong>false</strong>.</returns>
public bool Delete()
{
  bool blnResult = false;

  try
  {
    _StudyPlanCriterion_StudyType_ID.Validate("StudyPlanCriterion_StudyType_ID");
  }
  catch (DalException excp)
  {
    throw excp;
  }

  SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_StudyType_Delete", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, true);

  try
  {
    objConnection.Open();
    int rowsA↑ected = objCommand.ExecuteNonQuery();
    objConnection.Close();

    if (rowsA↑ected > 0)
      blnResult = true;
  }
  catch(SqlException e)
  {
    throw new DalException(e);
  }
  return blnResult;
}

#endregion //Delete Methods

#endregion //Public Methods

#region Private Methods

/// <summary>
/// Adds the relevant parameters to the specified <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see> object
/// setting the values to ones of the specified <see cref="StudyPlanning.DAL.StudyPlanCriteria.StudyType"/> object.
/// </summary>
/// <param name="studyType">
/// The <see cref="StudyPlanning.DAL.StudyPlanCriteria.StudyType"/> object containing the values that the parameters have to be
/// initialized to.
/// </param>
/// <param name="objCommand">
/// The <see cref="System.Data.SqlClient.SqlCommand"/> object to which parameters should be added.
/// </param>
/// <param name="isOriginal">
/// <strong>true</strong> if the parameters to be added represent original
/// data (for handling concurrent programming issues); otherwise, <strong>false</strong>.</param>
/// <returns>The <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see>
/// object with the relevant parameters added.</returns>
private static void AddParameters(StudyType studyType, SqlCommand objCommand, bool isOriginal)
{
  string paramName = "";
  SqlParameter objParam;

  paramName = "StudyPlanCriterion_StudyType_ID";
  objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er);
  objParam.Value = studyType._StudyPlanCriterion_StudyType_ID.Value;
```

```
      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      paramName = "StudyPlanCriterion_ID";
      objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er);
      objParam.Value = studyType._StudyPlanCriterion_ID.Value;
      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      paramName = "StudyType_ID";
      objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.Int);
      objParam.Value = studyType._StudyType_ID.Value;
      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      studyType._Log.AddParameters(objCommand, isOriginal);
    }

    /// <summary>
    /// Validates the current <see cref="StudyPlanning.DAL.StudyPlanCriteria.StudyType"/> object. If the
    /// value of either of the private properties is null a
    /// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
    /// </summary>
    private void Validate()
    {
      try
      {
        _StudyPlanCriterion_StudyType_ID.Validate("StudyPlanCriterion_StudyType_ID");
        _StudyPlanCriterion_ID.Validate("StudyPlanCriterion_ID");
        _StudyType_ID.Validate("StudyType_ID");
        _Log.Validate();
      }
      catch (DalException excp)
      {
        throw excp;
      }
    }
    #endregion //Private Methods
  }
}
```

1.23.8 WorkloadPeriods

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.StudyPlanCriteria.WorkloadPeriods
{
  /// <summary>
  /// Represents a studyPlanCriterion.
  /// </summary>
  public class WorkloadPeriod : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalGuid _StudyPlanCriterion_WorkloadPeriod_ID = new DalGuid(false);
    private DalGuid _StudyPlanCriterion_ID = new DalGuid(false);
    private DalGuid _Period_ID = new DalGuid(false);
    private DalGuid _Point_ID = new DalGuid(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.StudyPlanCriteria.WorkloadPeriods.WorkloadPeriod"/>
    /// class.
    /// </summary>
    public WorkloadPeriod() {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the StudyPlanCriterion_WorkloadPeriod_ID.
    /// </summary>
    /// <value></value>
    public DalGuid StudyPlanCriterion_WorkloadPeriod_ID
    {
      get { return _StudyPlanCriterion_WorkloadPeriod_ID; }
    }

    /// <summary>
    /// Gets the ID of the study plan criterion.
    /// </summary>
    public DalGuid StudyPlanCriterion_ID
    {
      get { return _StudyPlanCriterion_ID; }
    }
```

```
/// <summary>
/// Gets the period ID.
/// </summary>
public DalGuid Period_ID
{
  get { return _Period_ID; }
}

/// <summary>
/// Gets the point ID.
/// </summary>
public DalGuid Point_ID
{
  get { return _Point_ID; }
}

/// <summary>
/// Gets the data log of the course.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Public Methods

#region Create Methods

/// <summary>
/// Creates a StudyPlanCriterion_WorkloadPeriod object in the database using the values of the properties
/// of the current instance.
/// </summary>
public void Create()
{
  try
  {
    this.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }

  SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_WorkloadPeriod_Insert", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, false);

  try
  {
    objConnection.Open();
    objCommand.ExecuteNonQuery();
    objConnection.Close();
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
}

#endregion

#region Retrieve Methods

/// <summary>
/// Retrieves the StudyPlanCriterion_WorkloadPeriod from the database using the value of the
/// StudyPlanCriterion_WorkloadPeriod_ID property of the current instance
/// </summary>
public void Retrieve()
{
  try
  {
    this._StudyPlanCriterion_WorkloadPeriod_ID.Validate("StudyPlanCriterion_WorkloadPeriod_ID");
  }
  catch (DalException e)
  {
    throw e;
  }

  try
  {
    RetrieveExecute(_StudyPlanCriterion_WorkloadPeriod_ID.Value, this, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
}

/// <summary>
/// Retrieves the StudyPlanCriterion_WorkloadPeriod with the specified identifier from the database.
/// </summary>
/// <param name="studyPlanCriterion_WorkloadPeriod_ID"></param>
public static WorkloadPeriod Retrieve(Guid studyPlanCriterion_WorkloadPeriod_ID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);
```

```
  WorkloadPeriod objWorkloadPeriod = new WorkloadPeriod();

  try
  {
    RetrieveExecute(studyPlanCriterion_WorkloadPeriod_ID, objWorkloadPeriod, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
  return objWorkloadPeriod;
}

private static void RetrieveExecute(
  Guid studyPlanCriterion_WorkloadPeriod_ID,
  WorkloadPeriod workloadPeriod,
  SqlConnection sqlConnection)
{
  DataRow objDataRow;

  SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_WorkloadPeriod_Select", sqlConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  SqlParameter objParam;

  objParam = objCommand.Parameters.Add("@StudyPlanCriterion_WorkloadPeriod_ID", SqlDbType.UniqueIdenti↓er);
  objParam.Value = studyPlanCriterion_WorkloadPeriod_ID;

  DataSet objDataSet = new DataSet("StudyPlanCriterion_WorkloadPeriod");
  SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

  try
  {
    sqlConnection.Open();
    objAdap.Fill(objDataSet, "WorkloadPeriod");
    sqlConnection.Close();
    objDataRow = objDataSet.Tables["WorkloadPeriod"].Rows[0];
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
  workloadPeriod._StudyPlanCriterion_WorkloadPeriod_ID.Value = (Guid)objDataRow["StudyPlanCriterion_WorkloadPeriod_ID"];
  workloadPeriod._StudyPlanCriterion_ID.Value         = (Guid)objDataRow["StudyPlanCriterion_ID"];
  workloadPeriod._Period_ID.Value                 = (Guid)objDataRow["Period_ID"];
  workloadPeriod._Point_ID.Value                = (Guid)objDataRow["Point_ID"];
  workloadPeriod._Log.SetValues(objDataRow);
}

/// <summary>
/// Retrieves a point associated with a workload period in a study plan criterion.
/// </summary>
/// <param name="studyPlanCriterion_ID">Identification of the
/// <see cref="StudyPlanning.DAL.StudyPlanCriteria.StudyPlanCriterion"/>.</param>
/// <param name="period_ID">Identification of the
/// <see cref="StudyPlanning.DAL.Period"/>.</param>
/// <returns>A <see cref="StudyPlanning.DAL.Point"/> identification.</returns>
public static Guid GetPoint(Guid studyPlanCriterion_ID, Guid period_ID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_WorkloadPeriod_GetPoint", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  SqlParameter objParam = objCommand.Parameters.Add("@StudyPlanCriterion_ID", SqlDbType.UniqueIdenti↓er);
  objParam.Value = studyPlanCriterion_ID;

  objParam = objCommand.Parameters.Add("@Period_ID", SqlDbType.UniqueIdenti↓er);
  objParam.Value = period_ID;

  DataSet objDataSet = new DataSet("Result");
  SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

  try
  {
    objConnection.Open();
    objAdap.Fill(objDataSet, "Result");
    objConnection.Close();
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
  return (Guid)objDataSet.Tables["Result"].Rows[0]["Point_ID"];
}

/// <summary>
/// Retrieves the workload period identification associated with a
/// study plan criterion and a period.
/// </summary>
/// <param name="studyPlanCriterion_ID">Identification of the
/// <see cref="StudyPlanning.DAL.StudyPlanCriteria.StudyPlanCriterion"/>.</param>
/// <param name="period_ID">Identification of the
/// <see cref="StudyPlanning.DAL.Period"/>.</param>
/// <returns>A <see cref="StudyPlanning.DAL.StudyPlanCriteria.WorkloadPeriods.WorkloadPeriod"/> identification.</returns>
public static Guid GetID(Guid studyPlanCriterion_ID, Guid period_ID)
{
```

```
    string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
    SqlConnection objConnection = new SqlConnection(connString);

    SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_WorkloadPeriod_GetID", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam = objCommand.Parameters.Add("@StudyPlanCriterion_ID", SqlDbType.UniqueIdenti↓er);
    objParam.Value = studyPlanCriterion_ID;

    objParam = objCommand.Parameters.Add("@Period_ID", SqlDbType.UniqueIdenti↓er);
    objParam.Value = period_ID;

    DataSet objDataSet = new DataSet("Result");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
      objConnection.Open();
      objAdap.Fill(objDataSet, "Result");
      objConnection.Close();
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
    return (Guid)objDataSet.Tables["Result"].Rows[0]["StudyPlanCriterion_WorkloadPeriod_ID"];
  }

  #endregion //Retrieve Methods

  #region Update Methods

  /// <summary>
  /// Updates the current StudyPlanCriterion_WorkloadPeriod in the database using the original
  /// StudyPlanCriterion_WorkloadPeriod to resolve possible concurrency issues.
  /// </summary>
  /// <param name="originalWorkloadPeriod">The <see cref="StudyPlanning.DAL.StudyPlanCriteria.WorkloadPeriods.WorkloadPeriod"/> object.</
    param>
  /// <returns><strong>true</strong> if the update was succesfully executed; otherwise, <strong>false</strong>.</returns>
  public bool Update(WorkloadPeriod originalWorkloadPeriod)
  {
    bool blnResult = false;

    try
    {
      this.Validate();
    }
    catch (DalException excp)
    {
      throw excp;
    }

    SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_WorkloadPeriod_Update", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, false);
    AddParameters(originalWorkloadPeriod, objCommand, true);

    try
    {
      objConnection.Open();
      int rowsA↑ected = objCommand.ExecuteNonQuery();
      objConnection.Close();

      if (rowsA↑ected > 0)
        blnResult = true;
    }
    catch(SqlException excp)
    {
      throw new DalException(excp);
    }
    return blnResult;
  }

  #endregion //Update Methods

  #region Delete Methods

  /// <summary>
  /// Deletes the current StudyPlanCriterion_WorkloadPeriod from the database.
  /// </summary>
  /// <returns><strong>true</strong> if the deletion is succesfully executed; otherwise, <strong>false</strong>.</returns>
  public bool Delete()
  {
    bool blnResult = false;

    try
    {
      _StudyPlanCriterion_WorkloadPeriod_ID.Validate("StudyPlanCriterion_WorkloadPeriod_ID");
    }
    catch (DalException excp)
    {
      throw excp;
    }

    SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_WorkloadPeriod_Delete", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, true);
```

```
      try
      {
        objConnection.Open();
        int rowsA↑ected = objCommand.ExecuteNonQuery();
        objConnection.Close();

        if (rowsA↑ected > 0)
          blnResult = true;
      }
      catch(SqlException e)
      {
        throw new DalException(e);
      }
      return blnResult;
    }

    #endregion //Delete Methods

    #endregion //Public Methods

    #region Private Methods

    /// <summary>
    /// Adds the relevant parameters to the specified <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see> object
    /// setting the values to ones of the specified <see cref="StudyPlanning.DAL.StudyPlanCriteria.WorkloadPeriods.WorkloadPeriod"/> object.
    /// </summary>
    /// <param name="workloadPeriod">
    /// The <see cref="StudyPlanning.DAL.StudyPlanCriteria.WorkloadPeriods.WorkloadPeriod"/> object containing the values that the parameters
      have to be
    /// initialized to.
    /// </param>
    /// <param name="objCommand">
    /// The <see cref="System.Data.SqlClient.SqlCommand"/> object to which parameters should be added.
    /// </param>
    /// <param name="isOriginal">
    /// <strong>true</strong> if the parameters to be added represent original
    /// data (for handling concurrent programming issues); otherwise, <strong>false</strong>.</param>
    /// <returns>The <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see>
    /// object with the relevant parameters added.</returns>
    private static void AddParameters(WorkloadPeriod workloadPeriod, SqlCommand objCommand, bool isOriginal)
    {
      string paramName = "";
      SqlParameter objParam;

      paramName = "StudyPlanCriterion_WorkloadPeriod_ID";
      objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er);
      objParam.Value = workloadPeriod._StudyPlanCriterion_WorkloadPeriod_ID.Value;
      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      paramName = "StudyPlanCriterion_ID";
      objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er);
      objParam.Value = workloadPeriod._StudyPlanCriterion_ID.Value;
      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      paramName = "Period_ID";
      objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er);
      objParam.Value = workloadPeriod._Period_ID.Value;
      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      paramName = "Point_ID";
      objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er);
      objParam.Value = workloadPeriod._Point_ID.Value;
      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      workloadPeriod._Log.AddParameters(objCommand, isOriginal);
    }

    /// <summary>
    /// Validates the current <see cref="StudyPlanning.DAL.StudyPlanCriteria.WorkloadPeriods.WorkloadPeriod"/> object. If the
    /// value of either of the private properties is null a
    /// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
    /// </summary>
    private void Validate()
    {
      try
      {
        _StudyPlanCriterion_WorkloadPeriod_ID.Validate("StudyPlanCriterion_WorkloadPeriod_ID");
        _StudyPlanCriterion_ID.Validate("StudyPlanCriterion_ID");
        _Period_ID.Validate("Period_ID");
        _Point_ID.Validate("Point_ID");
        _Log.Validate();
      }
      catch (DalException excp)
      {
        throw excp;
      }
    }
    #endregion //Private Methods
  }
}
```

Module

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.StudyPlanCriteria.WorkloadPeriods
{
  /// <summary>
  /// Represents a StudyPlanCriterion_WorkloadPeriod_Module object.
  /// </summary>
  public class Module : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalGuid _StudyPlanCriterion_WorkloadPeriod_Module_ID = new DalGuid(false);
    private DalGuid _StudyPlanCriterion_WorkloadPeriod_ID = new DalGuid(false);
    private DalInt _Module_ID = new DalInt(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.StudyPlanCriteria.WorkloadPeriods.Module"/>
    /// class.
    /// </summary>
    public Module() {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the StudyPlanCriterion_WorkloadPeriod_Module_ID.
    /// </summary>
    /// <value></value>
    public DalGuid StudyPlanCriterion_WorkloadPeriod_Module_ID
    {
      get { return _StudyPlanCriterion_WorkloadPeriod_Module_ID; }
    }

    /// <summary>
    /// Gets the StudyPlanCriterion_WorkloadPeriod_ID.
    /// </summary>
    public DalGuid StudyPlanCriterion_WorkloadPeriod_ID
    {
      get { return _StudyPlanCriterion_WorkloadPeriod_ID; }
    }

    /// <summary>
    /// Gets the module ID.
    /// </summary>
    public DalInt Module_ID
    {
      get { return _Module_ID; }
    }

    /// <summary>
    /// Gets the data log of the course.
    /// </summary>
    public DataLog Log
    {
      get { return _Log; }
    }

    #endregion //Public Properties

    #region Public Methods

    #region Create Methods

    /// <summary>
    /// Creates a StudyPlanCriterion_WorkloadPeriod_Module object in the database using the values of the properties
    /// of the current instance.
    /// </summary>
    public void Create()
    {
      try
      {
        this.Validate();
      }
      catch (DalException excp)
      {
        throw excp;
      }

      SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_WorkloadPeriod_Module_Insert", objConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      AddParameters(this, objCommand, false);

      try
      {
```

```
      objConnection.Open();
      objCommand.ExecuteNonQuery();
      objConnection.Close();
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
  }

  #endregion

  #region Retrieve Methods

  /// <summary>
  /// Retrieves the StudyPlanCriterion_WorkloadPeriod_Module from the database using the value of the
  /// StudyPlanCriterion_WorkloadPeriod_Module_ID property of the current instance
  /// </summary>
  public void Retrieve()
  {
    try
    {
      this._StudyPlanCriterion_WorkloadPeriod_Module_ID.Validate("StudyPlanCriterion_WorkloadPeriod_Module_ID");
    }
    catch (DalException e)
    {
      throw e;
    }

    try
    {
      RetrieveExecute(_StudyPlanCriterion_WorkloadPeriod_Module_ID.Value, this, objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }
  }

  /// <summary>
  /// Retrieves the StudyPlanCriterion_WorkloadPeriod_Module with the specified identifier from the database.
  /// </summary>
  /// <param name="studyPlanCriterion_WorkloadPeriod_Module_ID"></param>
  public static Module Retrieve(Guid studyPlanCriterion_WorkloadPeriod_Module_ID)
  {
    string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
    SqlConnection objConnection = new SqlConnection(connString);

    Module objModule = new Module();

    try
    {
      RetrieveExecute(studyPlanCriterion_WorkloadPeriod_Module_ID, objModule, objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }
    return objModule;
  }

  private static void RetrieveExecute(
    Guid studyPlanCriterion_WorkloadPeriod_Module_ID,
    Module module,
    SqlConnection sqlConnection)
  {
    DataRow objDataRow;

    SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_WorkloadPeriod_Module_Select", sqlConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam;

    objParam = objCommand.Parameters.Add("@StudyPlanCriterion_WorkloadPeriod_Module_ID", SqlDbType.UniqueIdenti↓er);
    objParam.Value = studyPlanCriterion_WorkloadPeriod_Module_ID;

    DataSet objDataSet = new DataSet("StudyPlanCriterion_WorkloadPeriod_Module");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
      sqlConnection.Open();
      objAdap.Fill(objDataSet, "WorkloadPeriod_Module");
      sqlConnection.Close();
      objDataRow = objDataSet.Tables["WorkloadPeriod_Module"].Rows[0];
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
    module._StudyPlanCriterion_WorkloadPeriod_Module_ID.Value = (Guid)objDataRow["StudyPlanCriterion_WorkloadPeriod_Module_ID"];
    module._StudyPlanCriterion_WorkloadPeriod_ID.Value     = (Guid)objDataRow["StudyPlanCriterion_WorkloadPeriod_ID"];
    module._Module_ID.Value                = (int)objDataRow["Module_ID"];
    module._Log.SetValues(objDataRow);
  }

  /// <summary>
  /// Retrieves deselected modules for a workload period in a study plan criterion.
  /// </summary>
```

```
/// <param name="studyPlanCriterion_WorkloadPeriod_ID">Identification of the
/// <see cref="StudyPlanning.DAL.StudyPlanCriteria.WorkloadPeriods.WorkloadPeriod"/>.</param>
/// <returns>An array of <see cref="StudyPlanning.DAL.Module"/> identifications.</returns>
public static int[] GetDeselectedModules(Guid studyPlanCriterion_WorkloadPeriod_ID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_WorkloadPeriod_Module_GetModules", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  SqlParameter objParam = objCommand.Parameters.Add("@StudyPlanCriterion_WorkloadPeriod_ID", SqlDbType.UniqueIdenti↓er);
  objParam.Value = studyPlanCriterion_WorkloadPeriod_ID;

  DataSet objDataSet = new DataSet("Result");
  SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

  try
  {
    objConnection.Open();
    objAdap.Fill(objDataSet, "Result");
    objConnection.Close();
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }

  int intRows = objDataSet.Tables["Result"].Rows.Count;
  int[] modules = new int[intRows];

  for (int i=0; i < intRows; i++)
  {
    modules[i] = (int)objDataSet.Tables["Result"].Rows[i]["Module_ID"];
  }
  return modules;
}

#endregion //Retrieve Methods

#region Update Methods

/// <summary>
/// Updates the current StudyPlanCriterion_WorkloadPeriod_Module in the database using the original
/// StudyPlanCriterion_WorkloadPeriod_Module to resolve possible concurrency issues.
/// </summary>
/// <param name="originalModule">The <see cref="StudyPlanning.DAL.StudyPlanCriteria.WorkloadPeriods.Module"/> object.</param>
/// <returns><strong>true</strong> if the update was succesfully executed; otherwise, <strong>false</strong>.</returns>
public bool Update(Module originalModule)
{
  bool blnResult = false;

  try
  {
    this.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }

  SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_WorkloadPeriod_Module_Update", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, false);
  AddParameters(originalModule, objCommand, true);

  try
  {
    objConnection.Open();
    int rowsA↑ected = objCommand.ExecuteNonQuery();
    objConnection.Close();

    if (rowsA↑ected > 0)
      blnResult = true;
  }
  catch(SqlException excp)
  {
    throw new DalException(excp);
  }
  return blnResult;
}

#endregion //Update Methods

#region Delete Methods

/// <summary>
/// Deletes the current StudyPlanCriterion_WorkloadPeriod_Module from the database.
/// </summary>
/// <returns><strong>true</strong> if the deletion is succesfully executed; otherwise, <strong>false</strong>.</returns>
public bool Delete()
{
  bool blnResult = false;

  try
  {
    _StudyPlanCriterion_WorkloadPeriod_Module_ID.Validate("StudyPlanCriterion_WorkloadPeriod_Module_ID");
  }
```

```
      catch (DalException excp)
      {
        throw excp;
      }

      SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_WorkloadPeriod_Module_Delete", objConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      AddParameters(this, objCommand, true);

      try
      {
        objConnection.Open();
        int rowsA↑ected = objCommand.ExecuteNonQuery();
        objConnection.Close();

        if (rowsA↑ected > 0)
          blnResult = true;
      }
      catch(SqlException e)
      {
        throw new DalException(e);
      }
      return blnResult;
    }

    #endregion //Delete Methods

    #endregion //Public Methods

    #region Private Methods

    /// <summary>
    /// Adds the relevant parameters to the specified <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see> object
    /// setting the values to ones of the specified <see cref="StudyPlanning.DAL.StudyPlanCriteria.WorkloadPeriods.Module"/> object.
    /// </summary>
    /// <param name="workloadPeriodModule">
    /// The <see cref="StudyPlanning.DAL.StudyPlanCriteria.WorkloadPeriods.Module"/> object containing the values that the parameters have to be
    /// initialized to.
    /// </param>
    /// <param name="objCommand">
    /// The <see cref="System.Data.SqlClient.SqlCommand"/> object to which parameters should be added.
    /// </param>
    /// <param name="isOriginal">
    /// <strong>true</strong> if the parameters to be added represent original
    /// data (for handling concurrent programming issues); otherwise, <strong>false</strong>.</param>
    /// <returns>The <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see>
    /// object with the relevant parameters added.</returns>
    private static void AddParameters(Module workloadPeriodModule, SqlCommand objCommand, bool isOriginal)
    {
      string paramName = "";
      SqlParameter objParam;

      paramName = "StudyPlanCriterion_WorkloadPeriod_Module_ID";
      objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er);
      objParam.Value = workloadPeriodModule._StudyPlanCriterion_WorkloadPeriod_Module_ID.Value;
      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      paramName = "StudyPlanCriterion_WorkloadPeriod_ID";
      objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er);
      objParam.Value = workloadPeriodModule._StudyPlanCriterion_WorkloadPeriod_ID.Value;
      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      paramName = "Module_ID";
      objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.Int);
      objParam.Value = workloadPeriodModule._Module_ID.Value;
      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      workloadPeriodModule._Log.AddParameters(objCommand, isOriginal);
    }

    /// <summary>
    /// Validates the current <see cref="StudyPlanning.DAL.StudyPlanCriteria.WorkloadPeriods.Module"/> object. If the
    /// value of either of the private properties is null a
    /// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
    /// </summary>
    private void Validate()
    {
      try
      {
        _StudyPlanCriterion_WorkloadPeriod_Module_ID.Validate("StudyPlanCriterion_WorkloadPeriod_Module_ID");
        _StudyPlanCriterion_WorkloadPeriod_ID.Validate("StudyPlanCriterion_WorkloadPeriod_ID");
        _Module_ID.Validate("Module_ID");
        _Log.Validate();
      }
      catch (DalException excp)
      {
        throw excp;
      }
    }
    #endregion //Private Methods
  }
}
```

1.23.9 WorkloadPeriodTypes

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.StudyPlanCriteria.WorkloadPeriodTypes
{
  /// <summary>
  /// Represents a studyPlanCriterion.
  /// </summary>
  public class WorkloadPeriodType : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalGuid _StudyPlanCriterion_WorkloadPeriodType_ID = new DalGuid(false);
    private DalGuid _StudyPlanCriterion_ID = new DalGuid(false);
    private DalInt _PeriodType_ID = new DalInt(false);
    private DalGuid _Point_ID = new DalGuid(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.StudyPlanCriteria.WorkloadPeriodTypes.WorkloadPeriodType"/>
    /// class.
    /// </summary>
    public WorkloadPeriodType() {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the StudyPlanCriterion_WorkloadPeriodType_ID.
    /// </summary>
    /// <value></value>
    public DalGuid StudyPlanCriterion_WorkloadPeriodType_ID
    {
      get { return _StudyPlanCriterion_WorkloadPeriodType_ID; }
    }

    /// <summary>
    /// Gets the ID of the study plan criterion.
    /// </summary>
    public DalGuid StudyPlanCriterion_ID
    {
      get { return _StudyPlanCriterion_ID; }
    }

    /// <summary>
    /// Gets the period type ID.
    /// </summary>
    public DalInt PeriodType_ID
    {
      get { return _PeriodType_ID; }
    }

    /// <summary>
    /// Gets the point ID.
    /// </summary>
    public DalGuid Point_ID
    {
      get { return _Point_ID; }
    }

    /// <summary>
    /// Gets the data log of the course.
    /// </summary>
    public DataLog Log
    {
      get { return _Log; }
    }

    #endregion //Public Properties

    #region Public Methods

    #region Create Methods

    /// <summary>
    /// Creates a StudyPlanCriterion_WorkloadPeriodType object in the database using the values of the properties
    /// of the current instance.
    /// </summary>
    public void Create()
    {
      try
      {
        this.Validate();
      }
      catch (DalException excp)
      {
        throw excp;
```

```
    }

    SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_WorkloadPeriodType_Insert", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, false);

    try
    {
      objConnection.Open();
      objCommand.ExecuteNonQuery();
      objConnection.Close();
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
  }

  #endregion

  #region Retrieve Methods

  /// <summary>
  /// Retrieves the StudyPlanCriterion_WorkloadPeriodType from the database using the value of the
  /// StudyPlanCriterion_WorkloadPeriodType_ID property of the current instance
  /// </summary>
  public void Retrieve()
  {
    try
    {
      this._StudyPlanCriterion_WorkloadPeriodType_ID.Validate("StudyPlanCriterion_WorkloadPeriodType_ID");
    }
    catch (DalException e)
    {
      throw e;
    }

    try
    {
      RetrieveExecute(_StudyPlanCriterion_WorkloadPeriodType_ID.Value, this, objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }
  }

  /// <summary>
  /// Retrieves the StudyPlanCriterion_WorkloadPeriodType with the specified identifier from the database.
  /// </summary>
  /// <param name="studyPlanCriterion_WorkloadPeriodType_ID"></param>
  public static WorkloadPeriodType Retrieve(Guid studyPlanCriterion_WorkloadPeriodType_ID)
  {
    string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
    SqlConnection objConnection = new SqlConnection(connString);

    WorkloadPeriodType objWorkloadPeriodType = new WorkloadPeriodType();

    try
    {
      RetrieveExecute(studyPlanCriterion_WorkloadPeriodType_ID, objWorkloadPeriodType, objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }
    return objWorkloadPeriodType;
  }

  private static void RetrieveExecute(
    Guid studyPlanCriterion_WorkloadPeriodType_ID,
    WorkloadPeriodType workloadPeriodType,
    SqlConnection sqlConnection)
  {
    DataRow objDataRow;

    SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_WorkloadPeriodType_Select", sqlConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam;

    objParam = objCommand.Parameters.Add("@StudyPlanCriterion_WorkloadPeriodType_ID", SqlDbType.UniqueIdenti↓er);
    objParam.Value = studyPlanCriterion_WorkloadPeriodType_ID;

    DataSet objDataSet = new DataSet("StudyPlanCriterion_WorkloadPeriodType");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
      sqlConnection.Open();
      objAdap.Fill(objDataSet, "WorkloadPeriodType");
      sqlConnection.Close();
      objDataRow = objDataSet.Tables["WorkloadPeriodType"].Rows[0];
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
```

```
    workloadPeriodType._StudyPlanCriterion_WorkloadPeriodType_ID.Value = (Guid)objDataRow["
      StudyPlanCriterion_WorkloadPeriodType_ID"];
    workloadPeriodType._StudyPlanCriterion_ID.Value          = (Guid)objDataRow["StudyPlanCriterion_ID"];
    workloadPeriodType._PeriodType_ID.Value                  = (int)objDataRow["PeriodType_ID"];
    workloadPeriodType._Point_ID.Value                       = (Guid)objDataRow["Point_ID"];
    workloadPeriodType._Log.SetValues(objDataRow);
  }

  /// <summary>
  /// Retrieves a point associated with a workload period type in a study plan criterion.
  /// </summary>
  /// <param name="studyPlanCriterion_ID">Identification of the
  /// <see cref="StudyPlanning.DAL.StudyPlanCriteria.StudyPlanCriterion"/>.</param>
  /// <param name="periodType_ID">Identification of the
  /// <see cref="StudyPlanning.DAL.PeriodTypes.PeriodType"/>.</param>
  /// <returns>A <see cref="StudyPlanning.DAL.Point"/> identification.</returns>
  public static Guid GetPoint(Guid studyPlanCriterion_ID, int periodType_ID)
  {
    string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
    SqlConnection objConnection = new SqlConnection(connString);

    SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_WorkloadPeriodType_GetPoint", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam = objCommand.Parameters.Add("@StudyPlanCriterion_ID", SqlDbType.UniqueIdenti↓er);
    objParam.Value = studyPlanCriterion_ID;

    objParam = objCommand.Parameters.Add("@PeriodType_ID", SqlDbType.Int);
    objParam.Value = periodType_ID;

    DataSet objDataSet = new DataSet("Result");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
      objConnection.Open();
      objAdap.Fill(objDataSet, "Result");
      objConnection.Close();
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
    return (Guid)objDataSet.Tables["Result"].Rows[0]["Point_ID"];
  }

  /// <summary>
  /// Retrieves the workload period type identification associated with a
  /// study plan criterion and a period type.
  /// </summary>
  /// <param name="studyPlanCriterion_ID">Identification of the
  /// <see cref="StudyPlanning.DAL.StudyPlanCriteria.StudyPlanCriterion"/>.</param>
  /// <param name="periodType_ID">Identification of the
  /// <see cref="StudyPlanning.DAL.PeriodTypes.PeriodType"/>.</param>
  /// <returns>A <see cref="StudyPlanning.DAL.StudyPlanCriteria.WorkloadPeriodTypes.WorkloadPeriodType"/> identification.</returns>
  public static Guid GetID(Guid studyPlanCriterion_ID, int periodType_ID)
  {
    string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
    SqlConnection objConnection = new SqlConnection(connString);

    SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_WorkloadPeriodType_GetID", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam = objCommand.Parameters.Add("@StudyPlanCriterion_ID", SqlDbType.UniqueIdenti↓er);
    objParam.Value = studyPlanCriterion_ID;

    objParam = objCommand.Parameters.Add("@PeriodType_ID", SqlDbType.Int);
    objParam.Value = periodType_ID;

    DataSet objDataSet = new DataSet("Result");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
      objConnection.Open();
      objAdap.Fill(objDataSet, "Result");
      objConnection.Close();
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
    return (Guid)objDataSet.Tables["Result"].Rows[0]["StudyPlanCriterion_WorkloadPeriodType_ID"];
  }

  #endregion //Retrieve Methods

  #region Update Methods

  /// <summary>
  /// Updates the current StudyPlanCriterion_WorkloadPeriodType in the database using the original
  /// StudyPlanCriterion_WorkloadPeriodType to resolve possible concurrency issues.
  /// </summary>
  /// <param name="originalWorkloadPeriodType">The <see cref="StudyPlanning.DAL.StudyPlanCriteria.WorkloadPeriodTypes.WorkloadPeriodType
    "/> object.</param>
  /// <returns><strong>true</strong> if the update was succesfully executed; otherwise, <strong>false</strong>.</returns>
  public bool Update(WorkloadPeriodType originalWorkloadPeriodType)
  {
    bool blnResult = false;
```

```
    try
    {
      this.Validate();
    }
    catch (DalException excp)
    {
      throw excp;
    }

    SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_WorkloadPeriodType_Update", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, false);
    AddParameters(originalWorkloadPeriodType, objCommand, true);

    try
    {
      objConnection.Open();
      int rowsA↑ected = objCommand.ExecuteNonQuery();
      objConnection.Close();

      if (rowsA↑ected > 0)
        blnResult = true;
    }
    catch(SqlException excp)
    {
      throw new DalException(excp);
    }
    return blnResult;
  }

  #endregion //Update Methods

  #region Delete Methods

  /// <summary>
  /// Deletes the current StudyPlanCriterion_WorkloadPeriodType from the database.
  /// </summary>
  /// <returns><strong>true</strong> if the deletion is succesfully executed; otherwise, <strong>false</strong>.</returns>
  public bool Delete()
  {
    bool blnResult = false;

    try
    {
      _StudyPlanCriterion_WorkloadPeriodType_ID.Validate("StudyPlanCriterion_WorkloadPeriodType_ID");
    }
    catch (DalException excp)
    {
      throw excp;
    }

    SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_WorkloadPeriodType_Delete", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, true);

    try
    {
      objConnection.Open();
      int rowsA↑ected = objCommand.ExecuteNonQuery();
      objConnection.Close();

      if (rowsA↑ected > 0)
        blnResult = true;
    }
    catch(SqlException e)
    {
      throw new DalException(e);
    }
    return blnResult;
  }

  #endregion //Delete Methods

  #endregion //Public Methods

  #region Private Methods

  /// <summary>
  /// Adds the relevant parameters to the specified <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see> object
  /// setting the values to ones of the specified <see cref="StudyPlanning.DAL.StudyPlanCriteria.WorkloadPeriodTypes.WorkloadPeriodType"/> object
    .
  /// </summary>
  /// <param name="workloadPeriodType">
  /// The <see cref="StudyPlanning.DAL.StudyPlanCriteria.WorkloadPeriodTypes.WorkloadPeriodType"/> object containing the values that the
    parameters have to be
  /// initialized to.
  /// </param>
  /// <param name="objCommand">
  /// The <see cref="System.Data.SqlClient.SqlCommand"/> object to which parameters should be added.
  /// </param>
  /// <param name="isOriginal">
  /// <strong>true</strong> if the parameters to be added represent original
  /// data (for handling concurrent programming issues); otherwise, <strong>false</strong>.</param>
  /// <returns>The <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see>
  /// object with the relevant parameters added.</returns>
  private static void AddParameters(WorkloadPeriodType workloadPeriodType, SqlCommand objCommand, bool isOriginal)
```

```
    {
      string paramName = "";
      SqlParameter objParam;

      paramName = "StudyPlanCriterion_WorkloadPeriodType_ID";
      objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er);
      objParam.Value = workloadPeriodType._StudyPlanCriterion_WorkloadPeriodType_ID.Value;
      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      paramName = "StudyPlanCriterion_ID";
      objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er);
      objParam.Value = workloadPeriodType._StudyPlanCriterion_ID.Value;
      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      paramName = "PeriodType_ID";
      objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.Int);
      objParam.Value = workloadPeriodType._PeriodType_ID.Value;
      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      paramName = "Point_ID";
      objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er);
      objParam.Value = workloadPeriodType._Point_ID.Value;
      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      workloadPeriodType._Log.AddParameters(objCommand, isOriginal);
    }

    /// <summary>
    /// Validates the current <see cref="StudyPlanning.DAL.StudyPlanCriteria.WorkloadPeriodTypes.WorkloadPeriodType"/> object. If the
    /// value of either of the private properties is null a
    /// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
    /// </summary>
    private void Validate()
    {
      try
      {
        _StudyPlanCriterion_WorkloadPeriodType_ID.Validate("StudyPlanCriterion_WorkloadPeriodType_ID");
        _StudyPlanCriterion_ID.Validate("StudyPlanCriterion_ID");
        _PeriodType_ID.Validate("PeriodType_ID");
        _Point_ID.Validate("Point_ID");
        _Log.Validate();
      }
      catch (DalException excp)
      {
        throw excp;
      }
    }
    #endregion //Private Methods
  }
}
```

Module

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.StudyPlanCriteria.WorkloadPeriodTypes
{
  /// <summary>
  /// Represents a StudyPlanCriterion_WorkloadPeriodType_Module object.
  /// </summary>
  public class Module : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalGuid _StudyPlanCriterion_WorkloadPeriodType_Module_ID = new DalGuid(false);
    private DalGuid _StudyPlanCriterion_WorkloadPeriodType_ID = new DalGuid(false);
    private DalInt _Module_ID = new DalInt(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.StudyPlanCriteria.WorkloadPeriodTypes.Module"/>
    /// class.
    /// </summary>
    public Module() {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the StudyPlanCriterion_WorkloadPeriodType_Module_ID.
    /// </summary>
    /// <value></value>
```

```
public DalGuid StudyPlanCriterion_WorkloadPeriodType_Module_ID
{
  get { return _StudyPlanCriterion_WorkloadPeriodType_Module_ID; }
}

/// <summary>
/// Gets the StudyPlanCriterion_WorkloadPeriodType_ID.
/// </summary>
public DalGuid StudyPlanCriterion_WorkloadPeriodType_ID
{
  get { return _StudyPlanCriterion_WorkloadPeriodType_ID; }
}

/// <summary>
/// Gets the module ID.
/// </summary>
public DalInt Module_ID
{
  get { return _Module_ID; }
}

/// <summary>
/// Gets the data log of the course.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Public Methods

#region Create Methods

/// <summary>
/// Creates a StudyPlanCriterion_WorkloadPeriodType_Module object in the database using the values of the properties
/// of the current instance.
/// </summary>
public void Create()
{
  try
  {
    this.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }

  SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_WorkloadPeriodType_Module_Insert", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, false);

  try
  {
    objConnection.Open();
    objCommand.ExecuteNonQuery();
    objConnection.Close();
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
}

#endregion

#region Retrieve Methods

/// <summary>
/// Retrieves the StudyPlanCriterion_WorkloadPeriodType_Module from the database using the value of the
/// StudyPlanCriterion_WorkloadPeriodType_Module_ID property of the current instance
/// </summary>
public void Retrieve()
{
  try
  {
    this._StudyPlanCriterion_WorkloadPeriodType_Module_ID.Validate("StudyPlanCriterion_WorkloadPeriodType_Module_ID");
  }
  catch (DalException e)
  {
    throw e;
  }

  try
  {
    RetrieveExecute(_StudyPlanCriterion_WorkloadPeriodType_Module_ID.Value, this, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
}

/// <summary>
/// Retrieves the StudyPlanCriterion_WorkloadPeriodType_Module with the specified identifier from the database.
/// </summary>
```

```
/// <param name="studyPlanCriterion_WorkloadPeriodType_Module_ID"></param>
public static Module Retrieve(Guid studyPlanCriterion_WorkloadPeriodType_Module_ID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  Module objModule = new Module();

  try
  {
    RetrieveExecute(studyPlanCriterion_WorkloadPeriodType_Module_ID, objModule, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
  return objModule;
}

private static void RetrieveExecute(
  Guid studyPlanCriterion_WorkloadPeriodType_Module_ID,
  Module module,
  SqlConnection sqlConnection)
{
  DataRow objDataRow;

  SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_WorkloadPeriodType_Module_Select", sqlConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  SqlParameter objParam;

  objParam = objCommand.Parameters.Add("@StudyPlanCriterion_WorkloadPeriodType_Module_ID", SqlDbType.UniqueIdenti↓er);
  objParam.Value = studyPlanCriterion_WorkloadPeriodType_Module_ID;

  DataSet objDataSet = new DataSet("StudyPlanCriterion_WorkloadPeriodType_Module");
  SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

  try
  {
    sqlConnection.Open();
    objAdap.Fill(objDataSet, "WorkloadPeriodType_Module");
    sqlConnection.Close();
    objDataRow = objDataSet.Tables["WorkloadPeriodType_Module"].Rows[0];
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
  module._StudyPlanCriterion_WorkloadPeriodType_Module_ID.Value = (Guid)objDataRow["
    StudyPlanCriterion_WorkloadPeriodType_Module_ID"];
  module._StudyPlanCriterion_WorkloadPeriodType_ID.Value    = (Guid)objDataRow["StudyPlanCriterion_WorkloadPeriodType_ID"];
  module._Module_ID.Value                  = (int)objDataRow["Module_ID"];
  module._Log.SetValues(objDataRow);
}

/// <summary>
/// Retrieves deselected modules for a workload period type in a study plan criterion.
/// </summary>
/// <param name="studyPlanCriterion_WorkloadPeriodType_ID">Identification of the
/// <see cref="StudyPlanning.DAL.StudyPlanCriteria.WorkloadPeriodTypes.WorkloadPeriodType"/>.</param>
/// <returns>An array of <see cref="StudyPlanning.DAL.Module"/> identifications.</returns>
public static int[] GetDeselectedModules(Guid studyPlanCriterion_WorkloadPeriodType_ID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_WorkloadPeriodType_Module_GetModules", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  SqlParameter objParam = objCommand.Parameters.Add("@StudyPlanCriterion_WorkloadPeriodType_ID", SqlDbType.UniqueIdenti↓er);
  objParam.Value = studyPlanCriterion_WorkloadPeriodType_ID;

  DataSet objDataSet = new DataSet("Result");
  SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

  try
  {
    objConnection.Open();
    objAdap.Fill(objDataSet, "Result");
    objConnection.Close();
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }

  int intRows = objDataSet.Tables["Result"].Rows.Count;
  int[] modules = new int[intRows];

  for (int i=0; i < intRows; i++)
  {
    modules[i] = (int)objDataSet.Tables["Result"].Rows[i]["Module_ID"];
  }
  return modules;
}

#endregion //Retrieve Methods

#region Update Methods
```

```
/// <summary>
/// Updates the current StudyPlanCriterion_WorkloadPeriodType_Module in the database using the original
/// StudyPlanCriterion_WorkloadPeriodType_Module to resolve possible concurrency issues.
/// </summary>
/// <param name="originalModule">The <see cref="StudyPlanning.DAL.StudyPlanCriteria.WorkloadPeriodTypes.Module"/> object.</param>
/// <returns><strong>true</strong> if the update was succesfully executed; otherwise, <strong>false</strong>.</returns>
public bool Update(Module originalModule)
{
  bool blnResult = false;

  try
  {
    this.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }

  SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_WorkloadPeriodType_Module_Update", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, false);
  AddParameters(originalModule, objCommand, true);

  try
  {
    objConnection.Open();
    int rowsA↑ected = objCommand.ExecuteNonQuery();
    objConnection.Close();

    if (rowsA↑ected > 0)
      blnResult = true;
  }
  catch(SqlException excp)
  {
    throw new DalException(excp);
  }
  return blnResult;
}

#endregion //Update Methods

#region Delete Methods

/// <summary>
/// Deletes the current StudyPlanCriterion_WorkloadPeriodType_Module from the database.
/// </summary>
/// <returns><strong>true</strong> if the deletion is succesfully executed; otherwise, <strong>false</strong>.</returns>
public bool Delete()
{
  bool blnResult = false;

  try
  {
    _StudyPlanCriterion_WorkloadPeriodType_Module_ID.Validate("StudyPlanCriterion_WorkloadPeriodType_Module_ID");
  }
  catch (DalException excp)
  {
    throw excp;
  }

  SqlCommand objCommand = new SqlCommand("StudyPlanCriterion_WorkloadPeriodType_Module_Delete", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, true);

  try
  {
    objConnection.Open();
    int rowsA↑ected = objCommand.ExecuteNonQuery();
    objConnection.Close();

    if (rowsA↑ected > 0)
      blnResult = true;
  }
  catch(SqlException e)
  {
    throw new DalException(e);
  }
  return blnResult;
}

#endregion //Delete Methods

#endregion //Public Methods

#region Private Methods

/// <summary>
/// Adds the relevant parameters to the specified <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see> object
/// setting the values to ones of the specified <see cref="StudyPlanning.DAL.StudyPlanCriteria.WorkloadPeriodTypes.Module"/> object.
/// </summary>
/// <param name="workloadPeriodModule">
/// The <see cref="StudyPlanning.DAL.StudyPlanCriteria.WorkloadPeriodTypes.Module"/> object containing the values that the parameters have to
  be
/// initialized to.
/// </param>
```

```
    /// <param name="objCommand">
    /// The <see cref="System.Data.SqlClient.SqlCommand"/> object to which parameters should be added.
    /// </param>
    /// <param name="isOriginal">
    /// <strong>true</strong> if the parameters to be added represent original
    /// data (for handling concurrent programming issues); otherwise, <strong>false</strong>.</param>
    /// <returns>The <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see>
    /// object with the relevant parameters added.</returns>
    private static void AddParameters(Module workloadPeriodModule, SqlCommand objCommand, bool isOriginal)
    {
      string paramName = "";
      SqlParameter objParam;

      paramName = "StudyPlanCriterion_WorkloadPeriodType_Module_ID";
      objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er);
      objParam.Value = workloadPeriodModule._StudyPlanCriterion_WorkloadPeriodType_Module_ID.Value;
      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      paramName = "StudyPlanCriterion_WorkloadPeriodType_ID";
      objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er);
      objParam.Value = workloadPeriodModule._StudyPlanCriterion_WorkloadPeriodType_ID.Value;
      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      paramName = "Module_ID";
      objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.Int);
      objParam.Value = workloadPeriodModule._Module_ID.Value;
      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      workloadPeriodModule._Log.AddParameters(objCommand, isOriginal);
    }

    /// <summary>
    /// Validates the current <see cref="StudyPlanning.DAL.StudyPlanCriteria.WorkloadPeriodTypes.Module"/> object. If the
    /// value of either of the private properties is null a
    /// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
    /// </summary>
    private void Validate()
    {
      try
      {
        _StudyPlanCriterion_WorkloadPeriodType_Module_ID.Validate("StudyPlanCriterion_WorkloadPeriodType_Module_ID");
        _StudyPlanCriterion_WorkloadPeriodType_ID.Validate("StudyPlanCriterion_WorkloadPeriodType_ID");
        _Module_ID.Validate("Module_ID");
        _Log.Validate();
      }
      catch (DalException excp)
      {
        throw excp;
      }
    }
    #endregion //Private Methods
  }
}
```

1.24 StudyPlans

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.StudyPlans
{
  /// <summary>
  /// Represents a study plan.
  /// </summary>
  public class StudyPlan : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalGuid _StudyPlan_ID = new DalGuid(false);
    private DalString _Name = new DalString(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Initializes a new instance of the <see cref="StudyPlanning.DAL.StudyPlans.StudyPlan"/>
    /// class.
    /// </summary>
    public StudyPlan()
    {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties
```

```
/// <summary>
/// Gets the ID of study plan.
/// </summary>
public DalGuid StudyPlan_ID
{
  get { return _StudyPlan_ID; }
}

/// <summary>
/// Gets the name of the study plan.
/// </summary>
public DalString Name
{
  get { return _Name; }
}

/// <summary>
/// Gets the data log of the course.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Methods

/// <summary>
/// Creates a new study plan in the database using the values of the properties
/// of the current instance.
/// </summary>
public void Create()
{
  try
  {
    this.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }

  SqlCommand objCommand = new SqlCommand("StudyPlan_Insert", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, false);

  try
  {
    objConnection.Open();
    objCommand.ExecuteNonQuery();
    objConnection.Close();
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
}

#region Retrieve Methods

/// <summary>
/// Retrieves the study plan from the database using the value of the
/// StudyPlan_ID property of the current instance.
/// </summary>
public void Retrieve()
{
  try
  {
    this._StudyPlan_ID.Validate("StudyPlan_ID");
  }
  catch (DalException excp)
  {
    throw excp;
  }

  try
  {
      RetrieveExecute(_StudyPlan_ID.Value, this, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
}

/// <summary>
/// Retrieves the study plan with the specified identifier from the database.
/// </summary>
/// <param name="studyPlan_ID">The globally unique identifier of the study plan.</param>
public static StudyPlan Retrieve(Guid studyPlan_ID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  StudyPlan objSp = new StudyPlan();

  try
```

```
    {
      RetrieveExecute(studyPlan_ID, objSp, objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }

    return objSp;
  }

  private static void RetrieveExecute(
    Guid studyPlan_ID,
    StudyPlan objStudyPlan,
    SqlConnection sqlConnection)
  {
    SqlCommand objCommand = new SqlCommand("StudyPlan_Select", sqlConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam;

    objParam = objCommand.Parameters.Add("@StudyPlan_ID", SqlDbType.UniqueIdenti↓er, 16);
    objParam.Value = studyPlan_ID;

    DataSet objDataSet = new DataSet("StudyPlan");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
      sqlConnection.Open();
      objAdap.Fill(objDataSet, "StudyPlan");
      sqlConnection.Close();
      DataRow objDataRow = objDataSet.Tables["StudyPlan"].Rows[0];

      objStudyPlan.StudyPlan_ID.Value = (Guid)objDataRow["StudyPlan_ID"];
      objStudyPlan.Name.Value = (string)objDataRow["Name"];

      objStudyPlan.Log.SetValues(objDataRow);
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
  }

  #endregion //Retrive Methods

  /// <summary>
  /// Updates the current study plan in the database using the original
  /// study plan to resolve possible concurrency issues.
  /// </summary>
  /// <param name="originalStudyPlan">The original <see cref="StudyPlanning.DAL.StudyPlans.StudyPlan"/> object.</param>
  /// <returns><strong>true</strong> if the update was succesfully executed; otherwise, <strong>false</strong>.</returns>
  public bool Update(StudyPlan originalStudyPlan)
  {
    try
    {
      this.Validate();
    }
    catch (DalException excp)
    {
      throw excp;
    }

    bool blnResult = false;

    SqlCommand objCommand = new SqlCommand("StudyPlan_Update", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, false);
    AddParameters(originalStudyPlan, objCommand, true);

    int rowsA↑ected;

    try
    {
      objConnection.Open();
      rowsA↑ected = objCommand.ExecuteNonQuery();
      objConnection.Close();

      if (rowsA↑ected > 0)
        blnResult = true;
    }
    catch(SqlException excp)
    {
      throw new DalException(excp);
    }

    return blnResult;
  }

  /// <summary>
  /// Deletes the current study plan from the database.
  /// </summary>
  /// <returns><strong>true</strong> if the deletion is succesfully executed; otherwise, <strong>false</strong>.</returns>
  public bool Delete()
  {
    bool blnResult = false;
```

```
      try
      {
        this.Validate();
      }
      catch (DalException excp)
      {
        throw excp;
      }

      SqlCommand objCommand = new SqlCommand("StudyPlan_Delete", objConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      AddParameters(this, objCommand, true);

      try
      {
        objConnection.Open();
        int rowsA↑ected = objCommand.ExecuteNonQuery();
        objConnection.Close();

        if (rowsA↑ected > 0)
          blnResult = true;
      }
      catch(SqlException objEx)
      {
        throw new DalException(objEx);
      }

      return blnResult;
    }

    /// <summary>
    /// Adds the relevant parameters to the specified <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see> object
    /// setting the values to ones of the specified <see cref="StudyPlanning.DAL.StudyPlans.StudyPlan"/> object.
    /// </summary>
    /// <param name="studyPlan">
    /// The <see cref="StudyPlanning.DAL.StudyPlans.StudyPlan"/> object containing the values that the parameters have to be
    /// initialized to.
    /// </param>
    /// <param name="objCommand">
    /// The <see cref="System.Data.SqlClient.SqlCommand"/> object to which parameters should be added.
    /// </param>
    /// <param name="isOriginal">
    /// <strong>true</strong> if the parameters to be added represent original
    /// data (for handling concurrent programming issues); otherwise, <strong>false</strong>.</param>
    /// <returns>The <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see>
    /// object with the relevant parameters added.</returns>
    private static void AddParameters(StudyPlan studyPlan, SqlCommand objCommand, bool isOriginal)
    {
      string paramName = "";
      SqlParameter objParam;

      //StudyPlan_ID
      paramName = "StudyPlan_ID";
      objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      objParam.Value = studyPlan.StudyPlan_ID.Value;

      //Name
      paramName = "Name";
      objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.VarChar, 200);

      if (isOriginal)
        objParam.ParameterName = "@Original_" + paramName;

      objParam.Value = studyPlan.Name.Value;

      //Log
      objCommand = studyPlan.Log.AddParameters(objCommand, isOriginal);
    }

    /// <summary>
    /// Validates the current <see cref="StudyPlanning.DAL.StudyPlans.StudyPlan"/> object. If the
    /// value of either of the private properties is null a
    /// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
    /// </summary>
    private void Validate()
    {
      try
      {
        _StudyPlan_ID.Validate("StudyPlan_ID");
        _Name.Validate("Name");
        _Log.Validate();
      }
      catch (DalException excp)
      {
        throw excp;
      }
    }

    #endregion //Methods
  }
}
```

1.24.1 Period

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.StudyPlans
{
  /// <summary>
  /// Represents a period in a study plan.
  /// </summary>
  public class Period : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalGuid _StudyPlan_Period_ID = new DalGuid(false);
    private DalGuid _StudyPlan_ID = new DalGuid(false);
    private DalGuid _Period_ID = new DalGuid(true);
    private DalGuid _Project_ID = new DalGuid(true);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Initializes a new instance of the <see cref="StudyPlanning.DAL.StudyPlans.Period"/>
    /// class.
    /// </summary>
    public Period()
    {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the ID of study plan period.
    /// </summary>
    public DalGuid StudyPlan_Period_ID
    {
      get { return _StudyPlan_Period_ID; }
    }

    /// <summary>
    /// Gets the ID of study plan.
    /// </summary>
    public DalGuid StudyPlan_ID
    {
      get { return _StudyPlan_ID; }
    }

    /// <summary>
    /// Gets the ID of the period.
    /// </summary>
    public DalGuid Period_ID
    {
      get { return _Period_ID; }
    }

    /// <summary>
    /// Gets the ID of the project.
    /// </summary>
    public DalGuid Project_ID
    {
      get { return _Project_ID; }
    }

    /// <summary>
    /// Gets the data log of the course.
    /// </summary>
    public DataLog Log
    {
      get { return _Log; }
    }

    #endregion //Public Properties

    #region Methods

    /// <summary>
    /// Creates a new study plan period in the database using the values of the properties
    /// of the current instance.
    /// </summary>
    public void Create()
    {
      try
      {
        this.Validate();
      }
      catch (DalException excp)
      {
        throw excp;
      }
```

```
    SqlCommand objCommand = new SqlCommand("StudyPlan_Period_Insert", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, false);

    try
    {
      objConnection.Open();
      objCommand.ExecuteNonQuery();
      objConnection.Close();
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
  }

  #region Retrieve Methods

  /// <summary>
  /// Retrieves the study plan period from the database using the value of the
  /// StudyPlan_Period_ID property of the current instance.
  /// </summary>
  public void Retrieve()
  {
    try
    {
      this._StudyPlan_Period_ID.Validate("StudyPlan_Period_ID");
    }
    catch (DalException excp)
    {
      throw excp;
    }

    try
    {
        RetrieveExecute(_StudyPlan_Period_ID.Value, this, objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }
  }

  /// <summary>
  /// Retrieves the study plan period with the specified identifier from the database.
  /// </summary>
  /// <param name="studyPlan_Period_ID">The globally unique identifier of the study plan period.</param>
  public static Period Retrieve(Guid studyPlan_Period_ID)
  {
    string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
    SqlConnection objConnection = new SqlConnection(connString);

    Period objPeriod = new Period();

    try
    {
      RetrieveExecute(studyPlan_Period_ID, objPeriod, objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }

    return objPeriod;
  }

  private static void RetrieveExecute(
    Guid studyPlan_Period_ID,
    Period objPeriod,
    SqlConnection sqlConnection)
  {
    SqlCommand objCommand = new SqlCommand("StudyPlan_Period_Select", sqlConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam;

    objParam = objCommand.Parameters.Add("@StudyPlan_Period_ID", SqlDbType.UniqueIdenti↓er, 16);
    objParam.Value = studyPlan_Period_ID;

    DataSet objDataSet = new DataSet("StudyPlan_Period");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
      sqlConnection.Open();
      objAdap.Fill(objDataSet, "StudyPlan_Period");
      sqlConnection.Close();
      DataRow objDataRow = objDataSet.Tables["StudyPlan_Period"].Rows[0];

      objPeriod.StudyPlan_Period_ID.Value = (Guid)objDataRow["StudyPlan_Period_ID"];
      objPeriod.StudyPlan_ID.Value = (Guid)objDataRow["StudyPlan_ID"];

      if (objDataRow["Period_ID"].Equals(System.DBNull.Value))
        objPeriod.Period_ID.IsNull = true;
      else
        objPeriod.Period_ID.Value = (Guid)objDataRow["Period_ID"];

      if (objDataRow["Project_ID"].Equals(System.DBNull.Value))
```

```
        objPeriod.Project_ID.IsNull = true;
      else
        objPeriod.Project_ID.Value = (Guid)objDataRow["Project_ID"];

      objPeriod.Log.SetValues(objDataRow);
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
  }

  #endregion //Retrive Methods

  /// <summary>
  /// Updates the current study plan period in the database using the original
  /// study plan period to resolve possible concurrency issues.
  /// </summary>
  /// <param name="originalStudyPlan_Period">The original <see cref="StudyPlanning.DAL.StudyPlans.Period"/> object.</param>
  /// <returns><strong>true</strong> if the update was succesfully executed; otherwise, <strong>false</strong>.</returns>
  public bool Update(Period originalStudyPlan_Period)
  {
    try
    {
      this.Validate();
    }
    catch (DalException excp)
    {
      throw excp;
    }

    bool blnResult = false;

    SqlCommand objCommand = new SqlCommand("StudyPlan_Period_Update", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, false);
    AddParameters(originalStudyPlan_Period, objCommand, true);

    int rowsA↑ected;

    try
    {
      objConnection.Open();
      rowsA↑ected = objCommand.ExecuteNonQuery();
      objConnection.Close();

      if (rowsA↑ected > 0)
        blnResult = true;
    }
    catch(SqlException excp)
    {
      throw new DalException(excp);
    }

    return blnResult;
  }

  /// <summary>
  /// Deletes the current study plan period from the database.
  /// </summary>
  /// <returns><strong>true</strong> if the deletion is succesfully executed; otherwise, <strong>false</strong>.</returns>
  public bool Delete()
  {
    bool blnResult = false;

    try
    {
      this.Validate();
    }
    catch (DalException excp)
    {
      throw excp;
    }

    SqlCommand objCommand = new SqlCommand("StudyPlan_Period_Delete", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, true);

    try
    {
      objConnection.Open();
      int rowsA↑ected = objCommand.ExecuteNonQuery();
      objConnection.Close();

      if (rowsA↑ected > 0)
        blnResult = true;
    }
    catch(SqlException objEx)
    {
      throw new DalException(objEx);
    }

    return blnResult;
  }

  /// <summary>
  /// Adds the relevant parameters to the specified <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see> object
```

```
/// setting the values to ones of the specified <see cref="StudyPlanning.DAL.StudyPlans.Period"/> object.
/// </summary>
/// <param name="period">
/// The <see cref="StudyPlanning.DAL.StudyPlans.Period"/> object containing the values that the parameters have to be
/// initialized to.
/// </param>
/// <param name="objCommand">
/// The <see cref="System.Data.SqlClient.SqlCommand"/> object to which parameters should be added.
/// </param>
/// <param name="isOriginal">
/// <strong>true</strong> if the parameters to be added represent original
/// data (for handling concurrent programming issues); otherwise, <strong>false</strong>.</param>
/// <returns>The <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see>
/// object with the relevant parameters added.</returns>
private static void AddParameters(Period period, SqlCommand objCommand, bool isOriginal)
{
  string paramName = "";
  SqlParameter objParam;

  //StudyPlan_ID
  paramName = "StudyPlan_Period_ID";
  objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

  if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

  objParam.Value = period.StudyPlan_Period_ID.Value;

  //StudyPlan_ID
  paramName = "StudyPlan_ID";
  objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

  if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

  objParam.Value = period.StudyPlan_ID.Value;

  //Period_ID
  paramName = "Period_ID";
  objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

  if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

  if (period.Period_ID.IsNull)
    objParam.Value = System.DBNull.Value;
  else
    objParam.Value = period.Period_ID.Value;

  //Project_ID
  paramName = "Project_ID";
  objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

  if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

  if (period.Project_ID.IsNull)
    objParam.Value = System.DBNull.Value;
  else
    objParam.Value = period.Project_ID.Value;

  //Log
  objCommand = period.Log.AddParameters(objCommand, isOriginal);
}

/// <summary>
/// Validates the current <see cref="StudyPlanning.DAL.StudyPlans.Period"/> object. If the
/// value of either of the private properties is null a
/// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
/// </summary>
private void Validate()
{
  try
  {
    _StudyPlan_Period_ID.Validate("StudyPlan_Period_ID");
    _StudyPlan_ID.Validate("StudyPlan_ID");
    _Period_ID.Validate("Period_ID");
    _Project_ID.Validate("Project_ID");
    _Log.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }
}

/// <summary>
/// Retrieves a list of study plan period identifiers representing the periods of the study plan
/// having the specified ID.
/// </summary>
/// <param name="studyPlan_ID">Identification of the study plan for which to retrieve a list of
/// study plan period identifiers.</param>
/// <returns>An array (possibly empty) of study plan period identifiers.</returns>
public static Guid[] GetPeriods(Guid studyPlan_ID)
{
  Guid[] periods = null;

  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);
```

```
      SqlCommand objCommand = new SqlCommand("StudyPlan_Period_GetPeriods", objConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      SqlParameter objParam;

      objParam = objCommand.Parameters.Add("@StudyPlan_ID", SqlDbType.UniqueIdenti↓er, 16);
      objParam.Value = studyPlan_ID;

      DataSet objDataSet = new DataSet("Result");
      SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

      try
      {
        objConnection.Open();
        objAdap.Fill(objDataSet, "Result");
        objConnection.Close();

        int numRows = objDataSet.Tables["Result"].Rows.Count;
        periods = new Guid[numRows];

        for(int i=0; i<numRows; i++)
        {
          periods[i] = (Guid)objDataSet.Tables["Result"].Rows[i]["StudyPlan_Period_ID"];
        }
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }
      return periods;
    }

    #endregion //Methods
  }
}
```

1.24.2 PeriodCourse

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.StudyPlans
{
  /// <summary>
  /// Represents a course in a period of a study plan.
  /// </summary>
  public class PeriodCourse : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalGuid _StudyPlan_PeriodCourse_ID = new DalGuid(false);
    private DalGuid _StudyPlan_Period_ID = new DalGuid(false);
    private DalGuid _CourseVersion_ID = new DalGuid(false);
    private DalInt _Part = new DalInt(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Initializes a new instance of the <see cref="StudyPlanning.DAL.StudyPlans.PeriodCourse"/>
    /// class.
    /// </summary>
    public PeriodCourse()
    {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the ID of study plan period course.
    /// </summary>
    public DalGuid StudyPlan_PeriodCourse_ID
    {
      get { return _StudyPlan_PeriodCourse_ID; }
    }

    /// <summary>
    /// Gets the ID of the study plan period.
    /// </summary>
    public DalGuid StudyPlan_Period_ID
    {
      get { return _StudyPlan_Period_ID; }
    }

    /// <summary>
    /// Gets the ID of the course version.
    /// </summary>
    public DalGuid CourseVersion_ID
```

```
{
  get { return _CourseVersion_ID; }
}

/// <summary>
/// Gets the Part attribute.
/// </summary>
public DalInt Part
{
  get { return _Part; }
}

/// <summary>
/// Gets the data log of the course.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Methods

/// <summary>
/// Creates a new period course in the database using the values of the properties
/// of the current instance.
/// </summary>
public void Create()
{
  try
  {
    this.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }

  SqlCommand objCommand = new SqlCommand("StudyPlan_PeriodCourse_Insert", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, false);

  try
  {
    objConnection.Open();
    objCommand.ExecuteNonQuery();
    objConnection.Close();
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
}

#region Retrieve Methods

/// <summary>
/// Retrieves period course from the database using the value of the
/// StudyPlan_PeriodCourse_ID property of the current instance.
/// </summary>
public void Retrieve()
{
  try
  {
    this._StudyPlan_PeriodCourse_ID.Validate("StudyPlan_PeriodCourse_ID");
  }
  catch (DalException excp)
  {
    throw excp;
  }

  try
  {
      RetrieveExecute(_StudyPlan_PeriodCourse_ID.Value, this, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
}

/// <summary>
/// Retrieves the period course with the specified identifier from the database.
/// </summary>
/// <param name="studyPlan_PeriodCourse_ID">The globally unique identifier of the period course.</param>
public static PeriodCourse Retrieve(Guid studyPlan_PeriodCourse_ID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  PeriodCourse objPeriodCourse = new PeriodCourse();

  try
  {
    RetrieveExecute(studyPlan_PeriodCourse_ID, objPeriodCourse, objConnection);
  }
  catch (DalException objExc)
```

```
    {
      throw objExc;
    }

    return objPeriodCourse;
  }

  private static void RetrieveExecute(
    Guid studyPlan_PeriodCourse_ID,
    PeriodCourse objPeriodCourse,
    SqlConnection sqlConnection)
  {
    SqlCommand objCommand = new SqlCommand("StudyPlan_PeriodCourse_Select", sqlConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam;

    objParam = objCommand.Parameters.Add("@StudyPlan_PeriodCourse_ID", SqlDbType.UniqueIdenti↓er, 16);
    objParam.Value = studyPlan_PeriodCourse_ID;

    DataSet objDataSet = new DataSet("StudyPlan_PeriodCourse");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
      sqlConnection.Open();
      objAdap.Fill(objDataSet, "StudyPlan_PeriodCourse");
      sqlConnection.Close();
      DataRow objDataRow = objDataSet.Tables["StudyPlan_PeriodCourse"].Rows[0];

      objPeriodCourse.StudyPlan_PeriodCourse_ID.Value = (Guid)objDataRow["StudyPlan_PeriodCourse_ID"];
      objPeriodCourse.StudyPlan_Period_ID.Value = (Guid)objDataRow["StudyPlan_Period_ID"];
      objPeriodCourse.CourseVersion_ID.Value = (Guid)objDataRow["CourseVersion_ID"];
      objPeriodCourse.Part.Value = (int)objDataRow["Part"];

      objPeriodCourse.Log.SetValues(objDataRow);
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
  }

  #endregion //Retrive Methods

  /// <summary>
  /// Updates the current period course in the database using the original
  /// study plan period to resolve possible concurrency issues.
  /// </summary>
  /// <param name="originalStudyPlan_PeriodCourse">The original <see cref="StudyPlanning.DAL.StudyPlans.PeriodCourse"/> object.</param>
  /// <returns><strong>true</strong> if the update was succesfully executed; otherwise, <strong>false</strong>.</returns>
  public bool Update(PeriodCourse originalStudyPlan_PeriodCourse)
  {
    try
    {
      this.Validate();
    }
    catch (DalException excp)
    {
      throw excp;
    }

    bool blnResult = false;

    SqlCommand objCommand = new SqlCommand("StudyPlan_PeriodCourse_Update", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, false);
    AddParameters(originalStudyPlan_PeriodCourse, objCommand, true);

    int rowsA↑ected;

    try
    {
      objConnection.Open();
      rowsA↑ected = objCommand.ExecuteNonQuery();
      objConnection.Close();

      if (rowsA↑ected > 0)
        blnResult = true;
    }
    catch(SqlException excp)
    {
      throw new DalException(excp);
    }

    return blnResult;
  }

  /// <summary>
  /// Deletes the current period course from the database.
  /// </summary>
  /// <returns><strong>true</strong> if the deletion is succesfully executed; otherwise, <strong>false</strong>.</returns>
  public bool Delete()
  {
    bool blnResult = false;

    try
    {
```

```
      this.Validate();
    }
    catch (DalException excp)
    {
      throw excp;
    }

    SqlCommand objCommand = new SqlCommand("StudyPlan_PeriodCourse_Delete", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, true);

    try
    {
      objConnection.Open();
      int rowsA↑ected = objCommand.ExecuteNonQuery();
      objConnection.Close();

      if (rowsA↑ected > 0)
        blnResult = true;
    }
    catch(SqlException objEx)
    {
      throw new DalException(objEx);
    }

    return blnResult;
  }

  /// <summary>
  /// Adds the relevant parameters to the specified <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see> object
  /// setting the values to ones of the specified <see cref="StudyPlanning.DAL.StudyPlans.PeriodCourse"/> object.
  /// </summary>
  /// <param name="periodCourse">
  /// The <see cref="StudyPlanning.DAL.StudyPlans.PeriodCourse"/> object containing the values that the parameters have to be
  /// initialized to.
  /// </param>
  /// <param name="objCommand">
  /// The <see cref="System.Data.SqlClient.SqlCommand"/> object to which parameters should be added.
  /// </param>
  /// <param name="isOriginal">
  /// <strong>true</strong> if the parameters to be added represent original
  /// data (for handling concurrent programming issues); otherwise, <strong>false</strong>.</param>
  /// <returns>The <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see>
  /// object with the relevant parameters added.</returns>
  private static void AddParameters(PeriodCourse periodCourse, SqlCommand objCommand, bool isOriginal)
  {
    string paramName = "";
    SqlParameter objParam;

    //StudyPlan_PeriodCourse_ID
    paramName = "StudyPlan_PeriodCourse_ID";
    objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

    if (isOriginal)
      objParam.ParameterName = "@Original_" + paramName;

    objParam.Value = periodCourse.StudyPlan_PeriodCourse_ID.Value;

    //StudyPlan_Period_ID
    paramName = "StudyPlan_Period_ID";
    objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

    if (isOriginal)
      objParam.ParameterName = "@Original_" + paramName;

    objParam.Value = periodCourse.StudyPlan_Period_ID.Value;

    //CourseVersion_ID
    paramName = "CourseVersion_ID";
    objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

    if (isOriginal)
      objParam.ParameterName = "@Original_" + paramName;

    objParam.Value = periodCourse.CourseVersion_ID.Value;

    //Part
    paramName = "Part";
    objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.Int, 4);

    if (isOriginal)
      objParam.ParameterName = "@Original_" + paramName;

    objParam.Value = periodCourse.Part.Value;

    //Log
    objCommand = periodCourse.Log.AddParameters(objCommand, isOriginal);
  }

  /// <summary>
  /// Validates the current <see cref="StudyPlanning.DAL.StudyPlans.PeriodCourse"/> object. If the
  /// value of either of the private properties is null a
  /// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
  /// </summary>
  private void Validate()
  {
    try
    {
```

```
        _StudyPlan_PeriodCourse_ID.Validate("StudyPlan_PeriodCourse_ID");
        _StudyPlan_Period_ID.Validate("StudyPlan_Period_ID");
        _CourseVersion_ID.Validate("CourseVersion_ID");
        _Part.Validate("Part");
        _Log.Validate();
      }
      catch (DalException excp)
      {
        throw excp;
      }
    }

    /// <summary>
    /// Retrieves a list of row identifiers representing the courses of the study plan period
    /// having the specified ID.
    /// </summary>
    /// <param name="studyPlan_Period_ID">Identification of the study plan period for which to
    /// retrieve a list of row identifiers.</param>
    /// <returns>An array (possibly empty) of row identifiers.</returns>
    public static Guid[] GetCourses(Guid studyPlan_Period_ID)
    {
      Guid[] courses = null;

      string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
      SqlConnection objConnection = new SqlConnection(connString);

      SqlCommand objCommand = new SqlCommand("StudyPlan_Period_GetPeriods", objConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      SqlParameter objParam;

      objParam = objCommand.Parameters.Add("@StudyPlan_ID", SqlDbType.UniqueIdenti↓er, 16);
      objParam.Value = studyPlan_Period_ID;

      DataSet objDataSet = new DataSet("Result");
      SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

      try
      {
        objConnection.Open();
        objAdap.Fill(objDataSet, "Result");
        objConnection.Close();

        int numRows = objDataSet.Tables["Result"].Rows.Count;
        courses = new Guid[numRows];

        for(int i=0; i<numRows; i++)
        {
          courses[i] = (Guid)objDataSet.Tables["Result"].Rows[i]["StudyPlan_PeriodCourse_ID"];
        }
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }
      return courses;
    }

    #endregion //Methods
  }
}
```

1.25 StudyTypes

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.StudyTypes
{
  /// <summary>
  /// Represents a study type.
  /// </summary>
  public class StudyType : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalInt _StudyType_ID = new DalInt(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Initializes a new instance of the <see cref="StudyPlanning.DAL.StudyTypes.StudyType"/>
    /// class.
    /// </summary>
    public StudyType()
    {
      _Log = new DataLog();
    }
```

```
#endregion

#region Public Properties

/// <summary>
/// Gets the ID of study type.
/// </summary>
public DalInt StudyType_ID
{
  get { return _StudyType_ID; }
}

/// <summary>
/// Gets the data log of the study type.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Methods

#region Retrieve Methods

/// <summary>
/// Retrieves the study type from the database using the value of the
/// StudyType_ID property of the current instance.
/// </summary>
public void Retrieve()
{
  try
  {
    this._StudyType_ID.Validate("StudyType_ID");
  }
  catch (DalException excp)
  {
    throw excp;
  }

  try
  {
      RetrieveExecute(_StudyType_ID.Value, this, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
}

/// <summary>
/// Retrieves the study type with the specified identifier from the database.
/// </summary>
/// <param name="studyType_ID">The globally unique identifier of the study type.</param>
public static StudyType Retrieve(int studyType_ID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  StudyType objSt = new StudyType();

  try
  {
    RetrieveExecute(studyType_ID, objSt, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }

  return objSt;
}

private static void RetrieveExecute(
  int studyType_ID,
  StudyType objStudyType,
  SqlConnection sqlConnection)
{
  SqlCommand objCommand = new SqlCommand("StudyType_Select", sqlConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  SqlParameter objParam;

  objParam = objCommand.Parameters.Add("@StudyType_ID", SqlDbType.Int, 4);
  objParam.Value = studyType_ID;

  DataSet objDataSet = new DataSet("StudyType");
  SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

  try
  {
    sqlConnection.Open();
    objAdap.Fill(objDataSet, "StudyType");
    sqlConnection.Close();
    DataRow objDataRow = objDataSet.Tables["StudyType"].Rows[0];

    objStudyType.StudyType_ID.Value = (int)objDataRow["StudyType_ID"];
    objStudyType.Log.SetValues(objDataRow);
```

```
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }
    }

    #endregion //Retrive Methods

    #endregion //Methods
  }
}
```

1.25.1 ProjectType

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.StudyTypes
{
  /// <summary>
  /// Represents the StudyType_ProjectType table.
  /// </summary>
  public class ProjectType : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalInt _StudyType_ProjectType_ID = new DalInt(false);
    private DalInt _StudyTypeVersion_ID = new DalInt(false);
    private DalInt _ProjectType_ID = new DalInt(false);
    private DalInt _SequenceNumber = new DalInt(false);
    private DalInt _Months = new DalInt(false);
    private DalGuid _PlacementPoint_ID = new DalGuid(false);
    private DalGuid _WorkloadPoint_ID = new DalGuid(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Initializes a new instance of the <see cref="StudyPlanning.DAL.StudyTypes.ProjectType"/>
    /// class.
    /// </summary>
    public ProjectType()
    {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the StudyType_ProjectType_ID.
    /// </summary>
    public DalInt StudyType_ProjectType_ID
    {
      get { return _StudyType_ProjectType_ID; }
    }

    /// <summary>
    /// Gets the StudyTypeVersion_ID
    /// </summary>
    public DalInt StudyTypeVersion_ID
    {
      get { return _StudyTypeVersion_ID; }
    }

    /// <summary>
    /// Gets the ProjectType_ID.
    /// </summary>
    public DalInt ProjectType_ID
    {
      get { return _ProjectType_ID; }
    }

    /// <summary>
    /// Gets the SequenceNumber.
    /// </summary>
    public DalInt SequenceNumber
    {
      get { return _SequenceNumber; }
    }

    /// <summary>
    /// Gets the number of months.
    /// </summary>
    public DalInt Months
    {
      get { return _Months; }
    }
```

```
/// <summary>
/// Gets the ID of the point denoting the placement.
/// </summary>
public DalGuid PlacementPoint_ID
{
  get { return _PlacementPoint_ID; }
}

/// <summary>
/// Gets the ID of the point denoting the total workload
/// </summary>
public DalGuid WorkloadPoint_ID
{
  get { return _WorkloadPoint_ID; }
}

/// <summary>
/// Gets the data log of the StudyType_ProjectType.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Methods

#region Retrieve Methods

/// <summary>
/// Retrieves the StudyType_ProjectType from the database using the value of the
/// StudyType_ProjectType_ID property of the current instance.
/// </summary>
public void Retrieve()
{
  try
  {
    this._StudyType_ProjectType_ID.Validate("StudyType_ProjectType_ID");
  }
  catch (DalException excp)
  {
    throw excp;
  }

  try
  {
    RetrieveExecute(_StudyType_ProjectType_ID.Value, this, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
}

/// <summary>
/// Retrieves the StudyType_ProjectType with the specified identifier from the database.
/// </summary>
/// <param name="StudyType_ProjectType_ID">Identification of the
/// StudyType_ProjectType.</param>
public static ProjectType Retrieve(int StudyType_ProjectType_ID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  ProjectType objProjectType = new ProjectType();

  try
  {
    RetrieveExecute(StudyType_ProjectType_ID, objProjectType, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }

  return objProjectType;
}

private static void RetrieveExecute(
  int ProjectType_ID,
  ProjectType objProjectType,
  SqlConnection sqlConnection)
{
  SqlCommand objCommand = new SqlCommand("StudyType_ProjectType_Select", sqlConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  SqlParameter objParam;

  objParam = objCommand.Parameters.Add("@StudyType_ProjectType_ID", SqlDbType.Int);
  objParam.Value = ProjectType_ID;

  DataSet objDataSet = new DataSet("Result");
  SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

  DataRow objDataRow;

  try
  {
```

```
        sqlConnection.Open();
        objAdap.Fill(objDataSet, "Result");
        sqlConnection.Close();
        objDataRow = objDataSet.Tables["Result"].Rows[0];
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }

      objProjectType.StudyType_ProjectType_ID.Value = (int)objDataRow["StudyType_ProjectType_ID"];
      objProjectType.StudyTypeVersion_ID.Value   = (int)objDataRow["StudyTypeVersion_ID"];
      objProjectType.ProjectType_ID.Value      = (int)objDataRow["ProjectType_ID"];
      objProjectType.SequenceNumber.Value      = (int)objDataRow["SequenceNumber"];
      objProjectType.Months.Value          = (int)objDataRow["Months"];
      objProjectType.PlacementPoint_ID.Value    = (Guid)objDataRow["PlacementPoint_ID"];
      objProjectType.WorkloadPoint_ID.Value     = (Guid)objDataRow["WorkloadPoint_ID"];
      objProjectType.Log.SetValues(objDataRow);

    }

    #endregion //Retrive Methods

    #endregion //Methods

    /// <summary>
    /// Retrieves a list of project types which are associated with the specified
    /// study type version.
    /// </summary>
    /// <param name="ProjectType_ID">The ID of the study type version for which
    /// to retrieve a list of project types.</param>
    /// <returns>An array containing primary key IDs of the rows which are
    /// related to the specified study type version ID.</returns>
    public static int[] GetProjectTypes(int ProjectType_ID)
    {
      int[] projectTypes;

      string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
      SqlConnection objConnection = new SqlConnection(connString);

      SqlCommand objCommand = new SqlCommand("StudyType_ProjectType_GetProjectTypes", objConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      SqlParameter objParam;

      objParam = objCommand.Parameters.Add("@ProjectType_ID", SqlDbType.Int, 4);
      objParam.Value = ProjectType_ID;

      DataSet objDataSet = new DataSet("ProjectTypes");
      SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

      try
      {
        objConnection.Open();
        objAdap.Fill(objDataSet, "ProjectTypes");
        objConnection.Close();

        int numRows = objDataSet.Tables["ProjectTypes"].Rows.Count;
        projectTypes = new int[numRows];

        for(int i=0; i<numRows; i++)
        {
          projectTypes[i] = (int)objDataSet.Tables["ProjectTypes"].Rows[i]["StudyType_ProjectType_ID"];
        }
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }

      return projectTypes;
    }

  }
}
```

1.25.2 StudyTypeVersion

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.StudyTypes
{
  /// <summary>
  /// Represents a version of a study type.
  /// </summary>
  public class StudyTypeVersion : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalInt _StudyTypeVersion_ID = new DalInt(false);
    private DalInt _StudyType_ID = new DalInt(false);
    private DalInt _Version = new DalInt(false);
```

```
private DalStringLocalizable _Name = new DalStringLocalizable(false);
private DalGuid _Point_ID = new DalGuid(false);
private DataLog _Log;

#endregion //Private Properties

#region Constructors

/// <summary>
/// Initializes a new instance of the <see cref="StudyPlanning.DAL.StudyTypes.StudyTypeVersion"/>
/// class.
/// </summary>
public StudyTypeVersion()
{
  _Log = new DataLog();
}

#endregion

#region Public Properties

/// <summary>
/// Gets the ID of the study type version.
/// </summary>
public DalInt StudyTypeVersion_ID
{
  get { return _StudyTypeVersion_ID; }
}

/// <summary>
/// Gets the ID of the study type.
/// </summary>
public DalInt StudyType_ID
{
  get { return _StudyType_ID; }
}

/// <summary>
/// Gets the version number.
/// </summary>
public DalInt Version
{
  get { return _Version; }
}

/// <summary>
/// Gets the name of the study type version.
/// </summary>
public DalStringLocalizable Name
{
  get { return _Name; }
}

/// <summary>
/// Gets the ID of the point.
/// </summary>
public DalGuid Point_ID
{
  get { return _Point_ID; }
}

/// <summary>
/// Gets the data log of the study type version.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Methods

#region Retrieve Methods

/// <summary>
/// Retrieves the study type version from the database using the value of the
/// StudyTypeVersion_ID property of the current instance.
/// </summary>
public void Retrieve()
{
  try
  {
    this._StudyTypeVersion_ID.Validate("StudyTypeVersion_ID");
  }
  catch (DalException excp)
  {
    throw excp;
  }

  try
  {
      RetrieveExecute(_StudyTypeVersion_ID.Value, this, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
}
```

```
    /// <summary>
    /// Retrieves the study type version with the specified identifier from the database.
    /// </summary>
    /// <param name="studyTypeVersion_ID">The globally unique identifier of the study type version.</param>
    public static StudyTypeVersion Retrieve(int studyTypeVersion_ID)
    {
      string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
      SqlConnection objConnection = new SqlConnection(connString);

      StudyTypeVersion objStudyTypeVersion = new StudyTypeVersion();

      try
      {
        RetrieveExecute(studyTypeVersion_ID, objStudyTypeVersion, objConnection);
      }
      catch (DalException objExc)
      {
        throw objExc;
      }

      return objStudyTypeVersion;
    }

    private static void RetrieveExecute(
      int studyTypeVersion_ID,
      StudyTypeVersion objStudyTypeVersion,
      SqlConnection sqlConnection)
    {
      SqlCommand objCommand = new SqlCommand("StudyTypeVersion_Select", sqlConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      SqlParameter objParam;

      objParam = objCommand.Parameters.Add("@StudyTypeVersion_ID", SqlDbType.Int, 4);
      objParam.Value = studyTypeVersion_ID;

      DataSet objDataSet = new DataSet("StudyType_StudyTypeVersion");
      SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

      try
      {
        sqlConnection.Open();
        objAdap.Fill(objDataSet, "StudyType_StudyTypeVersion");
        sqlConnection.Close();
        DataRow objDataRow = objDataSet.Tables["StudyType_StudyTypeVersion"].Rows[0];

        objStudyTypeVersion.StudyTypeVersion_ID.Value = (int)objDataRow["StudyTypeVersion_ID"];
        objStudyTypeVersion.StudyType_ID.Value = (int)objDataRow["StudyType_ID"];
        objStudyTypeVersion.Version.Value = (int)objDataRow["Version"];
        objStudyTypeVersion.Name.LoadXml(objDataRow["Name"].ToString());
        objStudyTypeVersion.Point_ID.Value = (Guid)objDataRow["Point_ID"];
        objStudyTypeVersion.Log.SetValues(objDataRow);
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }
    }

    #endregion //Retrive Methods

    #endregion //Methods
  }
}
```

1.25.3 TechnicalLine

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.StudyTypes
{
  /// <summary>
  /// Represents a technical line of a study type.
  /// </summary>
  public class TechnicalLine : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalInt _StudyType_TechnicalLine_ID = new DalInt(false);
    private DalInt _StudyTypeVersion_ID = new DalInt(false);
    private DalInt _TechnicalLine_ID = new DalInt(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Initializes a new instance of the <see cref="StudyPlanning.DAL.StudyTypes.TechnicalLine"/>
    /// class.
    /// </summary>
    public TechnicalLine()
```

```
{
  _Log = new DataLog();
}

#endregion

#region Public Properties

/// <summary>
/// Gets the ID of the study type technical line.
/// </summary>
public DalInt StudyType_TechnicalLine_ID
{
  get { return _StudyType_TechnicalLine_ID; }
}

/// <summary>
/// Gets the ID of the study type version.
/// </summary>
public DalInt StudyTypeVersion_ID
{
  get { return _StudyTypeVersion_ID; }
}

/// <summary>
/// Gets the ID of the technical line.
/// </summary>
public DalInt TechnicalLine_ID
{
  get { return _TechnicalLine_ID; }
}

/// <summary>
/// Gets the data log of the study type technical line.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Methods

#region Retrieve Methods

/// <summary>
/// Retrieves the study type technical line from the database using the value of the
/// StudyType_TechnicalLine_ID property of the current instance.
/// </summary>
public void Retrieve()
{
  try
  {
    this._StudyType_TechnicalLine_ID.Validate("StudyType_TechnicalLine_ID");
  }
  catch (DalException excp)
  {
    throw excp;
  }

  try
  {
      RetrieveExecute(_StudyType_TechnicalLine_ID.Value, this, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
}

/// <summary>
/// Retrieves the study type technical line with the specified identifier from the database.
/// </summary>
/// <param name="studyType_TechnicalLine_ID">The globally unique identifier of the study type technical line.</param>
public static TechnicalLine Retrieve(int studyType_TechnicalLine_ID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  TechnicalLine objTechnicalLine = new TechnicalLine();

  try
  {
    RetrieveExecute(studyType_TechnicalLine_ID, objTechnicalLine, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }

  return objTechnicalLine;
}

private static void RetrieveExecute(
  int studyType_TechnicalLine_ID,
  TechnicalLine objTechnicalLine,
  SqlConnection sqlConnection)
{
```

```
      SqlCommand objCommand = new SqlCommand("StudyType_TechnicalLine_Select", sqlConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      SqlParameter objParam;

      objParam = objCommand.Parameters.Add("@StudyType_TechnicalLine_ID", SqlDbType.Int, 4);
      objParam.Value = studyType_TechnicalLine_ID;

      DataSet objDataSet = new DataSet("StudyType_TechnicalLine");
      SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

      try
      {
        sqlConnection.Open();
        objAdap.Fill(objDataSet, "StudyType_TechnicalLine");
        sqlConnection.Close();
        DataRow objDataRow = objDataSet.Tables["StudyType_TechnicalLine"].Rows[0];

        objTechnicalLine.StudyType_TechnicalLine_ID.Value = (int)objDataRow["StudyType_TechnicalLine_ID"];
        objTechnicalLine.StudyTypeVersion_ID.Value = (int)objDataRow["StudyTypeVersion_ID"];
        objTechnicalLine.TechnicalLine_ID.Value = (int)objDataRow["TechnicalLine_ID"];
        objTechnicalLine.Log.SetValues(objDataRow);
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }
    }

    #endregion //Retrive Methods

    #endregion //Methods
  }
}
```

1.26 TechnicalFields

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.TechnicalFields
{
  /// <summary>
  /// Represents the TechnicalField table.
  /// </summary>
  public class TechnicalField : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalInt _TechnicalField_ID = new DalInt(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.TechnicalFields.TechnicalField"/>
    /// class.
    /// </summary>
    public TechnicalField() {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the TechnicalField_ID.
    /// </summary>
    public DalInt TechnicalField_ID
    {
      get { return _TechnicalField_ID; }
    }

    /// <summary>
    /// Gets the data log of the table.
    /// </summary>
    public DataLog Log
    {
      get { return _Log; }
    }

    #endregion //Public Properties

    #region Public Methods

    #region Retrieve Methods

    /// <summary>
    /// Retrieves the TechnicalField from the database using the value of the
```

```
/// TechnicalField_ID property of the current instance
/// </summary>
public void Retrieve()
{
  try
  {
    this._TechnicalField_ID.Validate("TechnicalField_ID");
  }
  catch (DalException e)
  {
    throw e;
  }

  try
  {
    RetrieveExecute(_TechnicalField_ID.Value, this, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
}

/// <summary>
/// Retrieves the TechnicalField with the specified identifier from the database.
/// </summary>
/// <param name="technicalField_ID"></param>
public static TechnicalField Retrieve(int technicalField_ID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  TechnicalField objTechnicalField = new TechnicalField();

  try
  {
    RetrieveExecute(technicalField_ID, objTechnicalField, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
  return objTechnicalField;
}

private static void RetrieveExecute(
  int technicalField_ID,
  TechnicalField technicalField,
  SqlConnection sqlConnection)
{
  DataRow objDataRow;

  SqlCommand objCommand = new SqlCommand("TechnicalField_Select", sqlConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  SqlParameter objParam;

  objParam = objCommand.Parameters.Add("@TechnicalField_ID", SqlDbType.Int);
  objParam.Value = technicalField._TechnicalField_ID.Value;

  DataSet objDataSet = new DataSet("TechnicalField");
  SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

  try
  {
    sqlConnection.Open();
    objAdap.Fill(objDataSet, "TechnicalField");
    sqlConnection.Close();
    objDataRow = objDataSet.Tables["TechnicalField"].Rows[0];
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
  technicalField._TechnicalField_ID.Value = (int)objDataRow["TechnicalField_ID"];
  technicalField._Log.SetValues(objDataRow);
}

#endregion //Retrieve Methods

  #endregion //Public Methods

#region Private Methods

/// <summary>
/// Validates the current <see cref="StudyPlanning.DAL.TechnicalFields.TechnicalField"/> object. If the
/// value of either of the private properties is null a
/// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
/// </summary>
private void Validate()
{
  try
  {
    _TechnicalField_ID.Validate("TechnicalField_ID");
    _Log.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
```

```
      }
    }
    #endregion //Private Methods
  }
}
```

1.26.1 Course

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.TechnicalFields
{
  /// <summary>
  /// Represents the TechnicalField_Course table.
  /// </summary>
  public class Course : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalInt _TechnicalField_Course_ID = new DalInt(false);
    private DalInt _TechnicalFieldVersion_ID = new DalInt(false);
    private DalGuid _Course_ID = new DalGuid(true);
    private DalString _Number = new DalString(true);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.TechnicalFields.Course"/>
    /// class.
    /// </summary>
    public Course() {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the TechnicalField_Course_ID.
    /// </summary>
    public DalInt TechnicalField_Course_ID
    {
      get { return _TechnicalField_Course_ID; }
    }

    /// <summary>
    /// Gets the TechnicalFieldVersion_ID.
    /// </summary>
    public DalInt TechnicalFieldVersion_ID
    {
      get { return _TechnicalFieldVersion_ID; }
    }

    /// <summary>
    /// Gets the Course_ID.
    /// </summary>
    public DalGuid Course_ID
    {
      get { return _Course_ID; }
    }

    /// <summary>
    /// Gets the Number.
    /// </summary>
    public DalString Number
    {
      get { return _Number; }
    }

    /// <summary>
    /// Gets the data log of the table.
    /// </summary>
    public DataLog Log
    {
      get { return _Log; }
    }

    #endregion //Public Properties

    #region Public Methods

    #region Retrieve Methods

    /// <summary>
    /// Retrieves the TechnicalField_Course from the database using the value of the
    /// TechnicalField_Course_ID property of the current instance
    /// </summary>
    public void Retrieve()
    {
```

```
    try
    {
      this._TechnicalField_Course_ID.Validate("TechnicalField_Course_ID");
    }
    catch (DalException e)
    {
      throw e;
    }

    try
    {
      RetrieveExecute(_TechnicalField_Course_ID.Value, this, objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }
  }

  /// <summary>
  /// Retrieves the TechnicalField_Course with the specified identifier from the database.
  /// </summary>
  /// <param name="TechnicalField_Course_ID"></param>
  public static Course Retrieve(int TechnicalField_Course_ID)
  {
    string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
    SqlConnection objConnection = new SqlConnection(connString);

    Course objTechnicalField_Course = new Course();

    try
    {
      RetrieveExecute(TechnicalField_Course_ID, objTechnicalField_Course, objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }
    return objTechnicalField_Course;
  }

  private static void RetrieveExecute(
    int TechnicalField_Course_ID,
    Course TechnicalField_Course,
    SqlConnection sqlConnection)
  {
    DataRow objDataRow;

    SqlCommand objCommand = new SqlCommand("TechnicalField_Course_Select", sqlConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam;

    objParam = objCommand.Parameters.Add("@TechnicalField_Course_ID", SqlDbType.Int);
    objParam.Value = TechnicalField_Course._TechnicalField_Course_ID.Value;

    DataSet objDataSet = new DataSet("TechnicalField_Course");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
      sqlConnection.Open();
      objAdap.Fill(objDataSet, "TechnicalField_Course");
      sqlConnection.Close();
      objDataRow = objDataSet.Tables["TechnicalField_Course"].Rows[0];
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
    TechnicalField_Course._TechnicalField_Course_ID.Value = (int)objDataRow["TechnicalField_Course_ID"];
    TechnicalField_Course._TechnicalFieldVersion_ID.Value       = (int)objDataRow["TechnicalFieldVersion_ID"];

    if (objDataRow["Course_ID"].Equals(DBNull.Value))
      TechnicalField_Course._Course_ID.IsNull = true;
    else
      TechnicalField_Course._Course_ID.Value = (Guid)objDataRow["Course_ID"];

    if (objDataRow["Number"].Equals(DBNull.Value))
      TechnicalField_Course._Number.IsNull = true;
    else
      TechnicalField_Course._Number.Value = (string)objDataRow["Number"];

    TechnicalField_Course._Log.SetValues(objDataRow);
  }

  /// <summary>
  /// Retrieves a list of ids for a given technical field version.
  /// </summary>
  /// <param name="technicalFieldVersion_ID">Identification of the
  /// <see cref="StudyPlanning.DAL.TechnicalFields.TechnicalFieldVersion"/>.</param>
  /// <returns>An array of <see cref="StudyPlanning.DAL.TechnicalFields.Course"/> identifications.</returns>
  public static int[] GetIDFromVersion(int technicalFieldVersion_ID)
  {
    string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
    SqlConnection objConnection = new SqlConnection(connString);

    SqlCommand objCommand = new SqlCommand("TechnicalField_Course_GetIDFromVersion", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;
```

```
      SqlParameter objParam = objCommand.Parameters.Add("@TechnicalFieldVersion_ID", SqlDbType.Int);
      objParam.Value = technicalFieldVersion_ID;

      DataSet objDataSet = new DataSet("Result");
      SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

      try
      {
        objConnection.Open();
        objAdap.Fill(objDataSet, "Result");
        objConnection.Close();
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }
      int intRows = objDataSet.Tables["Result"].Rows.Count;
      int[] ids = new int[intRows];

      for(int i=0; i < intRows; i++)
      {
        ids[i] = (int)objDataSet.Tables["Result"].Rows[i]["TechnicalField_Course_ID"];
      }
      return ids;
    }

    #endregion //Retrieve Methods

      #endregion //Public Methods

    #region Private Methods

    /// <summary>
    /// Validates the current <see cref="StudyPlanning.DAL.TechnicalFields.Course"/> object. If the
    /// value of either of the private properties is null a
    /// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
    /// </summary>
    private void Validate()
    {
      try
      {
        _TechnicalField_Course_ID.Validate("TechnicalField_Course_ID");
        _Log.Validate();
      }
      catch (DalException excp)
      {
        throw excp;
      }
    }
    #endregion //Private Methods
  }
}
```

1.26.2 TechnicalFieldVersion

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.TechnicalFields
{
  /// <summary>
  /// Represents the TechnicalFieldVersion table.
  /// </summary>
  public class TechnicalFieldVersion : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalInt _TechnicalFieldVersion_ID = new DalInt(false);
    private DalInt _TechnicalField_ID = new DalInt(false);
    private DalInt _Version = new DalInt(false);
    private DalStringLocalizable _Name = new DalStringLocalizable(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.TechnicalFields.TechnicalFieldVersion"/>
    /// class.
    /// </summary>
    public TechnicalFieldVersion() {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the TechnicalFieldVersion_ID.
    /// </summary>
    public DalInt TechnicalFieldVersion_ID
```

```
{
  get { return _TechnicalFieldVersion_ID; }
}

/// <summary>
/// Gets the TechnicalField_ID.
/// </summary>
public DalInt TechnicalField_ID
{
  get { return _TechnicalField_ID; }
}

/// <summary>
/// Gets the Version.
/// </summary>
public DalInt Version
{
  get { return _Version; }
}

/// <summary>
/// Gets the Name.
/// </summary>
public DalStringLocalizable Name
{
  get { return _Name; }
}

/// <summary>
/// Gets the data log of the table.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Public Methods

#region Retrieve Methods

/// <summary>
/// Retrieves the TechnicalFieldVersion from the database using the value of the
/// TechnicalFieldVersion_ID property of the current instance
/// </summary>
public void Retrieve()
{
  try
  {
    this._TechnicalFieldVersion_ID.Validate("TechnicalFieldVersion_ID");
  }
  catch (DalException e)
  {
    throw e;
  }

  try
  {
    RetrieveExecute(_TechnicalFieldVersion_ID.Value, this, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
}

/// <summary>
/// Retrieves the TechnicalFieldVersion with the specified identifier from the database.
/// </summary>
/// <param name="technicalFieldVersion_ID"></param>
public static TechnicalFieldVersion Retrieve(int technicalFieldVersion_ID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  TechnicalFieldVersion objTechnicalFieldVersion = new TechnicalFieldVersion();

  try
  {
    RetrieveExecute(technicalFieldVersion_ID, objTechnicalFieldVersion, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
  return objTechnicalFieldVersion;
}

private static void RetrieveExecute(
  int technicalFieldVersion_ID,
  TechnicalFieldVersion technicalFieldVersion,
  SqlConnection sqlConnection)
{
  DataRow objDataRow;

  SqlCommand objCommand = new SqlCommand("TechnicalFieldVersion_Select", sqlConnection);
  objCommand.CommandType = CommandType.StoredProcedure;
```

```
      SqlParameter objParam;

      objParam = objCommand.Parameters.Add("@TechnicalFieldVersion_ID", SqlDbType.Int);
      objParam.Value = technicalFieldVersion._TechnicalFieldVersion_ID.Value;

      DataSet objDataSet = new DataSet("TechnicalFieldVersion");
      SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

      try
      {
        sqlConnection.Open();
        objAdap.Fill(objDataSet, "TechnicalFieldVersion");
        sqlConnection.Close();
        objDataRow = objDataSet.Tables["TechnicalFieldVersion"].Rows[0];
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }
      technicalFieldVersion._TechnicalFieldVersion_ID.Value = (int)objDataRow["TechnicalFieldVersion_ID"];
      technicalFieldVersion._TechnicalField_ID.Value     = (int)objDataRow["TechnicalField_ID"];
      technicalFieldVersion._Version.Value          = (int)objDataRow["Version"];
      technicalFieldVersion._Name.LoadXml((string)objDataRow["Name"]);
      technicalFieldVersion._Log.SetValues(objDataRow);
    }

    #endregion //Retrieve Methods

      #endregion //Public Methods

    #region Private Methods

    /// <summary>
    /// Validates the current <see cref="StudyPlanning.DAL.TechnicalFields.TechnicalFieldVersion"/> object. If the
    /// value of either of the private properties is null a
    /// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
    /// </summary>
    private void Validate()
    {
      try
      {
        _TechnicalFieldVersion_ID.Validate("TechnicalFieldVersion_ID");
        _Log.Validate();
      }
      catch (DalException excp)
      {
        throw excp;
      }
    }
    #endregion //Private Methods
  }
}
```

1.27 TechnicalLines

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.TechnicalLines
{
  /// <summary>
  /// Represents the TechnicalLine table.
  /// </summary>
  public class TechnicalLine : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalInt _TechnicalLine_ID = new DalInt(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.TechnicalLines.TechnicalLine"/>
    /// class.
    /// </summary>
    public TechnicalLine() {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the TechnicalLine_ID.
    /// </summary>
    public DalInt TechnicalLine_ID
    {
      get { return _TechnicalLine_ID; }
```

```
}

/// <summary>
/// Gets the data log of the table.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Public Methods

#region Retrieve Methods

/// <summary>
/// Retrieves the TechnicalLine from the database using the value of the
/// TechnicalLine_ID property of the current instance
/// </summary>
public void Retrieve()
{
  try
  {
    this._TechnicalLine_ID.Validate("TechnicalLine_ID");
  }
  catch (DalException e)
  {
    throw e;
  }

  try
  {
    RetrieveExecute(_TechnicalLine_ID.Value, this, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
}

/// <summary>
/// Retrieves the TechnicalLine with the specified identifier from the database.
/// </summary>
/// <param name="technicalLine_ID"></param>
public static TechnicalLine Retrieve(int technicalLine_ID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  TechnicalLine objTechnicalLine = new TechnicalLine();

  try
  {
    RetrieveExecute(technicalLine_ID, objTechnicalLine, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
  return objTechnicalLine;
}

private static void RetrieveExecute(
  int technicalLine_ID,
  TechnicalLine technicalLine,
  SqlConnection sqlConnection)
{
  DataRow objDataRow;

  SqlCommand objCommand = new SqlCommand("TechnicalLine_Select", sqlConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  SqlParameter objParam;

  objParam = objCommand.Parameters.Add("@TechnicalLine_ID", SqlDbType.Int);
  objParam.Value = technicalLine._TechnicalLine_ID.Value;

  DataSet objDataSet = new DataSet("TechnicalLine");
  SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

  try
  {
    sqlConnection.Open();
    objAdap.Fill(objDataSet, "TechnicalLine");
    sqlConnection.Close();
    objDataRow = objDataSet.Tables["TechnicalLine"].Rows[0];
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
  technicalLine._TechnicalLine_ID.Value = (int)objDataRow["TechnicalLine_ID"];
  technicalLine._Log.SetValues(objDataRow);
}

#endregion //Retrieve Methods

  #endregion //Public Methods
```

```
    #region Private Methods

    /// <summary>
    /// Validates the current <see cref="StudyPlanning.DAL.TechnicalLines.TechnicalLine"/> object. If the
    /// value of either of the private properties is null a
    /// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
    /// </summary>
    private void Validate()
    {
      try
      {
        _TechnicalLine_ID.Validate("TechnicalLine_ID");
        _Log.Validate();
      }
      catch (DalException excp)
      {
        throw excp;
      }
    }
    #endregion //Private Methods
  }
}
```

1.27.1 PrerequisiteCourse

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.TechnicalLines
{
  /// <summary>
  /// Represents the TechnicalLine_PrerequisiteCourse table.
  /// </summary>
  public class PrerequisiteCourse : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalInt _TechnicalLine_PrerequisiteCourse_ID = new DalInt(false);
    private DalInt _TechnicalLineVersion_ID = new DalInt(false);
    private DalGuid _Course_ID = new DalGuid(true);
    private DalString _Number = new DalString(true);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.TechnicalLines.PrerequisiteCourse"/>
    /// class.
    /// </summary>
    public PrerequisiteCourse() {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the TechnicalLine_PrerequisiteCourse_ID.
    /// </summary>
    public DalInt TechnicalLine_PrerequisiteCourse_ID
    {
      get { return _TechnicalLine_PrerequisiteCourse_ID; }
    }

    /// <summary>
    /// Gets the TechnicalLineVersion_ID.
    /// </summary>
    public DalInt TechnicalLineVersion_ID
    {
      get { return _TechnicalLineVersion_ID; }
    }

    /// <summary>
    /// Gets the Course_ID.
    /// </summary>
    public DalGuid Course_ID
    {
      get { return _Course_ID; }
    }

    /// <summary>
    /// Gets the Number.
    /// </summary>
    public DalString Number
    {
      get { return _Number; }
    }

    /// <summary>
```

```
/// Gets the data log of the table.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Public Methods

#region Retrieve Methods

/// <summary>
/// Retrieves the TechnicalLine_PrerequisiteCourse from the database using the value of the
/// TechnicalLine_PrerequisiteCourse_ID property of the current instance
/// </summary>
public void Retrieve()
{
  try
  {
    this._TechnicalLine_PrerequisiteCourse_ID.Validate("TechnicalLine_PrerequisiteCourse_ID");
  }
  catch (DalException e)
  {
    throw e;
  }

  try
  {
    RetrieveExecute(_TechnicalLine_PrerequisiteCourse_ID.Value, this, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
}

/// <summary>
/// Retrieves the TechnicalLine_PrerequisiteCourse with the specified identifier from the database.
/// </summary>
/// <param name="technicalLine_PrerequisiteCourse_ID"></param>
public static PrerequisiteCourse Retrieve(int technicalLine_PrerequisiteCourse_ID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  PrerequisiteCourse objTechnicalLine_PrerequisiteCourse = new PrerequisiteCourse();

  try
  {
    RetrieveExecute(technicalLine_PrerequisiteCourse_ID, objTechnicalLine_PrerequisiteCourse, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
  return objTechnicalLine_PrerequisiteCourse;
}

private static void RetrieveExecute(
  int technicalLine_PrerequisiteCourse_ID,
  PrerequisiteCourse technicalLine_PrerequisiteCourse,
  SqlConnection sqlConnection)
{
  DataRow objDataRow;

  SqlCommand objCommand = new SqlCommand("TechnicalLine_PrerequisiteCourse_Select", sqlConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  SqlParameter objParam;

  objParam = objCommand.Parameters.Add("@TechnicalLine_PrerequisiteCourse_ID", SqlDbType.Int);
  objParam.Value = technicalLine_PrerequisiteCourse._TechnicalLine_PrerequisiteCourse_ID.Value;

  DataSet objDataSet = new DataSet("TechnicalLine_PrerequisiteCourse");
  SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

  try
  {
    sqlConnection.Open();
    objAdap.Fill(objDataSet, "TechnicalLine_PrerequisiteCourse");
    sqlConnection.Close();
    objDataRow = objDataSet.Tables["TechnicalLine_PrerequisiteCourse"].Rows[0];
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
  technicalLine_PrerequisiteCourse._TechnicalLine_PrerequisiteCourse_ID.Value = (int)objDataRow["
    TechnicalLine_PrerequisiteCourse_ID"];
  technicalLine_PrerequisiteCourse._TechnicalLineVersion_ID.Value       = (int)objDataRow["TechnicalLineVersion_ID"];

  if (objDataRow["Course_ID"].Equals(DBNull.Value))
    technicalLine_PrerequisiteCourse._Course_ID.IsNull = true;
  else
    technicalLine_PrerequisiteCourse._Course_ID.Value = (Guid)objDataRow["Course_ID"];

  if (objDataRow["Number"].Equals(DBNull.Value))
```

```
        technicalLine_PrerequisiteCourse._Number.IsNull = true;
      else
        technicalLine_PrerequisiteCourse._Number.Value = (string)objDataRow["Number"];

      technicalLine_PrerequisiteCourse._Log.SetValues(objDataRow);
    }

    #endregion //Retrieve Methods

    /// <summary>
    /// Retrieves a list of prerequisite courses of the technical line version with the specified
    /// ID.
    /// </summary>
    /// <param name="technicalLineVersion_ID">The ID of the technical line version for which to retrieve a list of prerequisite courses.</param>
    /// <returns>An array containing IDs of the prerequisite courses of the specified technical line version.</returns>
    public static Guid[] GetCourses(int technicalLineVersion_ID)
    {
      Guid[] courses;

      string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
      SqlConnection objConnection = new SqlConnection(connString);

      SqlCommand objCommand = new SqlCommand("TechnicalLine_PrerequisiteCourse_GetCourses", objConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      SqlParameter objParam;

      objParam = objCommand.Parameters.Add("@TechnicalLineVersion_ID", SqlDbType.Int, 4);
      objParam.Value = technicalLineVersion_ID;

      DataSet objDataSet = new DataSet("Courses");
      SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

      try
      {
        objConnection.Open();
        objAdap.Fill(objDataSet, "Courses");
        objConnection.Close();

        int numRows = objDataSet.Tables["Courses"].Rows.Count;
        courses = new Guid[numRows];

        for(int i=0; i<numRows; i++)
        {
          courses[i] = (Guid)objDataSet.Tables["Courses"].Rows[i]["Course_ID"];
        }
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }

      return courses;
    }

    #endregion //Public Methods

    #region Private Methods

    /// <summary>
    /// Validates the current <see cref="StudyPlanning.DAL.TechnicalLines.PrerequisiteCourse"/> object. If the
    /// value of either of the private properties is null a
    /// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
    /// </summary>
    private void Validate()
    {
      try
      {
        _TechnicalLine_PrerequisiteCourse_ID.Validate("TechnicalLine_PrerequisiteCourse_ID");
        _Log.Validate();
      }
      catch (DalException excp)
      {
        throw excp;
      }
    }
    #endregion //Private Methods
  }
}
```

1.27.2 PrerequisiteTechnicalPackage

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.TechnicalLines
{
  /// <summary>
  /// Represents the TechnicalLine_PrerequisiteTechnicalPackage table.
  /// </summary>
  public class PrerequisiteTechnicalPackage : StudyPlanning.DAL.DbObject
  {
    #region Private Properties
```

```
private DalInt _TechnicalLine_PrerequisiteTechnicalPackage_ID = new DalInt(false);
private DalInt _TechnicalLineVersion_ID = new DalInt(false);
private DalInt _TechnicalPackage_ID = new DalInt(false);
private DataLog _Log;

#endregion //Private Properties

#region Constructors

/// <summary>
/// Creates a new instance of the <see cref="StudyPlanning.DAL.TechnicalLines.PrerequisiteTechnicalPackage"/>
/// class.
/// </summary>
public PrerequisiteTechnicalPackage() {
  _Log = new DataLog();
}

#endregion

#region Public Properties

/// <summary>
/// Gets the TechnicalLine_PrerequisiteTechnicalPackage_ID.
/// </summary>
public DalInt TechnicalLine_PrerequisiteTechnicalPackage_ID
{
  get { return _TechnicalLine_PrerequisiteTechnicalPackage_ID; }
}

/// <summary>
/// Gets the TechnicalLineVersion_ID.
/// </summary>
public DalInt TechnicalLineVersion_ID
{
  get { return _TechnicalLineVersion_ID; }
}

/// <summary>
/// Gets the TechnicalPackage_ID.
/// </summary>
public DalInt TechnicalPackage_ID
{
  get { return _TechnicalPackage_ID; }
}

/// <summary>
/// Gets the data log of the table.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Public Methods

#region Retrieve Methods

/// <summary>
/// Retrieves the TechnicalLine_PrerequisiteTechnicalPackage from the database using the value of the
/// TechnicalLine_PrerequisiteTechnicalPackage_ID property of the current instance
/// </summary>
public void Retrieve()
{
  try
  {
    this._TechnicalLine_PrerequisiteTechnicalPackage_ID.Validate("TechnicalLine_PrerequisiteTechnicalPackage_ID");
  }
  catch (DalException e)
  {
    throw e;
  }

  try
  {
    RetrieveExecute(_TechnicalLine_PrerequisiteTechnicalPackage_ID.Value, this, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
}

/// <summary>
/// Retrieves the TechnicalLine_PrerequisiteTechnicalPackage with the specified identifier from the database.
/// </summary>
/// <param name="technicalLine_PrerequisiteTechnicalPackage_ID"></param>
public static PrerequisiteTechnicalPackage Retrieve(int technicalLine_PrerequisiteTechnicalPackage_ID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  PrerequisiteTechnicalPackage objTechnicalLine_PrerequisiteTechnicalPackage = new PrerequisiteTechnicalPackage();

  try
  {
```

```
      RetrieveExecute(technicalLine_PrerequisiteTechnicalPackage_ID, objTechnicalLine_PrerequisiteTechnicalPackage,
        objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }
    return objTechnicalLine_PrerequisiteTechnicalPackage;
  }

  private static void RetrieveExecute(
    int technicalLine_PrerequisiteTechnicalPackage_ID,
    PrerequisiteTechnicalPackage technicalLine_PrerequisiteTechnicalPackage,
    SqlConnection sqlConnection)
  {
    DataRow objDataRow;

    SqlCommand objCommand = new SqlCommand("TechnicalLine_PrerequisiteTechnicalPackage_Select", sqlConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam;

    objParam = objCommand.Parameters.Add("@TechnicalLine_PrerequisiteTechnicalPackage_ID", SqlDbType.Int);
    objParam.Value = technicalLine_PrerequisiteTechnicalPackage._TechnicalLine_PrerequisiteTechnicalPackage_ID.Value;

    DataSet objDataSet = new DataSet("TechnicalLine_PrerequisiteTechnicalPackage");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
      sqlConnection.Open();
      objAdap.Fill(objDataSet, "TechnicalLine_PrerequisiteTechnicalPackage");
      sqlConnection.Close();
      objDataRow = objDataSet.Tables["TechnicalLine_PrerequisiteTechnicalPackage"].Rows[0];
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
    technicalLine_PrerequisiteTechnicalPackage._TechnicalLine_PrerequisiteTechnicalPackage_ID.Value = (int)objDataRow["
      TechnicalLine_PrerequisiteTechnicalPackage_ID"];
    technicalLine_PrerequisiteTechnicalPackage._TechnicalLineVersion_ID.Value          = (int)objDataRow["TechnicalLineVersion_ID"
      ];
    technicalLine_PrerequisiteTechnicalPackage._TechnicalPackage_ID.Value            = (int)objDataRow["TechnicalPackage_ID"];
    technicalLine_PrerequisiteTechnicalPackage._Log.SetValues(objDataRow);
  }

  #endregion //Retrieve Methods

  /// <summary>
  /// Retrieves a list of prerequisite technical packages of the technical line version with the specified
  /// ID.
  /// </summary>
  /// <param name="technicalLineVersion_ID">The ID of the technical line version for which to retrieve a list of prerequisite technical packages.</
    param>
  /// <returns>An array containing IDs of the prerequisite technical packages of the specified technical line version.</returns>
  public static int[] GetTechnicalPackages(int technicalLineVersion_ID)
  {
    int[] tps;

    string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
    SqlConnection objConnection = new SqlConnection(connString);

    SqlCommand objCommand = new SqlCommand("TechnicalLine_PrerequisiteTechnicalPackage_GetTechnicalPackages", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam;

    objParam = objCommand.Parameters.Add("@TechnicalLineVersion_ID", SqlDbType.Int, 4);
    objParam.Value = technicalLineVersion_ID;

    DataSet objDataSet = new DataSet("TechnicalPackages");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
      objConnection.Open();
      objAdap.Fill(objDataSet, "TechnicalPackages");
      objConnection.Close();

      int numRows = objDataSet.Tables["TechnicalPackages"].Rows.Count;
      tps = new int[numRows];

      for(int i=0; i<numRows; i++)
      {
        tps[i] = (int)objDataSet.Tables["TechnicalPackages"].Rows[i]["TechnicalPackage_ID"];
      }
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }

    return tps;
  }

  /// <summary>
  /// Retrieves the ID of the row with specified technical line version ID and technical package ID.
  /// </summary>
```

```
    /// <param name="technicalLineVersion_ID">The ID of the technical line version for which to retrieve the row ID.</param>
    /// <param name="technicalPackage_ID">The ID of the technical package for which to retrieve the row ID.</param>
    /// <returns>The integer ID of the row.</returns>
    public static int GetID(int technicalLineVersion_ID, int technicalPackage_ID)
    {
      int prerequisiteTechnicalPackage_ID;

      string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
      SqlConnection objConnection = new SqlConnection(connString);

      SqlCommand objCommand = new SqlCommand("TechnicalLine_PrerequisiteTechnicalPackage_GetIDFromTechnicalPackage", objConnection
        );
      objCommand.CommandType = CommandType.StoredProcedure;

      SqlParameter objParam;

      objParam = objCommand.Parameters.Add("@TechnicalLineVersion_ID", SqlDbType.Int, 4);
      objParam.Value = technicalLineVersion_ID;

      objParam = objCommand.Parameters.Add("@TechnicalPackage_ID", SqlDbType.Int, 4);
      objParam.Value = technicalPackage_ID;

      objParam = objCommand.Parameters.Add("@Technical_PrerequisiteTechnicalPackage", SqlDbType.Int, 4);
      objParam.Direction = ParameterDirection.Output;

      try
      {
        objConnection.Open();
        objCommand.ExecuteNonQuery();
        prerequisiteTechnicalPackage_ID = (int)objCommand.Parameters["@Technical_PrerequisiteTechnicalPackage"].Value;
        objConnection.Close();
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }

      return prerequisiteTechnicalPackage_ID;
    }

    #endregion //Public Methods

    #region Private Methods

    /// <summary>
    /// Validates the current <see cref="StudyPlanning.DAL.TechnicalLines.PrerequisiteTechnicalPackage"/> object. If the
    /// value of either of the private properties is null a
    /// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
    /// </summary>
    private void Validate()
    {
      try
      {
        _TechnicalLine_PrerequisiteTechnicalPackage_ID.Validate("TechnicalLine_PrerequisiteTechnicalPackage_ID");
        _Log.Validate();
      }
      catch (DalException excp)
      {
        throw excp;
      }
    }
    #endregion //Private Methods
  }
}
```

1.27.3 PrerequisiteTechnicalPackageCourse

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.TechnicalLines
{
  /// <summary>
  /// Represents the TechnicalLine_PrerequisiteTechnicalPackageCourse table
  /// </summary>
  public class PrerequisiteTechnicalPackageCourse : StudyPlanning.DAL.DbObject
  {
    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.TechnicalLines.PrerequisiteTechnicalPackageCourse"/>
    /// class.
    /// </summary>
    public PrerequisiteTechnicalPackageCourse() { }

    #endregion

    #region Public Methods

    /// <summary>
    /// Retrieves a list of prerequisite courses related to the specified prerequisite technical
    /// package ID.
    /// </summary>
```

```
    /// <param name="prerequisiteTechnicalPackage_ID">The ID of the prerequisite technical package for which to retrieve a list of prerequisite courses
      .</param>
    /// <returns>An array containing IDs of the prerequisite courses of the specified prerequisite technical package.</returns>
    public static Guid[] GetCourses(int prerequisiteTechnicalPackage_ID)
    {
      Guid[] courses;

      string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
      SqlConnection objConnection = new SqlConnection(connString);

      SqlCommand objCommand = new SqlCommand("TechnicalLine_PrerequisiteTechnicalPackageCourse_GetCourses", objConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      SqlParameter objParam;

      objParam = objCommand.Parameters.Add("@TechnicalLine_PrerequisiteTechnicalPackage_ID", SqlDbType.Int, 4);
      objParam.Value = prerequisiteTechnicalPackage_ID;

      DataSet objDataSet = new DataSet("Courses");
      SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

      try
      {
        objConnection.Open();
        objAdap.Fill(objDataSet, "Courses");
        objConnection.Close();

        int numRows = objDataSet.Tables["Courses"].Rows.Count;
        courses = new Guid[numRows];

        for(int i=0; i<numRows; i++)
        {
          courses[i] = (Guid)objDataSet.Tables["Courses"].Rows[i]["Course_ID"];
        }
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }

      return courses;
    }

    #endregion //Public Methods
  }
}
```

1.27.4 Specialization

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.TechnicalLines
{
  /// <summary>
  /// Represents the TechnicalLine_Specialization table.
  /// </summary>
  public class Specialization : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalInt _TechnicalLine_Specialization_ID = new DalInt(false);
    private DalInt _TechnicalLineVersion_ID = new DalInt(false);
    private DalInt _Specialization_ID = new DalInt(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.TechnicalLines.Specialization"/>
    /// class.
    /// </summary>
    public Specialization() {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the TechnicalLine_Specialization_ID.
    /// </summary>
    public DalInt TechnicalLine_Specialization_ID
    {
      get { return _TechnicalLine_Specialization_ID; }
    }

    /// <summary>
    /// Gets the TechnicalLineVersion_ID.
    /// </summary>
```

```
public DalInt TechnicalLineVersion_ID
{
  get { return _TechnicalLineVersion_ID; }
}

/// <summary>
/// Gets the Specialization_ID.
/// </summary>
public DalInt Specialization_ID
{
  get { return _Specialization_ID; }
}

/// <summary>
/// Gets the data log of the table.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Public Methods

#region Retrieve Methods

/// <summary>
/// Retrieves the TechnicalLine_Specialization from the database using the value of the
/// TechnicalLine_Specialization_ID property of the current instance
/// </summary>
public void Retrieve()
{
  try
  {
    this._TechnicalLine_Specialization_ID.Validate("TechnicalLine_Specialization_ID");
  }
  catch (DalException e)
  {
    throw e;
  }

  try
  {
    RetrieveExecute(_TechnicalLine_Specialization_ID.Value, this, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
}

/// <summary>
/// Retrieves the TechnicalLine_Specialization with the specified identifier from the database.
/// </summary>
/// <param name="technicalLine_Specialization_ID"></param>
public static Specialization Retrieve(int technicalLine_Specialization_ID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  Specialization objTechnicalLine_Specialization = new Specialization();

  try
  {
    RetrieveExecute(technicalLine_Specialization_ID, objTechnicalLine_Specialization, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
  return objTechnicalLine_Specialization;
}

private static void RetrieveExecute(
  int technicalLine_Specialization_ID,
  Specialization technicalLine_Specialization,
  SqlConnection sqlConnection)
{
  DataRow objDataRow;

  SqlCommand objCommand = new SqlCommand("TechnicalLine_Specialization_Select", sqlConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  SqlParameter objParam;

  objParam = objCommand.Parameters.Add("@TechnicalLine_Specialization_ID", SqlDbType.Int);
  objParam.Value = technicalLine_Specialization._TechnicalLine_Specialization_ID.Value;

  DataSet objDataSet = new DataSet("TechnicalLine_Specialization");
  SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

  try
  {
    sqlConnection.Open();
    objAdap.Fill(objDataSet, "TechnicalLine_Specialization");
    sqlConnection.Close();
    objDataRow = objDataSet.Tables["TechnicalLine_Specialization"].Rows[0];
  }
```

```
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }
      technicalLine_Specialization._TechnicalLine_Specialization_ID.Value = (int)objDataRow["TechnicalLine_Specialization_ID"];
      technicalLine_Specialization._TechnicalLineVersion_ID.Value     = (int)objDataRow["TechnicalLineVersion_ID"];
      technicalLine_Specialization._Specialization_ID.Value        = (int)objDataRow["Specialization_ID"];
      technicalLine_Specialization._Log.SetValues(objDataRow);
    }

    #endregion //Retrieve Methods

      #endregion //Public Methods

    #region Private Methods

    /// <summary>
    /// Validates the current <see cref="StudyPlanning.DAL.TechnicalLines.Specialization"/> object. If the
    /// value of either of the private properties is null a
    /// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
    /// </summary>
    private void Validate()
    {
      try
      {
        _TechnicalLine_Specialization_ID.Validate("TechnicalLine_Specialization_ID");
        _Log.Validate();
      }
      catch (DalException excp)
      {
        throw excp;
      }
    }
    #endregion //Private Methods
  }
}
```

1.27.5 TechnicalField

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.TechnicalLines
{
  /// <summary>
  /// Represents the TechnicalLine_TechnicalField table.
  /// </summary>
  public class TechnicalField : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalInt _TechnicalLine_TechnicalField_ID = new DalInt(false);
    private DalInt _TechnicalLineVersion_ID = new DalInt(false);
    private DalInt _TechnicalField_ID = new DalInt(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.TechnicalLines.TechnicalField"/>
    /// class.
    /// </summary>
    public TechnicalField() {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the TechnicalLine_TechnicalField_ID.
    /// </summary>
    public DalInt TechnicalLine_TechnicalField_ID
    {
      get { return _TechnicalLine_TechnicalField_ID; }
    }

    /// <summary>
    /// Gets the TechnicalLineVersion_ID.
    /// </summary>
    public DalInt TechnicalLineVersion_ID
    {
      get { return _TechnicalLineVersion_ID; }
    }

    /// <summary>
    /// Gets the TechnicalField_ID.
    /// </summary>
    public DalInt TechnicalField_ID
    {
      get { return _TechnicalField_ID; }
```

```
}

/// <summary>
/// Gets the data log of the table.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Public Methods

#region Retrieve Methods

/// <summary>
/// Retrieves the TechnicalLine_TechnicalField from the database using the value of the
/// TechnicalLine_TechnicalField_ID property of the current instance
/// </summary>
public void Retrieve()
{
  try
  {
    this._TechnicalLine_TechnicalField_ID.Validate("TechnicalLine_TechnicalField_ID");
  }
  catch (DalException e)
  {
    throw e;
  }

  try
  {
    RetrieveExecute(_TechnicalLine_TechnicalField_ID.Value, this, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
}

/// <summary>
/// Retrieves the TechnicalLine_TechnicalField with the specified identifier from the database.
/// </summary>
/// <param name="technicalLine_TechnicalField_ID"></param>
public static TechnicalField Retrieve(int technicalLine_TechnicalField_ID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  TechnicalField objTechnicalLine_TechnicalField = new TechnicalField();

  try
  {
    RetrieveExecute(technicalLine_TechnicalField_ID, objTechnicalLine_TechnicalField, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
  return objTechnicalLine_TechnicalField;
}

private static void RetrieveExecute(
  int technicalLine_TechnicalField_ID,
  TechnicalField technicalLine_TechnicalField,
  SqlConnection sqlConnection)
{
  DataRow objDataRow;

  SqlCommand objCommand = new SqlCommand("TechnicalLine_TechnicalField_Select", sqlConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  SqlParameter objParam;

  objParam = objCommand.Parameters.Add("@TechnicalLine_TechnicalField_ID", SqlDbType.Int);
  objParam.Value = technicalLine_TechnicalField._TechnicalLine_TechnicalField_ID.Value;

  DataSet objDataSet = new DataSet("TechnicalLine_TechnicalField");
  SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

  try
  {
    sqlConnection.Open();
    objAdap.Fill(objDataSet, "TechnicalLine_TechnicalField");
    sqlConnection.Close();
    objDataRow = objDataSet.Tables["TechnicalLine_TechnicalField"].Rows[0];
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
  technicalLine_TechnicalField._TechnicalLine_TechnicalField_ID.Value = (int)objDataRow["TechnicalLine_TechnicalField_ID"];
  technicalLine_TechnicalField._TechnicalLineVersion_ID.Value      = (int)objDataRow["TechnicalLineVersion_ID"];
  technicalLine_TechnicalField._TechnicalField_ID.Value         = (int)objDataRow["TechnicalField_ID"];
  technicalLine_TechnicalField._Log.SetValues(objDataRow);
}

#endregion //Retrieve Methods
```

```
      #endregion //Public Methods

    #region Private Methods

    /// <summary>
    /// Validates the current <see cref="StudyPlanning.DAL.TechnicalLines.TechnicalField"/> object. If the
    /// value of either of the private properties is null a
    /// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
    /// </summary>
    private void Validate()
    {
      try
      {
        _TechnicalLine_TechnicalField_ID.Validate("TechnicalLine_TechnicalField_ID");
        _Log.Validate();
      }
      catch (DalException excp)
      {
        throw excp;
      }
    }
    #endregion //Private Methods
  }
}
```

1.27.6 TechnicalLineVersion

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.TechnicalLines
{
  /// <summary>
  /// Represents the TechnicalLineVersion table.
  /// </summary>
  public class TechnicalLineVersion : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalInt _TechnicalLineVersion_ID = new DalInt(false);
    private DalInt _TechnicalLine_ID = new DalInt(false);
    private DalInt _Version = new DalInt(false);
    private DalStringLocalizable _Name = new DalStringLocalizable(false);
    private DalGuid _Point_ID = new DalGuid(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.TechnicalLines.TechnicalLineVersion"/>
    /// class.
    /// </summary>
    public TechnicalLineVersion() {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the TechnicalLineVersion_ID.
    /// </summary>
    public DalInt TechnicalLineVersion_ID
    {
      get { return _TechnicalLineVersion_ID; }
    }

    /// <summary>
    /// Gets the TechnicalLine_ID.
    /// </summary>
    public DalInt TechnicalLine_ID
    {
      get { return _TechnicalLine_ID; }
    }

    /// <summary>
    /// Gets the Version.
    /// </summary>
    public DalInt Version
    {
      get { return _Version; }
    }

    /// <summary>
    /// Gets the Name.
    /// </summary>
    public DalStringLocalizable Name
    {
      get { return _Name; }
    }
```

```
/// <summary>
/// Gets the Point_ID.
/// </summary>
public DalGuid Point_ID
{
  get { return _Point_ID; }
}

/// <summary>
/// Gets the data log of the table.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Public Methods

#region Retrieve Methods

/// <summary>
/// Retrieves the TechnicalLineVersion from the database using the value of the
/// TechnicalLineVersion_ID property of the current instance
/// </summary>
public void Retrieve()
{
  try
  {
    this._TechnicalLineVersion_ID.Validate("TechnicalLineVersion_ID");
  }
  catch (DalException e)
  {
    throw e;
  }

  try
  {
    RetrieveExecute(_TechnicalLineVersion_ID.Value, this, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
}

/// <summary>
/// Retrieves the TechnicalLineVersion with the specified identifier from the database.
/// </summary>
/// <param name="technicalLineVersion_ID"></param>
public static TechnicalLineVersion Retrieve(int technicalLineVersion_ID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  TechnicalLineVersion objTechnicalLineVersion = new TechnicalLineVersion();

  try
  {
    RetrieveExecute(technicalLineVersion_ID, objTechnicalLineVersion, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
  return objTechnicalLineVersion;
}

private static void RetrieveExecute(
  int technicalLineVersion_ID,
  TechnicalLineVersion technicalLineVersion,
  SqlConnection sqlConnection)
{
  DataRow objDataRow;

  SqlCommand objCommand = new SqlCommand("TechnicalLineVersion_Select", sqlConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  SqlParameter objParam;

  objParam = objCommand.Parameters.Add("@TechnicalLineVersion_ID", SqlDbType.Int);
  objParam.Value = technicalLineVersion._TechnicalLineVersion_ID.Value;

  DataSet objDataSet = new DataSet("TechnicalLineVersion");
  SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

  try
  {
    sqlConnection.Open();
    objAdap.Fill(objDataSet, "TechnicalLineVersion");
    sqlConnection.Close();
    objDataRow = objDataSet.Tables["TechnicalLineVersion"].Rows[0];
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
```

```
      technicalLineVersion._TechnicalLineVersion_ID.Value = (int)objDataRow["TechnicalLineVersion_ID"];
      technicalLineVersion._TechnicalLine_ID.Value     = (int)objDataRow["TechnicalLine_ID"];
      technicalLineVersion._Version.Value          = (int)objDataRow["Version"];
      technicalLineVersion._Name.LoadXml((string)objDataRow["Name"]);
      technicalLineVersion._Point_ID.Value         = (Guid)objDataRow["Point_ID"];
      technicalLineVersion._Log.SetValues(objDataRow);
    }

    #endregion //Retrieve Methods

      #endregion //Public Methods

    #region Private Methods

    /// <summary>
    /// Validates the current <see cref="StudyPlanning.DAL.TechnicalLines.TechnicalLineVersion"/> object. If the
    /// value of either of the private properties is null a
    /// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
    /// </summary>
    private void Validate()
    {
      try
      {
        _TechnicalLineVersion_ID.Validate("TechnicalLineVersion_ID");
        _Log.Validate();
      }
      catch (DalException excp)
      {
        throw excp;
      }
    }
    #endregion //Private Methods
  }
}
```

1.28 TechnicalPackages

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.TechnicalPackages
{
  /// <summary>
  /// Represents the TechnicalPackage table.
  /// </summary>
  public class TechnicalPackage : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalInt _TechnicalPackage_ID = new DalInt(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.TechnicalPackages.TechnicalPackage"/>
    /// class.
    /// </summary>
    public TechnicalPackage() {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the value of the TechnicalPackage_ID attribute.
    /// </summary>
    /// <value></value>
    public DalInt TechnicalPackage_ID
    {
      get { return _TechnicalPackage_ID; }
    }

    /// <summary>
    /// Gets the data log of the table.
    /// </summary>
    public DataLog Log
    {
      get { return _Log; }
    }

    #endregion //Public Properties

    #region Public Methods

    #region Retrieve Methods

    /// <summary>
```

```
    /// Retrieves the technical package from the database using the value of the
    /// TechnicalPackage_ID property of the current instance
    /// </summary>
    public void Retrieve()
    {
      DataRow objDataRow;

      try
      {
        _TechnicalPackage_ID.Validate("TechnicalPackage_ID");
      }
      catch (DalException excp)
      {
        throw excp;
      }

      try
      {
        objDataRow = RetrieveExecute(_TechnicalPackage_ID.Value, objConnection);
      }
      catch (DalException objExc)
      {
        throw objExc;
      }
      this._TechnicalPackage_ID.Value = Convert.ToInt32(objDataRow["TechnicalPackage_ID"]);
      this._Log.SetValues(objDataRow);
    }

    /// <summary>
    /// Retrieves the technical package with the specified TechnicalPackage_ID from the database.
    /// </summary>
    /// <param name="technicalPackage_ID">The value of the TechnicalPackage_ID.</param>
    public static TechnicalPackage Retrieve(int technicalPackage_ID)
    {
      DataRow objDataRow;
      string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
      SqlConnection objConnection = new SqlConnection(connString);

      TechnicalPackage objTechnicalPackage = new TechnicalPackage();

      try
      {
        objDataRow = RetrieveExecute(technicalPackage_ID, objConnection);
      }
      catch (DalException objExc)
      {
        throw objExc;
      }
      objTechnicalPackage._TechnicalPackage_ID.Value = Convert.ToInt32(objDataRow["TechnicalPackage_ID"]);
      objTechnicalPackage._Log.SetValues(objDataRow);
      return objTechnicalPackage;
    }

    private static DataRow RetrieveExecute(
      int technicalPackage_ID,
      SqlConnection sqlConnection)
    {
      SqlCommand objCommand = new SqlCommand("TechnicalPackage_Select", sqlConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      SqlParameter objParam;

      objParam = objCommand.Parameters.Add("@TechnicalPackage_ID", SqlDbType.Int);
      objParam.Value = technicalPackage_ID;

      DataSet objDataSet = new DataSet("TechnicalPackage");
      SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

      try
      {
        sqlConnection.Open();
        objAdap.Fill(objDataSet, "TechnicalPackage");
        sqlConnection.Close();
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }
      DataRow objDataRow = objDataSet.Tables["TechnicalPackage"].Rows[0];
      return objDataRow;
    }

    #endregion //Retrieve Methods

    #endregion //Public Methods

  }
}
```

1.28.1 FundamentalCourse

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;
```

```
namespace StudyPlanning.DAL.TechnicalPackages
{
  /// <summary>
  /// Represents the TechnicalPackage_FundamentalCourse table.
  /// </summary>
  public class FundamentalCourse : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalInt _TechnicalPackage_FundamentalCourse_ID = new DalInt(false);
    private DalInt _TechnicalPackageVersion_ID = new DalInt(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.TechnicalPackages.FundamentalCourse"/>
    /// class.
    /// </summary>
    public FundamentalCourse() {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the TechnicalPackage_FundamentalCourse_ID.
    /// </summary>
    /// <value></value>
    public DalInt TechnicalPackage_FundamentalCourse_ID
    {
      get { return _TechnicalPackage_FundamentalCourse_ID; }
    }

    /// <summary>
    /// Gets the TechnicalPackageVersion_ID.
    /// </summary>
    /// <value></value>
    public DalInt TechnicalPackageVersion_ID
    {
      get { return _TechnicalPackageVersion_ID; }
    }

    /// <summary>
    /// Gets the data log of the table.
    /// </summary>
    public DataLog Log
    {
      get { return _Log; }
    }

    #endregion //Public Properties

    #region Public Methods

    #region Retrieve Methods

    /// <summary>
    /// Retrieves the TechnicalPackage_FundamentalCourse object from the database using the value of the
    /// TechnicalPackage_FundamentalCourse_ID property of the current instance
    /// </summary>
    public void Retrieve()
    {
      DataRow objDataRow;

      try
      {
        _TechnicalPackage_FundamentalCourse_ID.Validate("TechnicalPackage_FundamentalCourse_ID");
      }
      catch (DalException excp)
      {
        throw excp;
      }

      try
      {
        objDataRow = RetrieveExecute(_TechnicalPackage_FundamentalCourse_ID.Value, objConnection);
      }
      catch (DalException objExc)
      {
        throw objExc;
      }

      this._TechnicalPackage_FundamentalCourse_ID.Value = Convert.ToInt32(objDataRow["TechnicalPackage_FundamentalCourse_ID"]);
      this._TechnicalPackageVersion_ID.Value      = Convert.ToInt32(objDataRow["TechnicalPackageVersion_ID"]);
      this._Log.SetValues(objDataRow);
    }

    /// <summary>
    /// Retrieves the TechnicalPackage_FundamentalCourse object with the specified identifier from the database.
    /// </summary>
    /// <param name="technicalPackage_FundamentalCourse_ID">The unique identification of the TechnicalPackage_FundamentalCourse object.</
      param>
    public static FundamentalCourse Retrieve(int technicalPackage_FundamentalCourse_ID)
    {
```

```
      DataRow objDataRow;
      string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
      SqlConnection objConnection = new SqlConnection(connString);

      FundamentalCourse objTP_FundamentalCourse = new FundamentalCourse();

      try
      {
        objDataRow = RetrieveExecute(technicalPackage_FundamentalCourse_ID, objConnection);
      }
      catch (DalException objExc)
      {
        throw objExc;
      }

      objTP_FundamentalCourse._TechnicalPackage_FundamentalCourse_ID.Value = Convert.ToInt32(objDataRow["
        TechnicalPackage_FundamentalCourse_ID"]);
      objTP_FundamentalCourse._TechnicalPackageVersion_ID.Value       = Convert.ToInt32(objDataRow["TechnicalPackageVersion_ID"]);
      objTP_FundamentalCourse._Log.SetValues(objDataRow);
      return objTP_FundamentalCourse;
    }

    private static DataRow RetrieveExecute(
      int technicalPackage_FundamentalCourse_ID,
      SqlConnection sqlConnection)
    {
      SqlCommand objCommand = new SqlCommand("TechnicalPackage_FundamentalCourse_Select", sqlConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      SqlParameter objParam;

      objParam = objCommand.Parameters.Add("@TechnicalPackage_FundamentalCourse_ID", SqlDbType.Int);
      objParam.Value = technicalPackage_FundamentalCourse_ID;

      DataSet objDataSet = new DataSet("TechnicalPackage_FundamentalCourse");
      SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

      try
      {
        sqlConnection.Open();
        objAdap.Fill(objDataSet, "TechnicalPackage_FundamentalCourse");
        sqlConnection.Close();
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }
      DataRow objDataRow = objDataSet.Tables["TechnicalPackage_FundamentalCourse"].Rows[0];
      return objDataRow;
    }

    /// <summary>
    /// Retrieves a list of the TechnicalPackage_FundamentalCourse_IDs associated with
    /// the <see cref="StudyPlanning.DAL.TechnicalPackages.TechnicalPackageVersion"/>.
    /// </summary>
    /// <param name="technicalPackageVersion_ID"> The identification of the
    /// <see cref="StudyPlanning.DAL.TechnicalPackages.TechnicalPackageVersion"/>.</param>
    public static int[] GetIDFromVersion(int technicalPackageVersion_ID)
    {
      string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
      SqlConnection objConnection = new SqlConnection(connString);

      SqlCommand objCommand = new SqlCommand("TechnicalPackage_FundamentalCourse_GetIDFromVersion", objConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      SqlParameter objParam = objCommand.Parameters.Add("@TechnicalPackageVersion_ID", SqlDbType.Int);
      objParam.Value = technicalPackageVersion_ID;

      DataSet objDataSet = new DataSet("TechnicalPackage_FundamentalCourse");
      SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

      try
      {
        objConnection.Open();
        objAdap.Fill(objDataSet, "TechnicalPackage_FundamentalCourse");
        objConnection.Close();
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }
      int intRows = objDataSet.Tables["TechnicalPackage_FundamentalCourse"].Rows.Count;
      int[] ids = new int[intRows];

      for(int i=0; i < intRows; i++)
      {
        ids[i] = (int)objDataSet.Tables["TechnicalPackage_FundamentalCourse"].Rows[i]["TechnicalPackage_FundamentalCourse_ID"];
      }
      return ids;
    }

    #endregion //Retrieve Methods

    #endregion //Public Methods

  }
}
```

1.28.2 FundamentalCourseItem

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.TechnicalPackages
{
  /// <summary>
  /// Represents the TechnicalPackage_FundamentalCourseItem table.
  /// </summary>
  public class FundamentalCourseItem : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalInt _TechnicalPackage_FundamentalCourseItem_ID = new DalInt(false);
    private DalInt _TechnicalPackage_FundamentalCourse_ID = new DalInt(false);
    private DalGuid _Course_ID = new DalGuid(true);
    private DalString _Number = new DalString(true);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.TechnicalPackages.FundamentalCourseItem"/>
    /// class.
    /// </summary>
    public FundamentalCourseItem() {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the TechnicalPackage_FundamentalCourseItem_ID.
    /// </summary>
    /// <value></value>
    public DalInt TechnicalPackage_FundamentalCourseItem_ID
    {
      get { return _TechnicalPackage_FundamentalCourseItem_ID; }
    }

    /// <summary>
    /// Gets the TechnicalPackage_FundamentalCourse_ID.
    /// </summary>
    /// <value></value>
    public DalInt TechnicalPackage_FundamentalCourse_ID
    {
      get { return _TechnicalPackage_FundamentalCourse_ID; }
    }

    /// <summary>
    /// Gets the Course_ID.
    /// </summary>
    /// <value></value>
    public DalGuid Course_ID
    {
      get { return _Course_ID; }
    }

    /// <summary>
    /// Gets the Number.
    /// </summary>
    /// <value></value>
    public DalString Number
    {
      get { return _Number; }
    }

    /// <summary>
    /// Gets the data log of the table.
    /// </summary>
    public DataLog Log
    {
      get { return _Log; }
    }

    #endregion //Public Properties

    #region Public Methods

    #region Retrieve Methods

    /// <summary>
    /// Retrieves the TechnicalPackage_FundamentalCourseItem object from the database using the value of the
    /// TechnicalPackage_FundamentalCourseItem_ID property of the current instance
    /// </summary>
    public void Retrieve()
    {
      DataRow objDataRow;

      try
      {
```

```
      _TechnicalPackage_FundamentalCourseItem_ID.Validate("TechnicalPackage_FundamentalCourseItem");
    }
    catch (DalException excp)
    {
      throw excp;
    }

    try
    {
      objDataRow = RetrieveExecute(_TechnicalPackage_FundamentalCourseItem_ID.Value, objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }

    this._TechnicalPackage_FundamentalCourseItem_ID.Value = Convert.ToInt32(objDataRow["TechnicalPackage_FundamentalCourseItem_ID
      "]);
    this._TechnicalPackage_FundamentalCourse_ID.Value   = Convert.ToInt32(objDataRow["TechnicalPackage_FundamentalCourse_ID"]);

    if (objDataRow["Course_ID"].Equals(System.DBNull.Value))
      this._Course_ID.IsNull = true;
    else
      this._Course_ID.Value = (Guid)objDataRow["Course_ID"];

    if (objDataRow["Number"].Equals(System.DBNull.Value))
      this._Number.IsNull = true;
    else
      this._Number.Value = (string)objDataRow["Number"];

    this._Log.SetValues(objDataRow);
  }

  /// <summary>
  /// Retrieves the TechnicalPackage_FundamentalCourseItem object with the specified identifier from the database.
  /// </summary>
  /// <param name="technicalPackage_FundamentalCourseItem_ID">The unique identification of the TechnicalPackage_FundamentalCourseItem object
    .</param>
  public static FundamentalCourseItem Retrieve(int technicalPackage_FundamentalCourseItem_ID)
  {
    DataRow objDataRow;
    string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
    SqlConnection objConnection = new SqlConnection(connString);

    FundamentalCourseItem objTP_FundamentalCourseItem = new FundamentalCourseItem();

    try
    {
      objDataRow = RetrieveExecute(technicalPackage_FundamentalCourseItem_ID, objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }

    objTP_FundamentalCourseItem._TechnicalPackage_FundamentalCourseItem_ID.Value = Convert.ToInt32(objDataRow["
      TechnicalPackage_FundamentalCourseItem_ID"]);
    objTP_FundamentalCourseItem._TechnicalPackage_FundamentalCourse_ID.Value   = Convert.ToInt32(objDataRow["
      TechnicalPackage_FundamentalCourse_ID"]);

    if (objDataRow["Course_ID"].Equals(System.DBNull.Value))
      objTP_FundamentalCourseItem._Course_ID.IsNull = true;
    else
      objTP_FundamentalCourseItem._Course_ID.Value = (Guid)objDataRow["Course_ID"];

    if (objDataRow["Number"].Equals(System.DBNull.Value))
      objTP_FundamentalCourseItem._Number.IsNull = true;
    else
      objTP_FundamentalCourseItem._Number.Value = (string)objDataRow["Number"];

    objTP_FundamentalCourseItem._Log.SetValues(objDataRow);
    return objTP_FundamentalCourseItem;
  }

  private static DataRow RetrieveExecute(
    int technicalPackage_FundamentalCourseItem_ID,
    SqlConnection sqlConnection)
  {
    SqlCommand objCommand = new SqlCommand("TechnicalPackage_FundamentalCourseItem_Select", sqlConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam;

    objParam = objCommand.Parameters.Add("@TechnicalPackage_FundamentalCourseItem_ID", SqlDbType.Int);
    objParam.Value = technicalPackage_FundamentalCourseItem_ID;

    DataSet objDataSet = new DataSet("TechnicalPackage_FundamentalCourseItem");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
      sqlConnection.Open();
      objAdap.Fill(objDataSet, "TechnicalPackage_FundamentalCourseItem");
      sqlConnection.Close();
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
```

```
      DataRow objDataRow = objDataSet.Tables["TechnicalPackage_FundamentalCourseItem"].Rows[0];
      return objDataRow;
    }

    /// <summary>
    /// Retrieves a list of the Course_IDs associated with
    /// the <see cref="StudyPlanning.DAL.TechnicalPackages.FundamentalCourseItem"/>.
    /// </summary>
    /// <param name="technicalPackage_FundamentalCourse_ID"> The identification of the
    /// <see cref="StudyPlanning.DAL.TechnicalPackages.FundamentalCourse"/>.</param>
    public static Guid[] GetCourses(int technicalPackage_FundamentalCourse_ID)
    {
      string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
      SqlConnection objConnection = new SqlConnection(connString);

      SqlCommand objCommand = new SqlCommand("TechnicalPackage_FundamentalCourseItem_GetCourses", objConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      SqlParameter objParam = objCommand.Parameters.Add("@TechnicalPackage_FundamentalCourse_ID", SqlDbType.Int);
      objParam.Value = technicalPackage_FundamentalCourse_ID;

      DataSet objDataSet = new DataSet("Courses");
      SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

      try
      {
        objConnection.Open();
        objAdap.Fill(objDataSet, "Courses");
        objConnection.Close();
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }
      int intRows = objDataSet.Tables["Courses"].Rows.Count;
      Guid[] courses = new Guid[intRows];

      for (int i=0; i < intRows; i++)
      {
        courses[i] = (Guid)objDataSet.Tables["Courses"].Rows[i]["Course_ID"];
      }
      return courses;
    }

    #endregion //Retrieve Methods

    #endregion //Public Methods

  }
}
```

1.28.3 Period

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Collections;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.TechnicalPackages
{
  /// <summary>
  /// Represents the TechnicalPackage_Period table.
  /// </summary>
  public class Period : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalInt _TechnicalPackage_Period_ID = new DalInt(false);
    private DalInt _TechnicalPackageVersion_ID = new DalInt(false);
    private DalGuid _Period_ID = new DalGuid(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.TechnicalPackages.Period"/>
    /// class.
    /// </summary>
    public Period() {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the TechnicalPackage_Period_ID.
    /// </summary>
    /// <value></value>
    public DalInt TechnicalPackage_Period_ID
    {
      get { return _TechnicalPackage_Period_ID; }
```

```
}

/// <summary>
/// Gets the TechnicalPackageVersion_ID
/// </summary>
/// <value></value>
public DalInt TechnicalPackageVersion_ID
{
  get { return _TechnicalPackageVersion_ID; }
}

/// <summary>
/// Gets the Period_ID.
/// </summary>
/// <value></value>
public DalGuid Period_ID
{
  get { return _Period_ID; }
}

/// <summary>
/// Gets the data log of the table.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Public Methods

#region Retrieve Methods

/// <summary>
/// Retrieves the technical package period from the database using the value of the
/// TechnicalPackage_Period_ID property of the current instance
/// </summary>
public void Retrieve()
{
  DataRow objDataRow;

  try
  {
    _TechnicalPackage_Period_ID.Validate("TechnicalPackage_Period_ID");
  }
  catch (DalException excp)
  {
    throw excp;
  }

  try
  {
    objDataRow = RetrieveExecute(_TechnicalPackage_Period_ID.Value, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
  this._TechnicalPackage_Period_ID.Value = Convert.ToInt32(objDataRow["TechnicalPackage_Period_ID"]);
  this._TechnicalPackageVersion_ID.Value = Convert.ToInt32(objDataRow["TechnicalPackageVersion_ID"]);
  this._Period_ID.Value          = (Guid)objDataRow["Period_ID"];
  this._Log.SetValues(objDataRow);
}

/// <summary>
/// Retrieves the technical package period with the specified identifier from the database.
/// </summary>
/// <param name="technicalPackage_Period_ID">The unique identification of the technical package period.</param>
public static Period Retrieve(int technicalPackage_Period_ID)
{
  DataRow objDataRow;
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  Period objTechnicalPackage_Period = new Period();

  try
  {
    objDataRow = RetrieveExecute(technicalPackage_Period_ID, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
  objTechnicalPackage_Period._TechnicalPackage_Period_ID.Value = Convert.ToInt32(objDataRow["TechnicalPackage_Period_ID"]);
  objTechnicalPackage_Period._TechnicalPackageVersion_ID.Value = Convert.ToInt32(objDataRow["TechnicalPackageVersion_ID"]);
  objTechnicalPackage_Period._Period_ID.Value          = (Guid)objDataRow["Period_ID"];
  objTechnicalPackage_Period._Log.SetValues(objDataRow);
  return objTechnicalPackage_Period;
}

private static DataRow RetrieveExecute(
  int technicalPackage_Period_ID,
  SqlConnection sqlConnection)
{
  SqlCommand objCommand = new SqlCommand("TechnicalPackage_Period_Select", sqlConnection);
  objCommand.CommandType = CommandType.StoredProcedure;
```

```
    SqlParameter objParam;

    objParam = objCommand.Parameters.Add("@TechnicalPackage_Period_ID", SqlDbType.Int);
    objParam.Value = technicalPackage_Period_ID;

    DataSet objDataSet = new DataSet("TechnicalPackage_Period");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
      sqlConnection.Open();
      objAdap.Fill(objDataSet, "TechnicalPackage_Period");
      sqlConnection.Close();
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
    DataRow objDataRow = objDataSet.Tables["TechnicalPackage_Period"].Rows[0];
    return objDataRow;
  }

  /// <summary>
  /// Retrieves a list of the Period_IDs associated with
  /// the <see cref="StudyPlanning.DAL.TechnicalPackages.TechnicalPackageVersion"/>.
  /// </summary>
  /// <param name="technicalPackageVersion_ID"> The identification of the
  /// <see cref="StudyPlanning.DAL.TechnicalPackages.TechnicalPackageVersion"/>.</param>
  public static Guid[] GetPeriods(int technicalPackageVersion_ID)
  {
    string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
    SqlConnection objConnection = new SqlConnection(connString);

    SqlCommand objCommand = new SqlCommand("TechnicalPackage_Period_GetPeriods", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam = objCommand.Parameters.Add("@TechnicalPackageVersion_ID", SqlDbType.Int);
    objParam.Value = technicalPackageVersion_ID;

    DataSet objDataSet = new DataSet("TechnicalPackage_Period");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
      objConnection.Open();
      objAdap.Fill(objDataSet, "TechnicalPackage_Period");
      objConnection.Close();
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
    int intRows = objDataSet.Tables["TechnicalPackage_Period"].Rows.Count;
    Guid[] periods = new Guid[intRows];

    for(int i=0; i < intRows; i++)
    {
      periods[i] = (Guid)objDataSet.Tables["TechnicalPackage_Period"].Rows[i]["Period_ID"];
    }
    return periods;
  }

  /// <summary>
  /// Retrieves the TechnicalPackage_Period_ID associated with
  /// the <see cref="StudyPlanning.DAL.TechnicalPackages.TechnicalPackageVersion"/> and
  /// the <see cref="StudyPlanning.DAL.Period"/>.
  /// </summary>
  /// <param name="technicalPackageVersion_ID"> The identification of the
  /// <see cref="StudyPlanning.DAL.TechnicalPackages.TechnicalPackageVersion"/>.</param>
  /// <param name="period_ID"> The identification of the <see cref="StudyPlanning.DAL.Period"/>.</param>
  public static int GetIDFromVersion(int technicalPackageVersion_ID, Guid period_ID)
  {
    string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
    SqlConnection objConnection = new SqlConnection(connString);

    SqlCommand objCommand = new SqlCommand("TechnicalPackage_Period_GetIDFromVersion", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam = objCommand.Parameters.Add("@TechnicalPackageVersion_ID", SqlDbType.Int);
    objParam.Value = technicalPackageVersion_ID;

    objParam = objCommand.Parameters.Add("@Period_ID", SqlDbType.UniqueIdenti↓er);
    objParam.Value = period_ID;

    DataSet objDataSet = new DataSet("Period");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
      objConnection.Open();
      objAdap.Fill(objDataSet, "Period");
      objConnection.Close();
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }

    int id = (int)objDataSet.Tables["Period"].Rows[0]["TechnicalPackage_Period_ID"];
```

```
      return id;
    }

    #endregion //Retrieve Methods

    #endregion //Public Methods

  }
}
```

1.28.4 PeriodCourse

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.TechnicalPackages
{
  /// <summary>
  /// Represents the TechnicalPackage_PeriodCourse table.
  /// </summary>
  public class PeriodCourse : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalInt _TechnicalPackage_PeriodCourse_ID = new DalInt(false);
    private DalInt _TechnicalPackage_Period_ID = new DalInt(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.TechnicalPackages.PeriodCourse"/>
    /// class.
    /// </summary>
    public PeriodCourse() {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the TechnicalPackage_PeriodCourse_ID.
    /// </summary>
    /// <value></value>
    public DalInt TechnicalPackage_PeriodCourse_ID
    {
      get { return _TechnicalPackage_PeriodCourse_ID; }
    }

    /// <summary>
    /// Gets the TechnicalPackage_Period_ID.
    /// </summary>
    /// <value></value>
    public DalInt TechnicalPackage_Period_ID
    {
      get { return _TechnicalPackage_Period_ID; }
    }

    /// <summary>
    /// Gets the data log of the table.
    /// </summary>
    public DataLog Log
    {
      get { return _Log; }
    }

    #endregion //Public Properties

    #region Public Methods

    #region Retrieve Methods

    /// <summary>
    /// Retrieves the TechnicalPackage_PeriodCourse object from the database using the value of the
    /// TechnicalPackage_PeriodCourse_ID property of the current instance
    /// </summary>
    public void Retrieve()
    {
      DataRow objDataRow;

      try
      {
        _TechnicalPackage_PeriodCourse_ID.Validate("TechnicalPackage_PeriodCourse_ID");
      }
      catch (DalException excp)
      {
        throw excp;
      }

      try
```

```
    {
      objDataRow = RetrieveExecute(_TechnicalPackage_PeriodCourse_ID.Value, objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }

    this._TechnicalPackage_PeriodCourse_ID.Value = Convert.ToInt32(objDataRow["TechnicalPackage_PeriodCourse_ID"]);
    this._TechnicalPackage_Period_ID.Value     = Convert.ToInt32(objDataRow["TechnicalPackage_Period_ID"]);
    this._Log.SetValues(objDataRow);
  }

  /// <summary>
  /// Retrieves the TechnicalPackage_PeriodCourse object with the specified identifier from the database.
  /// </summary>
  /// <param name="technicalPackage_PeriodCourse_ID">The unique identification of the TechnicalPackage_PeriodCourse object.</param>
  public static PeriodCourse Retrieve(int technicalPackage_PeriodCourse_ID)
  {
    DataRow objDataRow;
    string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
    SqlConnection objConnection = new SqlConnection(connString);

    PeriodCourse objTP_PeriodCourse = new PeriodCourse();

    try
    {
      objDataRow = RetrieveExecute(technicalPackage_PeriodCourse_ID, objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }

    objTP_PeriodCourse._TechnicalPackage_PeriodCourse_ID.Value = Convert.ToInt32(objDataRow["TechnicalPackage_PeriodCourse_ID"])
      ;
    objTP_PeriodCourse._TechnicalPackage_Period_ID.Value     = Convert.ToInt32(objDataRow["TechnicalPackage_Period_ID"]);
    objTP_PeriodCourse._Log.SetValues(objDataRow);
    return objTP_PeriodCourse;
  }

  private static DataRow RetrieveExecute(
    int technicalPackage_PeriodCourse_ID,
    SqlConnection sqlConnection)
  {
    SqlCommand objCommand = new SqlCommand("TechnicalPackage_PeriodCourse_Select", sqlConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam;

    objParam = objCommand.Parameters.Add("@TechnicalPackage_PeriodCourse_ID", SqlDbType.Int);
    objParam.Value = technicalPackage_PeriodCourse_ID;

    DataSet objDataSet = new DataSet("TechnicalPackage_PeriodCourse");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
      sqlConnection.Open();
      objAdap.Fill(objDataSet, "TechnicalPackage_PeriodCourse");
      sqlConnection.Close();
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
    DataRow objDataRow = objDataSet.Tables["TechnicalPackage_PeriodCourse"].Rows[0];
    return objDataRow;
  }

  /// <summary>
  /// Retrieves a list of TechnicalPackage_Course_ID associated with
  /// the <see cref="StudyPlanning.DAL.TechnicalPackages.Period"/>.
  /// </summary>
  /// <param name="technicalPackage_Period_ID"> The identification of the
  /// <see cref="StudyPlanning.DAL.TechnicalPackages.Period"/>.</param>
  public static int[] GetIDFromPeriod(int technicalPackage_Period_ID)
  {
    string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
    SqlConnection objConnection = new SqlConnection(connString);

    SqlCommand objCommand = new SqlCommand("TechnicalPackage_PeriodCourse_GetIDFromPeriod", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam = objCommand.Parameters.Add("@TechnicalPackage_Period_ID", SqlDbType.Int);
    objParam.Value = technicalPackage_Period_ID;

    DataSet objDataSet = new DataSet("Course");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
      objConnection.Open();
      objAdap.Fill(objDataSet, "Course");
      objConnection.Close();
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
```

```
      }
      int intRows = objDataSet.Tables["Course"].Rows.Count;
      int[] ids = new int[intRows];

      for (int i=0; i < intRows; i++)
      {
        ids[i] = (int)objDataSet.Tables["Course"].Rows[i]["TechnicalPackage_PeriodCourse_ID"];
      }

      return ids;
    }

    #endregion //Retrieve Methods

    #endregion //Public Methods

  }
}
```

1.28.5 PeriodCourseItem

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;
using StudyPlanning.DAL.Courses;

namespace StudyPlanning.DAL.TechnicalPackages
{
  /// <summary>
  /// Represents the TechnicalPackage_PeriodCourseItem table.
  /// </summary>
  public class PeriodCourseItem : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalInt _TechnicalPackage_PeriodCourseItem_ID = new DalInt(false);
    private DalInt _TechnicalPackage_PeriodCourse_ID = new DalInt(false);
    private DalGuid _Course_ID = new DalGuid(true);
    private DalInt _Part = new DalInt(false);
    private DalString _Number = new DalString(true);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.TechnicalPackages.PeriodCourseItem"/>
    /// class.
    /// </summary>
    public PeriodCourseItem() {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the TechnicalPackage_PeriodCourseItem_ID.
    /// </summary>
    /// <value></value>
    public DalInt TechnicalPackage_PeriodCourseItem_ID
    {
      get { return _TechnicalPackage_PeriodCourseItem_ID; }
    }

    /// <summary>
    /// Gets the TechnicalPackage_PeriodCourse_ID.
    /// </summary>
    /// <value></value>
    public DalInt TechnicalPackage_PeriodCourse_ID
    {
      get { return _TechnicalPackage_PeriodCourse_ID; }
    }

    /// <summary>
    /// Gets the Course_ID.
    /// </summary>
    /// <value></value>
    public DalGuid Course_ID
    {
      get { return _Course_ID; }
    }

    /// <summary>
    /// Gets the Part.
    /// </summary>
    /// <value></value>
    public DalInt Part
    {
      get { return _Part; }
    }

    /// <summary>
```

```
/// Gets the Number.
/// </summary>
/// <value></value>
public DalString Number
{
  get { return _Number; }
}

/// <summary>
/// Gets the data log of the table.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Public Methods

#region Retrieve Methods

/// <summary>
/// Retrieves the TechnicalPackage_PeriodCourseItem object from the database using the value of the
/// TechnicalPackage_PeriodCourseItem_ID property of the current instance
/// </summary>
public void Retrieve()
{
  DataRow objDataRow;

  try
  {
    _TechnicalPackage_PeriodCourseItem_ID.Validate("TechnicalPackage_PeriodCourseItem_ID");
  }
  catch (DalException excp)
  {
    throw excp;
  }

  try
  {
    objDataRow = RetrieveExecute(_TechnicalPackage_PeriodCourseItem_ID.Value, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }

  this._TechnicalPackage_PeriodCourseItem_ID.Value = Convert.ToInt32(objDataRow["TechnicalPackage_PeriodCourseItem_ID"]);
  this._TechnicalPackage_PeriodCourse_ID.Value   = Convert.ToInt32(objDataRow["TechnicalPackage_PeriodCourse_ID"]);

  if (objDataRow["Course_ID"].Equals(System.DBNull.Value))
    this._Course_ID.IsNull = true;
  else
    this._Course_ID.Value = (Guid)objDataRow["Course_ID"];

  this._Part.Value = Convert.ToInt32(objDataRow["Part"]);

  if (objDataRow["Number"].Equals(System.DBNull.Value))
    this._Number.IsNull = true;
  else
    this._Number.Value = (string)objDataRow["Number"];

  this._Log.SetValues(objDataRow);
}

/// <summary>
/// Retrieves the TechnicalPackage_PeriodCourseItem object with the specified identifier from the database.
/// </summary>
/// <param name="technicalPackage_PeriodCourseItem_ID">The unique identification of the TechnicalPackage_PeriodCourseItem object.</param>
public static PeriodCourseItem Retrieve(int technicalPackage_PeriodCourseItem_ID)
{
  DataRow objDataRow;
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  PeriodCourseItem objTP_PeriodCourseItem = new PeriodCourseItem();

  try
  {
    objDataRow = RetrieveExecute(technicalPackage_PeriodCourseItem_ID, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }

  objTP_PeriodCourseItem._TechnicalPackage_PeriodCourseItem_ID.Value = Convert.ToInt32(objDataRow["
    TechnicalPackage_PeriodCourseItem_ID"]);
  objTP_PeriodCourseItem._TechnicalPackage_PeriodCourse_ID.Value   = Convert.ToInt32(objDataRow["
    TechnicalPackage_PeriodCourse_ID"]);

  if (objDataRow["Course_ID"].Equals(System.DBNull.Value))
    objTP_PeriodCourseItem._Course_ID.IsNull = true;
  else
    objTP_PeriodCourseItem._Course_ID.Value = (Guid)objDataRow["Course_ID"];

  objTP_PeriodCourseItem._Part.Value = Convert.ToInt32(objDataRow["Part"]);
```

```
      if (objDataRow["Number"].Equals(System.DBNull.Value))
        objTP_PeriodCourseItem._Number.IsNull = true;
      else
        objTP_PeriodCourseItem._Number.Value = (string)objDataRow["Number"];

      objTP_PeriodCourseItem._Log.SetValues(objDataRow);
      return objTP_PeriodCourseItem;
    }

    private static DataRow RetrieveExecute(
      int technicalPackage_PeriodCourseItem_ID,
      SqlConnection sqlConnection)
    {
      SqlCommand objCommand = new SqlCommand("TechnicalPackage_PeriodCourseItem_Select", sqlConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      SqlParameter objParam;

      objParam = objCommand.Parameters.Add("@TechnicalPackage_PeriodCourseItem_ID", SqlDbType.Int);
      objParam.Value = technicalPackage_PeriodCourseItem_ID;

      DataSet objDataSet = new DataSet("TechnicalPackage_PeriodCourseItem");
      SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

      try
      {
        sqlConnection.Open();
        objAdap.Fill(objDataSet, "TechnicalPackage_PeriodCourseItem");
        sqlConnection.Close();
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }
      DataRow objDataRow = objDataSet.Tables["TechnicalPackage_PeriodCourseItem"].Rows[0];
      return objDataRow;
    }

    /// <summary>
    /// Retrieves a list of the <see cref="StudyPlanning.DAL.Courses.CoursePart"/> items associated with
    /// the <see cref="StudyPlanning.DAL.TechnicalPackages.PeriodCourseItem"/>.
    /// </summary>
    /// <param name="technicalPackage_PeriodCourse_ID"> The identification of the
    /// <see cref="StudyPlanning.DAL.TechnicalPackages.PeriodCourse"/>.</param>
    public static CoursePart[] GetCourses(int technicalPackage_PeriodCourse_ID)
    {
      string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
      SqlConnection objConnection = new SqlConnection(connString);

      SqlCommand objCommand = new SqlCommand("TechnicalPackage_PeriodCourseItem_GetCourses", objConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      SqlParameter objParam = objCommand.Parameters.Add("@TechnicalPackage_PeriodCourse_ID", SqlDbType.Int);
      objParam.Value = technicalPackage_PeriodCourse_ID;

      DataSet objDataSet = new DataSet("Courses");
      SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

      try
      {
        objConnection.Open();
        objAdap.Fill(objDataSet, "Courses");
        objConnection.Close();
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }
      int intRows = objDataSet.Tables["Courses"].Rows.Count;
      CoursePart[] courses = new CoursePart[intRows];

      for(int i=0; i < intRows; i++)
      {
        courses[i] = new CoursePart();
        courses[i].Course_ID = (Guid)objDataSet.Tables["Courses"].Rows[i]["Course_ID"];
        courses[i].Part   = (int)objDataSet.Tables["Courses"].Rows[i]["Part"];
      }
      return courses;
    }

    #endregion //Retrieve Methods

    #endregion //Public Methods

  }
}
```

1.28.6 PeriodOptionalCourse

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;
using StudyPlanning.DAL.Courses;
```

```
namespace StudyPlanning.DAL.TechnicalPackages
{
  /// <summary>
  /// Represents the TechnicalPackage_PeriodOptionalCourse table.
  /// </summary>
  public class PeriodOptionalCourse : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalInt _TechnicalPackage_PeriodOptionalCourse_ID = new DalInt(false);
    private DalInt _TechnicalPackage_Period_ID = new DalInt(false);
    private DalGuid _Course_ID = new DalGuid(true);
    private DalInt _Part = new DalInt(false);
    private DalString _Number = new DalString(true);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.TechnicalPackages.PeriodOptionalCourse"/>
    /// class.
    /// </summary>
    public PeriodOptionalCourse() {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the TechnicalPackage_PeriodOptionalCourse_ID.
    /// </summary>
    /// <value></value>
    public DalInt TechnicalPackage_PeriodOptionalCourse_ID
    {
      get { return _TechnicalPackage_PeriodOptionalCourse_ID; }
    }

    /// <summary>
    /// Gets the TechnicalPackage_Period_ID.
    /// </summary>
    /// <value></value>
    public DalInt TechnicalPackage_Period_ID
    {
      get { return _TechnicalPackage_Period_ID; }
    }

    /// <summary>
    /// Gets the Course_ID.
    /// </summary>
    /// <value></value>
    public DalGuid Course_ID
    {
      get { return _Course_ID; }
    }

    /// <summary>
    /// Gets the Part.
    /// </summary>
    /// <value></value>
    public DalInt Part
    {
      get { return _Part; }
    }

    /// <summary>
    /// Gets the Number.
    /// </summary>
    /// <value></value>
    public DalString Number
    {
      get { return _Number; }
    }

    /// <summary>
    /// Gets the data log of table.
    /// </summary>
    public DataLog Log
    {
      get { return _Log; }
    }

    #endregion //Public Properties

    #region Public Methods

    #region Retrieve Methods

    /// <summary>
    /// Retrieves the TechnicalPackage_PeriodOptionalCourse object from the database using the value of the
    /// TechnicalPackage_PeriodOptionalCourse_ID property of the current instance
    /// </summary>
    public void Retrieve()
    {
      DataRow objDataRow;
```

```
    try
    {
      _TechnicalPackage_PeriodOptionalCourse_ID.Validate("TechnicalPackage_PeriodOptionalCourse_ID");
    }
    catch (DalException excp)
    {
      throw excp;
    }

    try
    {
      objDataRow = RetrieveExecute(_TechnicalPackage_PeriodOptionalCourse_ID.Value, objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }

    this._TechnicalPackage_PeriodOptionalCourse_ID.Value = Convert.ToInt32(objDataRow["TechnicalPackage_PeriodOptionalCourse_ID"]);
    this._TechnicalPackage_Period_ID.Value        = Convert.ToInt32(objDataRow["TechnicalPackage_Period_ID"]);

    if (objDataRow["Course_ID"].Equals(System.DBNull.Value))
      this._Course_ID.IsNull = true;
    else
      this._Course_ID.Value = (Guid)objDataRow["Course_ID"];

    this._Part.Value = Convert.ToInt32(objDataRow["Part"]);

    if (objDataRow["Number"].Equals(System.DBNull.Value))
      this._Number.IsNull = true;
    else
      this._Number.Value = (string)objDataRow["Number"];

    this._Log.SetValues(objDataRow);
  }

  /// <summary>
  /// Retrieves the TechnicalPackage_PeriodOptionalCourse object with the specified identifier from the database.
  /// </summary>
  /// <param name="technicalPackage_PeriodOptionalCourse_ID">The unique identification of the TechnicalPackage_PeriodOptionalCourse object.</
    param>
  public static PeriodOptionalCourse Retrieve(int technicalPackage_PeriodOptionalCourse_ID)
  {
    DataRow objDataRow;
    string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
    SqlConnection objConnection = new SqlConnection(connString);

    PeriodOptionalCourse objTP_PeriodOptionalCourse = new PeriodOptionalCourse();

    try
    {
      objDataRow = RetrieveExecute(technicalPackage_PeriodOptionalCourse_ID, objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }

    objTP_PeriodOptionalCourse._TechnicalPackage_PeriodOptionalCourse_ID.Value = Convert.ToInt32(objDataRow["
      TechnicalPackage_PeriodOptionalCourse_ID"]);
    objTP_PeriodOptionalCourse._TechnicalPackage_Period_ID.Value   = Convert.ToInt32(objDataRow["TechnicalPackage_Period_ID"]);

    if (objDataRow["Course_ID"].Equals(System.DBNull.Value))
      objTP_PeriodOptionalCourse._Course_ID.IsNull = true;
    else
      objTP_PeriodOptionalCourse._Course_ID.Value = (Guid)objDataRow["Course_ID"];

    objTP_PeriodOptionalCourse._Part.Value = Convert.ToInt32(objDataRow["Part"]);

    if (objDataRow["Number"].Equals(System.DBNull.Value))
      objTP_PeriodOptionalCourse._Number.IsNull = true;
    else
      objTP_PeriodOptionalCourse._Number.Value = (string)objDataRow["Number"];

    objTP_PeriodOptionalCourse._Log.SetValues(objDataRow);
    return objTP_PeriodOptionalCourse;
  }

  private static DataRow RetrieveExecute(
    int technicalPackage_PeriodOptionalCourse_ID,
    SqlConnection sqlConnection)
  {
    SqlCommand objCommand = new SqlCommand("TechnicalPackage_PeriodOptionalCourse_Select", sqlConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam;

    objParam = objCommand.Parameters.Add("@TechnicalPackage_PeriodOptionalCourse_ID", SqlDbType.Int);
    objParam.Value = technicalPackage_PeriodOptionalCourse_ID;

    DataSet objDataSet = new DataSet("TechnicalPackage_PeriodOptionalCourse");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
      sqlConnection.Open();
      objAdap.Fill(objDataSet, "TechnicalPackage_PeriodOptionalCourse");
      sqlConnection.Close();
    }
```

```
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }
      DataRow objDataRow = objDataSet.Tables["TechnicalPackage_PeriodOptionalCourse"].Rows[0];
      return objDataRow;
    }

    /// <summary>
    /// Retrieves a list of the optional <see cref="StudyPlanning.DAL.Courses.CoursePart"/> items associated with
    /// the <see cref="StudyPlanning.DAL.TechnicalPackages.Period"/>.
    /// </summary>
    /// <param name="technicalPackage_Period_ID"> The identification of the
    /// <see cref="StudyPlanning.DAL.TechnicalPackages.Period"/>.</param>
    public static CoursePart[] GetCourses(int technicalPackage_Period_ID)
    {
      string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
      SqlConnection objConnection = new SqlConnection(connString);

      SqlCommand objCommand = new SqlCommand("TechnicalPackage_PeriodOptionalCourse_GetCourses", objConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      SqlParameter objParam = objCommand.Parameters.Add("@TechnicalPackage_Period_ID", SqlDbType.Int);
      objParam.Value = technicalPackage_Period_ID;

      DataSet objDataSet = new DataSet("Result");
      SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

      try
      {
        objConnection.Open();
        objAdap.Fill(objDataSet, "Result");
        objConnection.Close();
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }
      int intRows = objDataSet.Tables["Result"].Rows.Count;
      CoursePart[] courses = new CoursePart[intRows];

      for(int i=0; i < intRows; i++)
      {
        courses[i] = new CoursePart();
        courses[i].Course_ID = (Guid)objDataSet.Tables["Result"].Rows[i]["Course_ID"];
        courses[i].Part    = (int)objDataSet.Tables["Result"].Rows[i]["Part"];
      }
      return courses;
    }

    #endregion //Retrieve Methods

    #endregion //Public Methods

  }
}
```

1.28.7 Project

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.TechnicalPackages
{
  /// <summary>
  /// Represents the TechnicalPackage_Project table
  /// </summary>
  public class Project : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalInt _TechnicalPackage_Project_ID = new DalInt(false);
    private DalInt _TechnicalPackageVersion_ID = new DalInt(false);
    private DalGuid _Project_ID = new DalGuid(false);
    private DalGuid _Period_ID = new DalGuid(false);
    private DalGuid _Point_ID = new DalGuid(false);
    private DalInt _Months = new DalInt(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.TechnicalPackages.Project"/>
    /// class.
    /// </summary>
    public Project() {
      _Log = new DataLog();
    }

    #endregion
```

```
#region Public Properties

/// <summary>
/// Gets the TechnicalPackage_Project_ID.
/// </summary>
public DalInt TechnicalPackage_Project_ID
{
  get { return _TechnicalPackage_Project_ID; }
}

/// <summary>
/// Gets the TechnicalPackageVersion_ID
/// </summary>
public DalInt TechnicalPackageVersion_ID
{
  get { return _TechnicalPackageVersion_ID; }
}

/// <summary>
/// Gets the Project_ID
/// </summary>
public DalGuid Project_ID
{
  get { return _Project_ID; }
}

/// <summary>
/// Gets the Period_ID
/// </summary>
public DalGuid Period_ID
{
  get { return _Period_ID; }
}

/// <summary>
/// Gets the Point_ID.
/// </summary>
public DalGuid Point_ID
{
  get { return _Point_ID; }
}

/// <summary>
/// Gets the Point_ID.
/// </summary>
public DalInt Months
{
  get { return _Months; }
}

/// <summary>
/// Gets the data log of table.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Methods

#region Retrieve Methods

/// <summary>
/// Retrieves the project from the database using the value of the
/// TechnicalPackage_Project_ID property of the current instance.
/// </summary>
public void Retrieve()
{
  try
  {
    this._TechnicalPackage_Project_ID.Validate("TechnicalPackage_Project_ID");
  }
  catch (DalException excp)
  {
    throw excp;
  }

  try
  {
    RetrieveExecute(_TechnicalPackage_Project_ID.Value, this, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
}

/// <summary>
/// Retrieves the project with the specified identifier from the database.
/// </summary>
/// <param name="technicalPackageProject_ID">The globally unique identifier of the project.</param>
public static Project Retrieve(int technicalPackageProject_ID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  Project objPrj = new Project();
```

```
      try
      {
        RetrieveExecute(technicalPackageProject_ID, objPrj, objConnection);
      }
      catch (DalException objExc)
      {
        throw objExc;
      }

      return objPrj;
    }

    private static void RetrieveExecute(
      int technicalPackageProject_ID,
      Project objProject,
      SqlConnection sqlConnection)
    {
      SqlCommand objCommand = new SqlCommand("TechnicalPackage_Project_Select", sqlConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      SqlParameter objParam;

      objParam = objCommand.Parameters.Add("@TechnicalPackage_Project_ID", SqlDbType.Int);
      objParam.Value = technicalPackageProject_ID;

      DataSet objDataSet = new DataSet("Result");
      SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

      try
      {
        sqlConnection.Open();
        objAdap.Fill(objDataSet, "Result");
        sqlConnection.Close();
        DataRow objDataRow = objDataSet.Tables["Result"].Rows[0];

        objProject.TechnicalPackage_Project_ID.Value = (int)objDataRow["TechnicalPackage_Project_ID"];
        objProject.TechnicalPackageVersion_ID.Value = (int)objDataRow["TechnicalPackageVersion_ID"];
        objProject.Project_ID.Value = (Guid)objDataRow["Project_ID"];
        objProject.Period_ID.Value = (Guid)objDataRow["Period_ID"];
        objProject.Point_ID.Value = (Guid)objDataRow["Point_ID"];
        objProject.Months.Value = (int)objDataRow["Months"];
        objProject._Log.SetValues(objDataRow);
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }
    }

    #endregion //Retrive Methods

    #endregion //Methods

  }
}
```

1.28.8 TechnicalPackageVersion

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.TechnicalPackages
{
  /// <summary>
  /// Represents the TechnicalPackageVersion table.
  /// </summary>
  public class TechnicalPackageVersion : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalInt _TechnicalPackageVersion_ID = new DalInt(false);
    private DalInt _TechnicalPackage_ID = new DalInt(false);
    private DalInt _Version = new DalInt(false);
    private DalString _Number = new DalString(false);
    private DalStringLocalizable _Name = new DalStringLocalizable(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.TechnicalPackages.TechnicalPackageVersion"/>
    /// class.
    /// </summary>
    public TechnicalPackageVersion() {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties
```

```
/// <summary>
/// Gets the value of the TechnicalPackageVersion_ID attribute.
/// </summary>
/// <value></value>
public DalInt TechnicalPackageVersion_ID
{
  get { return _TechnicalPackageVersion_ID; }
}

/// <summary>
/// Gets the value of the TechnicalPackage_ID attribute .
/// </summary>
/// <value></value>
public DalInt TechnicalPackage_ID
{
  get { return _TechnicalPackage_ID; }
}

/// <summary>
/// Gets the value of the Version attribute.
/// </summary>
/// <value></value>
public DalInt Version
{
  get { return _Version; }
}

/// <summary>
/// Gets the value of the Number attribute.
/// </summary>
/// <value></value>
public DalString Number
{
  get { return _Number; }
}

/// <summary>
/// Gets the value of the Name attribute.
/// </summary>
/// <value></value>
public DalStringLocalizable Name
{
  get { return _Name; }
}

/// <summary>
/// Gets the data log of the table.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Public Methods

#region Retrieve Methods

/// <summary>
/// Retrieves the technical package version from the database using the value of the
/// TechnicalPackageVersion_ID property of the current instance.
/// </summary>
public void Retrieve()
{
  DataRow objDataRow;

  try
  {
    _TechnicalPackageVersion_ID.Validate("TechnicalPackageVersion_ID");
  }
  catch (DalException excp)
  {
    throw excp;
  }

  try
  {
    objDataRow = RetrieveExecute(_TechnicalPackageVersion_ID.Value, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }

  this._TechnicalPackageVersion_ID.Value = Convert.ToInt32(objDataRow["TechnicalPackageVersion_ID"]);
  this._TechnicalPackage_ID.Value     = Convert.ToInt32(objDataRow["TechnicalPackage_ID"]);
  this._Version.Value           = Convert.ToInt32(objDataRow["Version"]);
  this._Number.Value            = (string)objDataRow["Number"];
  this._Name.LoadXml((string)objDataRow["Name"]);
  this._Log.SetValues(objDataRow);
}

/// <summary>
/// Retrieves the technical package version with the specified identifier from the database.
/// </summary>
/// <param name="technicalPackageVersion_ID">The TechnicalPackageVersion_ID.</param>
public static TechnicalPackageVersion Retrieve(int technicalPackageVersion_ID)
```

```
    {
      DataRow objDataRow;
      string connString = System.Con|guration.Con|gurationSettings.AppSettings["sqlConnectionString"];
      SqlConnection objConnection = new SqlConnection(connString);

      TechnicalPackageVersion objTPVersion = new TechnicalPackageVersion();

      try
      {
        objDataRow = RetrieveExecute(technicalPackageVersion_ID, objConnection);
      }
      catch (DalException objExc)
      {
        throw objExc;
      }
      objTPVersion._TechnicalPackageVersion_ID.Value = Convert.ToInt32(objDataRow["TechnicalPackageVersion_ID"]);
      objTPVersion._TechnicalPackage_ID.Value     = Convert.ToInt32(objDataRow["TechnicalPackage_ID"]);
      objTPVersion._Version.Value           = Convert.ToInt32(objDataRow["Version"]);
      objTPVersion._Number.Value            = (string)objDataRow["Number"];
      objTPVersion._Name.LoadXml((string)objDataRow["Name"]);
      objTPVersion._Log.SetValues(objDataRow);
      return objTPVersion;
    }

    private static DataRow RetrieveExecute(
      int technicalPackageVersion_ID,
      SqlConnection sqlConnection)
    {
      SqlCommand objCommand = new SqlCommand("TechnicalPackageVersion_Select", sqlConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      SqlParameter objParam;

      objParam = objCommand.Parameters.Add("@TechnicalPackageVersion_ID", SqlDbType.Int);
      objParam.Value = technicalPackageVersion_ID;

      DataSet objDataSet = new DataSet("TechnicalPackageVersion");
      SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

      try
      {
        sqlConnection.Open();
        objAdap.Fill(objDataSet, "TechnicalPackageVersion");
        sqlConnection.Close();
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }
      DataRow objDataRow = objDataSet.Tables["TechnicalPackageVersion"].Rows[0];
      return objDataRow;
    }

    #endregion //Retrieve Methods

    #endregion //Public Methods

  }
}
```

1.29 Text

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;
using System.Collections;

namespace StudyPlanning.DAL
{
  /// <summary>
  /// Represents a course version.
  /// </summary>
  public class TextItem : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalInt _Text_ID = new DalInt(false);
    private DalInt _TextGroup_ID = new DalInt(false);
    private DalString _NumberInGroup = new DalString(false);
    private DalString _Culture_ID = new DalString(false);
    private DalString _Text = new DalString(true);
    private DalString _Description = new DalString(true);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Initializes a new instance of the
    /// <see cref="StudyPlanning.DAL.TextItem" />
    /// class.
    /// </summary>
```

```
public TextItem()
{
  _Log = new DataLog();
}

/// <summary>
/// Initializes a new instance of the <see cref="StudyPlanning.DAL.TextItem" />
/// class with the a text group ID, the number of the text in the group, and
/// a culture ID.
/// </summary>
/// <param name="textGroup_ID">The ID of the text group.</param>
/// <param name="numberInGroup">The number of the text in the group.</param>
/// <param name="culture_ID">The ID of the culture.</param>
public TextItem(
  int textGroup_ID,
  string numberInGroup,
  string culture_ID)
{
  this._TextGroup_ID.Value = textGroup_ID;
  this._NumberInGroup.Value = numberInGroup;
  this._Culture_ID.Value = culture_ID;
  _Log = new DataLog();
}

#endregion //Constructors

#region Public Properties

/// <summary>
/// Gets or sets the <see cref="StudyPlanning.DAL.DalInt">DalInt</see>
/// object for the text ID.
/// </summary>
public DalInt Text_ID
{
  get { return _Text_ID; }
  set { _Text_ID = value; }
}

/// <summary>
/// Gets or sets the <see cref="StudyPlanning.DAL.DalInt">DalInt</see>
/// object for the text group ID.
/// </summary>
public DalInt TextGroup_ID
{
  get { return _TextGroup_ID; }
  set { _TextGroup_ID = value; }
}

/// <summary>
/// Gets or sets the <see cref="StudyPlanning.DAL.DalString">DalString</see>
/// object for the text number in the group.
/// </summary>
public DalString NumberInGroup
{
  get { return _NumberInGroup; }
  set { _NumberInGroup = value; }
}

/// <summary>
/// Gets or sets the <see cref="StudyPlanning.DAL.DalString">DalString</see>
/// object for the culture ID.
/// </summary>
public DalString Culture_ID
{
  get { return _Culture_ID; }
  set { _Culture_ID = value; }
}

/// <summary>
/// Gets or sets the <see cref="StudyPlanning.DAL.DalString">DalString</see>
/// object for the text itself.
/// </summary>
public DalString Text
{
  get { return _Text; }
  set { _Text = value; }
}

/// <summary>
/// Gets or sets the <see cref="StudyPlanning.DAL.DataLog">DataLog</see>
/// object for the data log.
/// </summary>
public DataLog Log
{
  get { return _Log; }
  set { _Log = value; }
}

#endregion //Public Properties

#region Retrieve Methods

/// <summary>
/// Retrieves the text from the database using the value of the
/// Text_ID property of the current instance.
/// </summary>
public void Retrieve()
{
  try
```

```
    {
      _TextGroup_ID.Validate("TextGroup_ID");
      _NumberInGroup.Validate("NumberInGroup");
      _Culture_ID.Validate("Culture_ID");
    }
    catch (DalException objEx)
    {
      throw objEx;
    }

    try
    {
      DataRow objDataRow =
        RetrieveExecute(
          _TextGroup_ID.Value,
          NumberInGroup.Value,
          _Culture_ID.Value,
          objConnection
        );

      this._Text_ID.Value = Convert.ToInt32(objDataRow["Text_ID"]);
      this._TextGroup_ID.Value = Convert.ToInt32(objDataRow["TextGroup_ID"]);
      this._NumberInGroup.Value = (string)objDataRow["NumberInGroup"];
      this._Culture_ID.Value = (string)objDataRow["Culture_ID"];

      if (objDataRow["Text"].Equals(System.DBNull.Value))
        this._Text.IsNull = true;
      else
        this._Text.Value = (string)objDataRow["Text"];

      if (objDataRow["Description"].Equals(System.DBNull.Value))
        this._Description.IsNull = true;
      else
        this._Description.Value = (string)objDataRow["Description"];

      this._Log.Created.Value = (DateTime)objDataRow["Created"];
      this._Log.CreatedBy.Value = (Guid)objDataRow["CreatedBy"];
      this._Log.Updated.Value = (DateTime)objDataRow["Updated"];
      this._Log.UpdatedBy.Value = (Guid)objDataRow["UpdatedBy"];
    }
    catch (SqlException objExc)
    {
      throw objExc;
    }
  }

  /// <summary>
  /// Retrieves the database the text with the specified text group ID, number in
  /// the text group and culture ID.
  /// </summary>
  /// <param name="textGroup_ID">The ID of the text group.</param>
  /// <param name="textNumberInGroup">The number of the text in the group.</param>
  /// <param name="culture_ID">The ID of the culture.</param>
  /// <returns>A <see cref="StudyPlanning.DAL.TextItem">TextItem</see> object
  /// containing the text string and a possibly a description.
  /// </returns>
  public static TextItem Retrieve(int textGroup_ID, string textNumberInGroup, string culture_ID)
  {
    string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
    SqlConnection objConnection = new SqlConnection(connString);

    StudyPlanning.DAL.TextItem objText = new StudyPlanning.DAL.TextItem();

    try
    {
      DataRow objDataRow =
        RetrieveExecute(textGroup_ID, textNumberInGroup, culture_ID, objConnection);

      objText._Text_ID.Value = Convert.ToInt32(objDataRow["Text_ID"]);
      objText._TextGroup_ID.Value = Convert.ToInt32(objDataRow["TextGroup_ID"]);
      objText._NumberInGroup.Value = (string)objDataRow["NumberInGroup"];
      objText._Culture_ID.Value = (string)objDataRow["Culture_ID"];

      if (objDataRow["Text"].Equals(System.DBNull.Value))
        objText._Text.IsNull = true;
      else
        objText._Text.Value = (string)objDataRow["Text"];

      if (objDataRow["Description"].Equals(System.DBNull.Value))
        objText._Description.IsNull = true;
      else
        objText._Description.Value = (string)objDataRow["Description"];

      objText.Log.Created.Value = (DateTime)objDataRow["Created"];
      objText.Log.CreatedBy.Value = (Guid)objDataRow["CreatedBy"];
      objText.Log.Updated.Value = (DateTime)objDataRow["Updated"];
      objText.Log.UpdatedBy.Value = (Guid)objDataRow["UpdatedBy"];

      return objText;
    }
    catch (SqlException objExc)
    {
      throw objExc;
    }
  }

  private static DataRow RetrieveExecute(
    int textGroup_ID,
    string textNumberInGroup,
```

```
      string culture_ID,
      SqlConnection sqlConnection)
    {
      SqlCommand objCommand = new SqlCommand("Text_Select", sqlConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      SqlParameter objParam;

      objParam = objCommand.Parameters.Add("@TextGroup_ID", SqlDbType.Int, 4);
      objParam.Value = textGroup_ID;

      objParam = objCommand.Parameters.Add("@NumberInGroup", SqlDbType.VarChar, 25);
      objParam.Value = textNumberInGroup;

      objParam = objCommand.Parameters.Add("@Culture_ID", SqlDbType.VarChar, 10);
      objParam.Value = culture_ID;

      DataSet objDataSet = new DataSet("Text");
      SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

      try
      {
        sqlConnection.Open();
        objAdap.Fill(objDataSet, "Text");
        sqlConnection.Close();
        DataRow objDataRow = objDataSet.Tables["Text"].Rows[0];

        return objDataRow;
      }
      catch (SqlException objExc)
      {
        throw objExc;
      }
    }

    #endregion //Retrieve Methods

    /// <summary>
    /// Retrives all the texts in the specified text group in the
    /// specified culture.
    /// </summary>
    /// <param name="textGroup_ID">The ID of the text group for which
    /// to retrieve texts.</param>
    /// <param name="culture_ID">The ID of the culture for which to
    /// retrieve texts.</param>
    /// <returns>A <see cref="System.Collections.Hashtable"/> containing the texts
    /// with the text number as key and the text itself as value.
    /// </returns>
    public static Hashtable GetTextsInGroup(int textGroup_ID, string culture_ID)
    {
      Hashtable texts = new Hashtable();

      string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
      SqlConnection objConnection = new SqlConnection(connString);

      SqlCommand objCommand = new SqlCommand("Text_GetTextsInGroup", objConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      SqlParameter objParam;

      objParam = objCommand.Parameters.Add("@TextGroup_ID", SqlDbType.Int, 4);
      objParam.Value = textGroup_ID;

      objParam = objCommand.Parameters.Add("@Culture_ID", SqlDbType.VarChar, 10);
      objParam.Value = culture_ID;

      SqlDataReader objReader;

      try
      {
        objConnection.Open();
        objReader = objCommand.ExecuteReader();

        while (objReader.Read())
        {
          texts.Add((string)objReader["NumberInGroup"],(string)objReader["Text"]);
        }
        objConnection.Close();
      }
      catch (SqlException objExc)
      {
        throw objExc;
      }

      return texts;
    }
  }
}
```

1.30 Users

1.30.1 Login

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.Users
{
  /// <summary>
  /// Represents a user login.
  /// </summary>
  public class Login : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalGuid _User_Login_ID = new DalGuid(false);
    private DalGuid _User_ID = new DalGuid(false);
    private DalInt _User_LoginType_ID = new DalInt(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.Users.Login"/>
    /// class.
    /// </summary>
    public Login() {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the User_Login_ID.
    /// </summary>
    public DalGuid User_Login_ID
    {
      get { return _User_Login_ID; }
    }

    /// <summary>
    /// Gets the User_ID.
    /// </summary>
    public DalGuid User_ID
    {
      get { return _User_ID; }
    }

    /// <summary>
    /// Gets the User_LoginType_ID.
    /// </summary>
    public DalInt User_LoginType_ID
    {
      get { return _User_LoginType_ID; }
    }

    /// <summary>
    /// Gets the data log of the user.
    /// </summary>
    public DataLog Log
    {
      get { return _Log; }
    }

    #endregion //Public Properties

    #region Public Methods

    /// <summary>
    /// Creates a new user login in the database using the values of the properties
    /// of the current instance.
    /// </summary>
    public void Create()
    {
      try
      {
        this.Validate();
      }
      catch (DalException excp)
      {
        throw excp;
      }

      SqlCommand objCommand = new SqlCommand("User_Login_Insert", objConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      AddParameters(this, objCommand, false);

      try
      {
        objConnection.Open();
        objCommand.ExecuteNonQuery();
        objConnection.Close();
      }
      catch (SqlException objExc)
      {
```

```
      throw new DalException(objExc);
    }
  }

  #region Retrieve Methods

  /// <summary>
  /// Retrieves the user login from the database using the value of the
  /// User_Login_ID property of the current instance
  /// </summary>
  public void Retrieve()
  {
    try
    {
      this._User_Login_ID.Validate("User_Login_ID");
    }
    catch (DalException e)
    {
      throw e;
    }

    try
    {
      RetrieveExecute(_User_Login_ID.Value, this, objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }
  }

  /// <summary>
  /// Retrieves the user login with the specified identifier from the database.
  /// </summary>
  /// <param name="user_Login_ID">The ID of the user login to retrieve.</param>
  public static StudyPlanning.DAL.Users.Login Retrieve(Guid user_Login_ID)
  {
    string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
    SqlConnection objConnection = new SqlConnection(connString);

    StudyPlanning.DAL.Users.Login objLogin =
      new StudyPlanning.DAL.Users.Login();

    try
    {
      RetrieveExecute(user_Login_ID, objLogin, objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }
    return objLogin;
  }

  private static void RetrieveExecute(
    Guid user_Login_ID,
    StudyPlanning.DAL.Users.Login login,
    SqlConnection sqlConnection)
  {
    DataRow objDataRow;

    SqlCommand objCommand = new SqlCommand("User_Login_Select", sqlConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam;

    objParam = objCommand.Parameters.Add("@User_Login_ID", SqlDbType.UniqueIdenti↓er);
    objParam.Value = user_Login_ID;

    DataSet objDataSet = new DataSet("Login");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
      sqlConnection.Open();
      objAdap.Fill(objDataSet, "Login");
      sqlConnection.Close();
      objDataRow = objDataSet.Tables["Login"].Rows[0];
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
    login.User_Login_ID.Value = (Guid)objDataRow["User_Login_ID"];
    login.User_ID.Value = (Guid)objDataRow["User_ID"];
    login.User_LoginType_ID.Value = (int)objDataRow["User_LoginType_ID"];
    login.Log.SetValues(objDataRow);
  }

  #endregion //Retrieve Methods

  /// <summary>
  /// Updates the current user login in the database using the original
  /// user login data to resolve possible concurrency issues.
  /// </summary>
  /// <param name="originalLogin">The original <see cref="StudyPlanning.DAL.Users.Login"/> object.</param>
  /// <returns><strong>true</strong> if the update was succesfully executed; otherwise, <strong>false</strong>.</returns>
  public bool Update(StudyPlanning.DAL.Users.Login originalLogin)
  {
```

```
    try
    {
      this.Validate();
    }
    catch (DalException excp)
    {
      throw excp;
    }

    bool blnResult = false;

    SqlCommand objCommand = new SqlCommand("User_Login_Update", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, false);
    AddParameters(originalLogin, objCommand, true);

    int rowsA↑ected;

    try
    {
      objConnection.Open();
      rowsA↑ected = objCommand.ExecuteNonQuery();
      objConnection.Close();

      if (rowsA↑ected > 0)
        blnResult = true;
    }
    catch(SqlException excp)
    {
      throw new DalException(excp);
    }

    return blnResult;
  }

  /// <summary>
  /// Deletes the current user login from the database.
  /// </summary>
  /// <returns><strong>true</strong> if the deletion is succesfully executed; otherwise, <strong>false</strong>.</returns>
  public bool Delete()
  {
    bool blnResult = false;

    try
    {
      this.Validate();
    }
    catch (DalException excp)
    {
      throw excp;
    }

    SqlCommand objCommand = new SqlCommand("User_Login_Delete", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, true);

    try
    {
      objConnection.Open();
      int rowsA↑ected = objCommand.ExecuteNonQuery();
      objConnection.Close();

      if (rowsA↑ected > 0)
        blnResult = true;
    }
    catch(SqlException objEx)
    {
      throw new DalException(objEx);
    }

    return blnResult;
  }

  /// <summary>
  /// Adds the relevant parameters to the specified <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see> object
  /// setting the values to ones of the specified <see cref="StudyPlanning.DAL.Users.Login"/> object.
  /// </summary>
  /// <param name="login">
  /// The <see cref="StudyPlanning.DAL.Users.Login"/> object containing the values that the parameters have to be
  /// initialized to.
  /// </param>
  /// <param name="objCommand">
  /// The <see cref="System.Data.SqlClient.SqlCommand"/> object to which parameters should be added.
  /// </param>
  /// <param name="isOriginal">
  /// <strong>true</strong> if the parameters to be added represent original
  /// data (for handling concurrent programming issues); otherwise, <strong>false</strong>.</param>
  /// <returns>The <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see>
  /// object with the relevant parameters added.</returns>
  private static void AddParameters(Login login, SqlCommand objCommand, bool isOriginal)
  {
    string paramName = "";
    SqlParameter objParam;

    //User_Login_ID
    paramName = "User_Login_ID";
    objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);
```

```
    if (isOriginal)
      objParam.ParameterName = "@Original_" + paramName;

    objParam.Value = login.User_Login_ID.Value;

    //User_ID
    paramName = "User_ID";
    objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

    if (isOriginal)
      objParam.ParameterName = "@Original_" + paramName;

    objParam.Value = login.User_ID.Value;

    //User_LoginType_ID
    paramName = "User_LoginType_ID";
    objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.Int, 4);

    if (isOriginal)
      objParam.ParameterName = "@Original_" + paramName;

    objParam.Value = login.User_LoginType_ID.Value;

    //Log
    objCommand = login.Log.AddParameters(objCommand, isOriginal);
  }

  /// <summary>
  /// Validates the current <see cref="StudyPlanning.DAL.Users.Login"/> object. If the
  /// value of either of the private properties is null a
  /// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
  /// </summary>
  private void Validate()
  {
    try
    {
      _User_Login_ID.Validate("User_Login_ID");
      _User_ID.Validate("User_ID");
      _User_LoginType_ID.Validate("User_LoginType_ID");
      _Log.Validate();
    }
    catch (DalException excp)
    {
      throw excp;
    }
  }

  /// <summary>
  /// Gets the number of logins for the specified user, of the specified type,
  /// the specified number of minutes back in time using the specified
  /// current date and time.
  /// </summary>
  /// <param name="userID">The ID of the user for which to return logins.</param>
  /// <param name="dtmNow">The current date and time.</param>
  /// <param name="loginType">The type of login which the returned number should reflect.</param>
  /// <param name="withinMinutes">The number of minutes from now and back in time for which to get the number of logins.</param>
  /// <returns>The number of logins of the specified type for the specified user within the specified number of minutes.</returns>
  public static int GetNumberOfLogins(Guid userID, int loginType, DateTime dtmNow, int withinMinutes)
  {
    int numberOfLogins;

    string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
    SqlConnection objConnection = new SqlConnection(connString);

    SqlCommand objCommand = new SqlCommand("User_Login_GetNumberOfLogins", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam;

    objParam = objCommand.Parameters.Add("@User_ID", SqlDbType.UniqueIdenti↓er, 16);
    objParam.Value = userID;

    objParam = objCommand.Parameters.Add("@LoginType_ID", SqlDbType.Int, 4);
    objParam.Value = loginType;

    objParam = objCommand.Parameters.Add("@CurrentDateAndTime", SqlDbType.DateTime, 8);
    objParam.Value = dtmNow;

    objParam = objCommand.Parameters.Add("@WithinMinutes", SqlDbType.Int, 4);
    objParam.Value = withinMinutes;

    objParam = objCommand.Parameters.Add("@NumberOfLogins", SqlDbType.Int, 4);
    objParam.Direction = ParameterDirection.Output;

    try
    {
      objConnection.Open();
      objCommand.ExecuteNonQuery();
      numberOfLogins = (int)objCommand.Parameters["@NumberOfLogins"].Value;
      objConnection.Close();
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }

    return numberOfLogins;
  }
```

```
    #endregion //Public Methods

  }
}
```

1.30.2 PasswordHistory

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.Users
{
  /// <summary>
  /// Represents a previous password of a user.
  /// </summary>
  public class PasswordHistory : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalGuid _User_PasswordHistory_ID = new DalGuid(false);
    private DalGuid _User_ID = new DalGuid(false);
    private DalString _Password = new DalString(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.Users.PasswordHistory"/>
    /// class.
    /// </summary>
    public PasswordHistory() {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the User_PasswordHistory_ID.
    /// </summary>
    public DalGuid User_PasswordHistory_ID
    {
      get { return _User_PasswordHistory_ID; }
    }

    /// <summary>
    /// Gets the User_ID.
    /// </summary>
    public DalGuid User_ID
    {
      get { return _User_ID; }
    }

    /// <summary>
    /// Gets the Password.
    /// </summary>
    public DalString Password
    {
      get { return _Password; }
    }

    /// <summary>
    /// Gets the data log of the user.
    /// </summary>
    public DataLog Log
    {
      get { return _Log; }
    }

    #endregion //Public Properties

    #region Public Methods

    /// <summary>
    /// Creates a new previous password of a user in the database using the values of the properties
    /// of the current instance.
    /// </summary>
    public void Create()
    {
      try
      {
        this.Validate();
      }
      catch (DalException excp)
      {
        throw excp;
      }

      SqlCommand objCommand = new SqlCommand("User_PasswordHistory_Insert", objConnection);
```

```
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, false);

    try
    {
      objConnection.Open();
      objCommand.ExecuteNonQuery();
      objConnection.Close();
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
  }

  #region Retrieve Methods

  /// <summary>
  /// Retrieves the historic password of a user from the database using the value of the
  /// User_PasswordHistory_ID property of the current instance
  /// </summary>
  public void Retrieve()
  {
    try
    {
      this._User_PasswordHistory_ID.Validate("User_PasswordHistory_ID");
    }
    catch (DalException e)
    {
      throw e;
    }

    try
    {
      RetrieveExecute(_User_PasswordHistory_ID.Value, this, objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }
  }

  /// <summary>
  /// Retrieves the historic password of a user with the specified identifier from the database.
  /// </summary>
  /// <param name="user_PasswordHistory_ID">The ID of the historic password to retrieve.</param>
  public static StudyPlanning.DAL.Users.PasswordHistory Retrieve(Guid user_PasswordHistory_ID)
  {
    string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
    SqlConnection objConnection = new SqlConnection(connString);

    StudyPlanning.DAL.Users.PasswordHistory objPasswordHistory =
      new StudyPlanning.DAL.Users.PasswordHistory();

    try
    {
      RetrieveExecute(user_PasswordHistory_ID, objPasswordHistory, objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }
    return objPasswordHistory;
  }

  private static void RetrieveExecute(
    Guid user_PasswordHistory_ID,
    StudyPlanning.DAL.Users.PasswordHistory passwordHistory,
    SqlConnection sqlConnection)
  {
    DataRow objDataRow;

    SqlCommand objCommand = new SqlCommand("User_PasswordHistory_Select", sqlConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam;

    objParam = objCommand.Parameters.Add("@User_PasswordHistory_ID", SqlDbType.UniqueIdenti↓er);
    objParam.Value = user_PasswordHistory_ID;

    DataSet objDataSet = new DataSet("PasswordHistory");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
      sqlConnection.Open();
      objAdap.Fill(objDataSet, "PasswordHistory");
      sqlConnection.Close();
      objDataRow = objDataSet.Tables["PasswordHistory"].Rows[0];
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
    passwordHistory.User_PasswordHistory_ID.Value = (Guid)objDataRow["User_PasswordHistory_ID"];
    passwordHistory.User_ID.Value = (Guid)objDataRow["User_ID"];
    passwordHistory.Password.Value = (string)objDataRow["Password"];
    passwordHistory.Log.SetValues(objDataRow);
```

```
}

#endregion //Retrieve Methods

/// <summary>
/// Updates the current historic password in the database using the original
/// historic password to resolve possible concurrency issues.
/// </summary>
/// <param name="originalPasswordHistory">The original <see cref="StudyPlanning.DAL.Users.PasswordHistory"/> object.</param>
/// <returns><strong>true</strong> if the update was succesfully executed; otherwise, <strong>false</strong>.</returns>
public bool Update(StudyPlanning.DAL.Users.PasswordHistory originalPasswordHistory)
{
  try
  {
    this.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }

  bool blnResult = false;

  SqlCommand objCommand = new SqlCommand("User_PasswordHistory_Update", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, false);
  AddParameters(originalPasswordHistory, objCommand, true);

  int rowsA↑ected;

  try
  {
    objConnection.Open();
    rowsA↑ected = objCommand.ExecuteNonQuery();
    objConnection.Close();

    if (rowsA↑ected > 0)
      blnResult = true;
  }
  catch(SqlException excp)
  {
    throw new DalException(excp);
  }

  return blnResult;
}

/// <summary>
/// Deletes the current historic password from the database.
/// </summary>
/// <returns><strong>true</strong> if the deletion is succesfully executed; otherwise, <strong>false</strong>.</returns>
public bool Delete()
{
  bool blnResult = false;

  try
  {
    this.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }

  SqlCommand objCommand = new SqlCommand("User_PasswordHistory_Delete", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  AddParameters(this, objCommand, true);

  try
  {
    objConnection.Open();
    int rowsA↑ected = objCommand.ExecuteNonQuery();
    objConnection.Close();

    if (rowsA↑ected > 0)
      blnResult = true;
  }
  catch(SqlException objEx)
  {
    throw new DalException(objEx);
  }

  return blnResult;
}

/// <summary>
/// Adds the relevant parameters to the specified <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see> object
/// setting the values to ones of the specified <see cref="StudyPlanning.DAL.Users.PasswordHistory"/> object.
/// </summary>
/// <param name="passwordHistory">
/// The <see cref="StudyPlanning.DAL.Users.PasswordHistory"/> object containing the values that the parameters have to be
/// initialized to.
/// </param>
/// <param name="objCommand">
/// The <see cref="System.Data.SqlClient.SqlCommand"/> object to which parameters should be added.
/// </param>
/// <param name="isOriginal">
```

```
/// <strong>true</strong> if the parameters to be added represent original
/// data (for handling concurrent programming issues); otherwise, <strong>false</strong>.</param>
/// <returns>The <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see>
/// object with the relevant parameters added.</returns>
private static void AddParameters(PasswordHistory passwordHistory, SqlCommand objCommand, bool isOriginal)
{
  string paramName = "";
  SqlParameter objParam;

  //User_PasswordHistory_ID
  paramName = "User_PasswordHistory_ID";
  objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

  if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

  objParam.Value = passwordHistory.User_PasswordHistory_ID.Value;

  //User_ID
  paramName = "User_ID";
  objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

  if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

  objParam.Value = passwordHistory.User_ID.Value;

  //Password
  paramName = "Password";
  objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.VarChar, 50);

  if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

  objParam.Value = passwordHistory.Password.Value;

  //Log
  objCommand = passwordHistory.Log.AddParameters(objCommand, isOriginal);
}

/// <summary>
/// Validates the current <see cref="StudyPlanning.DAL.Users.PasswordHistory"/> object. If the
/// value of either of the private properties is null a
/// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
/// </summary>
private void Validate()
{
  try
  {
    _User_PasswordHistory_ID.Validate("User_PasswordHistory_ID");
    _User_ID.Validate("User_ID");
    _Password.Validate("Password");
    _Log.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }
}

/// <summary>
/// Retrieves a list of previous passwords of the specified user.
/// </summary>
/// <param name="userID">The ID of the user for which to retrieve the list of previous passwords.</param>
/// <returns>An array containing IDs of the users previous passwords.</returns>
public static Guid[] GetPreviousPasswords(Guid userID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  SqlCommand objCommand = new SqlCommand("User_PasswordHistory_GetPreviousPasswords", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  SqlParameter objParam = objCommand.Parameters.Add("@User_ID", SqlDbType.UniqueIdenti↓er, 16);
  objParam.Value = userID;

  DataSet objDataSet = new DataSet("Previous_Passwords");
  SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

  try
  {
    objConnection.Open();
    objAdap.Fill(objDataSet, "Previous_Passwords");
    objConnection.Close();
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
  int intRows = objDataSet.Tables["Previous_Passwords"].Rows.Count;
  Guid[] passwords = new Guid[intRows];

  for(int i=0; i < intRows; i++)
  {
    passwords[i] = (Guid)objDataSet.Tables["Previous_Passwords"].Rows[i]["User_PasswordHistory_ID"];
  }
  return passwords;
}
```

```
    #endregion //Public Methods

  }
}
```

1.30.3 SecurityRole

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.Users
{
  /// <summary>
  /// Represents a security role of a user.
  /// </summary>
  public class SecurityRole : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalGuid _User_SecurityRole_ID = new DalGuid(false);
    private DalGuid _User_ID = new DalGuid(false);
    private DalInt _SecurityRole_ID = new DalInt(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.Users.SecurityRole"/>
    /// class.
    /// </summary>
    public SecurityRole() {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the User_SecurityRole_ID.
    /// </summary>
    /// <value></value>
    public DalGuid User_SecurityRole_ID
    {
      get { return _User_SecurityRole_ID; }
    }

    /// <summary>
    /// Gets the User_ID.
    /// </summary>
    /// <value></value>
    public DalGuid User_ID
    {
      get { return _User_ID; }
    }

    /// <summary>
    /// Gets the SecurityRole_ID.
    /// </summary>
    public DalInt SecurityRole_ID
    {
      get { return _SecurityRole_ID; }
    }

    /// <summary>
    /// Gets the data log of the user.
    /// </summary>
    public DataLog Log
    {
      get { return _Log; }
    }

    #endregion //Public Properties

    #region Public Methods

    /// <summary>
    /// Creates a new user security role in the database using the values of the properties
    /// of the current instance.
    /// </summary>
    public void Create()
    {
      try
      {
        this.Validate();
      }
      catch (DalException excp)
      {
        throw excp;
      }
```

```
    SqlCommand objCommand = new SqlCommand("User_SecurityRole_Insert", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, false);

    try
    {
      objConnection.Open();
      objCommand.ExecuteNonQuery();
      objConnection.Close();
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
  }

  #region Retrieve Methods

  /// <summary>
  /// Retrieves the user security role from the database using the value of the
  /// User_SecurityRole_ID property of the current instance
  /// </summary>
  public void Retrieve()
  {
    try
    {
      this._User_SecurityRole_ID.Validate("User_SecurityRole_ID");
    }
    catch (DalException e)
    {
      throw e;
    }

    try
    {
      RetrieveExecute(_User_SecurityRole_ID.Value, this, objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }
  }

  /// <summary>
  /// Retrieves the user security role with the specified identifier from the database.
  /// </summary>
  /// <param name="user_SecurityRole_ID">The ID of the user security role.</param>
  public static StudyPlanning.DAL.Users.SecurityRole Retrieve(Guid user_SecurityRole_ID)
  {
    string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
    SqlConnection objConnection = new SqlConnection(connString);

    StudyPlanning.DAL.Users.SecurityRole objSecurityRole =
      new StudyPlanning.DAL.Users.SecurityRole();

    try
    {
      RetrieveExecute(user_SecurityRole_ID, objSecurityRole, objConnection);
    }
    catch (DalException objExc)
    {
      throw objExc;
    }
    return objSecurityRole;
  }

  private static void RetrieveExecute(
    Guid user_SecurityRole_ID,
    StudyPlanning.DAL.Users.SecurityRole userSecurityRole,
    SqlConnection sqlConnection)
  {
    DataRow objDataRow;

    SqlCommand objCommand = new SqlCommand("User_SecurityRole_Select", sqlConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    SqlParameter objParam;

    objParam = objCommand.Parameters.Add("@User_SecurityRole_ID", SqlDbType.UniqueIdenti↓er);
    objParam.Value = user_SecurityRole_ID;

    DataSet objDataSet = new DataSet("User_SecurityRole");
    SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

    try
    {
      sqlConnection.Open();
      objAdap.Fill(objDataSet, "User_SecurityRole");
      sqlConnection.Close();
      objDataRow = objDataSet.Tables["User_SecurityRole"].Rows[0];
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
    userSecurityRole._User_SecurityRole_ID.Value   = (Guid)objDataRow["User_SecurityRole_ID"];
    userSecurityRole._User_ID.Value  = (Guid)objDataRow["User_ID"];
    userSecurityRole._SecurityRole_ID.Value  = (int)objDataRow["SecurityRole_ID"];
```

```
    userSecurityRole._Log.SetValues(objDataRow);
  }

  #endregion //Retrieve Methods

  /// <summary>
  /// Updates the current user security role in the database using the original
  /// user security role data to resolve possible concurrency issues.
  /// </summary>
  /// <param name="originalSecurityRole">The original <see cref="StudyPlanning.DAL.Users.SecurityRole"/> object.</param>
  /// <returns><strong>true</strong> if the update was succesfully executed; otherwise, <strong>false</strong>.</returns>
  public bool Update(StudyPlanning.DAL.Users.SecurityRole originalSecurityRole)
  {
    try
    {
      this.Validate();
    }
    catch (DalException excp)
    {
      throw excp;
    }

    bool blnResult = false;

    SqlCommand objCommand = new SqlCommand("User_SecurityRole_Update", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, false);
    AddParameters(originalSecurityRole, objCommand, true);

    int rowsA↑ected;

    try
    {
      objConnection.Open();
      rowsA↑ected = objCommand.ExecuteNonQuery();
      objConnection.Close();

      if (rowsA↑ected > 0)
        blnResult = true;
    }
    catch(SqlException excp)
    {
      throw new DalException(excp);
    }

    return blnResult;
  }

  /// <summary>
  /// Deletes the current security role from the database.
  /// </summary>
  /// <returns><strong>true</strong> if the deletion is succesfully executed; otherwise, <strong>false</strong>.</returns>
  public bool Delete()
  {
    bool blnResult = false;

    try
    {
      this.Validate();
    }
    catch (DalException excp)
    {
      throw excp;
    }

    SqlCommand objCommand = new SqlCommand("User_SecurityRole_Delete", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, true);

    try
    {
      objConnection.Open();
      int rowsA↑ected = objCommand.ExecuteNonQuery();
      objConnection.Close();

      if (rowsA↑ected > 0)
        blnResult = true;
    }
    catch(SqlException objEx)
    {
      throw new DalException(objEx);
    }

    return blnResult;
  }

  /// <summary>
  /// Adds the relevant parameters to the specified <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see> object
  /// setting the values to ones of the specified <see cref="StudyPlanning.DAL.Users.SecurityRole"/> object.
  /// </summary>
  /// <param name="securityRole">
  /// The <see cref="StudyPlanning.DAL.Users.SecurityRole"/> object containing the values that the parameters have to be
  /// initialized to.
  /// </param>
  /// <param name="objCommand">
  /// The <see cref="System.Data.SqlClient.SqlCommand"/> object to which parameters should be added.
  /// </param>
```

```
/// <param name="isOriginal">
/// <strong>true</strong> if the parameters to be added represent original
/// data (for handling concurrent programming issues); otherwise, <strong>false</strong>.</param>
/// <returns>The <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see>
/// object with the relevant parameters added.</returns>
private static void AddParameters(StudyPlanning.DAL.Users.SecurityRole securityRole, SqlCommand objCommand, bool
  isOriginal)
{
  string paramName = "";
  SqlParameter objParam;

  //User_SecurityRole_ID
  paramName = "User_SecurityRole_ID";
  objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

  if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

  objParam.Value = securityRole.User_SecurityRole_ID.Value;

  //User_ID
  paramName = "User_ID";
  objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

  if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

  objParam.Value = securityRole.User_ID.Value;

  //SecurityRole_ID
  paramName = "SecurityRole_ID";
  objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.Int, 4);

  if (isOriginal)
    objParam.ParameterName = "@Original_" + paramName;

  objParam.Value = securityRole.SecurityRole_ID.Value;

  //Log
  objCommand = securityRole.Log.AddParameters(objCommand, isOriginal);
}

/// <summary>
/// Validates the current <see cref="StudyPlanning.DAL.Users.SecurityRole"/> object. If the
/// value of either of the private properties is null a
/// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
/// </summary>
private void Validate()
{
  try
  {
    _User_SecurityRole_ID.Validate("User_SecurityRole_ID");
    _User_ID.Validate("User_ID");
    _SecurityRole_ID.Validate("SecurityRole_ID");
    _Log.Validate();
  }
  catch (DalException excp)
  {
    throw excp;
  }
}

/// <summary>
/// Retrieves the security roles of the specified user.
/// </summary>
/// <param name="user_ID">The ID of the user for which to retrieve security roles.</param>
/// <returns>An array containing IDs of the security roles of the specified user.</returns>
public static int[] GetRolesFromUserID(Guid user_ID)
{
  int[] roles;

  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  SqlCommand objCommand = new SqlCommand("User_SecurityRole_GetRolesFromUserID", objConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  SqlParameter objParam;

  objParam = objCommand.Parameters.Add("@User_ID", SqlDbType.UniqueIdenti↓er, 16);
  objParam.Value = user_ID;

  DataSet objDataSet = new DataSet("Roles");
  SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

  try
  {
    objConnection.Open();
    objAdap.Fill(objDataSet, "Roles");
    objConnection.Close();

    int numRows = objDataSet.Tables["Roles"].Rows.Count;
    roles = new int[numRows];

    for(int i=0; i<numRows; i++)
    {
      roles[i] = (int)objDataSet.Tables["Roles"].Rows[i]["SecurityRole_ID"];
    }
  }
```

```
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }

      return roles;
    }

    #endregion //Public Methods

  }
}
```

1.30.4 User

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.Users
{
  /// <summary>
  /// Represents a user in the data access layer.
  /// </summary>
  public class User : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalGuid _User_ID = new DalGuid(false);
    private DalString _Username = new DalString(false);
    private DalString _Password = new DalString(false);
    private DalBool _ChangePasswordAtNextLogon = new DalBool(false);
    private DalBool _UserCannotChangePassword = new DalBool(false);
    private DalBool _PasswordNeverExpires = new DalBool(false);
    private DalDateTime _PasswordLastChanged = new DalDateTime(false);
    private DalBool _Locked = new DalBool(false);
    private DalDateTime _LockedAt = new DalDateTime(true);
    private DalBool _Disabled = new DalBool(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.DAL.Users.User"/>
    /// class.
    /// </summary>
    public User() {
      _Log = new DataLog();
    }

    #endregion

    #region Public Properties

    /// <summary>
    /// Gets the User_ID.
    /// </summary>
    /// <value></value>
    public DalGuid User_ID
    {
      get { return _User_ID; }
    }

    /// <summary>
    /// Gets the username.
    /// </summary>
    public DalString Username
    {
      get { return _Username; }
    }

    /// <summary>
    /// Gets the password.
    /// </summary>
    public DalString Password
    {
      get { return _Password; }
    }

    /// <summary>
    /// Gets an indication of whether the password should be changed at next logon.
    /// </summary>
    public DalBool ChangePasswordAtNextLogon
    {
      get { return _ChangePasswordAtNextLogon; }
    }

    /// <summary>
    /// Gets an indication of whether the user is allowed to change his
    /// password.
    /// </summary>
    public DalBool UserCannotChangePassword
    {
```

```
    get { return _UserCannotChangePassword; }
  }

  /// <summary>
  /// Gets an indication of whether the password does ever expires.
  /// </summary>
  public DalBool PasswordNeverExpires
  {
    get { return _PasswordNeverExpires; }
  }

  /// <summary>
  /// Gets the date and time at when the password was last changed.
  /// </summary>
  public DalDateTime PasswordLastChanged
  {
    get { return _PasswordLastChanged; }
  }

  /// <summary>
  /// Gets an indication of whether the user (account) is locked or not.
  /// </summary>
  public DalBool Locked
  {
    get {return _Locked; }
  }

  /// <summary>
  /// Gets the date and time at when the user (account) was locked.
  /// </summary>
  public DalDateTime LockedAt
  {
    get { return _LockedAt; }
  }

  /// <summary>
  /// Gets an indication of whether the user (account) has been disabled.
  /// </summary>
  public DalBool Disabled
  {
    get { return _Disabled; }
  }

  /// <summary>
  /// Gets the data log of the user.
  /// </summary>
  public DataLog Log
  {
    get { return _Log; }
  }

  #endregion //Public Properties

  #region Public Methods

  /// <summary>
  /// Creates a new user in the database using the values of the properties
  /// of the current instance.
  /// </summary>
  public void Create()
  {
    try
    {
      this.Validate();
    }
    catch (DalException excp)
    {
      throw excp;
    }

    SqlCommand objCommand = new SqlCommand("User_Insert", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, false);

    try
    {
      objConnection.Open();
      objCommand.ExecuteNonQuery();
      objConnection.Close();
    }
    catch (SqlException objExc)
    {
      throw new DalException(objExc);
    }
  }

  #region Retrieve Methods

  /// <summary>
  /// Retrieves the User from the database using the value of the
  /// User_ID property of the current instance
  /// </summary>
  public void Retrieve()
  {
    try
    {
      this._User_ID.Validate("User_ID");
    }
```

```
  catch (DalException e)
  {
    throw e;
  }

  try
  {
    RetrieveExecute(_User_ID.Value, this, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
}

/// <summary>
/// Retrieves the User with the specified identifier from the database.
/// </summary>
/// <param name="user_ID">The ID of the user to retrieve.</param>
public static StudyPlanning.DAL.Users.User Retrieve(Guid user_ID)
{
  string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  StudyPlanning.DAL.Users.User objUser =
    new StudyPlanning.DAL.Users.User();

  try
  {
    RetrieveExecute(user_ID, objUser, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
  return objUser;
}

private static void RetrieveExecute(
  Guid user_ID,
  StudyPlanning.DAL.Users.User user,
  SqlConnection sqlConnection)
{
  DataRow objDataRow;

  SqlCommand objCommand = new SqlCommand("User_Select", sqlConnection);
  objCommand.CommandType = CommandType.StoredProcedure;

  SqlParameter objParam;

  objParam = objCommand.Parameters.Add("@User_ID", SqlDbType.UniqueIdenti↓er);
  objParam.Value = user_ID;

  DataSet objDataSet = new DataSet("User");
  SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

  try
  {
    sqlConnection.Open();
    objAdap.Fill(objDataSet, "User");
    sqlConnection.Close();
    objDataRow = objDataSet.Tables["User"].Rows[0];
  }
  catch (SqlException objExc)
  {
    throw new DalException(objExc);
  }
  user.User_ID.Value   = (Guid)objDataRow["User_ID"];
  user.Username.Value  = (string)objDataRow["Username"];
  user.Password.Value  = (string)objDataRow["Password"];
  user.ChangePasswordAtNextLogon.Value = (bool)objDataRow["ChangePasswordAtNextLogon"];
  user.UserCannotChangePassword.Value = (bool)objDataRow["UserCannotChangePassword"];
  user.PasswordNeverExpires.Value = (bool)objDataRow["PasswordNeverExpires"];
  user.PasswordLastChanged.Value = (DateTime)objDataRow["PasswordLastChanged"];
  user.Locked.Value = (bool)objDataRow["Locked"];

  if (objDataRow["LockedAt"].Equals(System.DBNull.Value))
    user.LockedAt.IsNull = true;
  else
    user.LockedAt.Value = (DateTime)objDataRow["LockedAt"];

  user.Disabled.Value = (bool)objDataRow["Disabled"];
  user.Log.SetValues(objDataRow);
}

#endregion //Retrieve Methods

/// <summary>
/// Updates the current user in the database using the original
/// user data to resolve possible concurrency issues.
/// </summary>
/// <param name="originalUser">The original <see cref="StudyPlanning.DAL.Users.User"/> object.</param>
/// <returns><strong>true</strong> if the update was succesfully executed; otherwise, <strong>false</strong>.</returns>
public bool Update(StudyPlanning.DAL.Users.User originalUser)
{
  try
  {
    this.Validate();
  }
```

```
    catch (DalException excp)
    {
      throw excp;
    }

    bool blnResult = false;

    SqlCommand objCommand = new SqlCommand("User_Update", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, false);
    AddParameters(originalUser, objCommand, true);

    int rowsA↑ected;

    try
    {
      objConnection.Open();
      rowsA↑ected = objCommand.ExecuteNonQuery();
      objConnection.Close();

      if (rowsA↑ected > 0)
        blnResult = true;
    }
    catch(SqlException excp)
    {
      throw new DalException(excp);
    }

    return blnResult;
  }

  /// <summary>
  /// Deletes the current user from the database.
  /// </summary>
  /// <returns><strong>true</strong> if the deletion is succesfully executed; otherwise, <strong>false</strong>.</returns>
  public bool Delete()
  {
    bool blnResult = false;

    try
    {
      this.Validate();
    }
    catch (DalException excp)
    {
      throw excp;
    }

    SqlCommand objCommand = new SqlCommand("User_Delete", objConnection);
    objCommand.CommandType = CommandType.StoredProcedure;

    AddParameters(this, objCommand, true);

    try
    {
      objConnection.Open();
      int rowsA↑ected = objCommand.ExecuteNonQuery();
      objConnection.Close();

      if (rowsA↑ected > 0)
        blnResult = true;
    }
    catch(SqlException objEx)
    {
      throw new DalException(objEx);
    }

    return blnResult;
  }

  /// <summary>
  /// Adds the relevant parameters to the specified <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see> object
  /// setting the values to ones of the specified <see cref="StudyPlanning.DAL.Users.User"/> object.
  /// </summary>
  /// <param name="user">
  /// The <see cref="StudyPlanning.DAL.Users.User"/> object containing the values that the parameters have to be
  /// initialized to.
  /// </param>
  /// <param name="objCommand">
  /// The <see cref="System.Data.SqlClient.SqlCommand"/> object to which parameters should be added.
  /// </param>
  /// <param name="isOriginal">
  /// <strong>true</strong> if the parameters to be added represent original
  /// data (for handling concurrent programming issues); otherwise, <strong>false</strong>.</param>
  /// <returns>The <see cref="System.Data.SqlClient.SqlCommand">SqlCommand</see>
  /// object with the relevant parameters added.</returns>
  private static void AddParameters(User user, SqlCommand objCommand, bool isOriginal)
  {
    string paramName = "";
    SqlParameter objParam;

    //User_ID
    paramName = "User_ID";
    objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.UniqueIdenti↓er, 16);

    if (isOriginal)
      objParam.ParameterName = "@Original_" + paramName;
```

```
    objParam.Value = user.User_ID.Value;

    //Username
    paramName = "Username";
    objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.VarChar, 20);

    if (isOriginal)
      objParam.ParameterName = "@Original_" + paramName;

    objParam.Value = user.Username.Value;

    //Password
    paramName = "Password";
    objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.VarChar, 50);

    if (isOriginal)
      objParam.ParameterName = "@Original_" + paramName;

    objParam.Value = user.Password.Value;

    //ChangePasswordAtNextLogon
    paramName = "ChangePasswordAtNextLogon";
    objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.Bit, 1);

    if (isOriginal)
      objParam.ParameterName = "@Original_" + paramName;

    objParam.Value = user.ChangePasswordAtNextLogon.Value;

    //UserCannotChangePassword
    paramName = "UserCannotChangePassword";
    objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.Bit, 1);

    if (isOriginal)
      objParam.ParameterName = "@Original_" + paramName;

    objParam.Value = user.UserCannotChangePassword.Value;

    //PasswordNeverExpires
    paramName = "PasswordNeverExpires";
    objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.Bit, 1);

    if (isOriginal)
      objParam.ParameterName = "@Original_" + paramName;

    objParam.Value = user.PasswordNeverExpires.Value;

    //PasswordLastChanged
    paramName = "PasswordLastChanged";
    objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.DateTime, 8);

    if (isOriginal)
      objParam.ParameterName = "@Original_" + paramName;

    objParam.Value = user.PasswordLastChanged.Value;

    //Locked
    paramName = "Locked";
    objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.Bit, 1);

    if (isOriginal)
      objParam.ParameterName = "@Original_" + paramName;

    objParam.Value = user.Locked.Value;

    //LockedAt
    paramName = "LockedAt";
    objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.DateTime, 8);

    if (isOriginal)
      objParam.ParameterName = "@Original_" + paramName;

    if (user.LockedAt.IsNull)
      objParam.Value = System.DBNull.Value;
    else
      objParam.Value = user.LockedAt.Value;

    //Disabled
    paramName = "Disabled";
    objParam = objCommand.Parameters.Add("@" + paramName, SqlDbType.Bit, 1);

    if (isOriginal)
      objParam.ParameterName = "@Original_" + paramName;

    objParam.Value = user.Disabled.Value;

    //Log
    objCommand = user.Log.AddParameters(objCommand, isOriginal);
  }

  /// <summary>
  /// Validates the current <see cref="StudyPlanning.DAL.Users.User"/> object. If the
  /// value of either of the private properties is null a
  /// <see cref="StudyPlanning.DAL.DalException"/> is thrown.
  /// </summary>
  private void Validate()
  {
    try
    {
```

```
        _User_ID.Validate("User_ID");
        _Username.Validate("Username");
        _Password.Validate("Password");
        _ChangePasswordAtNextLogon.Validate("ChangePasswordAtNextLogon");
        _UserCannotChangePassword.Validate("UserCannotChangePassword");
        _PasswordNeverExpires.Validate("PasswordNeverExpires");
        _PasswordLastChanged.Validate("PasswordLastChanged");
        _Locked.Validate("Locked");
        _LockedAt.Validate("LockedAt");
        _Disabled.Validate("Disabled");
        _Log.Validate();
      }
      catch (DalException excp)
      {
        throw excp;
      }
    }

    /// <summary>
    /// Gets the ID of the user having the specified username.
    /// </summary>
    /// <param name="username">The username of the user.</param>
    /// <returns>The GUID of the user.</returns>
    public static Guid GetIDFromUsername(string username)
    {
      Guid userID;

      string connString = System.Con↓guration.Con↓gurationSettings.AppSettings["sqlConnectionString"];
      SqlConnection objConnection = new SqlConnection(connString);

      SqlCommand objCommand = new SqlCommand("User_GetIDFromUsername", objConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      SqlParameter objParam;

      objParam = objCommand.Parameters.Add("@Username", SqlDbType.VarChar, 20);
      objParam.Value = username;

      objParam = objCommand.Parameters.Add("@User_ID", SqlDbType.UniqueIdenti↓er, 16);
      objParam.Direction = ParameterDirection.Output;

      try
      {
        objConnection.Open();
        objCommand.ExecuteNonQuery();
        userID = (Guid)objCommand.Parameters["@User_ID"].Value;
        objConnection.Close();
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }

      return userID;
    }

    /// <summary>
    /// Clones the current <see cref="StudyPlanning.DAL.Users.User"/> object and
    /// returns an object of same type.
    /// </summary>
    /// <returns>A <see cref="StudyPlanning.DAL.Users.User"/> object.</returns>
    public User Clone()
    {
      return (User)this.MemberwiseClone();
    }

    #endregion //Public Methods

  }
}
```

1.31 Weekday

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL
{
  /// <summary>
  /// Represents the Weekday table.
  /// </summary>
  public class Weekday : StudyPlanning.DAL.DbObject
  {
    #region Private Properties

    private DalInt _Weekday_ID = new DalInt(false);
    private DalStringLocalizable _Name = new DalStringLocalizable(false);
    private DataLog _Log;

    #endregion //Private Properties

    #region Constructors
```

```
/// <summary>
/// Creates a new instance of the <see cref="StudyPlanning.DAL.Weekday">Weekday</see>
/// class.
/// </summary>
public Weekday() {
  _Log = new DataLog();
}

#endregion

#region Public Properties

/// <summary>
/// Gets the Weekday_ID.
/// </summary>
/// <value></value>
public DalInt Weekday_ID
{
  get { return _Weekday_ID; }
}

/// <summary>
/// Gets the Name.
/// </summary>
public DalStringLocalizable Name
{
  get { return _Name; }
}

/// <summary>
/// Gets the data log of the table.
/// </summary>
public DataLog Log
{
  get { return _Log; }
}

#endregion //Public Properties

#region Public Methods

#region Retrieve Methods

/// <summary>
/// Retrieves the weekday from the database using the value of the
/// Weekday_ID property of the current instance
/// </summary>
public void Retrieve()
{
  DataRow objDataRow;

  try
  {
    _Weekday_ID.Validate("Weekday_ID");
  }
  catch (DalException excp)
  {
    throw excp;
  }

  try
  {
    objDataRow = RetrieveExecute(_Weekday_ID.Value, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
  this._Weekday_ID.Value = Convert.ToInt32(objDataRow["Weekday_ID"]);
  this._Name.LoadXml((string)objDataRow["Name"]);
  this._Log.SetValues(objDataRow);
}

/// <summary>
/// Retrieves the weekday with the specified identifier from the database.
/// </summary>
/// <param name="weekday_ID">The unique identification of the weekday.</param>
public static Weekday Retrieve(int weekday_ID)
{
  DataRow objDataRow;
  string connString = System.Con|guration.Con|gurationSettings.AppSettings["sqlConnectionString"];
  SqlConnection objConnection = new SqlConnection(connString);

  Weekday objWeekday = new Weekday();

  try
  {
    objDataRow = RetrieveExecute(weekday_ID, objConnection);
  }
  catch (DalException objExc)
  {
    throw objExc;
  }
  objWeekday._Weekday_ID.Value = Convert.ToInt32(objDataRow["Weekday_ID"]);
  objWeekday._Name.LoadXml((string)objDataRow["Name"]);
  objWeekday._Log.SetValues(objDataRow);
  return objWeekday;
}
```

```
    private static DataRow RetrieveExecute(
      int weekday_ID,
      SqlConnection sqlConnection)
    {
      SqlCommand objCommand = new SqlCommand("Weekday_Select", sqlConnection);
      objCommand.CommandType = CommandType.StoredProcedure;

      SqlParameter objParam;

      objParam = objCommand.Parameters.Add("@Weekday_ID", SqlDbType.Int);
      objParam.Value = weekday_ID;

      DataSet objDataSet = new DataSet("Weekday");
      SqlDataAdapter objAdap = new SqlDataAdapter(objCommand);

      try
      {
        sqlConnection.Open();
        objAdap.Fill(objDataSet, "Weekday");
        sqlConnection.Close();
      }
      catch (SqlException objExc)
      {
        throw new DalException(objExc);
      }
      DataRow objDataRow = objDataSet.Tables["Weekday"].Rows[0];
      return objDataRow;
    }

    #endregion //Retrieve Methods

    #endregion //Public Methods

  }
}
```

Chapter 2

Data Tier Test

2.1 Courses

2.1.1 Course

course.aspx

```
<%@ Page language="c#" Src="Course.aspx.cs" CodeBehind="Course.aspx.cs" AutoEventWireup="false" Inherits="StudyPlanning.DAL.
  Courses.Test.CourseTest" trace="false" warningLevel="4" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" >
<HTML>
  <HEAD>
    <title>Course</title>
    <meta name="vs_showGrid" content="True">
    <meta name="GENERATOR" Content="Microsoft Visual Studio 7.0">
    <meta name="CODE_LANGUAGE" Content="C#">
    <meta name="vs_defaultClientScript" content="VBScript">
    <meta name="vs_targetSchema" content="http://schemas.microsoft.com/intellisense/ie5">
    <link rel="stylesheet" href="../../misc/style.css">
    <style>
    .textbox { WIDTH: 400px }
    .button { WIDTH: 120px }
    </style>
  </HEAD>
  <body>
    <P align="center"><STRONG><FONT size="4">Course</FONT></STRONG></P>
    <form id="form" method="post" runat="server">
      <TABLE id="Table2" cellSpacing="1" cellPadding="1" align="center" border="1">
        <TR>
          <TD align="middle">
            <P>
              <asp:Button id="btnCreate" runat="server" Text="Create" OnClick="btnCreate_Click" CssClass="button"></asp:Button><
                /P>
            <P>
              <asp:Button id="btnRetrieve" onclick="btnRetrieve_Click" runat="server" Text="Retrieve" CssClass="button"></asp:
                Button></P>
            <P>
              <asp:Button id="btnUpdate" onclick="btnUpdate_Click" runat="server" Text="Update" CssClass="button"></asp:Button></
                P>
            <P>
              <asp:Button id="btnDelete" onclick="btnDelete_Click" runat="server" Text="Delete" CssClass="button"></asp:Button></
                P>
            <DIV align="left">
              <asp:Button id="btnReset" onclick="btnReset_Click" runat="server" CssClass="button" Text="Reset"></asp:Button></
                DIV>
          </TD>
          <TD align="middle">
            <TABLE id="Table1" cellSpacing="1" cellPadding="1" width="500" border="1">
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Course_id</STRONG></TD>
                <TD><STRONG>
                    <asp:TextBox id="tbxCourse_ID" runat="server" CssClass="textbox"></asp:TextBox></STRONG></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Created</STRONG></TD>
                <TD>
                  <asp:TextBox id="tbxCreated" runat="server" CssClass="textbox"></asp:TextBox></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>CreatedBy</STRONG></TD>
```

```
            <TD>
              <asp:TextBox id="tbxCreatedBy" runat="server" CssClass="textbox"></asp:TextBox></TD>
          </TR>
          <TR>
            <TD style="WIDTH: 138px"><STRONG>Updated</STRONG></TD>
            <TD width="50%">
              <asp:TextBox id="tbxUpdated" runat="server" CssClass="textbox"></asp:TextBox></TD>
          </TR>
          <TR>
            <TD style="WIDTH: 138px"><STRONG>UpdatedBy</STRONG></TD>
            <TD>
              <asp:TextBox id="tbxUpdatedBy" runat="server" CssClass="textbox"></asp:TextBox></TD>
          </TR>
        </TABLE>
      </TD>
    </TR>
  </TABLE>
</form>
<P>
  <asp:Label id="lblError" Runat="server" ForeColor="MediumBlue"></asp:Label></P>
</FORM>
</body>
</HTML>
```

course.aspx.cs

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data.SqlClient;
using System.Data;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using System.Text;
using System.Web.Security;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.Courses.Test
{
  /// <summary>
  /// Summary description for course test page.
  /// </summary>
  public class CourseTest : System.Web.UI.Page
  {
    protected System.Web.UI.WebControls.Button btnCreate;
    protected System.Web.UI.WebControls.Button btnRetrieve;
    protected System.Web.UI.WebControls.Button btnUpdate;
    protected System.Web.UI.WebControls.Button btnDelete;
    protected System.Web.UI.WebControls.Button btnReset;
    protected System.Web.UI.WebControls.TextBox tbxCreated;
    protected System.Web.UI.WebControls.TextBox tbxCreatedBy;
    protected System.Web.UI.WebControls.TextBox tbxUpdated;
    protected System.Web.UI.WebControls.TextBox tbxUpdatedBy;
    protected System.Web.UI.WebControls.TextBox tbxCourse_ID;
    protected Label lblError;

    private void Page_Load(object sender, System.EventArgs e)
    {

    }

    private void InitializeComponent()
    {

    }

    private string writeError(DalException x)
    {
      string strError = "";

      strError += "Message: " + x.Message + "<br/>";
      strError += "Number: " + x.Number + "<br/>";
      strError += x.ToString();

      return strError;
    }

    protected void btnCreate_Click(object sender, System.EventArgs e)
    {
      lblError.Text = "";

      Course course = new Course();

      course.Course_ID.Value = Guid.NewGuid();

      if (tbxCreated.Text == "")
        course.Log.Created.Value = DateTime.Now;
      else
        course.Log.Created.Value = Convert.ToDateTime(tbxCreated.Text);

      if (tbxCreatedBy.Text.Equals(""))
        course.Log.CreatedBy.Value = Guid.NewGuid();
      else
```

```
    course.Log.CreatedBy.Value = new Guid(tbxCreatedBy.Text);

  if (tbxUpdated.Text == "")
    course.Log.Updated.Value = DateTime.Now;
  else
    course.Log.Updated.Value = Convert.ToDateTime(tbxUpdated.Text);

  if (tbxUpdatedBy.Text.Equals(""))
    course.Log.UpdatedBy.Value = Guid.NewGuid();
  else
    course.Log.UpdatedBy.Value = new Guid(tbxUpdatedBy.Text);

  try
  {
    course.Create();
    tbxCourse_ID.Text = course.Course_ID.Value.ToString();
  }
  catch (DalException x)
  {
    lblError.Text = writeError(x);
    return;
  }
}

protected void btnRetrieve_Click(object sender, System.EventArgs e)
{
  DataLog dl = new DataLog();
  lblError.Text = "";

  Course course = new Course();
  course.Course_ID.Value = new Guid(tbxCourse_ID.Text);

  try
  {
    course.Retrieve();
  }
  catch (DalException x)
  {
    lblError.Text = writeError(x);
    return;
  }
  tbxCreated.Text  = course.Log.Created.ToString();
  tbxCreatedBy.Text = course.Log.CreatedBy.ToString();
  tbxUpdated.Text  = course.Log.Updated.ToString();
  tbxUpdatedBy.Text = course.Log.UpdatedBy.ToString();
}

protected void btnUpdate_Click(object sender, System.EventArgs e)
{
  lblError.Text = "";
  Course orgCourse = new Course();
  Course newCourse = new Course();

  orgCourse.Course_ID.Value = new Guid(tbxCourse_ID.Text);

  newCourse.Course_ID.Value = orgCourse.Course_ID.Value;
  newCourse.Log.Created.Value = Convert.ToDateTime(tbxCreated.Text);
  newCourse.Log.CreatedBy.Value = new Guid(tbxCreatedBy.Text);
  newCourse.Log.Updated.Value = Convert.ToDateTime(tbxUpdated.Text);
  newCourse.Log.UpdatedBy.Value = new Guid(tbxUpdatedBy.Text);

  try
  {
    orgCourse.Retrieve();
    newCourse.Update(orgCourse);
  }
  catch (DalException x)
  {
    lblError.Text = writeError(x);
    return;
  }
}

protected void btnDelete_Click(object sender, System.EventArgs e)
{
  lblError.Text = "";
  DataLog log = new DataLog();
  Course course = new Course();

  course.Course_ID.Value = new Guid(tbxCourse_ID.Text);

  try
  {
    course.Retrieve();
    course.Delete();
  }
  catch (DalException x)
  {
    lblError.Text = writeError(x);
    return;
  }
}

protected void btnReset_Click(object sender, System.EventArgs e)
{
  lblError.Text   = "";
  tbxCourse_ID.Text = "";
  tbxCreated.Text  = "";
  tbxCreatedBy.Text = "";
```

```
      tbxUpdated.Text  = "";
      tbxUpdatedBy.Text = "";
    }

  }
}
```

2.1.2 CourseGrab

coursegrab.aspx

```
<%@ Page language="c#" Src="CourseVersion.aspx.cs" CodeBehind="CourseGrab.aspx.cs" AutoEventWireup="false" Inherits="StudyPlanning
  .DAL.Courses.Test.CourseGrabTest" trace="false" warningLevel="4" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" >
<HTML>
  <HEAD>
    <title>CourseGrab</title>
    <meta content="True" name="vs_showGrid">
    <meta content="Microsoft Visual Studio 7.0" name="GENERATOR">
    <meta content="C#" name="CODE_LANGUAGE">
    <meta content="JavaScript" name="vs_defaultClientScript">
    <meta content="http://schemas.microsoft.com/intellisense/ie5" name="vs_targetSchema">
    <LINK href="../../../misc/style.css" rel="stylesheet">
    <style>.textbox { MARGIN-TOP: 0px; DISPLAY: inline; WIDTH: 400px }
  .button { WIDTH: 120px }
    </style>
  </HEAD>
  <body>
    <br>
    <asp:Label id="lblError" ForeColor="MediumBlue" Runat="server"></asp:Label>
    <p><STRONG><FONT size="4">CourseGrab</FONT></STRONG></p>
    <form id="form" method="post" runat="server">
      <TABLE id="Table2" cellSpacing="1" cellPadding="1" align="center" border="1">
        <TR>
          <TD align="middle" valign="top">
            <br>
            <p><asp:button id="btnCreate" onclick="btnCreate_Click" runat="server" CssClass="button" Text="Create"></asp:button>
              </p>
            <p><asp:button id="btnRetrieve" onclick="btnRetrieve_Click" runat="server" CssClass="button" Text="Retrieve"></asp:
              button></p>
            <p><asp:button id="btnUpdate" onclick="btnUpdate_Click" runat="server" CssClass="button" Text="Update"></asp:button>
              </p>
            <p><asp:button id="btnDelete" onclick="btnDelete_Click" runat="server" CssClass="button" Text="Delete"></asp:button>
              </p>
            <p><asp:button id="btnReset" onclick="btnReset_Click" runat="server" CssClass="button" Text="Reset"></asp:button></
              p>
          </TD>
          <TD align="middle">
            <TABLE id="Table1" cellSpacing="1" cellPadding="1" border="1">
              <TR>
                <TD style="WIDTH: 138px"><STRONG>CourseGrab_ID</STRONG></TD>
                <TD>
                  <asp:textbox id="tbxCourseGrab_ID" runat="server" CssClass="textbox"></asp:textbox></TD>
                <TD></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>CourseVersion_ID</STRONG></TD>
                <TD><STRONG><asp:textbox id="tbxCourseVersion_ID" runat="server" CssClass="textbox"></asp:textbox></
                  STRONG></TD>
                <TD></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Course_ID</STRONG></TD>
                <TD><asp:textbox id="tbxCourse_ID" runat="server" CssClass="textbox" Rows="1"></asp:textbox></TD>
                <TD></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Number</STRONG></TD>
                <TD><STRONG><asp:textbox id="tbxNumber" runat="server" CssClass="textbox" Rows="1"></asp:textbox></
                  STRONG></TD>
                <TD></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>DtuVersion</STRONG></TD>
                <TD><STRONG>
                    <asp:textbox id="tbxDtuVersion" runat="server" CssClass="textbox" Rows="1"></asp:textbox></STRONG></
                    TD>
                <TD>
                  <asp:CheckBox id="chkDtuVersion" runat="server"></asp:CheckBox></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px" rowSpan="2"><STRONG>Schedule</STRONG></TD>
                <TD>
                  <P><STRONG>EN:</STRONG>
                    <asp:textbox id="tbxScheduleEn" runat="server" CssClass="textbox" Rows="2" TextMode="MultiLine"></asp:
                    textbox></P>
                </TD>
                <TD rowSpan="2">
                  <asp:CheckBox id="chkSchedule" runat="server"></asp:CheckBox></TD>
              </TR>
              <tr>
                <TD>
                  <P><STRONG>da-DK:</STRONG>
```

```
      <asp:textbox id="tbxScheduleDa" runat="server" CssClass="textbox" Rows="2" TextMode="MultiLine"></asp:
        textbox></P>
    </TD>
  </tr>
  <TR>
    <TD style="WIDTH: 138px"><STRONG>Duration</STRONG></TD>
    <TD><STRONG>
      <asp:textbox id="tbxDuration" runat="server" CssClass="textbox" Rows="2" TextMode="MultiLine"></asp:textbox
        ></STRONG></TD>
    <TD>
      <asp:CheckBox id="chkDuration" runat="server"></asp:CheckBox></TD>
  </TR>
  <TR>
    <TD style="WIDTH: 138px" rowSpan="2"><STRONG>ExaminationPlacement</STRONG></TD>
    <TD><STRONG>EN:
      <asp:textbox id="tbxExaminationPlacementEn" runat="server" CssClass="textbox" Rows="2" TextMode="MultiLine"><
        /asp:textbox></STRONG></TD>
    <TD rowSpan="2"><strong> 
      <asp:CheckBox id="chkExaminationPlacement" runat="server"></asp:CheckBox></strong></TD>
  </TR>
  <tr>
    <TD><STRONG>da-DK:
      <asp:textbox id="tbxExaminationPlacementDa" runat="server" CssClass="textbox" Rows="2" TextMode="MultiLine"><
        /asp:textbox></STRONG></TD>
  </tr>
  <TR>
    <TD style="WIDTH: 138px" rowSpan="2"><STRONG>ExaminationAids</STRONG></TD>
    <TD><STRONG>EN:
      <asp:textbox id="tbxExaminationAidsEn" runat="server" CssClass="textbox" Rows="2" TextMode="MultiLine"></asp:
        textbox></STRONG></TD>
    <TD rowSpan="2"><strong> 
      <asp:checkbox id="chkExaminationAids" runat="server"></asp:checkbox></strong></TD>
  </TR>
  <tr>
    <TD><STRONG>da-DK:
      <asp:textbox id="tbxExaminationAidsDa" runat="server" CssClass="textbox" Rows="2" TextMode="MultiLine"></asp:
        textbox></STRONG></TD>
  </tr>
  <TR>
    <TD style="WIDTH: 138px"><STRONG>PreviousCourseNumbers</STRONG></TD>
    <TD><STRONG>
      <asp:textbox id="tbxPreviousCourseNumbers" runat="server" CssClass="textbox" Rows="2" TextMode="MultiLine"></
        asp:textbox></STRONG></TD>
    <TD><strong> 
      <asp:checkbox id="chkPreviousCourseNumbers" runat="server"></asp:checkbox></strong></TD>
  </TR>
  <TR>
    <TD style="WIDTH: 138px"><STRONG>MandatoryPrerequisites</STRONG></TD>
    <TD><STRONG>
      <asp:textbox id="tbxMandatoryPrerequisites" runat="server" CssClass="textbox" Rows="2" TextMode="MultiLine"><
        /asp:textbox></STRONG></TD>
    <TD><strong> 
      <asp:checkbox id="chkMandatoryPrerequisites" runat="server"></asp:checkbox></strong></TD>
  </TR>
  <TR>
    <TD style="WIDTH: 138px"><STRONG>TechnicalPrerequisites</STRONG></TD>
    <TD>
      <asp:textbox id="tbxTechnicalPrerequisites" runat="server" CssClass="textbox" Rows="2" TextMode="MultiLine"></
        asp:textbox></TD>
    <TD><STRONG> 
      <asp:checkbox id="chkTechnicalPrerequisites" runat="server"></asp:checkbox></STRONG></TD>
  </TR>
  <TR>
    <TD style="WIDTH: 138px"><STRONG>DesirablePrerequisites</STRONG></TD>
    <TD>
      <asp:textbox id="tbxDesirablePrerequisites" runat="server" CssClass="textbox" Rows="2" TextMode="MultiLine"></
        asp:textbox></TD>
    <TD>
      <asp:checkbox id="chkDesirablePrerequisites" runat="server"></asp:checkbox></TD>
  </TR>
  <TR>
    <TD style="WIDTH: 138px"><STRONG>ExternalInstitutions</STRONG></TD>
    <TD>
      <asp:textbox id="tbxExternalInstitutions" runat="server" CssClass="textbox" Rows="2" TextMode="MultiLine"></asp
        :textbox></TD>
    <TD>
      <asp:checkbox id="chkExternalInstitutions" runat="server"></asp:checkbox></TD>
  </TR>
  <TR>
    <TD style="WIDTH: 138px"><STRONG>Created</STRONG></TD>
    <TD><asp:textbox id="tbxCreated" runat="server" CssClass="textbox"></asp:textbox></TD>
    <TD></TD>
  </TR>
  <TR>
    <TD style="WIDTH: 138px"><STRONG>CreatedBy</STRONG></TD>
    <TD><asp:textbox id="tbxCreatedBy" runat="server" CssClass="textbox"></asp:textbox></TD>
    <TD></TD>
  </TR>
  <TR>
    <TD style="WIDTH: 138px"><STRONG>Updated</STRONG></TD>
    <TD width="50%">
      <asp:TextBox id="tbxUpdated" runat="server" CssClass="textbox"></asp:TextBox></TD>
    <TD width="50%"></TD>
  </TR>
  <TR>
    <TD style="WIDTH: 138px"><STRONG>UpdatedBy</STRONG></TD>
    <TD>
      <asp:TextBox id="tbxUpdatedBy" runat="server" CssClass="textbox"></asp:TextBox></TD>
    <TD></TD>
```

```
            </TR>
          </TABLE>
        </TD>
      </TR>
    </TABLE>
  </form>
  <P> </P>
  </FORM>
 </body>
</HTML>
```

coursegrab.aspx.cs

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data.SqlClient;
using System.Data;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using System.Text;
using System.Web.Security;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.Courses.Test
{
  /// <summary>
  /// Test class for the <see cref="StudyPlanning.DAL.Courses.CourseGrab"/> class.
  /// </summary>
  public class CourseGrabTest : System.Web.UI.Page
  {
    protected System.Web.UI.WebControls.TextBox tbxCreated;
    protected System.Web.UI.WebControls.TextBox tbxCreatedBy;
    protected System.Web.UI.WebControls.TextBox tbxUpdated;
    protected System.Web.UI.WebControls.TextBox tbxUpdatedBy;
    protected System.Web.UI.WebControls.Button btnCreate;
    protected System.Web.UI.WebControls.Button btnRetrieve;
    protected System.Web.UI.WebControls.Button btnUpdate;
    protected System.Web.UI.WebControls.Button btnDelete;
    protected System.Web.UI.WebControls.Button btnReset;
    protected System.Web.UI.WebControls.TextBox tbxCourseVersion_ID;
    protected System.Web.UI.WebControls.TextBox tbxCourse_ID;
    protected System.Web.UI.WebControls.TextBox tbxNumber;
    protected System.Web.UI.WebControls.Label lblError;
    protected System.Web.UI.WebControls.TextBox tbxCourseGrab_ID;
    protected System.Web.UI.WebControls.TextBox tbxExaminationPlacementEn;
    protected System.Web.UI.WebControls.TextBox tbxExaminationPlacementDa;
    protected System.Web.UI.WebControls.TextBox tbxExaminationAidsEn;
    protected System.Web.UI.WebControls.TextBox tbxExaminationAidsDa;
    protected System.Web.UI.WebControls.TextBox tbxScheduleEn;
    protected System.Web.UI.WebControls.TextBox tbxScheduleDa;
    protected System.Web.UI.WebControls.TextBox tbxDuration;
    protected System.Web.UI.WebControls.TextBox tbxPreviousCourseNumbers;
    protected System.Web.UI.WebControls.TextBox tbxMandatoryPrerequisites;
    protected System.Web.UI.WebControls.TextBox tbxTechnicalPrerequisites;
    protected System.Web.UI.WebControls.TextBox tbxDesirablePrerequisites;
    protected System.Web.UI.WebControls.TextBox tbxExternalInstitutions;
    protected System.Web.UI.WebControls.CheckBox chkDtuVersion;
    protected System.Web.UI.WebControls.CheckBox chkSchedule;
    protected System.Web.UI.WebControls.CheckBox chkDuration;
    protected System.Web.UI.WebControls.CheckBox chkExaminationPlacement;
    protected System.Web.UI.WebControls.CheckBox chkExaminationAids;
    protected System.Web.UI.WebControls.CheckBox chkPreviousCourseNumbers;
    protected System.Web.UI.WebControls.CheckBox chkTechnicalPrerequisites;
    protected System.Web.UI.WebControls.CheckBox chkMandatoryPrerequisites;
    protected System.Web.UI.WebControls.CheckBox chkDesirablePrerequisites;
    protected System.Web.UI.WebControls.CheckBox chkExternalInstitutions;
    protected System.Web.UI.WebControls.TextBox tbxDtuVersion;

    private void Page_Load(object sender, System.EventArgs e)
    {

    }

    private void InitializeComponent()
    {
      this.Load += new System.EventHandler(this.Page_Load);

    }

    private string writeError(DalException x)
    {
      string strError = "";
      //strError += "<script language=\"JavaScript\">";
      //strError += "alert(\"" + x.ToString() + "\");";
      //strError += "</script>";
      strError += x.ToString();

      if (x.Number == 16)
        strError += "<br/><br/>" + x.Sql.ToString();
      //strError += "alert(\"TEST\");";
      //strError += "</script>";
      return strError;
```

```
}

protected void btnCreate_Click(object sender, System.EventArgs e)
{
  lblError.Text = "";

  CourseGrab cg = new CourseGrab();
  SetValues(cg, CrudType.Create);

  try
  {
    cg.Create();
    tbxCourse_CourseGrab_ID.Text = cg.Course_CourseGrab_ID.Value.ToString();
  }
  catch (DalException x)
  {
    lblError.Text = writeError(x);
    return;
  }
}

protected void btnRetrieve_Click(object sender, System.EventArgs e)
{
  lblError.Text = "";

  CourseGrab cg = new CourseGrab();
  cg.CourseGrab_ID.Value = new Guid(tbxCourseGrab_ID.Text);

  try
  {
    cg.Retrieve();
  }
  catch (DalException x)
  {
    lblError.Text = writeError(x);
    return;
  }

  tbxCourseGrab_ID.Text = cg.CourseGrab_ID.Value.ToString();
  tbxCourseVersion_ID.Text = cg.CourseVersion_ID.Value.ToString();
  tbxCourse_ID.Text = cg.Course_ID.Value.ToString();
  tbxNumber.Text = cg.Number.Value.ToString();

  if (cg.DtuVersion.IsNull)
  {
    chkDtuVersion.Checked = true;
    tbxDtuVersion.Text = "";
  }
  else
  {
    chkDtuVersion.Checked = false;
    tbxDtuVersion.Text = cg.DtuVersion.Value.ToString();
  }

  if (cg.Schedule.IsNull)
  {
    chkSchedule.Checked = true;
    tbxScheduleEn.Text = "";
    tbxScheduleDa.Text = "";
  }
  else
  {
    chkSchedule.Checked = false;
    tbxScheduleEn.Text = cg.Schedule["en-GB"];
    tbxScheduleDa.Text = cg.Schedule["da-DK"];
  }

  if (cg.Duration.IsNull)
  {
    chkDuration.Checked = true;
    tbxDuration.Text = "";
  }
  else
  {
    chkDuration.Checked = false;
    tbxDuration.Text = cg.Duration.Value.ToString();
  }

  if (cg.ExaminationPlacement.IsNull)
  {
    chkExaminationPlacement.Checked = true;
    tbxExaminationPlacementEn.Text = "";
    tbxExaminationPlacementDa.Text = "";
  }
  else
  {
    chkExaminationPlacement.Checked = false;
    tbxExaminationPlacementEn.Text = cg.ExaminationPlacement["en-GB"];
    tbxExaminationPlacementDa.Text = cg.ExaminationPlacement["da-DK"];
  }

  if (cg.ExaminationAids.IsNull)
  {
    chkExaminationAids.Checked = true;
    tbxExaminationAidsEn.Text = "";
    tbxExaminationAidsDa.Text = "";
  }
  else
```

```
    {
      chkExaminationAids.Checked = false;
      tbxExaminationAidsEn.Text = cg.ExaminationAids["en-GB"];
      tbxExaminationAidsDa.Text = cg.ExaminationAids["da-DK"];
    }

    if (cg.PreviousCourseNumbers.IsNull)
    {
      chkPreviousCourseNumbers.Checked = true;
      tbxPreviousCourseNumbers.Text = "";
    }
    else
    {
      chkPreviousCourseNumbers.Checked = false;
      tbxPreviousCourseNumbers.Text = cg.PreviousCourseNumbers.Value.ToString();
    }

    if (cg.MandatoryPrerequisites.IsNull)
    {
      chkMandatoryPrerequisites.Checked = true;
      tbxMandatoryPrerequisites.Text = "";
    }
    else
    {
      chkMandatoryPrerequisites.Checked = false;
      tbxMandatoryPrerequisites.Text = cg.MandatoryPrerequisites.Value.ToString();
    }

    if (cg.TechnicalPrerequisites.IsNull)
    {
      chkTechnicalPrerequisites.Checked = true;
      tbxTechnicalPrerequisites.Text = "";
    }
    else
    {
      chkTechnicalPrerequisites.Checked = false;
      tbxTechnicalPrerequisites.Text = cg.TechnicalPrerequisites.Value.ToString();
    }

    if (cg.DesirablePrerequisites.IsNull)
    {
      chkDesirablePrerequisites.Checked = true;
      tbxDesirablePrerequisites.Text = "";
    }
    else
    {
      chkDesirablePrerequisites.Checked = false;
      tbxDesirablePrerequisites.Text = cg.DesirablePrerequisites.Value.ToString();
    }

    if (cg.ExternalInstitutions.IsNull)
    {
      chkExternalInstitutions.Checked = true;
      tbxExternalInstitutions.Text = "";
    }
    else
    {
      chkExternalInstitutions.Checked = false;
      tbxExternalInstitutions.Text = cg.ExternalInstitutions.Value.ToString();
    }

    if (cg.LastUpdated.IsNull)
    {
      chkLastUpdated.Checked = true;
      tbxLastUpdated.Text = "";
    }
    else
    {
      chkLastUpdated.Checked = false;
      tbxLastUpdated.Text = cg.LastUpdated.Value.ToString();
    }

    tbxCreated.Text  = cg.Log.Created.Value.ToString();
    tbxCreatedBy.Text = cg.Log.CreatedBy.Value.ToString();
    tbxUpdated.Text  = cg.Log.Updated.Value.ToString();
    tbxUpdatedBy.Text = cg.Log.UpdatedBy.Value.ToString();
  }

  protected void btnUpdate_Click(object sender, System.EventArgs e)
  {
    CourseGrab orgCg = new CourseGrab();
    orgCg.CourseGrab_ID.Value = new Guid(tbxCourseGrab_ID.Text);

    CourseGrab cg = new CourseGrab();
    SetValues(cg, CrudType.Update);

    try
    {
      orgCg.Retrieve();
      cg.Update(orgCg);
    }
    catch (DalException x)
    {
      lblError.Text = writeError(x);
      return;
    }
  }

  protected void btnDelete_Click(object sender, System.EventArgs e)
```

```
{
  lblError.Text = "";

  CourseGrab cg = new CourseGrab();
  cg.CourseGrab_ID.Value = new Guid(tbxCourseGrab_ID.Text);

  try
  {
    cg.Retrieve();
    cg.Delete();
  }
  catch (DalException x)
  {
    lblError.Text = writeError(x);
    return;
  }
}

private void SetValues(CourseGrab courseGrab, CrudType type)
{
  if (tbxCourseGrab_ID.Text.Length > 0)
    courseGrab.CourseGrab_ID.Value = new Guid(tbxCourseGrab_ID.Text);
  else if (type.Equals(CrudType.Create))
    courseGrab.CourseGrab_ID.Value = Guid.NewGuid();

  if (tbxCourseVersion_ID.Text.Length > 0)
    courseGrab.CourseVersion_ID.Value = new Guid(tbxCourseVersion_ID.Text);

  if (tbxCourse_ID.Text.Length > 0)
    courseGrab.Course_ID.Value = new Guid(tbxCourse_ID.Text);

  if (tbxNumber.Text.Length > 0)
    courseGrab.Number.Value = tbxNumber.Text;

  if (chkDtuVersion.Checked)
    courseGrab.DtuVersion.IsNull = true;
  else if (tbxDtuVersion.Text.Length > 0)
    courseGrab.DtuVersion.Value = Convert.ToInt32(tbxDtuVersion.Text);

  if (chkSchedule.Checked)
    courseGrab.Schedule.IsNull = true;
  else
  {
    if (tbxScheduleEn.Text.Length > 0)
      courseGrab.Schedule["en-GB"] = tbxScheduleEn.Text;
    if (tbxScheduleDa.Text.Length > 0)
      courseGrab.Schedule["da-DK"] = tbxScheduleDa.Text;
  }

  if (chkDuration.Checked)
    courseGrab.Duration.IsNull = true;
  else if (tbxDuration.Text.Length > 0)
    courseGrab.Duration.Value = tbxDuration.Text;

  if (chkExaminationPlacement.Checked)
    courseGrab.ExaminationPlacement.IsNull = true;
  else
  {
    if (tbxExaminationPlacementEn.Text.Length > 0)
      courseGrab.ExaminationPlacement["en-GB"] = tbxExaminationPlacementEn.Text;
    if (tbxExaminationPlacementDa.Text.Length > 0)
      courseGrab.ExaminationPlacement["da-DK"] = tbxExaminationPlacementDa.Text;
  }

  if (chkExaminationAids.Checked)
    courseGrab.ExaminationAids.IsNull = true;
  else
  {
    if (tbxExaminationAidsEn.Text.Length > 0)
      courseGrab.ExaminationAids["en-GB"] = tbxExaminationAidsEn.Text;
    if (tbxExaminationAidsDa.Text.Length > 0)
      courseGrab.ExaminationAids["da-DK"] = tbxExaminationAidsDa.Text;
  }

  if (chkPreviousCourseNumbers.Checked)
    courseGrab.PreviousCourseNumbers.IsNull = true;
  else if (tbxPreviousCourseNumbers.Text.Length > 0)
    courseGrab.PreviousCourseNumbers.Value = tbxPreviousCourseNumbers.Text;

  if (chkMandatoryPrerequisites.Checked)
    courseGrab.MandatoryPrerequisites.IsNull = true;
  else if (tbxMandatoryPrerequisites.Text.Length > 0)
    courseGrab.MandatoryPrerequisites.Value = tbxMandatoryPrerequisites.Text;

  if (chkTechnicalPrerequisites.Checked)
    courseGrab.TechnicalPrerequisites.IsNull = true;
  else if (tbxTechnicalPrerequisites.Text.Length > 0)
    courseGrab.TechnicalPrerequisites.Value = tbxTechnicalPrerequisites.Text;

  if (chkDesirablePrerequisites.Checked)
    courseGrab.DesirablePrerequisites.IsNull = true;
  else if (tbxDesirablePrerequisites.Text.Length > 0)
    courseGrab.DesirablePrerequisites.Value = tbxDesirablePrerequisites.Text;

  if (chkExternalInstitutions.Checked)
    courseGrab.ExternalInstitutions.IsNull = true;
  else if (tbxExternalInstitutions.Text.Length > 0)
    courseGrab.ExternalInstitutions.Value = tbxExternalInstitutions.Text;
```

```
      //Log
      if (tbxCreated.Text.Length > 0)
        courseGrab.Log.Created.Value = Convert.ToDateTime(tbxCreated.Text);
      else if (type.Equals(CrudType.Create))
        courseGrab.Log.Created.Value = DateTime.Now;

      if (tbxCreatedBy.Text.Length > 0)
        courseGrab.Log.CreatedBy.Value = new Guid(tbxCreatedBy.Text);
      else if (type.Equals(CrudType.Create))
        courseGrab.Log.CreatedBy.Value = Guid.NewGuid();

      if (tbxUpdated.Text.Length > 0)
        courseGrab.Log.Updated.Value = Convert.ToDateTime(tbxUpdated.Text);
      else if (type.Equals(CrudType.Create))
        courseGrab.Log.Updated.Value = DateTime.Now;

      if (tbxUpdatedBy.Text.Length > 0)
        courseGrab.Log.UpdatedBy.Value = new Guid(tbxUpdatedBy.Text);
      else if (type.Equals(CrudType.Create))
        courseGrab.Log.UpdatedBy.Value = Guid.NewGuid();
    }

    private enum CrudType { Create, Update }

    protected void btnReset_Click(object sender, System.EventArgs e)
    {
      lblError.Text = "";

      tbxCourseVersion_ID.Text = "";
      tbxCourse_ID.Text = "";
      tbxNumber.Text = "";
      tbxDtuVersion.Text = "";
      tbxCourseGrab_ID.Text = "";
      tbxExaminationPlacementEn.Text = "";
      tbxExaminationPlacementDa.Text = "";
      tbxExaminationAidsEn.Text = "";
      tbxExaminationAidsDa.Text = "";
      tbxScheduleEn.Text = "";
      tbxScheduleDa.Text = "";
      tbxDuration.Text = "";
      tbxPreviousCourseNumbers.Text = "";
      tbxMandatoryPrerequisites.Text = "";
      tbxTechnicalPrerequisites.Text = "";
      tbxDesirablePrerequisites.Text = "";
      tbxExternalInstitutions.Text = "";

      chkDtuVersion.Checked = false;
      chkSchedule.Checked = false;
      chkDuration.Checked = false;
      chkExaminationPlacement.Checked = false;
      chkExaminationAids.Checked = false;
      chkPreviousCourseNumbers.Checked = false;
      chkTechnicalPrerequisites.Checked = false;
      chkMandatoryPrerequisites.Checked = false;
      chkDesirablePrerequisites.Checked = false;
      chkExternalInstitutions.Checked = false;

      tbxCreated.Text = "";
      tbxCreatedBy.Text = "";
      tbxUpdated.Text = "";
      tbxUpdatedBy.Text = "";
    }

  }
}
```

2.1.3 CourseVersion

courseversion.aspx

```
<%@ Page language="c#" Src="CourseVersion.aspx.cs" CodeBehind="CourseVersion.aspx.cs" AutoEventWireup="false" Inherits="
  StudyPlanning.DAL.Courses.Test.CourseVersionTest" trace="false" warningLevel="4" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" >
<HTML>
  <HEAD>
    <title>CourseVersion</title>
    <meta content="True" name="vs_showGrid">
    <meta content="Microsoft Visual Studio 7.0" name="GENERATOR">
    <meta content="C#" name="CODE_LANGUAGE">
    <meta content="JavaScript" name="vs_defaultClientScript">
    <meta content="http://schemas.microsoft.com/intellisense/ie5" name="vs_targetSchema">
    <LINK href="../../../misc/style.css" rel="stylesheet">
    <style>.textbox { MARGIN-TOP: 0px; DISPLAY: inline; WIDTH: 400px }
  .button { WIDTH: 120px }
  </style>
</HEAD>
  <body>
    <br>
    <asp:Label id="lblError" ForeColor="MediumBlue" Runat="server"></asp:Label>
    <p><STRONG><FONT size="4">CourseVersion</FONT></STRONG></p>
    <form id="form" method="post" runat="server">
      <TABLE id="Table2" cellSpacing="1" cellPadding="1" align="center" border="1">
        <TR>
```

```
<TD align="middle" valign="top">
  <br>
  <p><asp:button id="btnCreate" onclick="btnCreate_Click" runat="server" CssClass="button" Text="Create"></asp:button>
    </p>
  <p><asp:button id="btnRetrieve" onclick="btnRetrieve_Click" runat="server" CssClass="button" Text="Retrieve"></asp:
    button></p>
  <p><asp:button id="btnUpdate" onclick="btnUpdate_Click" runat="server" CssClass="button" Text="Update"></asp:button>
    </p>
  <p><asp:button id="btnDelete" onclick="btnDelete_Click" runat="server" CssClass="button" Text="Delete"></asp:button>
    </p>
  <p><asp:button id="btnReset" onclick="btnReset_Click" runat="server" CssClass="button" Text="Reset"></asp:button></
    p>
</TD>
<TD align="middle">
  <TABLE id="Table1" cellSpacing="1" cellPadding="1" border="1">
    <TR>
      <TD style="WIDTH: 138px"><STRONG>CourseVersion_ID</STRONG></TD>
      <TD><STRONG><asp:textbox id="tbxCourseVersion_ID" runat="server" CssClass="textbox"></asp:textbox></
        STRONG></TD>
      <TD></TD>
    </TR>
    <TR>
      <TD style="WIDTH: 138px"><STRONG>Course_ID</STRONG></TD>
      <TD><asp:textbox id="tbxCourse_ID" runat="server" CssClass="textbox" Rows="1"></asp:textbox></TD>
      <TD></TD>
    </TR>
    <TR>
      <TD style="WIDTH: 138px"><STRONG>Version</STRONG></TD>
      <TD><STRONG><asp:textbox id="tbxVersion" runat="server" CssClass="textbox" Rows="1"></asp:textbox></
        STRONG></TD>
      <TD></TD>
    </TR>
    <TR>
      <TD style="WIDTH: 138px"><STRONG>Number</STRONG></TD>
      <TD><STRONG><asp:textbox id="tbxNumber" runat="server" CssClass="textbox" Rows="1"></asp:textbox></
        STRONG></TD>
      <TD></TD>
    </TR>
    <TR>
      <TD style="WIDTH: 138px" rowSpan="2"><STRONG>Name</STRONG></TD>
      <TD><STRONG>EN:
          <asp:textbox id="tbxNameEn" runat="server" CssClass="textbox" Rows="1"></asp:textbox></STRONG></TD>
      <TD rowSpan="2"></TD>
    </TR>
    <tr>
      <TD><STRONG>da-DK:
          <asp:textbox id="tbxNameDa" runat="server" CssClass="textbox" Rows="1"></asp:textbox></STRONG></TD>
    </tr>
    <TR>
      <TD style="WIDTH: 138px"><STRONG>Language_ID</STRONG></TD>
      <TD><STRONG><asp:textbox id="tbxLanguage_ID" runat="server" CssClass="textbox"></asp:textbox></STRONG
        ></TD>
      <TD></TD>
    </TR>
    <TR>
      <TD style="WIDTH: 138px" rowSpan="2"><STRONG>TeachingForm</STRONG></TD>
      <TD><STRONG>EN:
          <asp:textbox id="tbxTeachingFormEn" runat="server" CssClass="textbox" Rows="2" TextMode="MultiLine"></asp:
            textbox></STRONG></TD>
      <TD rowSpan="2"><strong> </strong></TD>
    </TR>
    <tr>
      <TD><STRONG>da-DK:
          <asp:textbox id="tbxTeachingFormDa" runat="server" CssClass="textbox" Rows="2" TextMode="MultiLine"></asp:
            textbox></STRONG></TD>
    </tr>
    <TR>
      <TD style="WIDTH: 138px"><STRONG>Parts</STRONG></TD>
      <TD>
        <asp:textbox id="tbxParts" runat="server" CssClass="textbox"></asp:textbox></TD>
      <TD></TD>
    </TR>
    <TR>
      <TD style="WIDTH: 138px"><STRONG>AssessmentType_ID</STRONG></TD>
      <TD><asp:textbox id="tbxAssessmentType_ID" runat="server" CssClass="textbox"></asp:textbox></TD>
      <TD></TD>
    </TR>
    <TR>
      <TD style="WIDTH: 138px" rowSpan="2"><STRONG>EvaluationFormDescription</STRONG></TD>
      <TD><STRONG>EN:
          <asp:textbox id="tbxEvaluationFormDescriptionEn" runat="server" CssClass="textbox" Rows="2" TextMode="MultiLine"
            ></asp:textbox></STRONG></TD>
      <TD rowSpan="2"><strong>Null:
          <asp:checkbox id="chkEvaluationFormDescription" runat="server" Text="Null"></asp:checkbox></strong></TD>
    </TR>
    <tr>
      <TD><STRONG>da-DK:
          <asp:textbox id="tbxEvaluationFormDescriptionDa" runat="server" CssClass="textbox" Rows="2" TextMode="MultiLine"
            ></asp:textbox></STRONG></TD>
    </tr>
    <TR>
      <TD style="WIDTH: 138px"><STRONG>ParticipantLimitationMin</STRONG></TD>
      <TD><asp:textbox id="tbxParticipantLimitationMin" runat="server" CssClass="textbox" Width="28px"></asp:textbox>
        </TD>
      <TD><STRONG>Null:
          <asp:checkbox id="chkParticipantLimitationMin" runat="server" Text="Null"></asp:checkbox></STRONG></TD
            >
    </TR>
    <TR>
```

```
            <TD style="WIDTH: 138px"><STRONG>ParticipantLimitationMax</STRONG></TD>
            <TD>
              <asp:textbox id="tbxParticipantLimitationMax" runat="server" CssClass="textbox" Width="28px"></asp:textbox></
                TD>
            <TD><STRONG>Null:
                <asp:checkbox id="chkParticipantLimitationMax" runat="server" Text="Null"></asp:checkbox></STRONG></TD
                >
          </TR>
          <TR>
            <TD style="WIDTH: 138px" rowSpan="2"><STRONG>Objective</STRONG></TD>
            <TD><STRONG>EN:
                <asp:textbox id="tbxObjectiveEn" runat="server" CssClass="textbox" Rows="2" TextMode="MultiLine"></asp:
                  textbox></STRONG></TD>
            <TD rowSpan="2"><strong> </strong></TD>
          </TR>
          <tr>
            <TD><STRONG>da-DK:
                <asp:textbox id="tbxObjectiveDa" runat="server" CssClass="textbox" Rows="2" TextMode="MultiLine"></asp:
                  textbox></STRONG></TD>
          </tr>
          <TR>
            <TD style="WIDTH: 138px" rowSpan="2"><STRONG>Contents</STRONG></TD>
            <TD><STRONG>EN:
                <asp:textbox id="tbxContentsEn" runat="server" CssClass="textbox" Rows="2" TextMode="MultiLine"></asp:
                  textbox></STRONG></TD>
            <TD rowSpan="2"><strong> </strong></TD>
          </TR>
          <tr>
            <TD><STRONG>da-DK:
                <asp:textbox id="tbxContentsDa" runat="server" CssClass="textbox" Rows="2" TextMode="MultiLine"></asp:
                  textbox></STRONG></TD>
          </tr>
          <TR>
            <TD style="WIDTH: 138px" rowSpan="2"><STRONG>Remark</STRONG></TD>
            <TD><STRONG>EN:
                <asp:textbox id="tbxRemarkEn" runat="server" CssClass="textbox" Rows="2" TextMode="MultiLine"></asp:textbox
                  ></STRONG></TD>
            <TD rowSpan="2"><strong>Null:
                <asp:checkbox id="chkRemark" runat="server" Text="Null"></asp:checkbox></strong></TD>
          </TR>
          <tr>
            <TD><STRONG>da-DK:
                <asp:textbox id="tbxRemarkDa" runat="server" CssClass="textbox" Rows="2" TextMode="MultiLine"></asp:textbox
                  ></STRONG></TD>
          </tr>
          <TR>
            <TD style="WIDTH: 138px"><STRONG>Url</STRONG></TD>
            <TD>
              <asp:textbox id="tbxUrl" runat="server" CssClass="textbox" Rows="1"></asp:textbox></TD>
            <TD><STRONG>Null:
                <asp:checkbox id="chkUrl" runat="server" Text="Null"></asp:checkbox></STRONG></TD>
          </TR>
          <TR>
            <TD style="WIDTH: 138px"><STRONG>Created</STRONG></TD>
            <TD><asp:textbox id="tbxCreated" runat="server" CssClass="textbox"></asp:textbox></TD>
            <TD></TD>
          </TR>
          <TR>
            <TD style="WIDTH: 138px"><STRONG>CreatedBy</STRONG></TD>
            <TD><asp:textbox id="tbxCreatedBy" runat="server" CssClass="textbox"></asp:textbox></TD>
            <TD></TD>
          </TR>
          <TR>
            <TD style="WIDTH: 138px"><STRONG>Updated</STRONG></TD>
            <TD width="50%">
              <asp:TextBox id="tbxUpdated" runat="server" CssClass="textbox"></asp:TextBox></TD>
            <TD width="50%"></TD>
          </TR>
          <TR>
            <TD style="WIDTH: 138px"><STRONG>UpdatedBy</STRONG></TD>
            <TD>
              <asp:TextBox id="tbxUpdatedBy" runat="server" CssClass="textbox"></asp:TextBox></TD>
            <TD></TD>
          </TR>
        </TABLE>
      </TD>
    </TR>
  </TABLE>
 </form>
 <P> </P></FORM>
</body>
</HTML>
```

courseversion.aspx.cs

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data.SqlClient;
using System.Data;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using System.Text;
```

```
using System.Web.Security;
using StudyPlanning.DAL;

namespace StudyPlanning.DAL.Courses.Test
{
  /// <summary>
  /// Summary description for forgotPassword.
  /// </summary>
  public class CourseVersionTest : System.Web.UI.Page
  {
    protected System.Web.UI.WebControls.TextBox tbxCreated;
    protected System.Web.UI.WebControls.TextBox tbxCreatedBy;
    protected System.Web.UI.WebControls.TextBox tbxUpdated;
    protected System.Web.UI.WebControls.TextBox tbxUpdatedBy;
    protected System.Web.UI.WebControls.Button btnCreate;
    protected System.Web.UI.WebControls.Button btnRetrieve;
    protected System.Web.UI.WebControls.Button btnUpdate;
    protected System.Web.UI.WebControls.Button btnDelete;
    protected System.Web.UI.WebControls.Button btnReset;
    protected System.Web.UI.WebControls.TextBox tbxCourseVersion_ID;
    protected System.Web.UI.WebControls.TextBox tbxCourse_ID;
    protected System.Web.UI.WebControls.TextBox tbxVersion;
    protected System.Web.UI.WebControls.TextBox tbxNumber;
    protected System.Web.UI.WebControls.TextBox tbxTeachingFormDa;
    protected System.Web.UI.WebControls.TextBox tbxNameEn;
    protected System.Web.UI.WebControls.TextBox tbxNameDa;
    protected System.Web.UI.WebControls.TextBox tbxTeachingFormEn;
    protected System.Web.UI.WebControls.TextBox tbxAssessmentType_ID;
    protected System.Web.UI.WebControls.TextBox tbxParticipantLimitationMin;
    protected System.Web.UI.WebControls.TextBox tbxParticipantLimitationMax;
    protected System.Web.UI.WebControls.TextBox tbxObjectiveEn;
    protected System.Web.UI.WebControls.TextBox tbxObjectiveDa;
    protected System.Web.UI.WebControls.TextBox tbxContentsEn;
    protected System.Web.UI.WebControls.TextBox tbxContentsDa;
    protected System.Web.UI.WebControls.TextBox tbxRemarkEn;
    protected System.Web.UI.WebControls.CheckBox chkRemark;
    protected System.Web.UI.WebControls.TextBox tbxRemarkDa;
    protected System.Web.UI.WebControls.TextBox tbxUrl;
    protected System.Web.UI.WebControls.TextBox tbxEvaluationFormDescriptionEn;
    protected System.Web.UI.WebControls.TextBox tbxEvaluationFormDescriptionDa;
    protected System.Web.UI.WebControls.CheckBox chkUrl;
    protected System.Web.UI.WebControls.CheckBox chkParticipantLimitationMin;
    protected System.Web.UI.WebControls.CheckBox chkParticipantLimitationMax;
    protected System.Web.UI.WebControls.TextBox tbxLanguage_ID;
    protected System.Web.UI.WebControls.CheckBox chkEvaluationFormDescription;
    protected System.Web.UI.WebControls.Label lblError;
    protected System.Web.UI.WebControls.TextBox tbxParts;

    private void Page_Load(object sender, System.EventArgs e)
    {

    }

    private void InitializeComponent()
    {
      this.Load += new System.EventHandler(this.Page_Load);

    }

    private string writeError(DalException x)
    {
      string strError = "";
      //strError += "<script language=\"JavaScript\">";
      //strError += "alert(\"" + x.ToString() + "\");";
      //strError += "</script>";
      strError += x.ToString();

      if (x.Number == 16)
        strError += "<br/><br/>" + x.Sql.ToString();
      //strError += "alert(\"TEST\");";
      //strError += "</script>";
      return strError;

    }

    protected void btnCreate_Click(object sender, System.EventArgs e)
    {
      lblError.Text = "";

      CourseVersion cv = new CourseVersion();

      if (tbxCourseVersion_ID.Text.Length > 0)
        cv.CourseVersion_ID.Value = new Guid(tbxCourseVersion_ID.Text);
      else
        cv.CourseVersion_ID.Value = Guid.NewGuid();

      if (tbxCourse_ID.Text.Length > 0)
        cv.Course_ID.Value = new Guid(tbxCourse_ID.Text);
      else
        cv.Course_ID.Value = Guid.NewGuid();

      if (tbxVersion.Text.Length > 0)
        cv.Version.Value = Convert.ToInt32(tbxVersion.Text);

      if (tbxNumber.Text.Length > 0)
        cv.Number.Value = tbxNumber.Text;

      if (tbxNameEn.Text.Length > 0)
        cv.Name.Add("en-GB", tbxNameEn.Text);
```

```
if (tbxNameDa.Text.Length > 0)
  cv.Name.Add("da-DK", tbxNameDa.Text);

if (tbxLanguage_ID.Text.Length > 0)
  cv.Language_ID.Value = tbxLanguage_ID.Text;

if (tbxTeachingFormEn.Text.Length > 0)
  cv.TeachingForm.Add("en-GB", tbxTeachingFormEn.Text);

if (tbxTeachingFormDa.Text.Length > 0)
  cv.TeachingForm.Add("da-DK", tbxTeachingFormDa.Text);

if (tbxParts.Text.Length > 0)
  cv.Parts.Value = Convert.ToInt32(tbxParts.Text);

if (tbxAssessmentType_ID.Text.Length > 0)
  cv.AssessmentType_ID.Value = Convert.ToInt32(tbxAssessmentType_ID.Text);

if (!chkEvaluationFormDescription.Checked)
{
  if (tbxEvaluationFormDescriptionEn.Text.Length > 0)
    cv.EvaluationFormDescription.Add("en-GB", tbxEvaluationFormDescriptionEn.Text);

  if (tbxEvaluationFormDescriptionDa.Text.Length > 0)
    cv.EvaluationFormDescription.Add("da-DK", tbxEvaluationFormDescriptionDa.Text);
}
else
  cv.EvaluationFormDescription.IsNull = true;

if (!chkParticipantLimitationMin.Checked)
{
  if (tbxParticipantLimitationMin.Text.Length > 0)
    cv.ParticipantLimitationMin.Value = Convert.ToInt32(tbxParticipantLimitationMin.Text);
}
else
  cv.ParticipantLimitationMin.IsNull = true;

if (!chkParticipantLimitationMax.Checked)
{
  if (tbxParticipantLimitationMax.Text.Length > 0)
    cv.ParticipantLimitationMax.Value = Convert.ToInt32(tbxParticipantLimitationMax.Text);
}
else
  cv.ParticipantLimitationMax.IsNull = true;

if (tbxObjectiveEn.Text.Length > 0)
  cv.Objective.Add("en-GB", tbxObjectiveEn.Text);

if (tbxObjectiveDa.Text.Length > 0)
  cv.Objective.Add("da-DK", tbxObjectiveDa.Text);

if (tbxContentsEn.Text.Length > 0)
  cv.Contents.Add("en-GB", tbxContentsEn.Text);

if (tbxContentsDa.Text.Length > 0)
  cv.Contents.Add("da-DK", tbxContentsDa.Text);

if (!chkRemark.Checked)
{
  if (tbxRemarkEn.Text.Length > 0)
    cv.Remark.Add("en-GB", tbxRemarkEn.Text);

  if (tbxRemarkDa.Text.Length > 0)
    cv.Remark.Add("da-DK", tbxRemarkDa.Text);
}

if (!chkUrl.Checked)
{
  if (tbxUrl.Text.Length > 0)
    cv.Url.Value = tbxUrl.Text;
}

if (tbxCreated.Text.Length == 0)
  cv.Log.Created.Value = DateTime.Now;
else
  cv.Log.Created.Value = Convert.ToDateTime(tbxCreated.Text);

if (tbxCreatedBy.Text.Length == 0)
  cv.Log.CreatedBy.Value = Guid.NewGuid();
else
  cv.Log.CreatedBy.Value = new Guid(tbxCreatedBy.Text);

if (tbxUpdated.Text.Length == 0)
  cv.Log.Updated.Value = DateTime.Now;
else
  cv.Log.Updated.Value = Convert.ToDateTime(tbxUpdated.Text);

if (tbxUpdatedBy.Text.Length == 0)
  cv.Log.UpdatedBy.Value = Guid.NewGuid();
else
  cv.Log.UpdatedBy.Value = new Guid(tbxUpdatedBy.Text);

try
{
  cv.Create();
  tbxCourseVersion_ID.Text = cv.CourseVersion_ID.Value.ToString();
}
catch (DalException x)
```

```
    {
      lblError.Text = writeError(x);
      return;
    }
  }

  protected void btnRetrieve_Click(object sender, System.EventArgs e)
  {
    lblError.Text = "";

    CourseVersion cv = new CourseVersion();
    cv.CourseVersion_ID.Value = new Guid(tbxCourseVersion_ID.Text);

    try
    {
      cv.Retrieve();
    }
    catch (DalException x)
    {
      lblError.Text = writeError(x);
      return;
    }
    tbxCourse_ID.Text = cv.Course_ID.Value.ToString();
    tbxVersion.Text = cv.Version.Value.ToString();
    tbxNumber.Text = cv.Number.Value.ToString();

    tbxNameEn.Text = cv.Name["en-GB"];
    tbxNameDa.Text = cv.Name["da-DK"];

    tbxLanguage_ID.Text = cv.Language_ID.Value.ToString();

    tbxTeachingFormEn.Text = cv.TeachingForm["en-GB"];
    tbxTeachingFormDa.Text = cv.TeachingForm["da-DK"];

    tbxParts.Text = cv.Parts.Value.ToString();
    tbxAssessmentType_ID.Text = cv.AssessmentType_ID.Value.ToString();

    if (cv.EvaluationFormDescription.IsNull)
    {
      tbxEvaluationFormDescriptionEn.Text = "";
      tbxEvaluationFormDescriptionDa.Text = "";
      chkEvaluationFormDescription.Checked = true;
    }
    else
    {
      tbxEvaluationFormDescriptionEn.Text = cv.EvaluationFormDescription["en-GB"];
      tbxEvaluationFormDescriptionDa.Text = cv.EvaluationFormDescription["da-DK"];
      chkEvaluationFormDescription.Checked = false;
    }

    if (cv.ParticipantLimitationMin.IsNull)
    {
      tbxParticipantLimitationMin.Text = "";
      chkParticipantLimitationMin.Checked = true;
    }
    else
    {
      tbxParticipantLimitationMin.Text = cv.ParticipantLimitationMin.Value.ToString();
      chkParticipantLimitationMin.Checked = false;
    }

    if (cv.ParticipantLimitationMax.IsNull)
    {
      tbxParticipantLimitationMax.Text = "";
      chkParticipantLimitationMax.Checked = true;
    }
    else
    {
      tbxParticipantLimitationMax.Text = cv.ParticipantLimitationMax.Value.ToString();
      chkParticipantLimitationMax.Checked = true;
    }

    tbxObjectiveEn.Text = cv.Objective["en-GB"];
    tbxObjectiveDa.Text = cv.Objective["da-DK"];

    tbxContentsEn.Text = cv.Contents["en-GB"];
    tbxContentsDa.Text = cv.Contents["da-DK"];

    if (cv.Remark.IsNull)
    {
      tbxRemarkEn.Text = "";
      tbxRemarkDa.Text = "";
      chkRemark.Checked = true;
    }
    else
    {
      tbxRemarkEn.Text = cv.Remark["en-GB"];
      tbxRemarkDa.Text = cv.Remark["da-DK"];
      chkRemark.Checked = false;
    }

    if (cv.Url.IsNull)
    {
      tbxUrl.Text = "";
      chkUrl.Checked = true;
    }
    else
    {
      tbxUrl.Text = cv.Url.Value.ToString();
```

```
      chkUrl.Checked = false;
    }

    tbxCreated.Text  = cv.Log.Created.Value.ToString();
    tbxCreatedBy.Text = cv.Log.CreatedBy.Value.ToString();
    tbxUpdated.Text  = cv.Log.Updated.Value.ToString();
    tbxUpdatedBy.Text = cv.Log.UpdatedBy.Value.ToString();
  }

  protected void btnUpdate_Click(object sender, System.EventArgs e)
  {
    CourseVersion orgCv = new CourseVersion();
    orgCv.CourseVersion_ID.Value = new Guid(tbxCourseVersion_ID.Text);

    CourseVersion cv = new CourseVersion();
    cv.CourseVersion_ID.Value = new Guid(tbxCourseVersion_ID.Text);
    cv.Course_ID.Value = new Guid(tbxCourse_ID.Text);

    cv.Version.Value = Convert.ToInt32(tbxVersion.Text);
    cv.Number.Value = tbxNumber.Text;

    if (tbxNameEn.Text.Length > 0)
      cv.Name.Add("en-GB", tbxNameEn.Text);

    if (tbxNameDa.Text.Length > 0)
      cv.Name.Add("da-DK", tbxNameDa.Text);

    cv.Language_ID.Value = tbxLanguage_ID.Text;

    if (tbxTeachingFormEn.Text.Length > 0)
      cv.TeachingForm.Add("en-GB", tbxTeachingFormEn.Text);

    if (tbxTeachingFormDa.Text.Length > 0)
      cv.TeachingForm.Add("da-DK", tbxTeachingFormDa.Text);

    cv.Parts.Value = Convert.ToInt32(tbxParts.Text);
    cv.AssessmentType_ID.Value = Convert.ToInt32(tbxAssessmentType_ID.Text);

    if (!chkEvaluationFormDescription.Checked)
    {
      if (tbxEvaluationFormDescriptionEn.Text.Length > 0)
        cv.EvaluationFormDescription.Add("en-GB", tbxEvaluationFormDescriptionEn.Text);

      if (tbxEvaluationFormDescriptionDa.Text.Length > 0)
        cv.EvaluationFormDescription.Add("da-DK", tbxEvaluationFormDescriptionDa.Text);
    }
    else
      cv.EvaluationFormDescription.IsNull = true;

    if (!chkParticipantLimitationMin.Checked)
    {
      if (tbxParticipantLimitationMin.Text.Length > 0)
        cv.ParticipantLimitationMin.Value = Convert.ToInt32(tbxParticipantLimitationMin.Text);
    }
    else
      cv.ParticipantLimitationMin.IsNull = true;

    if (!chkParticipantLimitationMax.Checked)
    {
      if (tbxParticipantLimitationMax.Text.Length > 0)
        cv.ParticipantLimitationMax.Value = Convert.ToInt32(tbxParticipantLimitationMax.Text);
    }
    else
      cv.ParticipantLimitationMax.IsNull = true;

    if (tbxObjectiveEn.Text.Length > 0)
      cv.Objective.Add("en-GB", tbxObjectiveEn.Text);

    if (tbxObjectiveDa.Text.Length > 0)
      cv.Objective.Add("da-DK", tbxObjectiveDa.Text);

    if (tbxContentsEn.Text.Length > 0)
      cv.Contents.Add("en-GB", tbxContentsEn.Text);

    if (tbxContentsDa.Text.Length > 0)
      cv.Contents.Add("da-DK", tbxContentsDa.Text);

    if (!chkRemark.Checked)
    {
      if (tbxRemarkEn.Text.Length > 0)
        cv.Remark.Add("en-GB", tbxRemarkEn.Text);

      if (tbxRemarkDa.Text.Length > 0)
        cv.Remark.Add("da-DK", tbxRemarkDa.Text);
    }
    else
      cv.Remark.IsNull = true;

    if (!chkUrl.Checked)
    {
      if (tbxUrl.Text.Length > 0)
        cv.Url.Value = tbxUrl.Text;
    }
    else
      cv.Url.IsNull = true;

    cv.Log.Created.Value = Convert.ToDateTime(tbxCreated.Text);
    cv.Log.CreatedBy.Value = new Guid(tbxCreatedBy.Text);
    cv.Log.Updated.Value = Convert.ToDateTime(tbxUpdated.Text);
```

```
      cv.Log.UpdatedBy.Value = new Guid(tbxUpdatedBy.Text);

      try
      {
        orgCv.Retrieve();
        cv.Update(orgCv);
      }
      catch (DalException x)
      {
        lblError.Text = writeError(x);
        return;
      }
    }

    protected void btnDelete_Click(object sender, System.EventArgs e)
    {
      lblError.Text = "";

      CourseVersion cv = new CourseVersion ();
      cv.CourseVersion_ID.Value = new Guid(tbxCourseVersion_ID.Text);

      try
      {
        cv.Retrieve();
        cv.Delete();
      }
      catch (DalException x)
      {
        lblError.Text = writeError(x);
        return;
      }
    }

    protected void btnReset_Click(object sender, System.EventArgs e)
    {
      lblError.Text = "";

      tbxCourse_ID.Text = "";
      tbxCourseVersion_ID.Text = "";
      tbxVersion.Text = "";
      tbxNumber.Text = "";
      tbxNameEn.Text = "";
      tbxNameDa.Text = "";
      tbxLanguage_ID.Text = "";
      tbxTeachingFormEn.Text = "";
      tbxTeachingFormDa.Text = "";
      tbxParts.Text = "";
      tbxAssessmentType_ID.Text = "";
      tbxEvaluationFormDescriptionEn.Text = "";
      tbxEvaluationFormDescriptionDa.Text = "";
      tbxParticipantLimitationMin.Text = "";
      tbxParticipantLimitationMax.Text = "";
      tbxObjectiveEn.Text = "";
      tbxObjectiveDa.Text = "";
      tbxContentsEn.Text = "";
      tbxContentsDa.Text = "";
      tbxRemarkEn.Text = "";
      tbxRemarkDa.Text = "";
      tbxUrl.Text = "";
      tbxCreated.Text = "";
      tbxCreatedBy.Text = "";
      tbxUpdated.Text = "";
      tbxUpdatedBy.Text = "";

      chkEvaluationFormDescription.Checked = false;
      chkParticipantLimitationMin.Checked = false;
      chkParticipantLimitationMax.Checked = false;
      chkRemark.Checked = false;
      chkUrl.Checked = false;
    }

  }
}
```

2.1.4 Department

department.aspx

```
<%@ Page language="c#" Src="Department.aspx.cs" CodeBehind="Department.aspx.cs" AutoEventWireup="false" Inherits="StudyPlanning.
  DAL.Courses.Test.DepartmentTest" trace="false" warningLevel="4" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" >
<HTML>
  <HEAD>
    <title>Department</title>
    <meta content="True" name="vs_showGrid">
    <meta content="Microsoft Visual Studio 7.0" name="GENERATOR">
    <meta content="C#" name="CODE_LANGUAGE">
    <meta content="JavaScript" name="vs_defaultClientScript">
    <meta content="http://schemas.microsoft.com/intellisense/ie5" name="vs_targetSchema">
    <LINK href="../../../misc/style.css" rel="stylesheet">
    <style>.textbox { MARGIN-TOP: 0px; DISPLAY: inline; WIDTH: 400px }
  .button { WIDTH: 120px }
    </style>
```

```
  </HEAD>
  <body>
    <br>
    <asp:Label id="lblError" ForeColor="MediumBlue" Runat="server"></asp:Label>
    <p><STRONG><FONT size="4">Department</FONT></STRONG></p>
    <form id="form" method="post" runat="server">
      <TABLE id="Table2" cellSpacing="1" cellPadding="1" align="center" border="1">
        <TR>
          <TD align="middle" valign="top">
            <br>
            <p><asp:button id="btnCreate" onclick="btnCreate_Click" runat="server" CssClass="button" Text="Create"></asp:button>
              </p>
            <p><asp:button id="btnRetrieve" onclick="btnRetrieve_Click" runat="server" CssClass="button" Text="Retrieve"></asp:
              button></p>
            <p><asp:button id="btnUpdate" onclick="btnUpdate_Click" runat="server" CssClass="button" Text="Update"></asp:button>
              </p>
            <p><asp:button id="btnDelete" onclick="btnDelete_Click" runat="server" CssClass="button" Text="Delete"></asp:button>
              </p>
            <p><asp:button id="btnReset" onclick="btnReset_Click" runat="server" CssClass="button" Text="Reset"></asp:button></
              p>
          </TD>
          <TD align="middle">
            <TABLE id="Table1" cellSpacing="1" cellPadding="1" border="1">
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Course_Department_ID</STRONG></TD>
                <TD><STRONG><asp:textbox id="tbxCourse_Department_ID" runat="server" CssClass="textbox"></asp:textbox></
                  STRONG></TD>
                <TD></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Course_ID</STRONG></TD>
                <TD><asp:textbox id="tbxCourse_ID" runat="server" CssClass="textbox" Rows="1"></asp:textbox></TD>
                <TD></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Department_ID</STRONG></TD>
                <TD><STRONG><asp:textbox id="tbxDepartment_ID" runat="server" CssClass="textbox" Rows="1"></asp:textbox>
                  </STRONG></TD>
                <TD></TD>
              </TR>
              <tr>
              </tr>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Created</STRONG></TD>
                <TD><asp:textbox id="tbxCreated" runat="server" CssClass="textbox"></asp:textbox></TD>
                <TD></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>CreatedBy</STRONG></TD>
                <TD><asp:textbox id="tbxCreatedBy" runat="server" CssClass="textbox"></asp:textbox></TD>
                <TD></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Updated</STRONG></TD>
                <TD width="50%">
                  <asp:TextBox id="tbxUpdated" runat="server" CssClass="textbox"></asp:TextBox></TD>
                <TD width="50%"></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>UpdatedBy</STRONG></TD>
                <TD>
                  <asp:TextBox id="tbxUpdatedBy" runat="server" CssClass="textbox"></asp:TextBox></TD>
                <TD></TD>
              </TR>
            </TABLE>
          </TD>
        </TR>
      </TABLE>
    </form>
    <P> </P>
    </FORM>
  </body>
</HTML>
```

department.aspx.cs

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data.SqlClient;
using System.Data;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using System.Text;
using System.Web.Security;
using StudyPlanning.DAL;
using StudyPlanning.DAL.Courses;

namespace StudyPlanning.DAL.Courses.Test
{
  /// <summary>
  /// Test class for the <see cref="StudyPlanning.DAL.Courses.Department"/> class.
  /// </summary>
```

```
public class DepartmentTest : System.Web.UI.Page
{
  protected System.Web.UI.WebControls.TextBox tbxCreated;
  protected System.Web.UI.WebControls.TextBox tbxCreatedBy;
  protected System.Web.UI.WebControls.TextBox tbxUpdated;
  protected System.Web.UI.WebControls.TextBox tbxUpdatedBy;
  protected System.Web.UI.WebControls.Button btnCreate;
  protected System.Web.UI.WebControls.Button btnRetrieve;
  protected System.Web.UI.WebControls.Button btnUpdate;
  protected System.Web.UI.WebControls.Button btnDelete;
  protected System.Web.UI.WebControls.Button btnReset;
  protected System.Web.UI.WebControls.TextBox tbxCourse_Department_ID;
  protected System.Web.UI.WebControls.TextBox tbxCourse_ID;
  protected System.Web.UI.WebControls.TextBox tbxDepartment_ID;
  protected System.Web.UI.WebControls.Label lblError;

  private void Page_Load(object sender, System.EventArgs e)
  {

  }

  private void InitializeComponent()
  {
    this.Load += new System.EventHandler(this.Page_Load);

  }

  private string writeError(DalException x)
  {
    string strError = "";
    strError += x.ToString();

    if (x.Number == 16)
      strError += "<br/><br/>" + x.Sql.ToString();

    return strError;
  }

  protected void btnCreate_Click(object sender, System.EventArgs e)
  {
    lblError.Text = "";

    Department dp = new Department();
    SetValues(dp, CrudType.Create);

    try
    {
      dp.Create();
      tbxCourse_Department_ID.Text = dp.Course_Department_ID.Value.ToString();
    }
    catch (DalException x)
    {
      lblError.Text = writeError(x);
      return;
    }
  }

  protected void btnRetrieve_Click(object sender, System.EventArgs e)
  {
    lblError.Text = "";

    Department dp = new Department();
    dp.Course_Department_ID.Value = new Guid(tbxCourse_Department_ID.Text);

    try
    {
      dp.Retrieve();
    }
    catch (DalException x)
    {
      lblError.Text = writeError(x);
      return;
    }

    tbxCourse_Department_ID.Text = dp.Course_Department_ID.Value.ToString();
    tbxCourse_ID.Text = dp.Course_ID.Value.ToString();
    tbxDepartment_ID.Text = dp.Department_ID.Value.ToString();

    tbxCreated.Text  = dp.Log.Created.Value.ToString();
    tbxCreatedBy.Text = dp.Log.CreatedBy.Value.ToString();
    tbxUpdated.Text  = dp.Log.Updated.Value.ToString();
    tbxUpdatedBy.Text = dp.Log.UpdatedBy.Value.ToString();
  }

  protected void btnUpdate_Click(object sender, System.EventArgs e)
  {
    Department orgDp = new Department();
    orgDp.Course_Department_ID.Value = new Guid(tbxCourse_Department_ID.Text);

    Department dp = new Department();
    SetValues(dp, CrudType.Update);

    try
    {
      orgDp.Retrieve();
      dp.Update(orgDp);
    }
    catch (DalException x)
    {
```

```
        lblError.Text = writeError(x);
        return;
      }
    }

    protected void btnDelete_Click(object sender, System.EventArgs e)
    {
      lblError.Text = "";

      Department dp = new Department();
      dp.Course_Department_ID.Value = new Guid(tbxCourse_Department_ID.Text);

      try
      {
        dp.Retrieve();
        dp.Delete();
      }
      catch (DalException x)
      {
        lblError.Text = writeError(x);
        return;
      }
    }

    private void SetValues(Department department, CrudType type)
    {
      if (tbxCourse_Department_ID.Text.Length > 0)
        department.Course_Department_ID.Value = new Guid(tbxCourse_Department_ID.Text);
      else if (type.Equals(CrudType.Create))
        department.Course_Department_ID.Value = Guid.NewGuid();

      if (tbxCourse_ID.Text.Length > 0)
        department.Course_ID.Value = new Guid(tbxCourse_ID.Text);

      if (tbxDepartment_ID.Text.Length > 0)
        department.Department_ID.Value = Convert.ToInt32(tbxDepartment_ID.Text);

      //Log
      if (tbxCreated.Text.Length > 0)
        department.Log.Created.Value = Convert.ToDateTime(tbxCreated.Text);
      else if (type.Equals(CrudType.Create))
        department.Log.Created.Value = DateTime.Now;

      if (tbxCreatedBy.Text.Length > 0)
        department.Log.CreatedBy.Value = new Guid(tbxCreatedBy.Text);
      else if (type.Equals(CrudType.Create))
        department.Log.CreatedBy.Value = Guid.NewGuid();

      if (tbxUpdated.Text.Length > 0)
        department.Log.Updated.Value = Convert.ToDateTime(tbxUpdated.Text);
      else if (type.Equals(CrudType.Create))
        department.Log.Updated.Value = DateTime.Now;

      if (tbxUpdatedBy.Text.Length > 0)
        department.Log.UpdatedBy.Value = new Guid(tbxUpdatedBy.Text);
      else if (type.Equals(CrudType.Create))
        department.Log.UpdatedBy.Value = Guid.NewGuid();
    }

    private enum CrudType { Create, Update }

    protected void btnReset_Click(object sender, System.EventArgs e)
    {
      lblError.Text = "";

      tbxCourse_Department_ID.Text = "";
      tbxCourse_ID.Text = "";
      tbxDepartment_ID.Text = "";

      tbxCreated.Text = "";
      tbxCreatedBy.Text = "";
      tbxUpdated.Text = "";
      tbxUpdatedBy.Text = "";
    }

  }
}
```

2.1.5 EvaluationForm

evaluationform.aspx

```
<%@ Page language="c#" Src="EvaluationForm.aspx.cs" CodeBehind="EvaluationForm.aspx.cs" AutoEventWireup="false" Inherits="
 StudyPlanning.DAL.Courses.Test.EvaluationFormTest" trace="false" warningLevel="4" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" >
<HTML>
  <HEAD>
    <title>StudyType</title>
    <meta content="True" name="vs_showGrid">
    <meta content="Microsoft Visual Studio 7.0" name="GENERATOR">
    <meta content="C#" name="CODE_LANGUAGE">
    <meta content="JavaScript" name="vs_defaultClientScript">
    <meta content="http://schemas.microsoft.com/intellisense/ie5" name="vs_targetSchema">
```

```
    <LINK href="../../../misc/style.css" rel="stylesheet">
    <style>.textbox { MARGIN−TOP: 0px; DISPLAY: inline; WIDTH: 400px }
  .button { WIDTH: 120px }
    </style>
  </HEAD>
  <body>
    <br>
    <asp:Label id="lblError" ForeColor="MediumBlue" Runat="server"></asp:Label>
    <p><STRONG><FONT size="4">StudyType</FONT></STRONG></p>
    <form id="form" method="post" runat="server">
      <TABLE id="Table2" cellSpacing="1" cellPadding="1" align="center" border="1">
        <TR>
          <TD align="middle" valign="top">
            <br>
            <p><asp:button id="btnCreate" onclick="btnCreate_Click" runat="server" CssClass="button" Text="Create"></asp:button>
              </p>
            <p><asp:button id="btnRetrieve" onclick="btnRetrieve_Click" runat="server" CssClass="button" Text="Retrieve"></asp:
              button></p>
            <p><asp:button id="btnUpdate" onclick="btnUpdate_Click" runat="server" CssClass="button" Text="Update"></asp:button>
              </p>
            <p><asp:button id="btnDelete" onclick="btnDelete_Click" runat="server" CssClass="button" Text="Delete"></asp:button>
              </p>
            <p><asp:button id="btnReset" onclick="btnReset_Click" runat="server" CssClass="button" Text="Reset"></asp:button></
              p>
          </TD>
          <TD align="middle">
            <TABLE id="Table1" cellSpacing="1" cellPadding="1" border="1">
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Course_EvaluationForm_ID</STRONG></TD>
                <TD><STRONG><asp:textbox id="tbxCourse_EvaluationForm_ID" runat="server" CssClass="textbox"></asp:textbox><
                  /STRONG></TD>
                <TD></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>CourseVersion_ID</STRONG></TD>
                <TD><asp:textbox id="tbxCourseVersion_ID" runat="server" CssClass="textbox" Rows="1"></asp:textbox></TD>
                <TD></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>EvaluationForm_ID</STRONG></TD>
                <TD><STRONG><asp:textbox id="tbxEvaluationForm_ID" runat="server" CssClass="textbox" Rows="1"></asp:
                  textbox></STRONG></TD>
                <TD></TD>
              </TR>
              <tr>
              </tr>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Created</STRONG></TD>
                <TD><asp:textbox id="tbxCreated" runat="server" CssClass="textbox"></asp:textbox></TD>
                <TD></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>CreatedBy</STRONG></TD>
                <TD><asp:textbox id="tbxCreatedBy" runat="server" CssClass="textbox"></asp:textbox></TD>
                <TD></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Updated</STRONG></TD>
                <TD width="50%">
                  <asp:TextBox id="tbxUpdated" runat="server" CssClass="textbox"></asp:TextBox></TD>
                <TD width="50%"></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>UpdatedBy</STRONG></TD>
                <TD>
                  <asp:TextBox id="tbxUpdatedBy" runat="server" CssClass="textbox"></asp:TextBox></TD>
                <TD></TD>
              </TR>
            </TABLE>
          </TD>
        </TR>
      </TABLE>
    </form>
    <P> </P>
    </FORM>
  </body>
</HTML>
```

evaluationform.aspx.cs

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data.SqlClient;
using System.Data;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using System.Text;
using System.Web.Security;
using StudyPlanning.DAL;
using StudyPlanning.DAL.Courses;

namespace StudyPlanning.DAL.Courses.Test
```

```
{
  /// <summary>
  /// Summary description for forgotPassword.
  /// </summary>
  public class EvaluationFormTest : System.Web.UI.Page
  {
    protected System.Web.UI.WebControls.TextBox tbxCreated;
    protected System.Web.UI.WebControls.TextBox tbxCreatedBy;
    protected System.Web.UI.WebControls.TextBox tbxUpdated;
    protected System.Web.UI.WebControls.TextBox tbxUpdatedBy;
    protected System.Web.UI.WebControls.Button btnCreate;
    protected System.Web.UI.WebControls.Button btnRetrieve;
    protected System.Web.UI.WebControls.Button btnUpdate;
    protected System.Web.UI.WebControls.Button btnDelete;
    protected System.Web.UI.WebControls.Button btnReset;
    protected System.Web.UI.WebControls.TextBox tbxCourseVersion_ID;
    protected System.Web.UI.WebControls.TextBox tbxCourse_EvaluationForm_ID;
    protected System.Web.UI.WebControls.TextBox tbxEvaluationForm_ID;
    protected System.Web.UI.WebControls.Label lblError;

    private void Page_Load(object sender, System.EventArgs e)
    {

    }

    private void InitializeComponent()
    {
      this.Load += new System.EventHandler(this.Page_Load);

    }

    private string writeError(DalException x)
    {
      string strError = "";
      strError += x.ToString();

      if (x.Number == 16)
        strError += "<br/><br/>" + x.Sql.ToString();

      return strError;
    }

    protected void btnCreate_Click(object sender, System.EventArgs e)
    {
      lblError.Text = "";

      EvaluationForm ef = new EvaluationForm();
      SetValues(ef, CrudType.Create);

      try
      {
        ef.Create();
        tbxCourse_EvaluationForm_ID.Text = ef.Course_EvaluationForm_ID.Value.ToString();
      }
      catch (DalException x)
      {
        lblError.Text = writeError(x);
        return;
      }
    }

    protected void btnRetrieve_Click(object sender, System.EventArgs e)
    {
      lblError.Text = "";

      EvaluationForm ef = new EvaluationForm();
      ef.Course_EvaluationForm_ID.Value = new Guid(tbxCourse_EvaluationForm_ID.Text);

      try
      {
        ef.Retrieve();
      }
      catch (DalException x)
      {
        lblError.Text = writeError(x);
        return;
      }

      tbxCourse_EvaluationForm_ID.Text = ef.Course_EvaluationForm_ID.Value.ToString();
      tbxCourseVersion_ID.Text = ef.CourseVersion_ID.Value.ToString();
      tbxEvaluationForm_ID.Text = ef.EvaluationForm_ID.Value.ToString();

      tbxCreated.Text  = ef.Log.Created.Value.ToString();
      tbxCreatedBy.Text = ef.Log.CreatedBy.Value.ToString();
      tbxUpdated.Text  = ef.Log.Updated.Value.ToString();
      tbxUpdatedBy.Text = ef.Log.UpdatedBy.Value.ToString();
    }

    protected void btnUpdate_Click(object sender, System.EventArgs e)
    {
      EvaluationForm orgEf = new EvaluationForm();
      orgEf.Course_EvaluationForm_ID.Value = new Guid(tbxCourse_EvaluationForm_ID.Text);

      EvaluationForm ef = new EvaluationForm();
      SetValues(ef, CrudType.Update);

      try
      {
        orgEf.Retrieve();
```

```
        ef.Update(orgEf);
      }
      catch (DalException x)
      {
        lblError.Text = writeError(x);
        return;
      }
    }

    protected void btnDelete_Click(object sender, System.EventArgs e)
    {
      lblError.Text = "";

      EvaluationForm ef = new EvaluationForm();
      ef.Course_EvaluationForm_ID.Value = new Guid(tbxCourse_EvaluationForm_ID.Text);

      try
      {
        ef.Retrieve();
        ef.Delete();
      }
      catch (DalException x)
      {
        lblError.Text = writeError(x);
        return;
      }
    }

    private void SetValues(EvaluationForm evalForm, CrudType type)
    {
      if (tbxCourse_EvaluationForm_ID.Text.Length > 0)
        evalForm.Course_EvaluationForm_ID.Value = new Guid(tbxCourse_EvaluationForm_ID.Text);
      else if (type.Equals(CrudType.Create))
        evalForm.Course_EvaluationForm_ID.Value = Guid.NewGuid();

      if (tbxCourseVersion_ID.Text.Length > 0)
        evalForm.CourseVersion_ID.Value = new Guid(tbxCourseVersion_ID.Text);
      else if (type.Equals(CrudType.Create))
        evalForm.CourseVersion_ID.Value = Guid.NewGuid();

      if (tbxEvaluationForm_ID.Text.Length > 0)
        evalForm.EvaluationForm_ID.Value = Convert.ToInt32(tbxEvaluationForm_ID.Text);

      //Log
      if (tbxCreated.Text.Length > 0)
        evalForm.Log.Created.Value = Convert.ToDateTime(tbxCreated.Text);
      else if (type.Equals(CrudType.Create))
        evalForm.Log.Created.Value = DateTime.Now;

      if (tbxCreatedBy.Text.Length > 0)
        evalForm.Log.CreatedBy.Value = new Guid(tbxCreatedBy.Text);
      else if (type.Equals(CrudType.Create))
        evalForm.Log.CreatedBy.Value = Guid.NewGuid();

      if (tbxUpdated.Text.Length > 0)
        evalForm.Log.Updated.Value = Convert.ToDateTime(tbxUpdated.Text);
      else if (type.Equals(CrudType.Create))
        evalForm.Log.Updated.Value = DateTime.Now;

      if (tbxUpdatedBy.Text.Length > 0)
        evalForm.Log.UpdatedBy.Value = new Guid(tbxUpdatedBy.Text);
      else if (type.Equals(CrudType.Create))
        evalForm.Log.UpdatedBy.Value = Guid.NewGuid();
    }

    private enum CrudType { Create, Update }

    protected void btnReset_Click(object sender, System.EventArgs e)
    {
      lblError.Text = "";

      tbxCourse_EvaluationForm_ID.Text = "";
      tbxCourseVersion_ID.Text = "";
      tbxEvaluationForm_ID.Text = "";

      tbxCreated.Text = "";
      tbxCreatedBy.Text = "";
      tbxUpdated.Text = "";
      tbxUpdatedBy.Text = "";
    }

  }
}
```

2.1.6 Keyword

keyword.aspx

```
<%@ Page language="c#" Src="Keyword.aspx.cs" CodeBehind="Keyword.aspx.cs" AutoEventWireup="false" Inherits="StudyPlanning.DAL.
  Courses.Test.KeywordTest" trace="false" warningLevel="4" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" >
<HTML>
  <HEAD>
```

```
    <title>Keyword</title>
    <meta content="True" name="vs_showGrid">
    <meta content="Microsoft Visual Studio 7.0" name="GENERATOR">
    <meta content="C#" name="CODE_LANGUAGE">
    <meta content="JavaScript" name="vs_defaultClientScript">
    <meta content="http://schemas.microsoft.com/intellisense/ie5" name="vs_targetSchema">
    <LINK href="../../../misc/style.css" rel="stylesheet">
    <style>.textbox { MARGIN-TOP: 0px; DISPLAY: inline; WIDTH: 400px }
.button { WIDTH: 120px }
    </style>
  </HEAD>
  <body>
    <br>
    <asp:Label id="lblError" ForeColor="MediumBlue" Runat="server"></asp:Label>
    <p><STRONG><FONT size="4">Keyword</FONT></STRONG></p>
    <form id="form" method="post" runat="server">
      <TABLE id="Table2" cellSpacing="1" cellPadding="1" align="center" border="1">
        <TR>
          <TD align="middle" valign="top">
            <br>
            <p><asp:button id="btnCreate" onclick="btnCreate_Click" runat="server" CssClass="button" Text="Create"></asp:button>
              </p>
            <p><asp:button id="btnRetrieve" onclick="btnRetrieve_Click" runat="server" CssClass="button" Text="Retrieve"></asp:
              button></p>
            <p><asp:button id="btnUpdate" onclick="btnUpdate_Click" runat="server" CssClass="button" Text="Update"></asp:button>
              </p>
            <p><asp:button id="btnDelete" onclick="btnDelete_Click" runat="server" CssClass="button" Text="Delete"></asp:button>
              </p>
            <p><asp:button id="btnReset" onclick="btnReset_Click" runat="server" CssClass="button" Text="Reset"></asp:button></
              p>
          </TD>
          <TD align="middle">
            <TABLE id="Table1" cellSpacing="1" cellPadding="1" border="1">
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Course_Keyword_ID</STRONG></TD>
                <TD><STRONG><asp:textbox id="tbxCourse_Keyword_ID" runat="server" CssClass="textbox"></asp:textbox></
                  STRONG></TD>
                <TD></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>CourseVersion_ID</STRONG></TD>
                <TD><asp:textbox id="tbxCourseVersion_ID" runat="server" CssClass="textbox" Rows="1"></asp:textbox></TD>
                <TD></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Keyword_ID</STRONG></TD>
                <TD><STRONG><asp:textbox id="tbxKeyword_ID" runat="server" CssClass="textbox" Rows="1"></asp:textbox></
                  STRONG></TD>
                <TD></TD>
              </TR>
              <tr>
              </tr>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Created</STRONG></TD>
                <TD><asp:textbox id="tbxCreated" runat="server" CssClass="textbox"></asp:textbox></TD>
                <TD></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>CreatedBy</STRONG></TD>
                <TD><asp:textbox id="tbxCreatedBy" runat="server" CssClass="textbox"></asp:textbox></TD>
                <TD></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Updated</STRONG></TD>
                <TD width="50%">
                  <asp:TextBox id="tbxUpdated" runat="server" CssClass="textbox"></asp:TextBox></TD>
                <TD width="50%"></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>UpdatedBy</STRONG></TD>
                <TD>
                  <asp:TextBox id="tbxUpdatedBy" runat="server" CssClass="textbox"></asp:TextBox></TD>
                <TD></TD>
              </TR>
            </TABLE>
          </TD>
        </TR>
      </TABLE>
    </form>
    <P> </P>
    </FORM>
  </body>
</HTML>
```

keyword.aspx.cs

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data.SqlClient;
using System.Data;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
```

```
using System.Text;
using System.Web.Security;
using StudyPlanning.DAL;
using StudyPlanning.DAL.Courses;

namespace StudyPlanning.DAL.Courses.Test
{
  /// <summary>
  /// Test class for the <see cref="StudyPlanning.DAL.Courses.Keyword"/> class.
  /// </summary>
  public class KeywordTest : System.Web.UI.Page
  {
    protected System.Web.UI.WebControls.TextBox tbxCreated;
    protected System.Web.UI.WebControls.TextBox tbxCreatedBy;
    protected System.Web.UI.WebControls.TextBox tbxUpdated;
    protected System.Web.UI.WebControls.TextBox tbxUpdatedBy;
    protected System.Web.UI.WebControls.Button btnCreate;
    protected System.Web.UI.WebControls.Button btnRetrieve;
    protected System.Web.UI.WebControls.Button btnUpdate;
    protected System.Web.UI.WebControls.Button btnDelete;
    protected System.Web.UI.WebControls.Button btnReset;
    protected System.Web.UI.WebControls.TextBox tbxCourseVersion_ID;
    protected System.Web.UI.WebControls.TextBox tbxCourse_Keyword_ID;
    protected System.Web.UI.WebControls.TextBox tbxKeyword_ID;
    protected System.Web.UI.WebControls.Label lblError;

    private void Page_Load(object sender, System.EventArgs e)
    {

    }

    private void InitializeComponent()
    {
      this.Load += new System.EventHandler(this.Page_Load);

    }

    private string writeError(DalException x)
    {
      string strError = "";
      strError += x.ToString();

      if (x.Number == 16)
        strError += "<br/><br/>" + x.Sql.ToString();

      return strError;
    }

    protected void btnCreate_Click(object sender, System.EventArgs e)
    {
      lblError.Text = "";

      Keyword kw = new Keyword();
      SetValues(kw, CrudType.Create);

      try
      {
        kw.Create();
        tbxCourse_Keyword_ID.Text = kw.Course_Keyword_ID.Value.ToString();
      }
      catch (DalException x)
      {
        lblError.Text = writeError(x);
        return;
      }
    }

    protected void btnRetrieve_Click(object sender, System.EventArgs e)
    {
      lblError.Text = "";

      Keyword kw = new Keyword();
      kw.Course_Keyword_ID.Value = new Guid(tbxCourse_Keyword_ID.Text);

      try
      {
        kw.Retrieve();
      }
      catch (DalException x)
      {
        lblError.Text = writeError(x);
        return;
      }

      tbxCourse_Keyword_ID.Text = kw.Course_Keyword_ID.Value.ToString();
      tbxCourseVersion_ID.Text = kw.CourseVersion_ID.Value.ToString();
      tbxKeyword_ID.Text = kw.Keyword_ID.Value.ToString();

      tbxCreated.Text  = kw.Log.Created.Value.ToString();
      tbxCreatedBy.Text = kw.Log.CreatedBy.Value.ToString();
      tbxUpdated.Text  = kw.Log.Updated.Value.ToString();
      tbxUpdatedBy.Text = kw.Log.UpdatedBy.Value.ToString();
    }

    protected void btnUpdate_Click(object sender, System.EventArgs e)
    {
      Keyword orgKw = new Keyword();
      orgKw.Course_Keyword_ID.Value = new Guid(tbxCourse_Keyword_ID.Text);
```

```
      Keyword kw = new Keyword();
      SetValues(kw, CrudType.Update);

      try
      {
        orgKw.Retrieve();
        kw.Update(orgKw);
      }
      catch (DalException x)
      {
        lblError.Text = writeError(x);
        return;
      }
    }

    protected void btnDelete_Click(object sender, System.EventArgs e)
    {
      lblError.Text = "";

      Keyword kw = new Keyword();
      kw.Course_Keyword_ID.Value = new Guid(tbxCourse_Keyword_ID.Text);

      try
      {
        kw.Retrieve();
        kw.Delete();
      }
      catch (DalException x)
      {
        lblError.Text = writeError(x);
        return;
      }
    }

    private void SetValues(Keyword keyword, CrudType type)
    {
      if (tbxCourse_Keyword_ID.Text.Length > 0)
        keyword.Course_Keyword_ID.Value = new Guid(tbxCourse_Keyword_ID.Text);
      else if (type.Equals(CrudType.Create))
        keyword.Course_Keyword_ID.Value = Guid.NewGuid();

      keyword.CourseVersion_ID.Value = new Guid(tbxCourseVersion_ID.Text);
      keyword.Keyword_ID.Value = new Guid(tbxKeyword_ID.Text);

      //Log
      if (tbxCreated.Text.Length > 0)
        keyword.Log.Created.Value = Convert.ToDateTime(tbxCreated.Text);
      else if (type.Equals(CrudType.Create))
        keyword.Log.Created.Value = DateTime.Now;

      if (tbxCreatedBy.Text.Length > 0)
        keyword.Log.CreatedBy.Value = new Guid(tbxCreatedBy.Text);
      else if (type.Equals(CrudType.Create))
        keyword.Log.CreatedBy.Value = Guid.NewGuid();

      if (tbxUpdated.Text.Length > 0)
        keyword.Log.Updated.Value = Convert.ToDateTime(tbxUpdated.Text);
      else if (type.Equals(CrudType.Create))
        keyword.Log.Updated.Value = DateTime.Now;

      if (tbxUpdatedBy.Text.Length > 0)
        keyword.Log.UpdatedBy.Value = new Guid(tbxUpdatedBy.Text);
      else if (type.Equals(CrudType.Create))
        keyword.Log.UpdatedBy.Value = Guid.NewGuid();
    }

    private enum CrudType { Create, Update }

    protected void btnReset_Click(object sender, System.EventArgs e)
    {
      lblError.Text = "";

      tbxCourse_Keyword_ID.Text = "";
      tbxCourseVersion_ID.Text = "";
      tbxKeyword_ID.Text = "";

      tbxCreated.Text = "";
      tbxCreatedBy.Text = "";
      tbxUpdated.Text = "";
      tbxUpdatedBy.Text = "";
    }

  }
}
```

2.1.7 Lecturer

lecturer.aspx

```
<%@ Page language="c#" Src="Lecturer.aspx.cs" CodeBehind="Lecturer.aspx.cs" AutoEventWireup="false" Inherits="StudyPlanning.DAL.
 Courses.Test.LecturerTest" trace="false" warningLevel="4" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" >
<HTML>
```

```
<HEAD>
  <title>Lecturer</title>
  <meta content="True" name="vs_showGrid">
  <meta content="Microsoft Visual Studio 7.0" name="GENERATOR">
  <meta content="C#" name="CODE_LANGUAGE">
  <meta content="JavaScript" name="vs_defaultClientScript">
  <meta content="http://schemas.microsoft.com/intellisense/ie5" name="vs_targetSchema">
  <LINK href="../../../misc/style.css" rel="stylesheet">
  <style>.textbox { MARGIN-TOP: 0px; DISPLAY: inline; WIDTH: 400px }
.button { WIDTH: 120px }
  </style>
</HEAD>
<body>
  <br>
  <asp:Label id="lblError" ForeColor="MediumBlue" Runat="server"></asp:Label>
  <p><STRONG><FONT size="4">Lecturer</FONT></STRONG></p>
  <form id="form" method="post" runat="server">
    <TABLE id="Table2" cellSpacing="1" cellPadding="1" align="center" border="1">
      <TR>
        <TD align="middle" valign="top">
          <br>
          <p><asp:button id="btnCreate" onclick="btnCreate_Click" runat="server" CssClass="button" Text="Create"></asp:button>
            </p>
          <p><asp:button id="btnRetrieve" onclick="btnRetrieve_Click" runat="server" CssClass="button" Text="Retrieve"></asp:
            button></p>
          <p><asp:button id="btnUpdate" onclick="btnUpdate_Click" runat="server" CssClass="button" Text="Update"></asp:button>
            </p>
          <p><asp:button id="btnDelete" onclick="btnDelete_Click" runat="server" CssClass="button" Text="Delete"></asp:button>
            </p>
          <p><asp:button id="btnReset" onclick="btnReset_Click" runat="server" CssClass="button" Text="Reset"></asp:button></
            p>
        </TD>
        <TD align="middle">
          <TABLE id="Table1" cellSpacing="1" cellPadding="1" border="1">
            <TR>
              <TD style="WIDTH: 138px"><STRONG>Course_Lecturer_ID</STRONG></TD>
              <TD><STRONG><asp:textbox id="tbxCourse_Lecturer_ID" runat="server" CssClass="textbox"></asp:textbox></
                STRONG></TD>
              <TD></TD>
            </TR>
            <TR>
              <TD style="WIDTH: 138px"><STRONG>CourseVersion_ID</STRONG></TD>
              <TD><asp:textbox id="tbxCourseVersion_ID" runat="server" CssClass="textbox" Rows="1"></asp:textbox></TD>
              <TD></TD>
            </TR>
            <TR>
              <TD style="WIDTH: 138px"><STRONG>Lecturer_ID</STRONG></TD>
              <TD><STRONG><asp:textbox id="tbxLecturer_ID" runat="server" CssClass="textbox" Rows="1"></asp:textbox><
                /STRONG></TD>
              <TD></TD>
            </TR>
            <tr>
            </tr>
            <TR>
              <TD style="WIDTH: 138px"><STRONG>Created</STRONG></TD>
              <TD><asp:textbox id="tbxCreated" runat="server" CssClass="textbox"></asp:textbox></TD>
              <TD></TD>
            </TR>
            <TR>
              <TD style="WIDTH: 138px"><STRONG>CreatedBy</STRONG></TD>
              <TD><asp:textbox id="tbxCreatedBy" runat="server" CssClass="textbox"></asp:textbox></TD>
              <TD></TD>
            </TR>
            <TR>
              <TD style="WIDTH: 138px"><STRONG>Updated</STRONG></TD>
              <TD width="50%">
                <asp:TextBox id="tbxUpdated" runat="server" CssClass="textbox"></asp:TextBox></TD>
              <TD width="50%"></TD>
            </TR>
            <TR>
              <TD style="WIDTH: 138px"><STRONG>UpdatedBy</STRONG></TD>
              <TD>
                <asp:TextBox id="tbxUpdatedBy" runat="server" CssClass="textbox"></asp:TextBox></TD>
              <TD></TD>
            </TR>
          </TABLE>
        </TD>
      </TR>
    </TABLE>
  </form>
  <P> </P>
  </FORM>
 </body>
</HTML>
```

lecturer.aspx.cs

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data.SqlClient;
using System.Data;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
```

```
using System.Web.UI.HtmlControls;
using System.Text;
using System.Web.Security;
using StudyPlanning.DAL;
using StudyPlanning.DAL.Courses;

namespace StudyPlanning.DAL.Courses.Test
{
  /// <summary>
  /// Test class for the <see cref="StudyPlanning.DAL.Courses.Lecturer"/> class.
  /// </summary>
  public class LecturerTest : System.Web.UI.Page
  {
    protected System.Web.UI.WebControls.TextBox tbxCreated;
    protected System.Web.UI.WebControls.TextBox tbxCreatedBy;
    protected System.Web.UI.WebControls.TextBox tbxUpdated;
    protected System.Web.UI.WebControls.TextBox tbxUpdatedBy;
    protected System.Web.UI.WebControls.Button btnCreate;
    protected System.Web.UI.WebControls.Button btnRetrieve;
    protected System.Web.UI.WebControls.Button btnUpdate;
    protected System.Web.UI.WebControls.Button btnDelete;
    protected System.Web.UI.WebControls.Button btnReset;
    protected System.Web.UI.WebControls.TextBox tbxCourseVersion_ID;
    protected System.Web.UI.WebControls.TextBox tbxCourse_Lecturer_ID;
    protected System.Web.UI.WebControls.TextBox tbxLecturer_ID;
    protected System.Web.UI.WebControls.Label lblError;

    private void Page_Load(object sender, System.EventArgs e)
    {

    }

    private void InitializeComponent()
    {
      this.Load += new System.EventHandler(this.Page_Load);

    }

    private string writeError(DalException x)
    {
      string strError = "";
      strError += x.ToString();

      if (x.Number == 16)
        strError += "<br/><br/>" + x.Sql.ToString();

      return strError;
    }

    protected void btnCreate_Click(object sender, System.EventArgs e)
    {
      lblError.Text = "";

      Lecturer lc = new Lecturer();
      SetValues(lc, CrudType.Create);

      try
      {
        lc.Create();
        tbxCourse_Lecturer_ID.Text = lc.Course_Lecturer_ID.Value.ToString();
      }
      catch (DalException x)
      {
        lblError.Text = writeError(x);
        return;
      }
    }

    protected void btnRetrieve_Click(object sender, System.EventArgs e)
    {
      lblError.Text = "";

      Lecturer lc = new Lecturer();
      lc.Course_Lecturer_ID.Value = new Guid(tbxCourse_Lecturer_ID.Text);

      try
      {
        lc.Retrieve();
      }
      catch (DalException x)
      {
        lblError.Text = writeError(x);
        return;
      }

      tbxCourse_Lecturer_ID.Text = lc.Course_Lecturer_ID.Value.ToString();
      tbxCourseVersion_ID.Text = lc.CourseVersion_ID.Value.ToString();
      tbxLecturer_ID.Text = lc.Lecturer_ID.Value.ToString();

      tbxCreated.Text  = lc.Log.Created.Value.ToString();
      tbxCreatedBy.Text = lc.Log.CreatedBy.Value.ToString();
      tbxUpdated.Text  = lc.Log.Updated.Value.ToString();
      tbxUpdatedBy.Text = lc.Log.UpdatedBy.Value.ToString();
    }

    protected void btnUpdate_Click(object sender, System.EventArgs e)
    {
      Lecturer orgLc = new Lecturer();
      orgLc.Course_Lecturer_ID.Value = new Guid(tbxCourse_Lecturer_ID.Text);
```

```
      Lecturer lc = new Lecturer();
      SetValues(lc, CrudType.Update);

      try
      {
        orgLc.Retrieve();
        lc.Update(orgLc);
      }
      catch (DalException x)
      {
        lblError.Text = writeError(x);
        return;
      }
    }

    protected void btnDelete_Click(object sender, System.EventArgs e)
    {
      lblError.Text = "";

      Lecturer lc = new Lecturer();
      lc.Course_Lecturer_ID.Value = new Guid(tbxCourse_Lecturer_ID.Text);

      try
      {
        lc.Retrieve();
        lc.Delete();
      }
      catch (DalException x)
      {
        lblError.Text = writeError(x);
        return;
      }
    }

    private void SetValues(Lecturer lecturer, CrudType type)
    {
      if (tbxCourse_Lecturer_ID.Text.Length > 0)
        lecturer.Course_Lecturer_ID.Value = new Guid(tbxCourse_Lecturer_ID.Text);
      else if (type.Equals(CrudType.Create))
        lecturer.Course_Lecturer_ID.Value = Guid.NewGuid();

      lecturer.CourseVersion_ID.Value = new Guid(tbxCourseVersion_ID.Text);
      lecturer.Lecturer_ID.Value = new Guid(tbxLecturer_ID.Text);

      //Log
      if (tbxCreated.Text.Length > 0)
        lecturer.Log.Created.Value = Convert.ToDateTime(tbxCreated.Text);
      else if (type.Equals(CrudType.Create))
        lecturer.Log.Created.Value = DateTime.Now;

      if (tbxCreatedBy.Text.Length > 0)
        lecturer.Log.CreatedBy.Value = new Guid(tbxCreatedBy.Text);
      else if (type.Equals(CrudType.Create))
        lecturer.Log.CreatedBy.Value = Guid.NewGuid();

      if (tbxUpdated.Text.Length > 0)
        lecturer.Log.Updated.Value = Convert.ToDateTime(tbxUpdated.Text);
      else if (type.Equals(CrudType.Create))
        lecturer.Log.Updated.Value = DateTime.Now;

      if (tbxUpdatedBy.Text.Length > 0)
        lecturer.Log.UpdatedBy.Value = new Guid(tbxUpdatedBy.Text);
      else if (type.Equals(CrudType.Create))
        lecturer.Log.UpdatedBy.Value = Guid.NewGuid();
    }

    private enum CrudType { Create, Update }

    protected void btnReset_Click(object sender, System.EventArgs e)
    {
      lblError.Text = "";

      tbxCourse_Lecturer_ID.Text = "";
      tbxCourseVersion_ID.Text = "";
      tbxLecturer_ID.Text = "";

      tbxCreated.Text = "";
      tbxCreatedBy.Text = "";
      tbxUpdated.Text = "";
      tbxUpdatedBy.Text = "";
    }

  }
}
```

2.1.8 Period

period.aspx

```
<%@ Page language="c#" Src="Period.aspx.cs" CodeBehind="Period.aspx.cs" AutoEventWireup="false" Inherits="StudyPlanning.DAL.
  Courses.Test.PeriodTest" trace="false" warningLevel="4" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" >
```

```
<HTML>
  <HEAD>
    <title>Period</title>
    <meta content="True" name="vs_showGrid">
    <meta content="Microsoft Visual Studio 7.0" name="GENERATOR">
    <meta content="C#" name="CODE_LANGUAGE">
    <meta content="JavaScript" name="vs_defaultClientScript">
    <meta content="http://schemas.microsoft.com/intellisense/ie5" name="vs_targetSchema">
    <LINK href="../../../misc/style.css" rel="stylesheet">
    <style>.textbox { MARGIN-TOP: 0px; DISPLAY: inline; WIDTH: 400px }
  .button { WIDTH: 120px }
    </style>
  </HEAD>
  <body>
    <br>
    <asp:Label id="lblError" ForeColor="MediumBlue" Runat="server"></asp:Label>
    <p><STRONG><FONT size="4">Period</FONT></STRONG></p>
    <form id="form" method="post" runat="server">
      <TABLE id="Table2" cellSpacing="1" cellPadding="1" align="center" border="1">
        <TR>
          <TD align="middle" valign="top">
            <br>
            <p><asp:button id="btnCreate" onclick="btnCreate_Click" runat="server" CssClass="button" Text="Create"></asp:button>
              </p>
            <p><asp:button id="btnRetrieve" onclick="btnRetrieve_Click" runat="server" CssClass="button" Text="Retrieve"></asp:
              button></p>
            <p><asp:button id="btnUpdate" onclick="btnUpdate_Click" runat="server" CssClass="button" Text="Update"></asp:button>
              </p>
            <p><asp:button id="btnDelete" onclick="btnDelete_Click" runat="server" CssClass="button" Text="Delete"></asp:button>
              </p>
            <p><asp:button id="btnReset" onclick="btnReset_Click" runat="server" CssClass="button" Text="Reset"></asp:button></
              p>
          </TD>
          <TD align="middle">
            <TABLE id="Table1" cellSpacing="1" cellPadding="1" border="1">
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Course_Period_ID</STRONG></TD>
                <TD><STRONG><asp:textbox id="tbxCourse_Period_ID" runat="server" CssClass="textbox"></asp:textbox></
                  STRONG></TD>
                <TD></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>CourseVersion_ID</STRONG></TD>
                <TD><asp:textbox id="tbxCourseVersion_ID" runat="server" CssClass="textbox" Rows="1"></asp:textbox></TD>
                <TD></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Period_ID</STRONG></TD>
                <TD><STRONG><asp:textbox id="tbxPeriod_ID" runat="server" CssClass="textbox" Rows="1"></asp:textbox></
                  STRONG></TD>
                <TD></TD>
              </TR>
              <tr>
              </tr>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Created</STRONG></TD>
                <TD><asp:textbox id="tbxCreated" runat="server" CssClass="textbox"></asp:textbox></TD>
                <TD></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>CreatedBy</STRONG></TD>
                <TD><asp:textbox id="tbxCreatedBy" runat="server" CssClass="textbox"></asp:textbox></TD>
                <TD></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Updated</STRONG></TD>
                <TD width="50%">
                  <asp:TextBox id="tbxUpdated" runat="server" CssClass="textbox"></asp:TextBox></TD>
                <TD width="50%"></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>UpdatedBy</STRONG></TD>
                <TD>
                  <asp:TextBox id="tbxUpdatedBy" runat="server" CssClass="textbox"></asp:TextBox></TD>
                <TD></TD>
              </TR>
            </TABLE>
          </TD>
        </TR>
      </TABLE>
    </form>
    <P> </P>
    </FORM>
  </body>
</HTML>
```

period.aspx.cs

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data.SqlClient;
using System.Data;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
```

```
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using System.Text;
using System.Web.Security;
using StudyPlanning.DAL;
using StudyPlanning.DAL.Courses;

namespace StudyPlanning.DAL.Courses.Test
{
  /// <summary>
  /// Summary description for forgotPassword.
  /// </summary>
  public class PeriodTest : System.Web.UI.Page
  {
    protected System.Web.UI.WebControls.TextBox tbxCreated;
    protected System.Web.UI.WebControls.TextBox tbxCreatedBy;
    protected System.Web.UI.WebControls.TextBox tbxUpdated;
    protected System.Web.UI.WebControls.TextBox tbxUpdatedBy;
    protected System.Web.UI.WebControls.Button btnCreate;
    protected System.Web.UI.WebControls.Button btnRetrieve;
    protected System.Web.UI.WebControls.Button btnUpdate;
    protected System.Web.UI.WebControls.Button btnDelete;
    protected System.Web.UI.WebControls.Button btnReset;
    protected System.Web.UI.WebControls.TextBox tbxCourseVersion_ID;
    protected System.Web.UI.WebControls.TextBox tbxCourse_Period_ID;
    protected System.Web.UI.WebControls.TextBox tbxPeriod_ID;
    protected System.Web.UI.WebControls.Label lblError;

    private void Page_Load(object sender, System.EventArgs e)
    {

    }

    private void InitializeComponent()
    {
      this.Load += new System.EventHandler(this.Page_Load);

    }

    private string writeError(DalException x)
    {
      string strError = "";
      strError += x.ToString();

      if (x.Number == 16)
        strError += "<br/><br/>" + x.Sql.ToString();

      return strError;
    }

    protected void btnCreate_Click(object sender, System.EventArgs e)
    {
      lblError.Text = "";

      Period pr = new Period();
      SetValues(pr, CrudType.Create);

      try
      {
        pr.Create();
        tbxCourse_Period_ID.Text = pr.Course_Period_ID.Value.ToString();
      }
      catch (DalException x)
      {
        lblError.Text = writeError(x);
        return;
      }
    }

    protected void btnRetrieve_Click(object sender, System.EventArgs e)
    {
      lblError.Text = "";

      Period pr = new Period();
      pr.Course_Period_ID.Value = new Guid(tbxCourse_Period_ID.Text);

      try
      {
        pr.Retrieve();
      }
      catch (DalException x)
      {
        lblError.Text = writeError(x);
        return;
      }

      tbxCourse_Period_ID.Text = pr.Course_Period_ID.Value.ToString();
      tbxCourseVersion_ID.Text = pr.CourseVersion_ID.Value.ToString();
      tbxPeriod_ID.Text = pr.Period_ID.Value.ToString();

      tbxCreated.Text  = pr.Log.Created.Value.ToString();
      tbxCreatedBy.Text = pr.Log.CreatedBy.Value.ToString();
      tbxUpdated.Text  = pr.Log.Updated.Value.ToString();
      tbxUpdatedBy.Text = pr.Log.UpdatedBy.Value.ToString();
    }

    protected void btnUpdate_Click(object sender, System.EventArgs e)
    {
      Period orgPr = new Period();
```

```
      orgPr.Course_Period_ID.Value = new Guid(tbxCourse_Period_ID.Text);

      Period pr = new Period();
      SetValues(pr, CrudType.Update);

      try
      {
        orgPr.Retrieve();
        pr.Update(orgPr);
      }
      catch (DalException x)
      {
        lblError.Text = writeError(x);
        return;
      }
    }

    protected void btnDelete_Click(object sender, System.EventArgs e)
    {
      lblError.Text = "";

      Period pr = new Period();
      pr.Course_Period_ID.Value = new Guid(tbxCourse_Period_ID.Text);

      try
      {
        pr.Retrieve();
        pr.Delete();
      }
      catch (DalException x)
      {
        lblError.Text = writeError(x);
        return;
      }
    }

    private void SetValues(Period period, CrudType type)
    {
      if (tbxCourse_Period_ID.Text.Length > 0)
        period.Course_Period_ID.Value = new Guid(tbxCourse_Period_ID.Text);
      else if (type.Equals(CrudType.Create))
        period.Course_Period_ID.Value = Guid.NewGuid();

      if (tbxCourseVersion_ID.Text.Length > 0)
        period.CourseVersion_ID.Value = new Guid(tbxCourseVersion_ID.Text);
      else if (type.Equals(CrudType.Create))
        period.CourseVersion_ID.Value = Guid.NewGuid();

      if (tbxPeriod_ID.Text.Length > 0)
        period.Period_ID.Value = new Guid(tbxPeriod_ID.Text);
      else if (type.Equals(CrudType.Create))
        period.Period_ID.Value = Guid.NewGuid();

      //Log
      if (tbxCreated.Text.Length > 0)
        period.Log.Created.Value = Convert.ToDateTime(tbxCreated.Text);
      else if (type.Equals(CrudType.Create))
        period.Log.Created.Value = DateTime.Now;

      if (tbxCreatedBy.Text.Length > 0)
        period.Log.CreatedBy.Value = new Guid(tbxCreatedBy.Text);
      else if (type.Equals(CrudType.Create))
        period.Log.CreatedBy.Value = Guid.NewGuid();

      if (tbxUpdated.Text.Length > 0)
        period.Log.Updated.Value = Convert.ToDateTime(tbxUpdated.Text);
      else if (type.Equals(CrudType.Create))
        period.Log.Updated.Value = DateTime.Now;

      if (tbxUpdatedBy.Text.Length > 0)
        period.Log.UpdatedBy.Value = new Guid(tbxUpdatedBy.Text);
      else if (type.Equals(CrudType.Create))
        period.Log.UpdatedBy.Value = Guid.NewGuid();
    }

    private enum CrudType { Create, Update }

    protected void btnReset_Click(object sender, System.EventArgs e)
    {
      lblError.Text = "";

      tbxCourse_Period_ID.Text = "";
      tbxCourseVersion_ID.Text = "";
      tbxPeriod_ID.Text = "";

      tbxCreated.Text = "";
      tbxCreatedBy.Text = "";
      tbxUpdated.Text = "";
      tbxUpdatedBy.Text = "";
    }

  }
}
```

2.1.9 PeriodModule

periodmodule.aspx

```
<%@ Page language="c#" src="PeriodModule.aspx.cs" CodeBehind="PeriodModule.aspx.cs" AutoEventWireup="false" Inherits="
 StudyPlanning.DAL.Courses.Test.ModuleTest" trace="false" warningLevel="4" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" >
<HTML>
  <HEAD>
    <title>PeriodModule</title>
    <meta content="True" name="vs_showGrid">
    <meta content="Microsoft Visual Studio 7.0" name="GENERATOR">
    <meta content="C#" name="CODE_LANGUAGE">
    <meta content="JavaScript" name="vs_defaultClientScript">
    <meta content="http://schemas.microsoft.com/intellisense/ie5" name="vs_targetSchema">
    <LINK href="../../../misc/style.css" rel="stylesheet">
    <style>.textbox { MARGIN-TOP: 0px; DISPLAY: inline; WIDTH: 400px }
  .button { WIDTH: 120px }
    </style>
  </HEAD>
  <body>
    <br>
    <asp:Label id="lblError" ForeColor="MediumBlue" Runat="server"></asp:Label>
    <p><STRONG><FONT size="4">PeriodModule</FONT></STRONG></p>
    <form id="form" method="post" runat="server">
      <TABLE id="Table2" cellSpacing="1" cellPadding="1" align="center" border="1">
        <TR>
          <TD align="middle" valign="top">
            <br>
            <p><asp:button id="btnCreate" onclick="btnCreate_Click" runat="server" CssClass="button" Text="Create"></asp:button>
              </p>
            <p><asp:button id="btnRetrieve" onclick="btnRetrieve_Click" runat="server" CssClass="button" Text="Retrieve"></asp:
              button></p>
            <p><asp:button id="btnUpdate" onclick="btnUpdate_Click" runat="server" CssClass="button" Text="Update"></asp:button>
              </p>
            <p><asp:button id="btnDelete" onclick="btnDelete_Click" runat="server" CssClass="button" Text="Delete"></asp:button>
              </p>
            <p><asp:button id="btnReset" onclick="btnReset_Click" runat="server" CssClass="button" Text="Reset"></asp:button></
              p>
          </TD>
          <TD align="middle">
            <TABLE id="Table1" cellSpacing="1" cellPadding="1" border="1">
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Course_Period_Module_ID</STRONG></TD>
                <TD><STRONG><asp:textbox id="tbxCourse_Period_Module_ID" runat="server" CssClass="textbox"></asp:textbox><
                  /STRONG></TD>
                <TD></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Course_Period_ID</STRONG></TD>
                <TD><asp:textbox id="tbxCourse_Period_ID" runat="server" CssClass="textbox" Rows="1"></asp:textbox></TD>
                <TD></TD>
              </TR>
              <tr>
              </tr>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Created</STRONG></TD>
                <TD><asp:textbox id="tbxCreated" runat="server" CssClass="textbox"></asp:textbox></TD>
                <TD></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>CreatedBy</STRONG></TD>
                <TD><asp:textbox id="tbxCreatedBy" runat="server" CssClass="textbox"></asp:textbox></TD>
                <TD></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Updated</STRONG></TD>
                <TD width="50%">
                  <asp:TextBox id="tbxUpdated" runat="server" CssClass="textbox"></asp:TextBox></TD>
                <TD width="50%"></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>UpdatedBy</STRONG></TD>
                <TD>
                  <asp:TextBox id="tbxUpdatedBy" runat="server" CssClass="textbox"></asp:TextBox></TD>
                <TD></TD>
              </TR>
            </TABLE>
          </TD>
        </TR>
      </TABLE>
    </form>
    <P> </P>
    </FORM>
  </body>
</HTML>
```

periodmodule.aspx.cs

```
using System;
using System.Collections;
using System.ComponentModel;
```

```
using System.Data.SqlClient;
using System.Data;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using System.Text;
using System.Web.Security;
using StudyPlanning.DAL;
using StudyPlanning.DAL.Courses;

namespace StudyPlanning.DAL.Courses.Test
{
  /// <summary>
  /// Summary description for forgotPassword.
  /// </summary>
  public class ModuleTest : System.Web.UI.Page
  {
    protected System.Web.UI.WebControls.TextBox tbxCreated;
    protected System.Web.UI.WebControls.TextBox tbxCreatedBy;
    protected System.Web.UI.WebControls.TextBox tbxUpdated;
    protected System.Web.UI.WebControls.TextBox tbxUpdatedBy;
    protected System.Web.UI.WebControls.Button btnCreate;
    protected System.Web.UI.WebControls.Button btnRetrieve;
    protected System.Web.UI.WebControls.Button btnUpdate;
    protected System.Web.UI.WebControls.Button btnDelete;
    protected System.Web.UI.WebControls.Button btnReset;
    protected System.Web.UI.WebControls.TextBox tbxCourse_Period_Module_ID;
    protected System.Web.UI.WebControls.TextBox tbxCourse_Period_ID;
    protected System.Web.UI.WebControls.Label lblError;

    private void Page_Load(object sender, System.EventArgs e)
    {

    }

    private void InitializeComponent()
    {
      this.Load += new System.EventHandler(this.Page_Load);

    }

    private string writeError(DalException x)
    {
      string strError = "";
      strError += x.ToString();

      if (x.Number == 16)
        strError += "<br/><br/>" + x.Sql.ToString();

      return strError;
    }

    protected void btnCreate_Click(object sender, System.EventArgs e)
    {
      lblError.Text = "";

      PeriodModule md = new PeriodModule();
      SetValues(md, CrudType.Create);

      try
      {
        md.Create();
        tbxCourse_Period_Module_ID.Text = md.Course_Period_Module_ID.Value.ToString();
      }
      catch (DalException x)
      {
        lblError.Text = writeError(x);
        return;
      }
    }

    protected void btnRetrieve_Click(object sender, System.EventArgs e)
    {
      lblError.Text = "";

      PeriodModule md = new PeriodModule();
      md.Course_Period_Module_ID.Value = new Guid(tbxCourse_Period_Module_ID.Text);

      try
      {
        md.Retrieve();
      }
      catch (DalException x)
      {
        lblError.Text = writeError(x);
        return;
      }

      tbxCourse_Period_Module_ID.Text = md.Course_Period_Module_ID.Value.ToString();
      tbxCourse_Period_ID.Text = md.Course_Period_ID.Value.ToString();

      tbxCreated.Text  = md.Log.Created.Value.ToString();
      tbxCreatedBy.Text = md.Log.CreatedBy.Value.ToString();
      tbxUpdated.Text  = md.Log.Updated.Value.ToString();
      tbxUpdatedBy.Text = md.Log.UpdatedBy.Value.ToString();
    }
```

```
    protected void btnUpdate_Click(object sender, System.EventArgs e)
    {
      PeriodModule orgMd = new PeriodModule();
      orgMd.Course_Period_Module_ID.Value = new Guid(tbxCourse_Period_Module_ID.Text);

      PeriodModule md = new PeriodModule();
      SetValues(md, CrudType.Update);

      try
      {
        orgMd.Retrieve();
        md.Update(orgMd);
      }
      catch (DalException x)
      {
        lblError.Text = writeError(x);
        return;
      }
    }

    protected void btnDelete_Click(object sender, System.EventArgs e)
    {
      lblError.Text = "";

      PeriodModule md = new PeriodModule();
      md.Course_Period_Module_ID.Value = new Guid(tbxCourse_Period_Module_ID.Text);

      try
      {
        md.Retrieve();
        md.Delete();
      }
      catch (DalException x)
      {
        lblError.Text = writeError(x);
        return;
      }
    }

    private void SetValues(PeriodModule module, CrudType type)
    {
      if (tbxCourse_Period_Module_ID.Text.Length > 0)
        module.Course_Period_Module_ID.Value = new Guid(tbxCourse_Period_Module_ID.Text);
      else if (type.Equals(CrudType.Create))
        module.Course_Period_Module_ID.Value = Guid.NewGuid();

      if (tbxCourse_Period_ID.Text.Length > 0)
        module.Course_Period_ID.Value = new Guid(tbxCourse_Period_ID.Text);
      else if (type.Equals(CrudType.Create))
        module.Course_Period_ID.Value = Guid.NewGuid();

      //Log
      if (tbxCreated.Text.Length > 0)
        module.Log.Created.Value = Convert.ToDateTime(tbxCreated.Text);
      else if (type.Equals(CrudType.Create))
        module.Log.Created.Value = DateTime.Now;

      if (tbxCreatedBy.Text.Length > 0)
        module.Log.CreatedBy.Value = new Guid(tbxCreatedBy.Text);
      else if (type.Equals(CrudType.Create))
        module.Log.CreatedBy.Value = Guid.NewGuid();

      if (tbxUpdated.Text.Length > 0)
        module.Log.Updated.Value = Convert.ToDateTime(tbxUpdated.Text);
      else if (type.Equals(CrudType.Create))
        module.Log.Updated.Value = DateTime.Now;

      if (tbxUpdatedBy.Text.Length > 0)
        module.Log.UpdatedBy.Value = new Guid(tbxUpdatedBy.Text);
      else if (type.Equals(CrudType.Create))
        module.Log.UpdatedBy.Value = Guid.NewGuid();
    }

    private enum CrudType { Create, Update }

    protected void btnReset_Click(object sender, System.EventArgs e)
    {
      lblError.Text = "";

      tbxCourse_Period_Module_ID.Text = "";
      tbxCourse_Period_ID.Text = "";

      tbxCreated.Text = "";
      tbxCreatedBy.Text = "";
      tbxUpdated.Text = "";
      tbxUpdatedBy.Text = "";
    }

  }
}
```

2.1.10 PeriodModuleItem

periodmoduleitem.aspx

```
<%@ Page language="c#" src="PeriodModuleItem.aspx.cs" CodeBehind="PeriodModuleItem.aspx.cs" AutoEventWireup="false" Inherits="
 StudyPlanning.DAL.Courses.Test.PeriodModuleItemTest" trace="false" warningLevel="4" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" >
<HTML>
  <HEAD>
    <title>PeriodModuleItem</title>
    <meta content="True" name="vs_showGrid">
    <meta content="Microsoft Visual Studio 7.0" name="GENERATOR">
    <meta content="C#" name="CODE_LANGUAGE">
    <meta content="JavaScript" name="vs_defaultClientScript">
    <meta content="http://schemas.microsoft.com/intellisense/ie5" name="vs_targetSchema">
    <LINK href="../../../misc/style.css" rel="stylesheet">
    <style>.textbox { MARGIN-TOP: 0px; DISPLAY: inline; WIDTH: 400px }
  .button { WIDTH: 120px }
    </style>
  </HEAD>
  <body>
    <br>
    <asp:Label id="lblError" ForeColor="MediumBlue" Runat="server"></asp:Label>
    <p><STRONG><FONT size="4">PeriodModuleItem</FONT></STRONG></p>
    <form id="form" method="post" runat="server">
      <TABLE id="Table2" cellSpacing="1" cellPadding="1" align="center" border="1">
        <TR>
          <TD align="middle" valign="top">
            <br>
            <p><asp:button id="btnCreate" onclick="btnCreate_Click" runat="server" CssClass="button" Text="Create"></asp:button>
              </p>
            <p><asp:button id="btnRetrieve" onclick="btnRetrieve_Click" runat="server" CssClass="button" Text="Retrieve"></asp:
              button></p>
            <p><asp:button id="btnUpdate" onclick="btnUpdate_Click" runat="server" CssClass="button" Text="Update"></asp:button>
              </p>
            <p><asp:button id="btnDelete" onclick="btnDelete_Click" runat="server" CssClass="button" Text="Delete"></asp:button>
              </p>
            <p><asp:button id="btnReset" onclick="btnReset_Click" runat="server" CssClass="button" Text="Reset"></asp:button></
              p>
          </TD>
          <TD align="middle">
            <TABLE id="Table1" cellSpacing="1" cellPadding="1" border="1">
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Course_Period_ModuleItem_ID</STRONG></TD>
                <TD><STRONG><asp:textbox id="tbxCourse_Period_ModuleItem_ID" runat="server" CssClass="textbox"></asp:textbox
                  ></STRONG></TD>
                <TD></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Course_Period_Module_ID</STRONG></TD>
                <TD><asp:textbox id="tbxCourse_Period_Module_ID" runat="server" CssClass="textbox" Rows="1"></asp:textbox></
                  TD>
                <TD></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Module_id</STRONG></TD>
                <TD>
                  <asp:TextBox id="tbxModule_ID" runat="server" CssClass="textbox"></asp:TextBox></TD>
                <TD></TD>
              </TR>
              <tr>
              </tr>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Created</STRONG></TD>
                <TD><asp:textbox id="tbxCreated" runat="server" CssClass="textbox"></asp:textbox></TD>
                <TD></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>CreatedBy</STRONG></TD>
                <TD><asp:textbox id="tbxCreatedBy" runat="server" CssClass="textbox"></asp:textbox></TD>
                <TD></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Updated</STRONG></TD>
                <TD width="50%">
                  <asp:TextBox id="tbxUpdated" runat="server" CssClass="textbox"></asp:TextBox></TD>
                <TD width="50%"></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>UpdatedBy</STRONG></TD>
                <TD>
                  <asp:TextBox id="tbxUpdatedBy" runat="server" CssClass="textbox"></asp:TextBox></TD>
                <TD></TD>
              </TR>
            </TABLE>
          </TD>
        </TR>
      </TABLE>
    </form>
    <P> </P>
    </FORM>
  </body>
</HTML>
```

periodmoduleitem.aspx.cs

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data.SqlClient;
```

```
using System.Data;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using System.Text;
using System.Web.Security;
using StudyPlanning.DAL;
using StudyPlanning.DAL.Courses;

namespace StudyPlanning.DAL.Courses.Test
{
  /// <summary>
  /// Summary description for forgotPassword.
  /// </summary>
  public class PeriodModuleItemTest : System.Web.UI.Page
  {
    protected System.Web.UI.WebControls.TextBox tbxCreated;
    protected System.Web.UI.WebControls.TextBox tbxCreatedBy;
    protected System.Web.UI.WebControls.TextBox tbxUpdated;
    protected System.Web.UI.WebControls.TextBox tbxUpdatedBy;
    protected System.Web.UI.WebControls.Button btnCreate;
    protected System.Web.UI.WebControls.Button btnRetrieve;
    protected System.Web.UI.WebControls.Button btnUpdate;
    protected System.Web.UI.WebControls.Button btnDelete;
    protected System.Web.UI.WebControls.Button btnReset;
    protected System.Web.UI.WebControls.TextBox tbxModule_ID;
    protected System.Web.UI.WebControls.TextBox tbxCourse_Period_ModuleItem_ID;
    protected System.Web.UI.WebControls.TextBox tbxCourse_Period_Module_ID;
    protected System.Web.UI.WebControls.Label lblError;

    private void Page_Load(object sender, System.EventArgs e)
    {

    }

    private void InitializeComponent()
    {
      this.Load += new System.EventHandler(this.Page_Load);

    }

    private string writeError(DalException x)
    {
      string strError = "";
      strError += x.ToString();

      if (x.Number == 16)
        strError += "<br/><br/>" + x.Sql.ToString();

      return strError;
    }

    protected void btnCreate_Click(object sender, System.EventArgs e)
    {
      lblError.Text = "";

      PeriodModuleItem md = new PeriodModuleItem();
      SetValues(md, CrudType.Create);

      try
      {
        md.Create();
        tbxCourse_Period_ModuleItem_ID.Text = md.Course_Period_ModuleItem_ID.Value.ToString();
      }
      catch (DalException x)
      {
        lblError.Text = writeError(x);
        return;
      }
    }

    protected void btnRetrieve_Click(object sender, System.EventArgs e)
    {
      lblError.Text = "";

      PeriodModuleItem md = new PeriodModuleItem();
      md.Course_Period_ModuleItem_ID.Value = new Guid(tbxCourse_Period_ModuleItem_ID.Text);

      try
      {
        md.Retrieve();
      }
      catch (DalException x)
      {
        lblError.Text = writeError(x);
        return;
      }

      tbxCourse_Period_ModuleItem_ID.Text = md.Course_Period_ModuleItem_ID.Value.ToString();
      tbxCourse_Period_Module_ID.Text = md.Course_Period_Module_ID.Value.ToString();
      tbxModule_ID.Text = md.Module_ID.Value.ToString();

      tbxCreated.Text  = md.Log.Created.Value.ToString();
      tbxCreatedBy.Text = md.Log.CreatedBy.Value.ToString();
      tbxUpdated.Text  = md.Log.Updated.Value.ToString();
      tbxUpdatedBy.Text = md.Log.UpdatedBy.Value.ToString();
    }
```

```
    protected void btnUpdate_Click(object sender, System.EventArgs e)
    {
      PeriodModuleItem orgMd = new PeriodModuleItem();
      orgMd.Course_Period_ModuleItem_ID.Value = new Guid(tbxCourse_Period_ModuleItem_ID.Text);

      PeriodModuleItem md = new PeriodModuleItem();
      SetValues(md, CrudType.Update);

      try
      {
        orgMd.Retrieve();
        md.Update(orgMd);
      }
      catch (DalException x)
      {
        lblError.Text = writeError(x);
        return;
      }
    }

    protected void btnDelete_Click(object sender, System.EventArgs e)
    {
      lblError.Text = "";

      PeriodModuleItem md = new PeriodModuleItem();
      md.Course_Period_ModuleItem_ID.Value = new Guid(tbxCourse_Period_ModuleItem_ID.Text);

      try
      {
        md.Retrieve();
        md.Delete();
      }
      catch (DalException x)
      {
        lblError.Text = writeError(x);
        return;
      }
    }

    private void SetValues(PeriodModuleItem module, CrudType type)
    {
      if (tbxCourse_Period_ModuleItem_ID.Text.Length > 0)
        module.Course_Period_ModuleItem_ID.Value = new Guid(tbxCourse_Period_ModuleItem_ID.Text);
      else if (type.Equals(CrudType.Create))
        module.Course_Period_ModuleItem_ID.Value = Guid.NewGuid();

      if (tbxCourse_Period_Module_ID.Text.Length > 0)
        module.Course_Period_Module_ID.Value = new Guid(tbxCourse_Period_Module_ID.Text);
      else if (type.Equals(CrudType.Create))
        module.Course_Period_Module_ID.Value = Guid.NewGuid();

      if (tbxModule_ID.Text.Length > 0)
        module.Module_ID.Value = Convert.ToInt32(tbxModule_ID.Text);

      //Log
      if (tbxCreated.Text.Length > 0)
        module.Log.Created.Value = Convert.ToDateTime(tbxCreated.Text);
      else if (type.Equals(CrudType.Create))
        module.Log.Created.Value = DateTime.Now;

      if (tbxCreatedBy.Text.Length > 0)
        module.Log.CreatedBy.Value = new Guid(tbxCreatedBy.Text);
      else if (type.Equals(CrudType.Create))
        module.Log.CreatedBy.Value = Guid.NewGuid();

      if (tbxUpdated.Text.Length > 0)
        module.Log.Updated.Value = Convert.ToDateTime(tbxUpdated.Text);
      else if (type.Equals(CrudType.Create))
        module.Log.Updated.Value = DateTime.Now;

      if (tbxUpdatedBy.Text.Length > 0)
        module.Log.UpdatedBy.Value = new Guid(tbxUpdatedBy.Text);
      else if (type.Equals(CrudType.Create))
        module.Log.UpdatedBy.Value = Guid.NewGuid();
    }

    private enum CrudType { Create, Update }

    protected void btnReset_Click(object sender, System.EventArgs e)
    {
      lblError.Text = "";

      tbxCourse_Period_ModuleItem_ID.Text = "";
      tbxCourse_Period_Module_ID.Text = "";
      tbxModule_ID.Text = "";

      tbxCreated.Text = "";
      tbxCreatedBy.Text = "";
      tbxUpdated.Text = "";
      tbxUpdatedBy.Text = "";
    }

  }
}
```

2.1.11 Point

point.aspx

```
<%@ Page language="c#" Src="Point.aspx.cs" CodeBehind="Point.aspx.cs" AutoEventWireup="false" Inherits="StudyPlanning.DAL.Courses.
 Test.PointTest" trace="false" warningLevel="4" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" >
<HTML>
  <HEAD>
    <title>Course: Point</title>
    <meta content="True" name="vs_showGrid">
    <meta content="Microsoft Visual Studio 7.0" name="GENERATOR">
    <meta content="C#" name="CODE_LANGUAGE">
    <meta content="JavaScript" name="vs_defaultClientScript">
    <meta content="http://schemas.microsoft.com/intellisense/ie5" name="vs_targetSchema">
    <LINK href="../../../misc/style.css" rel="stylesheet">
    <style>.textbox { MARGIN-TOP: 0px; DISPLAY: inline; WIDTH: 400px }
  .button { WIDTH: 120px }
  </style>
</HEAD>
  <body>
    <br>
    <asp:Label id="lblError" ForeColor="MediumBlue" Runat="server"></asp:Label>
    <p><STRONG><FONT size="4">Course: Point</FONT></STRONG></p>
    <form id="form" method="post" runat="server">
      <TABLE id="Table2" cellSpacing="1" cellPadding="1" align="center" border="1">
        <TR>
          <TD align="middle" valign="top">
            <br>
            <p><asp:button id="btnCreate" onclick="btnCreate_Click" runat="server" CssClass="button" Text="Create"></asp:button>
              </p>
            <p><asp:button id="btnRetrieve" onclick="btnRetrieve_Click" runat="server" CssClass="button" Text="Retrieve"></asp:
              button></p>
            <p><asp:button id="btnUpdate" onclick="btnUpdate_Click" runat="server" CssClass="button" Text="Update"></asp:button>
              </p>
            <p><asp:button id="btnDelete" onclick="btnDelete_Click" runat="server" CssClass="button" Text="Delete"></asp:button>
              </p>
            <p><asp:button id="btnReset" onclick="btnReset_Click" runat="server" CssClass="button" Text="Reset"></asp:button></
              p>
          </TD>
          <TD align="middle">
            <TABLE id="Table1" cellSpacing="1" cellPadding="1" border="1">
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Course_Point_ID</STRONG></TD>
                <TD><STRONG><asp:textbox id="tbxCourse_Point_ID" runat="server" CssClass="textbox"></asp:textbox></
                  STRONG></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>CourseVersion_ID</STRONG></TD>
                <TD><asp:textbox id="tbxCourseVersion_ID" runat="server" CssClass="textbox" Rows="1"></asp:textbox></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Part</STRONG></TD>
                <TD><STRONG><asp:textbox id="tbxPart" runat="server" CssClass="textbox" Rows="1"></asp:textbox></
                  STRONG></TD>
              </TR>
              <tr>
              </tr>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Point_ID</STRONG></TD>
                <TD>
                  <asp:textbox id="tbxPoint_ID" runat="server" CssClass="textbox" Rows="1"></asp:textbox></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Created</STRONG></TD>
                <TD><asp:textbox id="tbxCreated" runat="server" CssClass="textbox"></asp:textbox></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>CreatedBy</STRONG></TD>
                <TD><asp:textbox id="tbxCreatedBy" runat="server" CssClass="textbox"></asp:textbox></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Updated</STRONG></TD>
                <TD width="50%">
                  <asp:TextBox id="tbxUpdated" runat="server" CssClass="textbox"></asp:TextBox></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>UpdatedBy</STRONG></TD>
                <TD>
                  <asp:TextBox id="tbxUpdatedBy" runat="server" CssClass="textbox"></asp:TextBox></TD>
              </TR>
            </TABLE>
          </TD>
        </TR>
      </TABLE>
    </form>
    <P> </P></FORM>
  </body>
</HTML>
```

point.aspx.cs

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data.SqlClient;
using System.Data;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using System.Text;
using System.Web.Security;
using StudyPlanning.DAL;
using StudyPlanning.DAL.Courses;

namespace StudyPlanning.DAL.Courses.Test
{
  /// <summary>
  /// Test class for the <see cref="StudyPlanning.DAL.Courses.Point"/> class.
  /// </summary>
  public class PointTest : System.Web.UI.Page
  {
    protected System.Web.UI.WebControls.TextBox tbxCreated;
    protected System.Web.UI.WebControls.TextBox tbxCreatedBy;
    protected System.Web.UI.WebControls.TextBox tbxUpdated;
    protected System.Web.UI.WebControls.TextBox tbxUpdatedBy;
    protected System.Web.UI.WebControls.Button btnCreate;
    protected System.Web.UI.WebControls.Button btnRetrieve;
    protected System.Web.UI.WebControls.Button btnUpdate;
    protected System.Web.UI.WebControls.Button btnDelete;
    protected System.Web.UI.WebControls.Button btnReset;
    protected System.Web.UI.WebControls.TextBox tbxCourseVersion_ID;
    protected System.Web.UI.WebControls.TextBox tbxPoint_ID;
    protected System.Web.UI.WebControls.TextBox tbxCourse_Point_ID;
    protected System.Web.UI.WebControls.TextBox tbxPart;
    protected System.Web.UI.WebControls.Label lblError;

    private void Page_Load(object sender, System.EventArgs e)
    {

    }

    private void InitializeComponent()
    {
      this.Load += new System.EventHandler(this.Page_Load);

    }

    private string writeError(DalException x)
    {
      string strError = "";
      strError += x.ToString();

      if (x.Number == 16)
        strError += "<br/><br/>" + x.Sql.ToString();

      return strError;
    }

    protected void btnCreate_Click(object sender, System.EventArgs e)
    {
      lblError.Text = "";

      StudyPlanning.DAL.Courses.Point point = new StudyPlanning.DAL.Courses.Point();
      SetValues(point, CrudType.Create);

      try
      {
        point.Create();
        tbxCourse_Point_ID.Text = point.Course_Point_ID.Value.ToString();
      }
      catch (DalException x)
      {
        lblError.Text = writeError(x);
        return;
      }
    }

    protected void btnRetrieve_Click(object sender, System.EventArgs e)
    {
      lblError.Text = "";

      StudyPlanning.DAL.Courses.Point point = new StudyPlanning.DAL.Courses.Point();
      point.Course_Point_ID.Value = new Guid(tbxCourse_Point_ID.Text);

      try
      {
        point.Retrieve();
      }
      catch (DalException x)
      {
        lblError.Text = writeError(x);
        return;
      }

      tbxCourse_Point_ID.Text = point.Course_Point_ID.Value.ToString();
      tbxCourseVersion_ID.Text = point.CourseVersion_ID.Value.ToString();
      tbxPart.Text = point.Part.Value.ToString();
      tbxPoint_ID.Text = point.Point_ID.Value.ToString();
```

```
    tbxCreated.Text  = point.Log.Created.Value.ToString();
    tbxCreatedBy.Text = point.Log.CreatedBy.Value.ToString();
    tbxUpdated.Text  = point.Log.Updated.Value.ToString();
    tbxUpdatedBy.Text = point.Log.UpdatedBy.Value.ToString();
  }

  protected void btnUpdate_Click(object sender, System.EventArgs e)
  {
    StudyPlanning.DAL.Courses.Point orgPoint = new StudyPlanning.DAL.Courses.Point();
    orgPoint.Course_Point_ID.Value = new Guid(tbxCourse_Point_ID.Text);

    StudyPlanning.DAL.Courses.Point point = new StudyPlanning.DAL.Courses.Point();
    SetValues(point, CrudType.Update);

    try
    {
      orgPoint.Retrieve();
      point.Update(orgPoint);
    }
    catch (DalException x)
    {
      lblError.Text = writeError(x);
      return;
    }
  }

  protected void btnDelete_Click(object sender, System.EventArgs e)
  {
    lblError.Text = "";

    StudyPlanning.DAL.Courses.Point point = new StudyPlanning.DAL.Courses.Point();
    point.Course_Point_ID.Value = new Guid(tbxCourse_Point_ID.Text);

    try
    {
      point.Retrieve();
      point.Delete();
    }
    catch (DalException x)
    {
      lblError.Text = writeError(x);
      return;
    }
  }

  private void SetValues(StudyPlanning.DAL.Courses.Point point, CrudType type)
  {
    if (tbxCourse_Point_ID.Text.Length > 0)
      point.Course_Point_ID.Value = new Guid(tbxCourse_Point_ID.Text);
    else if (type.Equals(CrudType.Create))
      point.Course_Point_ID.Value = Guid.NewGuid();

    if (tbxCourseVersion_ID.Text.Length > 0)
      point.CourseVersion_ID.Value = new Guid(tbxCourseVersion_ID.Text);

    if (tbxPart.Text.Length > 0)
      point.Part.Value = Convert.ToInt32(tbxPart.Text);

    if (tbxPoint_ID.Text.Length > 0)
    point.Point_ID.Value = new Guid(tbxPoint_ID.Text);

    //Log
    if (tbxCreated.Text.Length > 0)
      point.Log.Created.Value = Convert.ToDateTime(tbxCreated.Text);
    else if (type.Equals(CrudType.Create))
      point.Log.Created.Value = DateTime.Now;

    if (tbxCreatedBy.Text.Length > 0)
      point.Log.CreatedBy.Value = new Guid(tbxCreatedBy.Text);
    else if (type.Equals(CrudType.Create))
      point.Log.CreatedBy.Value = Guid.NewGuid();

    if (tbxUpdated.Text.Length > 0)
      point.Log.Updated.Value = Convert.ToDateTime(tbxUpdated.Text);
    else if (type.Equals(CrudType.Create))
      point.Log.Updated.Value = DateTime.Now;

    if (tbxUpdatedBy.Text.Length > 0)
      point.Log.UpdatedBy.Value = new Guid(tbxUpdatedBy.Text);
    else if (type.Equals(CrudType.Create))
      point.Log.UpdatedBy.Value = Guid.NewGuid();
  }

  private enum CrudType { Create, Update }

  protected void btnReset_Click(object sender, System.EventArgs e)
  {
    lblError.Text = "";

    tbxCourse_Point_ID.Text = "";
    tbxCourseVersion_ID.Text = "";
    tbxPart.Text = "";
    tbxPoint_ID.Text = "";

    tbxCreated.Text = "";
    tbxCreatedBy.Text = "";
    tbxUpdated.Text = "";
    tbxUpdatedBy.Text = "";
```

```
    }
  }
}
```

2.1.12 RecommendedPlacement

recommendedplacement.aspx

```
<%@ Page language="c#" Src="RecommendedPlacement.aspx.cs" CodeBehind="RecommendedPlacement.aspx.cs" AutoEventWireup="false"
  Inherits="StudyPlanning.DAL.Courses.Test.RecommendedPlacementTest" trace="false" warningLevel="4" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" >
<HTML>
  <HEAD>
    <title>Course: RecommendePlacement</title>
    <meta content="True" name="vs_showGrid">
    <meta content="Microsoft Visual Studio 7.0" name="GENERATOR">
    <meta content="C#" name="CODE_LANGUAGE">
    <meta content="JavaScript" name="vs_defaultClientScript">
    <meta content="http://schemas.microsoft.com/intellisense/ie5" name="vs_targetSchema">
    <LINK href="../../../misc/style.css" rel="stylesheet">
    <style>.textbox { MARGIN-TOP: 0px; DISPLAY: inline; WIDTH: 400px }
  .button { WIDTH: 120px }
    </style>
  </HEAD>
  <body>
    <br>
    <asp:Label id="lblError" ForeColor="MediumBlue" Runat="server"></asp:Label>
    <p><STRONG><FONT size="4">Course: RecommendePlacement</FONT></STRONG></p>
    <form id="form" method="post" runat="server">
      <TABLE id="Table2" cellSpacing="1" cellPadding="1" align="center" border="1">
        <TR>
          <TD align="middle" valign="top">
            <br>
            <p><asp:button id="btnCreate" onclick="btnCreate_Click" runat="server" CssClass="button" Text="Create"></asp:button>
              </p>
            <p><asp:button id="btnRetrieve" onclick="btnRetrieve_Click" runat="server" CssClass="button" Text="Retrieve"></asp:
              button></p>
            <p><asp:button id="btnUpdate" onclick="btnUpdate_Click" runat="server" CssClass="button" Text="Update"></asp:button>
              </p>
            <p><asp:button id="btnDelete" onclick="btnDelete_Click" runat="server" CssClass="button" Text="Delete"></asp:button>
              </p>
            <p><asp:button id="btnReset" onclick="btnReset_Click" runat="server" CssClass="button" Text="Reset"></asp:button></
              p>
          </TD>
          <TD align="middle">
            <TABLE id="Table1" cellSpacing="1" cellPadding="1" border="1">
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Course_RecommendedPlacement_ID</STRONG></TD>
                <TD><STRONG><asp:textbox id="tbxCourse_RecommendedPlacement_ID" runat="server" CssClass="textbox"></asp:
                  textbox></STRONG></TD>
                <TD></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>CourseVersion_ID</STRONG></TD>
                <TD><asp:textbox id="tbxCourseVersion_ID" runat="server" CssClass="textbox" Rows="1"></asp:textbox></TD>
                <TD></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>StudyType_ID</STRONG></TD>
                <TD><STRONG><asp:textbox id="tbxStudyType_ID" runat="server" CssClass="textbox" Rows="1"></asp:textbox>
                  </STRONG></TD>
                <TD></TD>
              </TR>
              <tr>
              </tr>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Point_ID</STRONG></TD>
                <TD>
                  <asp:textbox id="tbxPoint_ID" runat="server" CssClass="textbox" Rows="1"></asp:textbox></TD>
                <TD>
                  <asp:CheckBox id="chkPoint_ID" runat="server"></asp:CheckBox></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>RecommendedPlacementConcept_ID</STRONG></TD>
                <TD>
                  <asp:textbox id="tbxRecommendedPlacementConcept_ID" runat="server" CssClass="textbox" Rows="1"></asp:textbox></
                    TD>
                <TD>
                  <asp:CheckBox id="chkRecommendedPlacementConcept_ID" runat="server"></asp:CheckBox></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Created</STRONG></TD>
                <TD><asp:textbox id="tbxCreated" runat="server" CssClass="textbox"></asp:textbox></TD>
                <TD></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>CreatedBy</STRONG></TD>
                <TD><asp:textbox id="tbxCreatedBy" runat="server" CssClass="textbox"></asp:textbox></TD>
                <TD></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Updated</STRONG></TD>
                <TD width="50%">
```

```
            <asp:TextBox id="tbxUpdated" runat="server" CssClass="textbox"></asp:TextBox></TD>
          <TD width="50%"></TD>
        </TR>
        <TR>
          <TD style="WIDTH: 138px"><STRONG>UpdatedBy</STRONG></TD>
          <TD>
            <asp:TextBox id="tbxUpdatedBy" runat="server" CssClass="textbox"></asp:TextBox></TD>
          <TD></TD>
        </TR>
      </TABLE>
    </TD>
  </TR>
</TABLE>
</form>
<P> </P>
</FORM>
</body>
</HTML>
```

recommendedplacement.aspx.cs

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data.SqlClient;
using System.Data;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using System.Text;
using System.Web.Security;
using StudyPlanning.DAL;
using StudyPlanning.DAL.Courses;

namespace StudyPlanning.DAL.Courses.Test
{
  /// <summary>
  /// Test class for the <see cref="StudyPlanning.DAL.Courses.RecommendedPlacement"/> class.
  /// </summary>
  public class RecommendedPlacementTest : System.Web.UI.Page
  {
    protected System.Web.UI.WebControls.TextBox tbxCreated;
    protected System.Web.UI.WebControls.TextBox tbxCreatedBy;
    protected System.Web.UI.WebControls.TextBox tbxUpdated;
    protected System.Web.UI.WebControls.TextBox tbxUpdatedBy;
    protected System.Web.UI.WebControls.Button btnCreate;
    protected System.Web.UI.WebControls.Button btnRetrieve;
    protected System.Web.UI.WebControls.Button btnUpdate;
    protected System.Web.UI.WebControls.Button btnDelete;
    protected System.Web.UI.WebControls.Button btnReset;
    protected System.Web.UI.WebControls.TextBox tbxCourseVersion_ID;
    protected System.Web.UI.WebControls.TextBox tbxStudyType_ID;
    protected System.Web.UI.WebControls.TextBox tbxPoint_ID;
    protected System.Web.UI.WebControls.TextBox tbxCourse_RecommendedPlacement_ID;
    protected System.Web.UI.WebControls.TextBox tbxRecommendedPlacementConcept_ID;
    protected System.Web.UI.WebControls.CheckBox chkPoint_ID;
    protected System.Web.UI.WebControls.CheckBox chkRecommendedPlacementConcept_ID;
    protected System.Web.UI.WebControls.Label lblError;

    private void Page_Load(object sender, System.EventArgs e)
    {

    }

    private void InitializeComponent()
    {
      this.Load += new System.EventHandler(this.Page_Load);

    }

    private string writeError(DalException x)
    {
      string strError = "";
      strError += x.ToString();

      if (x.Number == 16)
        strError += "<br/><br/>" + x.Sql.ToString();

      return strError;
    }

    protected void btnCreate_Click(object sender, System.EventArgs e)
    {
      lblError.Text = "";

      RecommendedPlacement rp = new RecommendedPlacement();
      SetValues(rp, CrudType.Create);

      try
      {
        rp.Create();
        tbxCourse_RecommendedPlacement_ID.Text = rp.Course_RecommendedPlacement_ID.Value.ToString();
      }
      catch (DalException x)
```

```
    {
      lblError.Text = writeError(x);
      return;
    }
  }

  protected void btnRetrieve_Click(object sender, System.EventArgs e)
  {
    lblError.Text = "";

    RecommendedPlacement rp = new RecommendedPlacement();
    rp.Course_RecommendedPlacement_ID.Value = new Guid(tbxCourse_RecommendedPlacement_ID.Text);

    try
    {
      rp.Retrieve();
    }
    catch (DalException x)
    {
      lblError.Text = writeError(x);
      return;
    }

    tbxCourse_RecommendedPlacement_ID.Text = rp.Course_RecommendedPlacement_ID.Value.ToString();
    tbxCourseVersion_ID.Text = rp.CourseVersion_ID.Value.ToString();
    tbxStudyType_ID.Text = rp.StudyType_ID.Value.ToString();

    if (rp.Point_ID.IsNull)
    {
      chkPoint_ID.Checked = true;
      tbxPoint_ID.Text = "";
    }
    else
    {
      chkPoint_ID.Checked = false;
      tbxPoint_ID.Text = rp.Point_ID.Value.ToString();
    }

    if (rp.RecommendedPlacementConcept_ID.IsNull)
    {
      chkRecommendedPlacementConcept_ID.Checked = true;
      tbxRecommendedPlacementConcept_ID.Text = "";
    }
    else
    {
      chkRecommendedPlacementConcept_ID.Checked = false;
      tbxRecommendedPlacementConcept_ID.Text = rp.RecommendedPlacementConcept_ID.Value.ToString();
    }

    tbxCreated.Text  = rp.Log.Created.Value.ToString();
    tbxCreatedBy.Text = rp.Log.CreatedBy.Value.ToString();
    tbxUpdated.Text  = rp.Log.Updated.Value.ToString();
    tbxUpdatedBy.Text = rp.Log.UpdatedBy.Value.ToString();
  }

  protected void btnUpdate_Click(object sender, System.EventArgs e)
  {
    RecommendedPlacement orgRp = new RecommendedPlacement();
    orgRp.Course_RecommendedPlacement_ID.Value = new Guid(tbxCourse_RecommendedPlacement_ID.Text);

    RecommendedPlacement rp = new RecommendedPlacement();
    SetValues(rp, CrudType.Update);

    try
    {
      orgRp.Retrieve();
      rp.Update(orgRp);
    }
    catch (DalException x)
    {
      lblError.Text = writeError(x);
      return;
    }
  }

  protected void btnDelete_Click(object sender, System.EventArgs e)
  {
    lblError.Text = "";

    RecommendedPlacement rp = new RecommendedPlacement();
    rp.Course_RecommendedPlacement_ID.Value = new Guid(tbxCourse_RecommendedPlacement_ID.Text);

    try
    {
      rp.Retrieve();
      rp.Delete();
    }
    catch (DalException x)
    {
      lblError.Text = writeError(x);
      return;
    }
  }

  private void SetValues(RecommendedPlacement rp, CrudType type)
  {
    if (tbxCourse_RecommendedPlacement_ID.Text.Length > 0)
      rp.Course_RecommendedPlacement_ID.Value = new Guid(tbxCourse_RecommendedPlacement_ID.Text);
    else if (type.Equals(CrudType.Create))
```

```
            rp.Course_RecommendedPlacement_ID.Value = Guid.NewGuid();

          rp.CourseVersion_ID.Value = new Guid(tbxCourseVersion_ID.Text);
          rp.StudyType_ID.Value = Convert.ToInt32(tbxStudyType_ID.Text);

          if (chkPoint_ID.Checked)
            rp.Point_ID.IsNull = true;
          else
            rp.Point_ID.Value = new Guid(tbxPoint_ID.Text);

          if (chkRecommendedPlacementConcept_ID.Checked)
            rp.RecommendedPlacementConcept_ID.IsNull = true;
          else
            rp.RecommendedPlacementConcept_ID.Value = Convert.ToInt32(tbxRecommendedPlacementConcept_ID.Text);

          //Log
          if (tbxCreated.Text.Length > 0)
            rp.Log.Created.Value = Convert.ToDateTime(tbxCreated.Text);
          else if (type.Equals(CrudType.Create))
            rp.Log.Created.Value = DateTime.Now;

          if (tbxCreatedBy.Text.Length > 0)
            rp.Log.CreatedBy.Value = new Guid(tbxCreatedBy.Text);
          else if (type.Equals(CrudType.Create))
            rp.Log.CreatedBy.Value = Guid.NewGuid();

          if (tbxUpdated.Text.Length > 0)
            rp.Log.Updated.Value = Convert.ToDateTime(tbxUpdated.Text);
          else if (type.Equals(CrudType.Create))
            rp.Log.Updated.Value = DateTime.Now;

          if (tbxUpdatedBy.Text.Length > 0)
            rp.Log.UpdatedBy.Value = new Guid(tbxUpdatedBy.Text);
          else if (type.Equals(CrudType.Create))
            rp.Log.UpdatedBy.Value = Guid.NewGuid();
        }

        private enum CrudType { Create, Update }

        protected void btnReset_Click(object sender, System.EventArgs e)
        {
          lblError.Text = "";

          tbxCourse_RecommendedPlacement_ID.Text = "";
          tbxCourseVersion_ID.Text = "";
          tbxStudyType_ID.Text = "";
          tbxPoint_ID.Text = "";
          tbxRecommendedPlacementConcept_ID.Text = "";

          chkPoint_ID.Checked = false;
          chkRecommendedPlacementConcept_ID.Checked = false;

          tbxCreated.Text = "";
          tbxCreatedBy.Text = "";
          tbxUpdated.Text = "";
          tbxUpdatedBy.Text = "";
        }

      }
    }
```

2.1.13 RelationCourse

relationcourse.aspx

```
<%@ Page language="c#" Src="RelationCourse.aspx.cs" CodeBehind="RelationCourse.aspx.cs" AutoEventWireup="false" Inherits="
  StudyPlanning.DAL.Courses.Test.RelationCourseTest" trace="false" warningLevel="4" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" >
<HTML>
  <HEAD>
    <title>Course_RelationCourse</title>
    <meta content="True" name="vs_showGrid">
    <meta content="Microsoft Visual Studio 7.0" name="GENERATOR">
    <meta content="C#" name="CODE_LANGUAGE">
    <meta content="JavaScript" name="vs_defaultClientScript">
    <meta content="http://schemas.microsoft.com/intellisense/ie5" name="vs_targetSchema">
    <LINK href="../../../misc/style.css" rel="stylesheet">
    <style>.textbox { MARGIN-TOP: 0px; DISPLAY: inline; WIDTH: 400px }
  .button { WIDTH: 120px }
    </style>
  </HEAD>
  <body>
    <br>
    <asp:Label id="lblError" ForeColor="MediumBlue" Runat="server"></asp:Label>
    <p><STRONG><FONT size="4">Course_RelationCourse</FONT></STRONG></p>
    <form id="form" method="post" runat="server">
      <TABLE id="Table2" cellSpacing="1" cellPadding="1" align="center" border="1">
        <TR>
          <TD align="middle" valign="top">
            <br>
            <p><asp:button id="btnCreate" onclick="btnCreate_Click" runat="server" CssClass="button" Text="Create"></asp:button>
              </p>
            <p><asp:button id="btnRetrieve" onclick="btnRetrieve_Click" runat="server" CssClass="button" Text="Retrieve"></asp:
              button></p>
```

```
            <p><asp:button id="btnUpdate" onclick="btnUpdate_Click" runat="server" CssClass="button" Text="Update"></asp:button>
              </p>
            <p><asp:button id="btnDelete" onclick="btnDelete_Click" runat="server" CssClass="button" Text="Delete"></asp:button>
              </p>
            <p><asp:button id="btnReset" onclick="btnReset_Click" runat="server" CssClass="button" Text="Reset"></asp:button></
              p>
          </TD>
          <TD align="middle">
            <TABLE id="Table1" cellSpacing="1" cellPadding="1" border="1">
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Course_RelationCourse_ID</STRONG></TD>
                <TD><STRONG><asp:textbox id="tbxCourse_RelationCourse_ID" runat="server" CssClass="textbox"></asp:textbox><
                  /STRONG></TD>
                <TD></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>CourseVersion_ID</STRONG></TD>
                <TD><asp:textbox id="tbxCourseVersion_ID" runat="server" CssClass="textbox" Rows="1"></asp:textbox></TD>
                <TD></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Course_RelationCourseType_ID</STRONG></TD>
                <TD><STRONG><asp:textbox id="tbxCourse_RelationCourseType_ID" runat="server" CssClass="textbox" Rows="1"></
                  asp:textbox></STRONG></TD>
                <TD></TD>
              </TR>
              <tr>
              </tr>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Created</STRONG></TD>
                <TD><asp:textbox id="tbxCreated" runat="server" CssClass="textbox"></asp:textbox></TD>
                <TD></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>CreatedBy</STRONG></TD>
                <TD><asp:textbox id="tbxCreatedBy" runat="server" CssClass="textbox"></asp:textbox></TD>
                <TD></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Updated</STRONG></TD>
                <TD width="50%">
                  <asp:TextBox id="tbxUpdated" runat="server" CssClass="textbox"></asp:TextBox></TD>
                <TD width="50%"></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>UpdatedBy</STRONG></TD>
                <TD>
                  <asp:TextBox id="tbxUpdatedBy" runat="server" CssClass="textbox"></asp:TextBox></TD>
                <TD></TD>
              </TR>
            </TABLE>
          </TD>
        </TR>
      </TABLE>
    </form>
    <P> </P>
    </FORM>
  </body>
</HTML>
```

relationcourse.aspx.cs

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data.SqlClient;
using System.Data;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using System.Text;
using System.Web.Security;
using StudyPlanning.DAL;
using StudyPlanning.DAL.Courses;

namespace StudyPlanning.DAL.Courses.Test
{
  /// <summary>
  /// Test class for the <see cref="StudyPlanning.DAL.Courses.RelationCourse"/> class.
  /// </summary>
  public class RelationCourseTest : System.Web.UI.Page
  {
    protected System.Web.UI.WebControls.TextBox tbxCreated;
    protected System.Web.UI.WebControls.TextBox tbxCreatedBy;
    protected System.Web.UI.WebControls.TextBox tbxUpdated;
    protected System.Web.UI.WebControls.TextBox tbxUpdatedBy;
    protected System.Web.UI.WebControls.Button btnCreate;
    protected System.Web.UI.WebControls.Button btnRetrieve;
    protected System.Web.UI.WebControls.Button btnUpdate;
    protected System.Web.UI.WebControls.Button btnDelete;
    protected System.Web.UI.WebControls.Button btnReset;
    protected System.Web.UI.WebControls.TextBox tbxCourseVersion_ID;
    protected System.Web.UI.WebControls.TextBox tbxCourse_RelationCourse_ID;
    protected System.Web.UI.WebControls.TextBox tbxCourse_RelationCourseType_ID;
```

```
    protected System.Web.UI.WebControls.Label lblError;

    private void Page_Load(object sender, System.EventArgs e)
    {

    }

    private void InitializeComponent()
    {
      this.Load += new System.EventHandler(this.Page_Load);

    }

    private string writeError(DalException x)
    {
      string strError = "";
      strError += x.ToString();

      if (x.Number == 16)
        strError += "<br/><br/>" + x.Sql.ToString();

      return strError;
    }

    protected void btnCreate_Click(object sender, System.EventArgs e)
    {
      lblError.Text = "Not implemented";
    }

    protected void btnRetrieve_Click(object sender, System.EventArgs e)
    {
      lblError.Text = "";

      RelationCourse rc = new RelationCourse();
      rc.Course_RelationCourse_ID.Value = new Guid(tbxCourse_RelationCourse_ID.Text);

      try
      {
        rc.Retrieve();
      }
      catch (DalException x)
      {
        lblError.Text = writeError(x);
        return;
      }

      tbxCourse_RelationCourse_ID.Text = rc.Course_RelationCourse_ID.Value.ToString();
      tbxCourseVersion_ID.Text = rc.CourseVersion_ID.Value.ToString();
      tbxCourse_RelationCourseType_ID.Text = rc.Course_RelationCourseType_ID.Value.ToString();

      tbxCreated.Text  = rc.Log.Created.Value.ToString();
      tbxCreatedBy.Text = rc.Log.CreatedBy.Value.ToString();
      tbxUpdated.Text  = rc.Log.Updated.Value.ToString();
      tbxUpdatedBy.Text = rc.Log.UpdatedBy.Value.ToString();
    }

    protected void btnUpdate_Click(object sender, System.EventArgs e)
    {
      lblError.Text = "Not implemented";
    }

    protected void btnDelete_Click(object sender, System.EventArgs e)
    {
      lblError.Text = "Not implemented";
    }

    private void SetValues(RelationCourse relationCourse, CrudType type) {}

    private enum CrudType { Create, Update }

    protected void btnReset_Click(object sender, System.EventArgs e)
    {
      lblError.Text = "";

      tbxCourse_RelationCourse_ID.Text = "";
      tbxCourseVersion_ID.Text = "";
      tbxCourse_RelationCourseType_ID.Text = "";

      tbxCreated.Text = "";
      tbxCreatedBy.Text = "";
      tbxUpdated.Text = "";
      tbxUpdatedBy.Text = "";
    }

  }
}
```

2.1.14 RelationCourseItem

relationcourseitem.aspx

```
<%@ Page language="c#" Src="RelationCourseItem.aspx.cs" CodeBehind="RelationCourseItem.aspx.cs" AutoEventWireup="false" Inherits="
  StudyPlanning.DAL.Courses.Test.RelationCourseItemTest" trace="false" warningLevel="4" %>
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" >
<HTML>
  <HEAD>
    <title>Course_RelationCourseItem</title>
    <meta content="True" name="vs_showGrid">
    <meta content="Microsoft Visual Studio 7.0" name="GENERATOR">
    <meta content="C#" name="CODE_LANGUAGE">
    <meta content="JavaScript" name="vs_defaultClientScript">
    <meta content="http://schemas.microsoft.com/intellisense/ie5" name="vs_targetSchema">
    <LINK href="../../../misc/style.css" rel="stylesheet">
    <style>.textbox { MARGIN-TOP: 0px; DISPLAY: inline; WIDTH: 400px }
  .button { WIDTH: 120px }
    </style>
  </HEAD>
  <body>
    <br>
    <asp:Label id="lblError" ForeColor="MediumBlue" Runat="server"></asp:Label>
    <p><STRONG><FONT size="4">Course_RelationCourseItem</FONT></STRONG></p>
    <form id="form" method="post" runat="server">
      <TABLE id="Table2" cellSpacing="1" cellPadding="1" align="center" border="1">
        <TR>
          <TD align="middle" valign="top">
            <br>
            <p><asp:button id="btnCreate" onclick="btnCreate_Click" runat="server" CssClass="button" Text="Create"></asp:button>
              </p>
            <p><asp:button id="btnRetrieve" onclick="btnRetrieve_Click" runat="server" CssClass="button" Text="Retrieve"></asp:
              button></p>
            <p><asp:button id="btnUpdate" onclick="btnUpdate_Click" runat="server" CssClass="button" Text="Update"></asp:button>
              </p>
            <p><asp:button id="btnDelete" onclick="btnDelete_Click" runat="server" CssClass="button" Text="Delete"></asp:button>
              </p>
            <p><asp:button id="btnReset" onclick="btnReset_Click" runat="server" CssClass="button" Text="Reset"></asp:button></
              p>
          </TD>
          <TD align="middle">
            <TABLE id="Table1" cellSpacing="1" cellPadding="1" border="1">
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Course_RelationCourseItem_ID</STRONG></TD>
                <TD>
                  <asp:textbox id="tbxCourse_RelationCourseItem_ID" runat="server" CssClass="textbox"></asp:textbox></TD>
                <TD></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Course_RelationCourse_ID</STRONG></TD>
                <TD><STRONG><asp:textbox id="tbxCourse_RelationCourse_ID" runat="server" CssClass="textbox"></asp:textbox><
                  /STRONG></TD>
                <TD></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Course_ID</STRONG></TD>
                <TD><asp:textbox id="tbxCourse_ID" runat="server" CssClass="textbox" Rows="1"></asp:textbox></TD>
                <TD>
                  <asp:CheckBox id="chkCourse_ID" runat="server"></asp:CheckBox></TD>
              </TR>
              <tr>
              </tr>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>CourseNumber</STRONG></TD>
                <TD>
                  <asp:textbox id="tbxCourseNumber" runat="server" CssClass="textbox" Rows="1"></asp:textbox></TD>
                <TD>
                  <asp:CheckBox id="chkCourseNumber" runat="server"></asp:CheckBox></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Created</STRONG></TD>
                <TD><asp:textbox id="tbxCreated" runat="server" CssClass="textbox"></asp:textbox></TD>
                <TD></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>CreatedBy</STRONG></TD>
                <TD><asp:textbox id="tbxCreatedBy" runat="server" CssClass="textbox"></asp:textbox></TD>
                <TD></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Updated</STRONG></TD>
                <TD width="50%">
                  <asp:TextBox id="tbxUpdated" runat="server" CssClass="textbox"></asp:TextBox></TD>
                <TD width="50%"></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>UpdatedBy</STRONG></TD>
                <TD>
                  <asp:TextBox id="tbxUpdatedBy" runat="server" CssClass="textbox"></asp:TextBox></TD>
                <TD></TD>
              </TR>
            </TABLE>
          </TD>
        </TR>
      </TABLE>
    </form>
    <P> </P>
    </FORM>
  </body>
</HTML>
```

relationcourseitem.aspx.cs

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data.SqlClient;
using System.Data;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using System.Text;
using System.Web.Security;
using StudyPlanning.DAL;
using StudyPlanning.DAL.Courses;

namespace StudyPlanning.DAL.Courses.Test
{
  /// <summary>
  /// Test class for the <see cref="StudyPlanning.DAL.Courses.RelationCourseItem"/> class.
  /// </summary>
  public class RelationCourseItemTest : System.Web.UI.Page
  {
    protected System.Web.UI.WebControls.TextBox tbxCreated;
    protected System.Web.UI.WebControls.TextBox tbxCreatedBy;
    protected System.Web.UI.WebControls.TextBox tbxUpdated;
    protected System.Web.UI.WebControls.TextBox tbxUpdatedBy;
    protected System.Web.UI.WebControls.Button btnCreate;
    protected System.Web.UI.WebControls.Button btnRetrieve;
    protected System.Web.UI.WebControls.Button btnUpdate;
    protected System.Web.UI.WebControls.Button btnDelete;
    protected System.Web.UI.WebControls.Button btnReset;
    protected System.Web.UI.WebControls.TextBox tbxCourse_RelationCourse_ID;
    protected System.Web.UI.WebControls.TextBox tbxCourse_ID;
    protected System.Web.UI.WebControls.TextBox tbxCourse_RelationCourseItem_ID;
    protected System.Web.UI.WebControls.TextBox tbxCourseNumber;
    protected System.Web.UI.WebControls.CheckBox chkCourse_ID;
    protected System.Web.UI.WebControls.CheckBox chkCourseNumber;
    protected System.Web.UI.WebControls.Label lblError;

    private void Page_Load(object sender, System.EventArgs e)
    {

    }

    private void InitializeComponent()
    {
      this.Load += new System.EventHandler(this.Page_Load);

    }

    private string writeError(DalException x)
    {
      string strError = "";
      strError += x.ToString();

      if (x.Number == 16)
        strError += "<br/><br/>" + x.Sql.ToString();

      return strError;
    }

    protected void btnCreate_Click(object sender, System.EventArgs e)
    {
      lblError.Text = "Not implemented";
    }

    protected void btnRetrieve_Click(object sender, System.EventArgs e)
    {
      lblError.Text = "";

      RelationCourseItem rci = new RelationCourseItem();
      rci.Course_RelationCourseItem_ID.Value = new Guid(tbxCourse_RelationCourseItem_ID.Text);

      try
      {
        rci.Retrieve();
      }
      catch (DalException x)
      {
        lblError.Text = writeError(x);
        return;
      }

      tbxCourse_RelationCourseItem_ID.Text = rci.Course_RelationCourseItem_ID.Value.ToString();
      tbxCourse_RelationCourse_ID.Text = rci.Course_RelationCourse_ID.Value.ToString();

      if (rci.Course_ID.IsNull)
      {
        chkCourse_ID.Checked = true;
        tbxCourse_ID.Text = "";
      }
      else
      {
        chkCourse_ID.Checked = false;
        tbxCourse_ID.Text = rci.Course_ID.Value.ToString();
      }

      if (rci.CourseNumber.IsNull)
      {
```

```
        chkCourseNumber.Checked = true;
        tbxCourseNumber.Text = "";
      }
      else
      {
        chkCourseNumber.Checked = false;
        tbxCourseNumber.Text = rci.CourseNumber.Value.ToString();
      }

      tbxCreated.Text  = rci.Log.Created.Value.ToString();
      tbxCreatedBy.Text = rci.Log.CreatedBy.Value.ToString();
      tbxUpdated.Text  = rci.Log.Updated.Value.ToString();
      tbxUpdatedBy.Text = rci.Log.UpdatedBy.Value.ToString();
    }

    protected void btnUpdate_Click(object sender, System.EventArgs e)
    {
      lblError.Text = "Not implemented";
    }

    protected void btnDelete_Click(object sender, System.EventArgs e)
    {
      lblError.Text = "Not implemented";
    }

    private void SetValues(RelationCourseItem relationCourseItem, CrudType type) {}

    private enum CrudType { Create, Update }

    protected void btnReset_Click(object sender, System.EventArgs e)
    {
      lblError.Text = "";

      tbxCourse_RelationCourseItem_ID.Text = "";
      tbxCourse_RelationCourse_ID.Text = "";
      tbxCourse_ID.Text = "";
      tbxCourseNumber.Text = "";

      chkCourse_ID.Checked = false;
      chkCourseNumber.Checked = false;

      tbxCreated.Text = "";
      tbxCreatedBy.Text = "";
      tbxUpdated.Text = "";
      tbxUpdatedBy.Text = "";
    }

  }
}
```

2.1.15 StudyType

studytype.aspx

```
<%@ Page language="c#" Src="StudyType.aspx.cs" CodeBehind="StudyType.aspx.cs" AutoEventWireup="false" Inherits="StudyPlanning.DAL.
 Courses.Test.StudyTypeTest" trace="false" warningLevel="4" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" >
<HTML>
  <HEAD>
    <title>StudyType</title>
    <meta content="True" name="vs_showGrid">
    <meta content="Microsoft Visual Studio 7.0" name="GENERATOR">
    <meta content="C#" name="CODE_LANGUAGE">
    <meta content="JavaScript" name="vs_defaultClientScript">
    <meta content="http://schemas.microsoft.com/intellisense/ie5" name="vs_targetSchema">
    <LINK href="../../../misc/style.css" rel="stylesheet">
    <style>.textbox { MARGIN-TOP: 0px; DISPLAY: inline; WIDTH: 400px }
  .button { WIDTH: 120px }
    </style>
  </HEAD>
  <body>
    <br>
    <asp:Label id="lblError" ForeColor="MediumBlue" Runat="server"></asp:Label>
    <p><STRONG><FONT size="4">StudyType</FONT></STRONG></p>
    <form id="form" method="post" runat="server">
      <TABLE id="Table2" cellSpacing="1" cellPadding="1" align="center" border="1">
        <TR>
          <TD align="middle" valign="top">
            <br>
            <p><asp:button id="btnCreate" onclick="btnCreate_Click" runat="server" CssClass="button" Text="Create"></asp:button>
              </p>
            <p><asp:button id="btnRetrieve" onclick="btnRetrieve_Click" runat="server" CssClass="button" Text="Retrieve"></asp:
              button></p>
            <p><asp:button id="btnUpdate" onclick="btnUpdate_Click" runat="server" CssClass="button" Text="Update"></asp:button>
              </p>
            <p><asp:button id="btnDelete" onclick="btnDelete_Click" runat="server" CssClass="button" Text="Delete"></asp:button>
              </p>
            <p><asp:button id="btnReset" onclick="btnReset_Click" runat="server" CssClass="button" Text="Reset"></asp:button></
              p>
          </TD>
          <TD align="middle">
            <TABLE id="Table1" cellSpacing="1" cellPadding="1" border="1">
              <TR>
```

```
            <TD style="WIDTH: 138px"><STRONG>Course_StudyType_ID</STRONG></TD>
            <TD><STRONG><asp:textbox id="tbxCourse_StudyType_ID" runat="server" CssClass="textbox"></asp:textbox></
              STRONG></TD>
            <TD></TD>
          </TR>
          <TR>
            <TD style="WIDTH: 138px"><STRONG>CourseVersion_ID</STRONG></TD>
            <TD><asp:textbox id="tbxCourseVersion_ID" runat="server" CssClass="textbox" Rows="1"></asp:textbox></TD>
            <TD></TD>
          </TR>
          <TR>
            <TD style="WIDTH: 138px"><STRONG>StudyType_ID</STRONG></TD>
            <TD><STRONG><asp:textbox id="tbxStudyType_ID" runat="server" CssClass="textbox" Rows="1"></asp:textbox>
              </STRONG></TD>
            <TD>
              <asp:CheckBox id="chkStudyType_ID" runat="server"></asp:CheckBox></TD>
          </TR>
          <TR>
            <TD style="WIDTH: 138px"><STRONG>Course_StudyTypeCategory_ID</STRONG></TD>
            <TD><STRONG><asp:textbox id="tbxCourse_StudyTypeCategory_ID" runat="server" CssClass="textbox" Rows="1"></
              asp:textbox></STRONG></TD>
            <TD>
              <asp:CheckBox id="chkCourse_StudyTypeCategory_ID" runat="server"></asp:CheckBox></TD>
          </TR>
          <tr>
          </tr>
          <TR>
            <TD style="WIDTH: 138px"><STRONG>Created</STRONG></TD>
            <TD><asp:textbox id="tbxCreated" runat="server" CssClass="textbox"></asp:textbox></TD>
            <TD></TD>
          </TR>
          <TR>
            <TD style="WIDTH: 138px"><STRONG>CreatedBy</STRONG></TD>
            <TD><asp:textbox id="tbxCreatedBy" runat="server" CssClass="textbox"></asp:textbox></TD>
            <TD></TD>
          </TR>
          <TR>
            <TD style="WIDTH: 138px"><STRONG>Updated</STRONG></TD>
            <TD width="50%">
              <asp:TextBox id="tbxUpdated" runat="server" CssClass="textbox"></asp:TextBox></TD>
            <TD width="50%"></TD>
          </TR>
          <TR>
            <TD style="WIDTH: 138px"><STRONG>UpdatedBy</STRONG></TD>
            <TD>
              <asp:TextBox id="tbxUpdatedBy" runat="server" CssClass="textbox"></asp:TextBox></TD>
            <TD></TD>
          </TR>
        </TABLE>
      </TD>
    </TR>
  </TABLE>
 </form>
 <P> </P>
 </FORM>
</body>
</HTML>
```

studytype.aspx.cs

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data.SqlClient;
using System.Data;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using System.Text;
using System.Web.Security;
using StudyPlanning.DAL;
using StudyPlanning.DAL.Courses;

namespace StudyPlanning.DAL.Courses.Test
{
  /// <summary>
  /// Summary description for forgotPassword.
  /// </summary>
  public class StudyTypeTest : System.Web.UI.Page
  {
    protected System.Web.UI.WebControls.TextBox tbxCreated;
    protected System.Web.UI.WebControls.TextBox tbxCreatedBy;
    protected System.Web.UI.WebControls.TextBox tbxUpdated;
    protected System.Web.UI.WebControls.TextBox tbxUpdatedBy;
    protected System.Web.UI.WebControls.Button btnCreate;
    protected System.Web.UI.WebControls.Button btnRetrieve;
    protected System.Web.UI.WebControls.Button btnUpdate;
    protected System.Web.UI.WebControls.Button btnDelete;
    protected System.Web.UI.WebControls.Button btnReset;
    protected System.Web.UI.WebControls.TextBox tbxCourseVersion_ID;
    protected System.Web.UI.WebControls.TextBox tbxCourse_StudyType_ID;
    protected System.Web.UI.WebControls.TextBox tbxStudyType_ID;
    protected System.Web.UI.WebControls.TextBox tbxCourse_StudyTypeCategory_ID;
    protected System.Web.UI.WebControls.CheckBox chkStudyType_ID;
```

```
protected System.Web.UI.WebControls.CheckBox chkCourse_StudyTypeCategory_ID;
protected System.Web.UI.WebControls.Label lblError;

private void Page_Load(object sender, System.EventArgs e)
{

}

private void InitializeComponent()
{
  this.Load += new System.EventHandler(this.Page_Load);

}

private string writeError(DalException x)
{
  string strError = "";
  strError += x.ToString();

  if (x.Number == 16)
    strError += "<br/><br/>" + x.Sql.ToString();

  return strError;
}

protected void btnCreate_Click(object sender, System.EventArgs e)
{
  lblError.Text = "";

  StudyType st = new StudyType();
  SetValues(st, CrudType.Create);

  try
  {
    st.Create();
    tbxCourse_StudyType_ID.Text = st.Course_StudyType_ID.Value.ToString();
  }
  catch (DalException x)
  {
    lblError.Text = writeError(x);
    return;
  }
}

protected void btnRetrieve_Click(object sender, System.EventArgs e)
{
  lblError.Text = "";

  StudyType st = new StudyType();
  st.Course_StudyType_ID.Value = new Guid(tbxCourse_StudyType_ID.Text);

  try
  {
    st.Retrieve();
  }
  catch (DalException x)
  {
    lblError.Text = writeError(x);
    return;
  }

  tbxCourse_StudyType_ID.Text = st.Course_StudyType_ID.Value.ToString();
  tbxCourseVersion_ID.Text = st.CourseVersion_ID.Value.ToString();

  if (st.StudyType_ID.IsNull)
  {
    chkStudyType_ID.Checked = true;
    tbxStudyType_ID.Text = "";
  }
  else
  {
    chkStudyType_ID.Checked = false;
    tbxStudyType_ID.Text = st.StudyType_ID.Value.ToString();
  }

  if (st.Course_StudyTypeCategory_ID.IsNull)
  {
    chkCourse_StudyTypeCategory_ID.Checked = true;
    tbxCourse_StudyTypeCategory_ID.Text = "";
  }
  else
  {
    chkCourse_StudyTypeCategory_ID.Checked = false;
    tbxCourse_StudyTypeCategory_ID.Text = st.Course_StudyTypeCategory_ID.Value.ToString();
  }

  tbxCreated.Text  = st.Log.Created.Value.ToString();
  tbxCreatedBy.Text = st.Log.CreatedBy.Value.ToString();
  tbxUpdated.Text  = st.Log.Updated.Value.ToString();
  tbxUpdatedBy.Text = st.Log.UpdatedBy.Value.ToString();
}

protected void btnUpdate_Click(object sender, System.EventArgs e)
{
  StudyType orgSt = new StudyType();
  orgSt.Course_StudyType_ID.Value = new Guid(tbxCourse_StudyType_ID.Text);

  StudyType st = new StudyType();
  SetValues(st, CrudType.Update);
```

```
    try
    {
      orgSt.Retrieve();
      st.Update(orgSt);
    }
    catch (DalException x)
    {
      lblError.Text = writeError(x);
      return;
    }
  }

  protected void btnDelete_Click(object sender, System.EventArgs e)
  {
    lblError.Text = "";

    StudyType st = new StudyType();
    st.Course_StudyType_ID.Value = new Guid(tbxCourse_StudyType_ID.Text);

    try
    {
      st.Retrieve();
      st.Delete();
    }
    catch (DalException x)
    {
      lblError.Text = writeError(x);
      return;
    }
  }

  private void SetValues(StudyType studyType, CrudType type)
  {
    if (tbxCourse_StudyType_ID.Text.Length > 0)
      studyType.Course_StudyType_ID.Value = new Guid(tbxCourse_StudyType_ID.Text);
    else if (type.Equals(CrudType.Create))
      studyType.Course_StudyType_ID.Value = Guid.NewGuid();

    if (tbxCourseVersion_ID.Text.Length > 0)
      studyType.CourseVersion_ID.Value = new Guid(tbxCourseVersion_ID.Text);
    //else if (type.Equals(CrudType.Create))
    // studyType.CourseVersion_ID.Value = Guid.NewGuid();

    if (chkStudyType_ID.Checked)
      studyType.StudyType_ID.IsNull = true;
    else
    {
      if (tbxStudyType_ID.Text.Length > 0)
        studyType.StudyType_ID.Value = Convert.ToInt32(tbxStudyType_ID.Text);
    }

    if (chkCourse_StudyTypeCategory_ID.Checked)
      studyType.Course_StudyTypeCategory_ID.IsNull = true;
    else
    {
      if (tbxCourse_StudyTypeCategory_ID.Text.Length > 0)
        studyType.Course_StudyTypeCategory_ID.Value = Convert.ToInt32(tbxCourse_StudyTypeCategory_ID.Text);
    }

    //Log
    if (tbxCreated.Text.Length > 0)
      studyType.Log.Created.Value = Convert.ToDateTime(tbxCreated.Text);
    else if (type.Equals(CrudType.Create))
      studyType.Log.Created.Value = DateTime.Now;

    if (tbxCreatedBy.Text.Length > 0)
      studyType.Log.CreatedBy.Value = new Guid(tbxCreatedBy.Text);
    else if (type.Equals(CrudType.Create))
      studyType.Log.CreatedBy.Value = Guid.NewGuid();

    if (tbxUpdated.Text.Length > 0)
      studyType.Log.Updated.Value = Convert.ToDateTime(tbxUpdated.Text);
    else if (type.Equals(CrudType.Create))
      studyType.Log.Updated.Value = DateTime.Now;

    if (tbxUpdatedBy.Text.Length > 0)
      studyType.Log.UpdatedBy.Value = new Guid(tbxUpdatedBy.Text);
    else if (type.Equals(CrudType.Create))
      studyType.Log.UpdatedBy.Value = Guid.NewGuid();
  }

  private enum CrudType { Create, Update }

  protected void btnReset_Click(object sender, System.EventArgs e)
  {
    lblError.Text = "";

    tbxCourse_StudyType_ID.Text = "";
    tbxCourseVersion_ID.Text = "";
    tbxStudyType_ID.Text = "";
    tbxCourse_StudyTypeCategory_ID.Text = "";

    chkStudyType_ID.Checked = false;
    chkCourse_StudyTypeCategory_ID.Checked = false;

    tbxCreated.Text = "";
    tbxCreatedBy.Text = "";
    tbxUpdated.Text = "";
```

```
      tbxUpdatedBy.Text = "";
    }

  }
}
```

2.1.16 StudyTypeCategory

studytypecategory.aspx

```
<%@ Page language="c#" Src="StudyTypeCategory.aspx.cs" CodeBehind="StudyTypeCategory.aspx.cs" AutoEventWireup="false" Inherits="
 StudyPlanning.DAL.Courses.Test.StudyTypeCategoryTest" trace="false" warningLevel="4" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" >
<HTML>
  <HEAD>
    <title>StudyTypeCategory</title>
    <meta content="True" name="vs_showGrid">
    <meta content="Microsoft Visual Studio 7.0" name="GENERATOR">
    <meta content="C#" name="CODE_LANGUAGE">
    <meta content="JavaScript" name="vs_defaultClientScript">
    <meta content="http://schemas.microsoft.com/intellisense/ie5" name="vs_targetSchema">
    <LINK href="../../../misc/style.css" rel="stylesheet">
    <style>.textbox { MARGIN-TOP: 0px; DISPLAY: inline; WIDTH: 400px }
  .button { WIDTH: 120px }
    </style>
  </HEAD>
  <body>
    <br>
    <asp:Label id="lblError" ForeColor="MediumBlue" Runat="server"></asp:Label>
    <p><STRONG><FONT size="4">StudyTypeCategory</FONT></STRONG></p>
    <form id="form" method="post" runat="server">
      <TABLE id="Table2" cellSpacing="1" cellPadding="1" align="center" border="1">
        <TR>
          <TD align="middle" valign="top">
            <br>
            <p><asp:button id="btnCreate" onclick="btnCreate_Click" runat="server" CssClass="button" Text="Create"></asp:button>
              </p>
            <p><asp:button id="btnRetrieve" onclick="btnRetrieve_Click" runat="server" CssClass="button" Text="Retrieve"></asp:
              button></p>
            <p><asp:button id="btnUpdate" onclick="btnUpdate_Click" runat="server" CssClass="button" Text="Update"></asp:button>
              </p>
            <p><asp:button id="btnDelete" onclick="btnDelete_Click" runat="server" CssClass="button" Text="Delete"></asp:button>
              </p>
            <p><asp:button id="btnReset" onclick="btnReset_Click" runat="server" CssClass="button" Text="Reset"></asp:button></
              p>
          </TD>
          <TD align="middle">
            <TABLE id="Table1" cellSpacing="1" cellPadding="1" border="1">
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Course_StudyTypeCategory_ID</STRONG></TD>
                <TD><STRONG><asp:textbox id="tbxCourse_StudyTypeCategory_ID" runat="server" CssClass="textbox" Rows="1"></
                  asp:textbox></STRONG></TD>
                <TD></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>StudyType_ID</STRONG></TD>
                <TD><STRONG>
                    <asp:textbox id="tbxStudyType_ID" runat="server" CssClass="textbox" Rows="1"></asp:textbox></STRONG></
                      TD>
                <TD></TD>
              </TR>
              <tr>
              </tr>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Name</STRONG></TD>
                <TD>
                  <P>English:
                    <asp:textbox id="tbxNameEn" runat="server" CssClass="textbox"></asp:textbox></P>
                  <P>Danish:
                    <asp:textbox id="tbxNameDa" runat="server" CssClass="textbox"></asp:textbox></P>
                </TD>
                <TD></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Created</STRONG></TD>
                <TD><asp:textbox id="tbxCreated" runat="server" CssClass="textbox"></asp:textbox></TD>
                <TD></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>CreatedBy</STRONG></TD>
                <TD><asp:textbox id="tbxCreatedBy" runat="server" CssClass="textbox"></asp:textbox></TD>
                <TD></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Updated</STRONG></TD>
                <TD width="50%">
                  <asp:TextBox id="tbxUpdated" runat="server" CssClass="textbox"></asp:TextBox></TD>
                <TD width="50%"></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>UpdatedBy</STRONG></TD>
                <TD>
                  <asp:TextBox id="tbxUpdatedBy" runat="server" CssClass="textbox"></asp:TextBox></TD>
```

```
            <TD></TD>
          </TR>
        </TABLE>
      </TD>
    </TR>
  </TABLE>
</form>
<P> </P>
</FORM>
</body>
</HTML>
```

studytypecategory.aspx.cs

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data.SqlClient;
using System.Data;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using System.Text;
using System.Web.Security;
using StudyPlanning.DAL;
using StudyPlanning.DAL.Courses;

namespace StudyPlanning.DAL.Courses.Test
{
  /// <summary>
  /// Test class for the <see cref="StudyPlanning.DAL.Courses.StudyTypeCategory"/> class.
  /// </summary>
  public class StudyTypeCategoryTest : System.Web.UI.Page
  {
    protected System.Web.UI.WebControls.TextBox tbxCreated;
    protected System.Web.UI.WebControls.TextBox tbxCreatedBy;
    protected System.Web.UI.WebControls.TextBox tbxUpdated;
    protected System.Web.UI.WebControls.TextBox tbxUpdatedBy;
    protected System.Web.UI.WebControls.Button btnCreate;
    protected System.Web.UI.WebControls.Button btnRetrieve;
    protected System.Web.UI.WebControls.Button btnUpdate;
    protected System.Web.UI.WebControls.Button btnDelete;
    protected System.Web.UI.WebControls.Button btnReset;
    protected System.Web.UI.WebControls.TextBox tbxCourse_StudyTypeCategory_ID;
    protected System.Web.UI.WebControls.TextBox tbxStudyType_ID;
    protected System.Web.UI.WebControls.TextBox tbxNameEn;
    protected System.Web.UI.WebControls.TextBox tbxNameDa;
    protected System.Web.UI.WebControls.Label lblError;

    private void Page_Load(object sender, System.EventArgs e)
    {

    }

    private void InitializeComponent()
    {
      this.Load += new System.EventHandler(this.Page_Load);

    }

    private string writeError(DalException x)
    {
      string strError = "";
      strError += x.ToString();

      if (x.Number == 16)
        strError += "<br/><br/>" + x.Sql.ToString();

      return strError;
    }

    protected void btnCreate_Click(object sender, System.EventArgs e)
    {
      lblError.Text = "Not implemented";
    }

    protected void btnRetrieve_Click(object sender, System.EventArgs e)
    {
      lblError.Text = "";

      StudyTypeCategory stc = new StudyTypeCategory();
      stc.Course_StudyTypeCategory_ID.Value = Convert.ToInt32(tbxCourse_StudyTypeCategory_ID.Text);

      try
      {
        stc.Retrieve();
      }
      catch (DalException x)
      {
        lblError.Text = writeError(x);
        return;
      }

      tbxCourse_StudyTypeCategory_ID.Text = stc.Course_StudyTypeCategory_ID.Value.ToString();
```

```
      tbxStudyType_ID.Text = stc.StudyType_ID.Value.ToString();
      tbxNameEn.Text = stc.Name["en-GB"].ToString();
      tbxNameDa.Text = stc.Name["da-DK"].ToString();

      tbxCreated.Text  = stc.Log.Created.Value.ToString();
      tbxCreatedBy.Text = stc.Log.CreatedBy.Value.ToString();
      tbxUpdated.Text  = stc.Log.Updated.Value.ToString();
      tbxUpdatedBy.Text = stc.Log.UpdatedBy.Value.ToString();
    }

    protected void btnUpdate_Click(object sender, System.EventArgs e)
    {
      lblError.Text = "Not implemented";
    }

    protected void btnDelete_Click(object sender, System.EventArgs e)
    {
      lblError.Text = "Not implemented";
    }

    private void SetValues(StudyType studyType, CrudType type) {}

    private enum CrudType { Create, Update }

    protected void btnReset_Click(object sender, System.EventArgs e)
    {
      lblError.Text = "";

      tbxCourse_StudyTypeCategory_ID.Text = "";
      tbxStudyType_ID.Text = "";
      tbxNameEn.Text = "";
      tbxNameDa.Text = "";

      tbxCreated.Text = "";
      tbxCreatedBy.Text = "";
      tbxUpdated.Text = "";
      tbxUpdatedBy.Text = "";
    }

  }
}
```

Chapter 3

Business Tier

3.1 BizTypes

3.1.1 BizException

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;

namespace StudyPlanning.Biz
{
  /// <summary>
  /// Represents an exception in the business services tier.
  /// </summary>
  public class BizException : System.SystemException
  {
    #region Private Properties
    private int _Number;
    private string _Message;
    private string _PropertyName = "";
    private string _PropertyValue = "";
    private DalException _Dal;
    #endregion //Private Properties

    #region Public Properties

    /// <summary>
    /// Gets or sets a number that identifies the type of error.
    /// </summary>
    public int Number
    {
      get { return _Number; }
      set { _Number = value; }
    }

    /// <summary>
    /// Gets the text describing the error.
    /// </summary>
    /// <remarks>Invoke the <strong>RetrieveMessage</strong> method in order
    /// to set the <strong>Message</strong> property.</remarks>
    public override string Message
    {
      get { return _Message; }
    }

    /// <summary>
    /// Gets or sets a string containing the name of the property that caused
    /// the exception to be thrown.
    /// </summary>
    public string PropertyName
    {
      get { return _PropertyName; }
      set { _PropertyName = value; }
    }

    /// <summary>
    /// Gets or sets a string containing the value of the property that caused
    /// the exception to be thrown.
    /// </summary>
    public string PropertyValue
    {
```

```
    get { return _PropertyValue; }
    set { _PropertyValue = value; }
  }

  /// <summary>
  /// Gets or sets the <see cref="StudyPlanning.DAL.DalException"/>
  /// that caused the exception to be thrown.
  /// </summary>
  public DalException Dal
  {
    get { return _Dal; }
    set { _Dal = value; }
  }

  #endregion

  #region Constructors

  /// <summary>
  /// Initializes a new instance of the <see cref="StudyPlanning.Biz.BizException"/>
  /// class.
  /// </summary>
  public BizException() {}

  /// <summary>
  /// Initializes a new instance of the <see cref="StudyPlanning.Biz.BizException"/>
  /// class with the specified error number.
  /// </summary>
  /// <param name="errorNumber">The number that identifies the type of error.</param>
  public BizException(int errorNumber)
  {
    _Number = errorNumber;
    RetrieveMessage();
    _Dal = null;
  }

  /// <summary>
  /// Initializes a new instance of the <see cref="StudyPlanning.Biz.BizException"/>
  /// class with the specified error number and property name.
  /// </summary>
  /// <param name="errorNumber">The number that identifies the type of error.</param>
  /// <param name="propertyName">The name of the property that causes the error.</param>
  public BizException(int errorNumber, string propertyName)
  {
    _Number = errorNumber;
    _PropertyName = propertyName;
    RetrieveMessage();
    _Dal = null;
  }

  /// <summary>
  /// Initializes a new instance of the <see cref="StudyPlanning.Biz.BizException"/>
  /// class with the specified error number, attribute name and attribute value.
  /// </summary>
  /// <param name="errorNumber">The number that identifies the type of error.</param>
  /// <param name="propertyName">The name of the property that causes the error.</param>
  /// <param name="propertyValue">The value of the property.</param>
  public BizException(int errorNumber, string propertyName, string propertyValue)
  {
    _Number = errorNumber;
    _PropertyName = propertyName;
    _PropertyValue = propertyValue;
    RetrieveMessage();
    _Dal = null;
  }

  /// <summary>
  /// Initializes a new instance of the <see cref="StudyPlanning.Biz.BizException" />
  /// class with the specified <see cref="StudyPlanning.DAL.DalException" /> object.
  /// </summary>
  /// <param name="objException">The <see cref="StudyPlanning.DAL.DalException" />
  /// object which caused the <see cref="StudyPlanning.Biz.BizException" /> to
  /// be thrown.
  /// </param>
  public BizException(DalException objException)
  {
    _Dal = objException;
    _Number = 4;
    RetrieveMessage();
  }

  /// <summary>
  /// Initializes a new instance of the <see cref="StudyPlanning.Biz.BizException" />
  /// class with the specified <see cref="StudyPlanning.DAL.DalException" /> object,
  /// attribute name and attribute value.
  /// </summary>
  /// <param name="objException">The <see cref="StudyPlanning.DAL.DalException" />
  /// object which has caused the <see cref="StudyPlanning.Biz.BizException" /> to
  /// be thrown.
  /// </param>
  /// <param name="propertyName">The name of the property that causes the error.</param>
  /// <param name="propertyValue">The value of the property.</param>
  public BizException(DalException objException, string propertyName, string propertyValue)
  {
    _Dal = objException;
    _Number = 4;
    _PropertyName = propertyName;
    _PropertyValue = propertyValue;
    RetrieveMessage();
```

```
    }

    /// <summary>
    /// Initializes a new instance of the <see cref="StudyPlanning.Biz.BizException" />
    /// class with the specified text.
    /// </summary>
    /// <param name="message">A text that describes the exception.</param>
    public BizException(string message)
    {
      _Message = message;
    }

    #endregion //Constructors

    #region Methods

    /// <summary>
    /// Retrieves the message corresponding to the error number of
    /// the Error property of the current instance.
    /// </summary>
    private void RetrieveMessage()
    {
      const int TextGroup_ID = 710;
      const string Culture_ID = "en-GB";

      if (_Number == 4)
      {
        string strText;

        if (_PropertyName.Length > 0 && _PropertyValue.Length > 0)
        {
          StudyPlanning.Biz.TextItem objText = StudyPlanning.Biz.TextItem.Retrieve(TextGroup_ID, Convert.ToString(_Number)
            + "_B", Culture_ID);
          strText = objText.Text;

          strText = strText.Replace("@@V1", _PropertyName);
          strText = strText.Replace("@@V2", _PropertyValue);
        }
        else if (_PropertyName.Length > 0)
        {
          StudyPlanning.Biz.TextItem objText = StudyPlanning.Biz.TextItem.Retrieve(TextGroup_ID, Convert.ToString(_Number)
            + "_C", Culture_ID);
          strText = objText.Text;
          strText = strText.Replace("@@V1", _PropertyName);
        }
        else
        {
          StudyPlanning.Biz.TextItem objText = StudyPlanning.Biz.TextItem.Retrieve(TextGroup_ID, Convert.ToString(_Number)
            + "_A", Culture_ID);
          strText = objText.Text;
        }

        _Message = strText;
      }
      else
      {
        StudyPlanning.Biz.TextItem objText = StudyPlanning.Biz.TextItem.Retrieve(TextGroup_ID, Convert.ToString(_Number),
          Culture_ID);
        _Message = objText.Text;
      }
    }

    #endregion //Methods
  }
}
```

3.1.2 BizObject

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Con↓guration;

namespace StudyPlanning.Biz
{
  /// <summary>
  /// Base class for the business layer (also known as business rules tier).
  /// The class is declared as abstract which means that this class can act as
  /// a basis for other classes, but cannot be instantiated.
  /// </summary>
  public abstract class BizObject
  {

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.Biz.BizObject"/> class.
    /// </summary>
    public BizObject()
    {
    }

  }
}
```

3.1.3 BizStringLocalizable

```
using System;
using System.Collections;
using System.Xml;

namespace StudyPlanning.Biz
{
  /// <summary>
  /// Represents a localizable text (series of Unicode characters)
  /// in the business services tier.
  /// </summary>
  public class BizStringLocalizable : StudyPlanning.Biz.BizType
  {
    private Hashtable _Values = new Hashtable();

    /// <summary>
    /// Initializes a new instance of the <see cref="StudyPlanning.Biz.BizStringLocalizable"/>
    /// class.
    /// </summary>
    /// <param name="allowNull">The value of this parameter decides whether to allow nulls.</param>
    public BizStringLocalizable(bool allowNull)
    {
      this._allowNull = allowNull;
    }

    /// <summary>
    /// Gets or sets the text for the specified culture.
    /// </summary>
    public string this[string culture_ID]
    {
      get
      {
        string strValue;
        strValue = Convert.ToString(_Values[culture_ID]);
        return strValue;
      }

      set
      {
        _Values[culture_ID] = value;
      }
    }

    /// <summary>
    /// Gets a text containing an xml representation of the localized string.
    /// </summary>
    public string Value
    {
      get
      {
        string strXml = "";
        IDictionaryEnumerator enumValues = _Values.GetEnumerator();

        //XmlDocument xmlDoc = new XmlDocument();
        //xmlDoc.CreateElement("cultures");
        strXml = "<cultures>";

        while (enumValues.MoveNext())
        {
          strXml += "<culture>";
          strXml += "<cultureID>" + enumValues.Key + "</cultureID>";
          strXml += "<value>" + enumValues.Value + "</value>";
          strXml += "</culture>";

          //XmlElement elemCulure = xmlDoc.CreateElement("culture");
          //(System.Xml.XmlElement)elemCulture.
        }
        strXml += "</cultures>";

        return strXml;
      }
    }

    /// <summary>
    /// Gets the number of cultures into which the string has been localized.
    /// </summary>
    public int Count
    {
      get { return _Values.Count; }
    }

    /// <summary>
    /// Localizes the string into the specified culture with the specified text.
    /// </summary>
    /// <param name="culture_ID">The culture for which the string should be
    /// localized into.</param>
    /// <param name="text">The text of the string for the specified culture.</param>
    public void Add(string culture_ID, string text)
    {
      _Values.Add(culture_ID, text);
      _isNull = false;
    }

    /// <summary>
    /// Removes the localization of the string for the specified culture.
    /// </summary>
    /// <param name="culture_ID">The culture for which the string should no
```

```
    /// longer be localized into.</param>
    public void Remove(string culture_ID)
    {
      _Values.Remove(culture_ID);
      if (_Values.Count == 0)
      {
        _isNull = true;
      }
    }

    /// <summary>
    /// Determines whether the string is localized into a specific culture.
    /// </summary>
    /// <param name="culture_ID">The culture for which it should be determined if the
    /// string has been localized into.</param>
    /// <returns><strong>true</strong>, if the string is localized into the specified culture;
    /// otherwise, <strong>false</strong>.</returns>
    public bool Contains(string culture_ID)
    {
      if (_Values.Contains(culture_ID))
        return true;
      else
        return false;
    }

    /// <summary>
    /// Loads an xml document from the specified string.
    /// </summary>
    /// <param name="xml">String containing the XML document to load.</param>
    public void LoadXml(string xml)
    {
      XmlDocument xDoc = new XmlDocument();
      xDoc.LoadXml(xml);
      XmlNode xmlRoot = xDoc.FirstChild;

      foreach(XmlNode xn in xmlRoot.ChildNodes)
      {
        if (_Values.Contains(xn.ChildNodes[0].InnerText))
        {
          /*
           * The string has already been localized into the current
           * culture and the text for the current culture is just
           * updated.
           */
          _Values[xn.ChildNodes[0].InnerText] = xn.ChildNodes[1].InnerXml;
        }
        else
        {
          _Values.Add(xn.ChildNodes[0].InnerText, xn.ChildNodes[1].InnerXml);

        }
      }

      if (_Values.Count == 0)
      {
        _isNull = true;
      }
      else
      {
        _isNull = false;
      }
    }

  }
}
```

3.1.4 BizType

```
using System;

namespace StudyPlanning.Biz
{
  /// <summary>
  /// Represents a type in the business rules layer.
  /// </summary>
  public abstract class BizType
  {
    /// <summary>
    /// Indicates whether the value of the type is null.
    /// </summary>
    protected bool _isNull = true;

    /// <summary>
    /// Indicates whether the value of the type is allowed to be null.
    /// </summary>
    protected bool _allowNull = false;

    /// <summary>
    /// Validates the type. If the property AllowNull is
    /// <strong>false</strong> and the property IsNull is
    /// <strong>true</strong> a <see cref="StudyPlanning.Biz.BizException"/>
    /// is thrown.
    /// </summary>
    /// <exception cref="StudyPlanning.Biz.BizException">Throws a
    /// <see cref="StudyPlanning.Biz.BizException" /> if the property
```

```
    /// AllowNull is false and the property IsNull is true.
    /// </exception>
    public void Validate()
    {
      if (!_allowNull && _isNull)
      {
        BizException objEx = new BizException(4);
        throw objEx;
      }
    }

    /// <summary>
    /// Validates the type. If the property AllowNull is
    /// <strong>false</strong> and the property IsNull is
    /// <strong>true</strong> a <see cref="StudyPlanning.Biz.BizException"/>
    /// is thrown.
    /// </summary>
    /// <exception cref="StudyPlanning.Biz.BizException">Throws a
    /// <see cref="StudyPlanning.Biz.BizException" /> if the property
    /// AllowNull is false and the property IsNull is true.
    /// </exception>
    public void Validate(string propertyName)
    {
      if (!_allowNull && _isNull)
      {
        BizException objEx = new BizException(4, propertyName);
        throw objEx;
      }
    }

    /// <summary>
    /// Gets or sets a value indicating whether the value of
    /// the type is null.
    /// </summary>
    /// <remarks>The <strong>IsNull</strong> property may only
    /// be set to <strong>true</strong> - if set to <strong>
    /// false</strong> a <see cref="StudyPlanning.Biz.BizException"/>
    /// is thrown.
    /// </remarks>
    public bool IsNull
    {
      get { return _isNull; }
      set
      {
        if (Convert.ToBoolean(value))
        {
          _isNull = true;
        }
        else
        {
          BizException objEx = new BizException(4);
          throw objEx;
        }
      }
    }

    /// <summary>
    /// Gets a value indicating whether the value of the type
    /// is allowed to be null.
    /// </summary>
    public bool AllowNull
    {
      get { return _allowNull; }
    }

  }
}
```

3.2 Course

```
using System;
using System.Collections;
using StudyPlanning.DAL;
using StudyPlanning.DAL.Students;
using StudyPlanning.DAL.Courses;

namespace StudyPlanning.Biz
{
  /// <summary>
  /// Represents a course in the business services tier.
  /// </summary>
  public class Course : StudyPlanning.Biz.BizObject
  {

    /// <summary>
    /// Retrieves a list of point blocking courses represented by Course_IDs for a given CourseVersion.
    /// </summary>
    /// <param name="courseVersion_ID">Identification of the <see cref="StudyPlanning.DAL.Courses.CourseVersion"/>.</param>
    public static Guid[][] GetPointBlockingCourses(Guid courseVersion_ID)
    {
      Guid[] ids = null;
      Guid[][] courses = null;

      try
```

```
    {
      ids = RelationCourse.GetIDFromVersion(courseVersion_ID, 0);
    }
    catch (DalException e)
    {
      if (e.Number == 8) // No point blocking courses - no error
        ids = new Guid[0];
      else
        throw new BizException(e, "CourseVersion_ID", courseVersion_ID.ToString());
    }

    courses = new Guid[ids.GetLength(0)][];

    for (int i=0; i < ids.GetLength(0); i++)
    {
      try
      {
        courses[i] = RelationCourseItem.GetCourses(ids[i]);
      }
      catch (DalException e)
      {
        if (e.Number == 8) // No point blocking courses (which are not null ) - no error
          courses[i] = new Guid[0];
        else
          throw new BizException(e, "Course_RelationCourse_ID", ids[i].ToString());
      }
    }
    return courses;
  }

  /// <summary>
  /// Retrieves a list of mandatory prerequisite courses represented by Course_IDs for a given CourseVersion.
  /// </summary>
  /// <param name="courseVersion_ID">Identification of the <see cref="StudyPlanning.DAL.Courses.CourseVersion"/>.</param>
  public static Guid[][] GetMandatoryPrerequisiteCourses(Guid courseVersion_ID)
  {
    Guid[] ids = null;
    ArrayList courses = new ArrayList();

    try
    {
      ids = RelationCourse.GetIDFromVersion(courseVersion_ID, 1);
    }
    catch (DalException e)
    {
      if (e.Number == 8) // No mandatory prerequisite courses - no error
        ids = new Guid[0];
      else
        throw new BizException(e, "CourseVersion_ID", courseVersion_ID.ToString());
    }

    Guid[] relationCourseItems = null;
    for (int i=0; i < ids.GetLength(0); i++)
    {
      try
      {
        relationCourseItems = RelationCourseItem.GetCourses(ids[i]);
      }
      catch (DalException e)
      {
        if (e.Number.Equals(8))
          relationCourseItems = new Guid[0];
        else
          throw new BizException(e, "Course_RelationCourse_ID", ids[i].ToString());
      }

      if (relationCourseItems.Length > 0)
        courses.Add(relationCourseItems);
    }
    return (Guid[][])courses.ToArray(Type.GetType("System.Guid[]"));
  }

  /// <summary>
  /// Retrieves a list of technical prerequisite courses represented by Course_IDs for a given CourseVersion.
  /// </summary>
  /// <param name="courseVersion_ID">Identification of the <see cref="StudyPlanning.DAL.Courses.CourseVersion"/>.</param>
  public static Guid[][] GetTechnicalPrerequisiteCourses(Guid courseVersion_ID)
  {
    Guid[] ids = null;
    ArrayList courses = new ArrayList();

    try
    {
      ids = RelationCourse.GetIDFromVersion(courseVersion_ID, 2);
    }
    catch (DalException e)
    {
      if (e.Number == 8) // No technical prerequisite courses - no error
        ids = new Guid[0];
      else
        throw new BizException(e, "CourseVersion_ID", courseVersion_ID.ToString());
    }

    Guid[] relationCourseItems = null;
    for (int i=0; i < ids.GetLength(0); i++)
    {
      try
      {
        relationCourseItems = RelationCourseItem.GetCourses(ids[i]);
```

```
      }
      catch (DalException e)
      {
        if (e.Number.Equals(8))
          relationCourseItems = new Guid[0];
        else
          throw new BizException(e, "Course_RelationCourse_ID", ids[i].ToString());
      }

      if (relationCourseItems.Length > 0)
        courses.Add(relationCourseItems);
    }
    return (Guid[][])courses.ToArray(Type.GetType("System.Guid[]"));
  }

  /// <summary>
  /// Retrieves a list of desirable prerequisite courses represented by Course_IDs for a given CourseVersion.
  /// </summary>
  /// <param name="courseVersion_ID">Identification of the <see cref="StudyPlanning.DAL.Courses.CourseVersion"/>.</param>
  public static Guid[][] GetDesirablePrerequisiteCourses(Guid courseVersion_ID)
  {
    Guid[] ids = null;
    Guid[][] courses = null;

    try
    {
      ids = RelationCourse.GetIDFromVersion(courseVersion_ID, 3);
    }
    catch (DalException e)
    {
      if (e.Number == 8) // No desirable prerequisite courses - no error
        ids = new Guid[0];
      else
        throw new BizException(e, "CourseVersion_ID", courseVersion_ID.ToString());
    }

    courses = new Guid[ids.GetLength(0)][];

    for (int i=0; i < ids.GetLength(0); i++)
    {
      try
      {
        courses[i] = RelationCourseItem.GetCourses(ids[i]);
      }
      catch (DalException e)
      {
        throw new BizException(e, "Course_RelationCourse_ID", ids[i].ToString());
      }
    }
    return courses;
  }

  /// <summary>
  /// Gets a list of prerequisite courses of the specified prerequisite type for
  /// the specified course version.
  /// </summary>
  /// <param name="courseVersion_ID">The ID of the course version for which to get a list.</param>
  /// <param name="type">The type of prerequisite courses to build a list of.</param>
  /// <returns>An array of course identifiers.</returns>
  public static Guid[][] GetPrerequisites(Guid courseVersion_ID, PrerequisiteCourseType type)
  {
    Guid[][] prerequisites = null;

    try
    {
      if (type.Equals(PrerequisiteCourseType.Desirable))
      {
        prerequisites = StudyPlanning.Biz.Course.GetDesirablePrerequisiteCourses(courseVersion_ID);
      }
      else if (type.Equals(PrerequisiteCourseType.Mandatory))
      {
        prerequisites = StudyPlanning.Biz.Course.GetMandatoryPrerequisiteCourses(courseVersion_ID);
      }
      else if (type.Equals(PrerequisiteCourseType.Technical))
      {
        prerequisites = StudyPlanning.Biz.Course.GetTechnicalPrerequisiteCourses(courseVersion_ID);
      }
    }
    catch (BizException bizEx)
    {
      throw bizEx;
    }

    return prerequisites;
  }

  /// <summary>
  /// Gets a list of prerequisite courses of the specified prerequisite type for
  /// the specified course.
  /// </summary>
  /// <param name="course_ID">The ID of the course for which to get a list (the newest version is utilized).</param>
  /// <param name="type">The type of prerequisite courses to build a list of.</param>
  /// <returns>An array of course identifiers.</returns>
  public static Guid[][] GetPrerequisitesFromCourseID(Guid course_ID, PrerequisiteCourseType type)
  {
    Guid courseVersion_ID = Guid.Empty;
    try
    {
      courseVersion_ID = StudyPlanning.Biz.Course.GetNewestVersion(course_ID);
```

```
    }
    catch (BizException bizEx)
    {
      throw bizEx;
    }

    Guid[][] prerequisites = null;
    try
    {
      prerequisites = GetPrerequisites(courseVersion_ID, type);
    }
    catch (BizException bizEx)
    {
      throw bizEx;
    }

    return prerequisites;
  }

  /*
  /// <summary>
  /// Retrieves a list of represented by Module_IDs for a course version.
  /// </summary>
  /// <param name="courseVersion_ID">Identification of the
  /// <see cref="StudyPlanning.DAL.Courses.CourseVersion"/>.</param>
  /// <returns>An array of <see cref="StudyPlanning.DAL.Module"/> identifications.</returns>
  public static int[] GetModules(Guid courseVersion_ID)
  {
    int[] modules = null;

    try
    {
      modules = StudyPlanning.DAL.Courses.Module.GetModules(courseVersion_ID);
    }
    catch (DalException e)
    {
      throw new BizException(e, "CourseVersion_ID", courseVersion_ID.ToString());
    }
    return modules;
  }

*/
  /// <summary>
  /// Retrieves a list of the lecturers associated with a course version.
  /// </summary>
  /// <param name="courseVersion_ID">Identification of the
  /// <see cref="StudyPlanning.DAL.Courses.CourseVersion"/>.</param>
  /// <returns>An array containing <see cref="StudyPlanning.DAL.Lecturer"/> identifications.</returns>
  public static Guid[] GetLecturers(Guid courseVersion_ID)
  {
    Guid[] lecturers = null;

    try
    {
      lecturers = StudyPlanning.DAL.Courses.Lecturer.GetLecturers(courseVersion_ID);
    }
    catch (DalException e)
    {
      throw new BizException(e, "CourseVersion_ID", courseVersion_ID.ToString());
    }
    return lecturers;
  }

  /// <summary>
  /// Retrieves the workload for a course version.
  /// </summary>
  /// <param name="courseVersion_ID">Identification of the
  /// <see cref="StudyPlanning.DAL.Courses.CourseVersion"/>.</param>
  /// <param name="part">The course version part.</param>
  /// <returns>A floating point number representing the workload of the
  /// course version part.</returns>
  public static float GetWorkload(Guid courseVersion_ID, int part)
  {
    Guid pointID;
    try
    {
      pointID = StudyPlanning.DAL.Courses.Point.GetPoint(courseVersion_ID, part);
    }
    catch (DalException e)
    {
      throw new BizException(e);
    }

    StudyPlanning.DAL.Point point = new StudyPlanning.DAL.Point();
    point.Point_ID.Value = pointID;

    try
    {
      point.Retrieve();
    }
    catch (DalException e)
    {
      throw new BizException(e, "Point_ID", point.ToString());
    }

    return point.PointMin.Value;
  }

  /// <summary>
```

```
/// Retrieves the recommended placement for a course and studytype, expressed as a lower and upper limit of
/// points.
/// </summary>
/// <param name="courseVersion_ID">Identification of the CourseVersion.</param>
/// <param name="studyType_ID">Identification of the StudyType.</param>
/// <returns>An array containing the lower and upper limits of point.</returns>
public static float[] GetRecommendedPlacement(Guid courseVersion_ID, int studyType_ID)
{
  float[] recommendedPlacement = new float[2];

  Guid courseRecommendedPlacement_ID = Guid.Empty;
  try
  {
    courseRecommendedPlacement_ID = RecommendedPlacement.GetID(courseVersion_ID, studyType_ID);
  }
  catch (DalException e)
  {
    if (e.Number.Equals(8))
      return null;
    else
      throw new BizException(e, "CourseVersion_ID / StudyType_ID",
        courseVersion_ID + " / " + studyType_ID);
  }

  RecommendedPlacement rp = new RecommendedPlacement();
  rp.Course_RecommendedPlacement_ID.Value = courseRecommendedPlacement_ID;
  try
  {
    rp.Retrieve();
  }
  catch (DalException e)
  {
    throw new BizException(e, "Course_RecommendedPlacement_ID", courseRecommendedPlacement_ID.ToString());
  }

  Guid point_ID = Guid.Empty;

  //If point_id is null - retrieve the concept.
  if (rp.Point_ID.IsNull)
  {
    int rpcStudyType_ID = 0;
    try
    {
      rpcStudyType_ID = StudyPlanning.DAL.RecommendedPlacementConcepts.StudyType.GetID(
        rp.RecommendedPlacementConcept_ID.Value, studyType_ID);
    }
    catch (DalException e)
    {
      throw new BizException(e, "RecommendedPlacementConcept_ID / StudyType_ID",
        rp.RecommendedPlacementConcept_ID.Value + " / " + studyType_ID);
    }

    StudyPlanning.DAL.RecommendedPlacementConcepts.StudyType rpcStudyType = new StudyPlanning.DAL.
      RecommendedPlacementConcepts.StudyType();
    rpcStudyType.RecommendedPlacementConcept_StudyType_ID.Value = rpcStudyType_ID;
    try
    {
      rpcStudyType.Retrieve();
    }
    catch (DalException e)
    {
      throw new BizException(e, "RecommendedPlacementConcept_StudyType_ID", rpcStudyType_ID.ToString());
    }

    point_ID = rpcStudyType.Point_ID.Value;
  }
  else
  {
    point_ID = rp.Point_ID.Value;
  }

  //Retrieve the point object.
  StudyPlanning.DAL.Point point = new StudyPlanning.DAL.Point();
  point.Point_ID.Value = point_ID;
  try
  {
    point.Retrieve();
  }
  catch (DalException e)
  {
    throw new BizException(e, "Point_ID", point_ID.ToString());
  }

  //Return recommended placement.
  recommendedPlacement[0] = point.PointMin.Value;
  recommendedPlacement[1] = point.PointMax.Value;
  return recommendedPlacement;
}

/// <summary>
/// Retrieves the newest version of a course.
/// </summary>
/// <param name="course_ID">Identification of the
/// <see cref="StudyPlanning.DAL.Courses.Course"/>.</param>
/// <returns>A <see cref="StudyPlanning.DAL.Courses.CourseVersion"/> identification.
/// </returns>
public static Guid GetNewestVersion(Guid course_ID)
{
  Guid courseVersion_ID = new Guid();
```

```
    try
    {
      courseVersion_ID = CourseVersion.GetNewestVersion(course_ID);
    }
    catch (DalException e)
    {
      throw new BizException(e, "Course_ID", course_ID.ToString());
    }
    return courseVersion_ID;
  }

  /// <summary>
  /// Retrieves a list of periods in which a given <see cref="StudyPlanning.Biz.CoursePart"/> is taught.
  /// </summary>
  /// <param name="coursePart">The coursepart object for which to retrieve the periods.</param>
  /// <returns>An array of <see cref="StudyPlanning.DAL.Period"/> identifications.</returns>
  public static Guid[] GetPeriods(CoursePart coursePart)
  {
    Guid[] periods = new Guid[0];
    try
    {
      periods = StudyPlanning.DAL.Courses.Period.GetPeriods(coursePart.CourseVersion_ID, coursePart.Part);
    }
    catch (DalException e)
    {
      //if (e.Number != 8)
        throw new BizException(e, "CourseVersion_ID / Part",
          coursePart.CourseVersion_ID.ToString() + " / " + coursePart.Part.ToString());
    }
    return periods;
  }

  /// <summary>
  /// Determines whether a coursepart is taught in a given period.
  /// </summary>
  /// <param name="coursePart">A <see cref="StudyPlanning.Biz.CoursePart"/> object.</param>
  /// <param name="period">Identification of the <see cref="StudyPlanning.DAL.Period"/>.</param>
  /// <returns><strong>True</strong> if the coursepart is taught in the specified period,
  /// otherwise <strong>false</strong> is returned.</returns>
  public static bool TaughtInPeriod(CoursePart coursePart, Guid period)
  {
    try
    {
      Guid course_period_id = StudyPlanning.DAL.Courses.Period.GetID(
        coursePart.CourseVersion_ID,
        coursePart.Part,
        period);
    }
    catch (DalException e)
    {
      if (e.Number == 8)
        return false;
      else
        throw new BizException(e, "CourseVersion_ID / Part / Period_ID",
          coursePart.CourseVersion_ID + " / " + coursePart.Part + " / " + period);
    }
    return true;
  }

  /// <summary>
  /// Retrieves the modules in which a given coursepart is taught in a given period.
  /// </summary>
  /// <param name="coursePart">A <see cref="StudyPlanning.Biz.CoursePart"/> object.</param>
  /// <param name="period">Identification of the <see cref="StudyPlanning.DAL.Period"/>.</param>
  /// <returns>A two-dimensional array containing identifications of
  /// <see cref="StudyPlanning.DAL.Module"/> alternatives.</returns>
  public static int[][] GetModules(CoursePart coursePart, Guid period)
  {
    Guid coursePeriod = Guid.Empty;
    try
    {
      coursePeriod = StudyPlanning.DAL.Courses.Period.GetID(
        coursePart.CourseVersion_ID,
        coursePart.Part,
        period);
    }
    catch (DalException e)
    {
      throw new BizException(e, "CourseVersion_ID / Part / Period_ID",
        coursePart.CourseVersion_ID + " / " + coursePart.Part + " / " + period);
    }

    Guid[] cpModuleIds = null;
    try
    {
      cpModuleIds = StudyPlanning.DAL.Courses.PeriodModule.GetID(coursePeriod);
    }
    catch (DalException e)
    {
      throw new BizException(e, "Course_Period_ID", coursePeriod.ToString());
    }

    Guid[][] cpModuleItemIds = new Guid[cpModuleIds.GetLength(0)][];
    for(int i=0; i < cpModuleIds.GetLength(0); i++)
    {
      try
      {
        cpModuleItemIds[i] = StudyPlanning.DAL.Courses.PeriodModuleItem.GetID(cpModuleIds[i]);
      }
```

```
      catch (DalException e)
      {
        throw new BizException(e, "Course_Period_Module_ID", cpModuleIds[i].ToString());
      }
    }

    int[][] modules = new int[cpModuleItemIds.GetLength(0)][];
    for(int i=0; i < modules.GetLength(0); i++)
    {
      modules[i] = new int[cpModuleItemIds[i].GetLength(0)];
      for(int j=0; j < modules[i].GetLength(0); j++)
      {
        StudyPlanning.DAL.Courses.PeriodModuleItem cpmi = new StudyPlanning.DAL.Courses.PeriodModuleItem();
        cpmi.Course_Period_ModuleItem_ID.Value = cpModuleItemIds[i][j];
        try
        {
          cpmi.Retrieve();
        }
        catch (DalException e)
        {
          throw new BizException(e, "Course_Period_ModuleItem_ID", cpModuleItemIds[i][j].ToString());
        }
        modules[i][j] = cpmi.Module_ID.Value;
      }
    }
    return modules;
  }

  /// <summary>
  /// Retrieves a list of study types to which a course applies.
  /// </summary>
  /// <param name="courseVersion_ID">Identification of the course version.</param>
  /// <returns>An array of study type identifications</returns>
  public static int[] GetStudyTypes(Guid courseVersion_ID)
  {
    int[] studytypes = null;

    try
    {
      studytypes = StudyType.GetStudyTypes(courseVersion_ID);
    }
    catch (DalException e)
    {
      if (e.Number.Equals(8))
        return new int[0];
      else
        throw new BizException(e, "CourseVersion_ID", courseVersion_ID.ToString());
    }
    return studytypes;
  }

  /// <summary>
  /// Retrives a list of those course versions which match the specified keywords
  /// in the specified culture.
  /// </summary>
  /// <param name="keywords">The keywords for which to match courses.</param>
  /// <param name="culture_ID">The ID of the culture for which to match courses.</param>
  /// <returns>
  /// An array containing IDs of those course versions which are related to one or more
  /// of the specified keywords.<br/>
  /// <br/>
  /// How to intepret the returned array:
  /// <ul>
  ///  <li>At index <strong>0</strong> of the first dimension of the returned array is contained
  ///  those course versions which match <strong>most</strong> of the specified keywords.</li>
  ///  <li>At index <strong>1</strong> of the first dimension of the returned array is contained
  /// those course versions which match <strong>second most</strong> of the specified keywords.</li>
  ///  <li>Etc.</li>
  /// </ul>
  /// </returns>
  public static Guid[][] GetKeywordMatchingCourses(string[] keywords, string culture_ID)
  {
    Guid[][] mCourses;
    System.Collections.Hashtable matchingCourses = new System.Collections.Hashtable(10);
    System.Collections.SortedList matchCount = new System.Collections.SortedList();

    foreach(string keyword in keywords)
    {
      Guid[] matchingKeywords = null;

      try
      {
        matchingKeywords = StudyPlanning.DAL.Keyword.SearchForKeywords(keyword, culture_ID);
      }
      catch (DalException dalEx)
      {
        throw new BizException(dalEx);
      }

      foreach(Guid keyword_ID in matchingKeywords)
      {
        Guid[] courses;
        try
        {
          courses = StudyPlanning.DAL.Courses.Keyword.GetCoursesFromKeyword(keyword_ID);
        }
        catch (DalException dalEx)
        {
          throw new BizException(dalEx);
```

```
        }

        foreach(Guid courseVersion_ID in courses)
        {
          if (matchingCourses.ContainsKey(courseVersion_ID))
          {
            int curCount = 0;
            curCount = Convert.ToInt32(matchingCourses[courseVersion_ID]);

            if (matchCount.ContainsKey(curCount))
            {
              if (Convert.ToInt32(matchCount[curCount]) == 1) //it should be able to become less than 1!
              {
                matchCount.Remove(curCount);
              }
              else
              {
                matchCount[curCount] = Convert.ToInt32(matchCount[curCount]) - 1;
              }
            }
            else
            {
              /*
               * According to the way the method is designed it should not
               * be possible to reach this case and an exception is therefore
               * thrown.
               */
              throw new BizException("The hashtable 'matchCount' unexpectedly did not contain key " +
                curCount + ".");
            }

            matchingCourses[courseVersion_ID] = Convert.ToInt32(matchingCourses[courseVersion_ID]) + 1;

            curCount = Convert.ToInt32(matchingCourses[courseVersion_ID]);

            if (matchCount.ContainsKey(curCount))
            {
              matchCount[curCount] = Convert.ToInt32(matchCount[curCount]) + 1;
            }
            else
            {
              matchCount.Add(curCount, 1);
            }
          }
          else
          {
            matchingCourses.Add(courseVersion_ID, 1);

            if (matchCount.ContainsKey(1))
            {
              matchCount[1] = Convert.ToInt32(matchCount[1]) + 1;
            }
            else
            {
              matchCount.Add(1, 1);
            }
          }
        } //for each course version ID
      } //for each matching keyword
    } //for each specified keyword

    mCourses = new Guid[matchCount.Count][];

    int h = matchCount.Count-1;
    for(int i=0; i < matchCount.Count; i++)
    {
      mCourses[i] = new Guid[Convert.ToInt32(matchCount[matchCount.GetKey(h)])];

      int index = 0;

      System.Collections.IDictionaryEnumerator cEnum = matchingCourses.GetEnumerator();

      while (cEnum.MoveNext())
      {
        Guid courseVersion_ID = (Guid)cEnum.Key;

        if (Convert.ToInt32(cEnum.Value) == Convert.ToInt32(matchCount.GetKey(h)))
        {
          mCourses[i][index] = courseVersion_ID;
          if (index == mCourses[i].Length)
            break;

          index++;

          #region Remark
          /*
           * It would be more effective to remove the key-and-value pair having the
           * current courseVersion_ID as key, however, it is not allowed to remove
           * elements from an enumerator...
           */
          #endregion
        }
      }
      h--;
    }

    return mCourses;
  }
```

```
/// <summary>
/// Produces an array of <see cref="StudyPlanning.Biz.CoursePart"/> objects from the
/// specified array of globally unique identifiers of courses.
/// </summary>
/// <param name="courses">An array of globally unique identifiers of courses.</param>
/// <param name="type">Denotes the type of 1st parameter
/// <ul>
/// <li><strong>1</strong> the array contains course identifiers.</li>
/// <li><strong>2</strong> the array contains courseversion identifiers.</li>
/// </ul>
/// </param>
/// <returns>An array of <see cref="StudyPlanning.Biz.CoursePart"/> objects.</returns>
public static CoursePart[] GetCourseParts(Guid[] courses, int type)
{
  ArrayList courseParts = new ArrayList();
  Guid courseVersion_ID = Guid.Empty;

  foreach(Guid id in courses)
  {
    if (type.Equals(1))
    {
      try
      {
        courseVersion_ID = StudyPlanning.Biz.Course.GetNewestVersion(id);
      }
      catch (DalException e)
      {
        throw new BizException(e, "Course_ID", id.ToString());
      }
    } else
      courseVersion_ID = id;

    StudyPlanning.DAL.Courses.CourseVersion objCv =
      new StudyPlanning.DAL.Courses.CourseVersion();

    objCv.CourseVersion_ID.Value = courseVersion_ID;

    try
    {
      objCv.Retrieve();
    }
    catch (DalException e)
    {
      throw new BizException(e, "CourseVersion_ID", courseVersion_ID.ToString());
    }

    int part = objCv.Parts.Value;
    while(part > 0)
    {
      CoursePart cp = new CoursePart();
      cp.Course_ID = objCv.Course_ID.Value;
      cp.CourseVersion_ID = courseVersion_ID;
      cp.Part = part;
      cp.TotalParts = objCv.Parts.Value;
      cp.Workload = StudyPlanning.Biz.Course.GetWorkload(courseVersion_ID, part);
      courseParts.Add(cp);
      part--;
    }
  }
  return (CoursePart[])courseParts.ToArray(Type.GetType("StudyPlanning.Biz.CoursePart"));
}

/// <summary>
/// Gets a list of the different combinations of prerequisites courses to
/// the specified course version.
/// </summary>
/// <param name="courseVersion_ID">The ID of the course version for which to get a list.</param>
/// <param name="type">The type of course prerequisite which to make a list of.</param>
/// <param name="chain">An indication of whether to get a list of combinations representing
/// the chain of prerequisites.</param>
/// <returns>An array containing the combinations of prerequisite courses.</returns>
public static Guid[][] GetPrerequisiteCourseCombinations(Guid courseVersion_ID, PrerequisiteCourseType type, bool chain)
{
  Guid[][] prerequisites = null;

  if (type.Equals(PrerequisiteCourseType.Desirable))
  {
    prerequisites = StudyPlanning.Biz.Course.GetDesirablePrerequisiteCourses(courseVersion_ID);
  }
  else if (type.Equals(PrerequisiteCourseType.Mandatory))
  {
    prerequisites = StudyPlanning.Biz.Course.GetMandatoryPrerequisiteCourses(courseVersion_ID);
  }
  else if (type.Equals(PrerequisiteCourseType.Technical))
  {
    prerequisites = StudyPlanning.Biz.Course.GetTechnicalPrerequisiteCourses(courseVersion_ID);
  }

  Guid[][] emptyComb = new Guid[0][];
  Guid[][] resultingCombinations = PrerequisiteCourseCombinationsExecute(emptyComb, prerequisites, type, chain);

  resultingCombinations = ReducePrerequisiteCourseCombinations(resultingCombinations);
  return resultingCombinations;
}

#region Prerequisite Course Combinations Methods

/// <summary>
/// Generates a list of combinations of the specified course prerequisites. Auxiliary
```

```
/// method to the <code>GetPrerequisiteCourseCombinations</code> method. The
/// method builds the list by recursively invoking itself.
/// </summary>
/// <param name="currentCombinations">The current set of combinations — possibly
/// the empty set.</param>
/// <param name="prerequisites">The course prerequisites which to generate combinations of.</param>
/// <param name="type">The type of course prerequisite which to generate a list of.</param>
/// <param name="chain">An indication of whether to generate a list of combinations representing
/// the chain of prerequisites.</param>
/// <returns>An array containing the combinations of prerequisite courses.</returns>
private static Guid[][] PrerequisiteCourseCombinationsExecute(
  Guid[][] currentCombinations,
  Guid[][] prerequisites,
  PrerequisiteCourseType type,
  bool chain)
{
  if (prerequisites.GetLength(0) == 0)
  {
    return currentCombinations;
  }
  else if (prerequisites.GetLength(0) == 1)
  {
    return PrerequisiteCourseCombine(currentCombinations, (Guid[])prerequisites[0], type, chain);
  }
  else
  {
    Guid[][] newPrerequisites = new Guid[prerequisites.Length-1][];

    for (int i=0; i < newPrerequisites.GetLength(0); i++)
    {
      newPrerequisites[i] = new Guid[prerequisites[i+1].Length];
      for (int j=0; j < newPrerequisites[i].Length; j++)
      {
        newPrerequisites[i][j] = prerequisites[i+1][j];
      }
    }

    Guid[][] newCombinations = PrerequisiteCourseCombine(currentCombinations, (Guid[])prerequisites[0], type, chain);
    currentCombinations = null;
    currentCombinations = PrerequisiteCourseCombinationsExecute(newCombinations, newPrerequisites, type, chain);
    return currentCombinations;
  }
}

/// <summary>
/// Combines the disjunction list with the specified set of combinations. Auxiliary
/// method to the <code>PrerequisiteCourseCombinationsExecute</code> method.
/// </summary>
/// <param name="currentCombinations">The set of combinations which the disjunction list should be combined with.</param>
/// <param name="disjunctionList">The disjunction list which should be combined with the set of combinations.</param>
/// <param name="type">The type of course prerequisites which to make a list of.</param>
/// <param name="chain">An indication of whether to get a list of combinations representing
/// the chain of prerequisites.</param>
/// <returns>An array containing the combinations of the specified disjunction list and
/// the specified set of existing combinations.</returns>
private static Guid[][] PrerequisiteCourseCombine(Guid[][] currentCombinations, Guid[] disjunctionList,
  PrerequisiteCourseType type, bool chain)
{
  int numberOfCombinations = currentCombinations.GetLength(0);
  int numberOfDisjunctions = disjunctionList.Length;

  Guid[][] newCombinations;

  if (numberOfCombinations == 0) //base case
  {
    #region Base Case
    if (chain)
    {
      #region Chain
      newCombinations = new Guid[0][];

      for (int i=0; i < numberOfDisjunctions; i++)
      {
        Guid[][] prerequisites = null;
        try
        {
          prerequisites = GetPrerequisitesFromCourseID(disjunctionList[i], type);
        }
        catch (BizException bizEx)
        {
          throw bizEx;
        }

        if (prerequisites.GetLength(0) == 0) //current course has no prerequisites
        {
          Guid[][] fracComb = new Guid[1][];
          fracComb[0] = new Guid[1];
          fracComb[0][0] = disjunctionList[i];
          newCombinations = MergeCombinations(newCombinations, fracComb);
        }
        else
        {
          Guid[][] preqCombinations = PrerequisiteCourseCombinationsExecute(new Guid[0][], prerequisites, type, chain);
          Guid[][] tempCombinations = new Guid[preqCombinations.GetLength(0)][];

          for (int j=0; j < preqCombinations.GetLength(0); j++)
          {
            tempCombinations[j] = new Guid[preqCombinations[j].Length + 1];
            tempCombinations[j][0] = disjunctionList[i];
```

```
            for (int g=0; g < preqCombinations[j].Length; g++)
            {
              tempCombinations[j][g+1] = preqCombinations[j][g];
            }
          }

          newCombinations = MergeCombinations(newCombinations, tempCombinations);
        }
      }
      #endregion //Chain
    }
    else
    {
      newCombinations = new Guid[numberOfDisjunctions][];
      for (int i=0; i < newCombinations.Length; i++)
      {
        newCombinations[i] = new Guid[1];
        newCombinations[i][0] = disjunctionList[i];
      }
    }
    #endregion
  }
  else
  {
    #region Non-base Case
    if (chain)
    {
      #region Chain
      newCombinations = new Guid[0][];

      for (int i=0; i < numberOfDisjunctions; i++)
      {
        Guid[][] prerequisites = null;
        try
        {
          prerequisites = GetPrerequisitesFromCourseID(disjunctionList[i], type);
        }
        catch (BizException bizEx)
        {
          throw bizEx;
        }

        if (prerequisites.GetLength(0) == 0)
        {
          Guid[][] tempCombinations = new Guid[currentCombinations.GetLength(0)][];
          for (int r=0; r < tempCombinations.GetLength(0); r++)
          {
            tempCombinations[r] = new Guid[currentCombinations[r].Length+1];
            tempCombinations[r][0] = disjunctionList[i];
            for (int s=0; s < currentCombinations[r].Length; s++)
            {
              tempCombinations[r][s+1] = currentCombinations[r][s];
            }
          }

          newCombinations = MergeCombinations(newCombinations, tempCombinations);
        }
        else
        {
          Guid[][] preqCombinations = PrerequisiteCourseCombinationsExecute(new Guid[0][], prerequisites, type, chain);
          Guid[][] tempCombinations = new Guid[preqCombinations.GetLength(0)][];

          for (int j=0; j < preqCombinations.GetLength(0); j++)
          {
            tempCombinations[j] = new Guid[preqCombinations[j].Length + 1];
            tempCombinations[j][0] = disjunctionList[i];

            for (int g=0; g < preqCombinations[j].Length; g++)
            {
              tempCombinations[j][g+1] = preqCombinations[j][g];
            }
          }

          Guid[][] ↓nalComb = new Guid[tempCombinations.GetLength(0)*currentCombinations.GetLength(0)][];
          int k=0;

          for (int m=0; m < tempCombinations.GetLength(0); m++)
          {
            for (int n=0; n < currentCombinations.GetLength(0); n++)
            {
              ↓nalComb[k] = new Guid[tempCombinations[m].Length + currentCombinations[n].Length];

              for (int p=0; p < tempCombinations[m].Length; p++)
              {
                ↓nalComb[k][p] = tempCombinations[m][p];
              }

              for (int q=0; q < currentCombinations[n].Length; q++)
              {
                ↓nalComb[k][q+tempCombinations[m].Length] = currentCombinations[n][q];
              }

              k++;
            }
          }

          newCombinations = MergeCombinations(newCombinations, ↓nalComb);
        }
```

```
        }
        #endregion
      }
      else
      {
        #region Non-chain
        newCombinations = new Guid[numberOfCombinations*numberOfDisjunctions][];

        int k=0;

        for (int j=0; j < numberOfDisjunctions; j++)
        {
          for (int i=0; i < numberOfCombinations; i++)
          {
            newCombinations[k] = new Guid[currentCombinations[j].Length + 1];
            newCombinations[k][0] = disjunctionList[j];

            for (int g=0; g < currentCombinations[j].Length; g++)
            {
              newCombinations[k][g+1] = currentCombinations[i][g];
            }

            k++;
          }
        }
        #endregion //Non-chain
      }
      #endregion //Non-base Case
    }
    return newCombinations;
  } // end of method PrerequisiteCourseCombine

  /// <summary>
  /// Eliminates any repeating courses in the specified combinations.
  /// </summary>
  /// <param name="combinations">The combinations from which to eliminate repeating courses.</param>
  /// <returns>An array containing combinations of courses containing
  /// no repeating courses within the individual combination.</returns>
  private static Guid[][] ReducePrerequisiteCourseCombinations(Guid[][] combinations)
  {
    ArrayList tComb = new ArrayList(combinations.GetLength(0));

    for (int i=0; i < combinations.GetLength(0); i++)
    {
      tComb.Add(new ArrayList(combinations[i].Length));

      for (int j=0; j < combinations[i].Length; j++)
      {
        ArrayList curList = (ArrayList)tComb[i];
        Guid course_ID = combinations[i][j];

        if (!curList.Contains(course_ID))
        {
          curList.Add(combinations[i][j]);
        }
      }
    }

    Guid[][] reducedComb = new Guid[tComb.Count][];

    for (int i=0; i < tComb.Count; i++)
    {
      reducedComb[i] = new Guid[((ArrayList)tComb[i]).Count];
      for (int j=0; j < ((ArrayList)tComb[i]).Count; j++)
      {
        reducedComb[i][j] = (Guid)((ArrayList)tComb[i])[j];
      }
    }

    return reducedComb;
  }

  /// <summary>
  /// Merges two sets of combinations.
  /// </summary>
  /// <param name="mainComb">The first combination.</param>
  /// <param name="fracComb">The second combination.</param>
  /// <returns>An array containing the merge of the two specified
  /// sets of combinations.</returns>
  private static Guid[][] MergeCombinations(Guid[][] mainComb, Guid[][] fracComb)
  {
    int mainCombLength = mainComb.GetLength(0);

    int fracCombLength = fracComb.GetLength(0);

    int newLength = mainCombLength + fracCombLength;

    Guid[][] mergedCombinations = new Guid[newLength][];

    for (int i=0; i < mainComb.GetLength(0); i++)
    {
      mergedCombinations[i] = new Guid[mainComb[i].Length];
      for(int j=0; j < mainComb[i].Length; j++)
      {
        mergedCombinations[i][j] = mainComb[i][j];
      }
    }

    for (int i=0; i < fracCombLength; i++)
```

```
      {
        mergedCombinations[i+mainCombLength] = new Guid[fracComb[i].Length];
        for(int j=0; j < fracComb[i].Length; j++)
        {
          mergedCombinations[i+mainCombLength][j] = fracComb[i][j];
        }
      }

      return mergedCombinations;
    }

    #endregion //Prerequisite Course Combinations Methods

    /// <summary>
    /// Takes the prerequisite combinations of some course and sorts these in order of
    /// ascending complexity taking into account the specified passed, going and
    /// planned courses of some student. That is, those prerequisite combinations
    /// being easiest for student to fulfill will be placed first in the returned
    /// list etc.<br/>
    /// The specified passed, going and planned will be filtered away from the returned
    /// prerequisite combinations.
    /// </summary>
    /// <param name="passedGoingPlannedCourses">The passed, going and planned courses of the student.</param>
    /// <param name="prerequisiteCombinations">The prerequisite combinations of some course.</param>
    /// <returns>An array containing the sorted and filtered combinations of prerequisite courses.</returns>
    public static Guid[][] SortCombinationsByComplexity(Hashtable passedGoingPlannedCourses, Guid[][] prerequisiteCombinations
      )
    {
      SortedList matches = new SortedList(10);
      Hashtable matchGroups = new Hashtable(10);
      Hashtable reducedCombinations = new Hashtable(prerequisiteCombinations.GetLength(0));

      for (int i=0; i < prerequisiteCombinations.GetLength(0); i++)
      {
        int count = 0;
        reducedCombinations.Add(i, new ArrayList());

        for (int j=0; j < prerequisiteCombinations[i].Length; j++)
        {
          bool check = passedGoingPlannedCourses.Contains(prerequisiteCombinations[i][j]);
          if (!check)
          {
            count++;
            ((ArrayList)reducedCombinations[i]).Add(prerequisiteCombinations[i][j]);
          }
        }

        if (!matches.Contains(count))
          matches.Add(count, "");

        if (matchGroups.Contains(count))
        {
          ((ArrayList)matchGroups[count]).Add(i);
        }
        else
        {
          matchGroups.Add(count, new ArrayList());
          ((ArrayList)matchGroups[count]).Add(i);
        }
      }

      Guid[][] sortedCombinations = new Guid[prerequisiteCombinations.GetLength(0)][];

      int m=0;

      for (int k=0; k < matches.Count; k++)
      {
        ArrayList curComb = (ArrayList)matchGroups[matches.GetKey(k)];

        for (int q=0; q < curComb.Count; q++)
        {
          ArrayList reducComb = (ArrayList)reducedCombinations[(int)curComb[q]];
          int length = reducComb.Count;
          sortedCombinations[m] = new Guid[length];

          for (int p=0; p < length; p++)
          {
            sortedCombinations[m][p] = (Guid)reducComb[p];
          }

          m++;
        }
      }

      return sortedCombinations;
    }

  }

  /// <summary>
  /// Specifies the different types of course prerequisites.
  /// </summary>
  public enum PrerequisiteCourseType
  {
    /// <summary>
    /// Indicates that the type of prerequisite course is mandatory.
    /// </summary>
    Mandatory=1,
```

```
    /// <summary>
    /// Indicates that the type of prerequisite course is technical.
    /// </summary>
    Technical=2,

    /// <summary>
    /// Indicates that the type of prerequisite course is desirable.
    /// </summary>
    Desirable=3
  };
}
```

3.2.1 Grab

```
using System;
using System.Collections;
using System.Text;
using System.Text.RegularExpressions;
using StudyPlanning.DAL;
using StudyPlanning.DAL.Courses;

namespace StudyPlanning.Biz.Courses.Grab
{
  /// <summary>
  /// Grabs courses from the DTU course catalogue.
  /// </summary>
  public class CourseGrabber
  {
    private Hashtable Document = new Hashtable();

    private StudyPlanning.DAL.Courses.Course course =
      new StudyPlanning.DAL.Courses.Course();

    private StudyPlanning.DAL.Courses.CourseVersion courseVersion =
      new StudyPlanning.DAL.Courses.CourseVersion();

    private StudyPlanning.DAL.Courses.CourseGrab courseGrab =
      new StudyPlanning.DAL.Courses.CourseGrab();

    private StudyPlanning.DAL.Courses.Point coursePoint =
      new StudyPlanning.DAL.Courses.Point();

    private Guid userID = new Guid("{25B22D49-8DDD-406f-A6AA-09E0DE40E3DC}");
    private DateTime dtmNow = DateTime.Now;

    private string DtuCourseNumber;
    private float CreditPoints;
    private string DepartmentID;
    private ArrayList CreatedKeywords = new ArrayList(5);

    //private ArrayList Keywords = new ArrayList();
    //private ArrayList Departments = new ArrayList();
    //private ArrayList Lecturers = new ArrayList();
    //private ArrayList RecommendedPlacements = new ArrayList();

    #region Constructors

    public CourseGrabber() {}

    public CourseGrabber(string courseNumber, string departmentID)
    {
      this.DtuCourseNumber = courseNumber;
      this.DepartmentID = departmentID;
    }

    #endregion

    public static void GrabCourses()
    {
      //TODO: Retrieve list of departments from database.
      //Preliminary hard-coding of departments
      string[] arrDepartment = new string[15];
      //string[] arrDepartment = new string[1];

      arrDepartment[0] = "27";
      arrDepartment[1] = "11";
      arrDepartment[2] = "13";
      arrDepartment[3] = "34";
      arrDepartment[4] = "02";
      arrDepartment[5] = "10";
      arrDepartment[6] = "28";
      arrDepartment[7] = "01";

      arrDepartment[8] = "41";
      arrDepartment[9] = "42";
      arrDepartment[10] = "26";
      arrDepartment[11] = "33";
      arrDepartment[12] = "12";
      arrDepartment[13] = "31";
      arrDepartment[14] = "83";

      String strUri;

      foreach(string dnumber in arrDepartment)
      {
        strUri = "http://www.kurser.dtu.dk/search/search-result.asp?lstYearGroup=2003-08-01&lstDepartment=" + Convert.ToInt16(dnumber);
```

```
    System.Net.WebRequest objWr = System.Net.WebRequest.Create(strUri);
    System.Net.WebResponse objWrs = objWr.GetResponse();
    System.IO.Stream objStream = objWrs.GetResponseStream();
    System.IO.StreamReader objReader = new System.IO.StreamReader(objStream);

    String strResponse = objReader.ReadToEnd();
    //String strPattern = @"<A.*>(\d{5})</A>";
    String strPattern = @"<A.*coursecode=(?<fullyQualifiedCourseNumber>(\d{5}-\d{1})*)"">(\d{5})</A>";

    MatchCollection objMatches = Regex.Matches(strResponse, strPattern, RegexOptions.IgnoreCase);

    if (objMatches.Count > 0)
    {
      string fullyQuali↓edCourseNumber = "";
      foreach(Match objMatch in objMatches)
      {
        fullyQuali↓edCourseNumber = objMatch.Groups["fullyQualifiedCourseNumber"].Value.ToString();
        StudyPlanning.Biz.Courses.Grab.CourseGrabber course =
          new CourseGrabber(fullyQuali↓edCourseNumber, dnumber);
        course.GrabCourse();
      }
    }

    //StudyPlanning.Biz.Courses.Grab.CourseGrabber course = new StudyPlanning.Biz.Courses.Grab.CourseGrabber("02647-2"); //27258-1
    //course.GrabCourse();
  }

}

public void GrabCourse(string fullyQuali↓edCourseNumber)
{
  this.DtuCourseNumber = fullyQuali↓edCourseNumber;
  GrabCourseExecute();
}

public void GrabCourse()
{
  GrabCourseExecute();
}

public void GrabCourseExecute()
{
  GetDocument("da", "da-DK");
  GetDocument("en-gb", "en");

  string strPattern;
  strPattern = @"Institut:</td>(.|\n)*?";
  strPattern += @"<td class=""value"">(?<num>([0-9]{2}))([\s]{1})(?<name>[\w|\W|.|\s]*?)</td>";
  //String strPattern = @"<label class=""header_white"">(?<number>\d{5})(?<fill>[\n\t{6,7}\s{0,1}\n\t{6,7})(?<name>[\w|\W|.|\s]*)</label
    >";

  //strPattern = @"(?<num>([0-9]{2}))([\s]{1})(?<name>[a-zA-Z|\s|\W]*?)<br/>";

  MatchCollection objMatches = Regex.Matches(this.Document["da-DK"].ToString(), strPattern, RegexOptions.Singleline);
  if (objMatches.Count > 0)
  {
    if (objMatches[0].Groups["num"].ToString().Equals(this.DepartmentID))
    {
      objMatches = Regex.Matches(this.Document["da-DK"].ToString(), strPattern, RegexOptions.Singleline);

      course.Course_ID.Value = Guid.NewGuid();

      course.Log.Created.Value = dtmNow;
      course.Log.CreatedBy.Value = userID;
      course.Log.Updated.Value = dtmNow;
      course.Log.UpdatedBy.Value = userID;

      try
      {
        course.Create();
      }
      catch (DalException ex)
      {
        throw new BizException(ex);
      }

      GrabNameAndNumber();
      GrabLanguage();
      GrabPoint();
      GrabTeachingForm();
      courseVersion.Parts.Value = 1; //parts is manually set to 1
      GrabAssessmentType();
      GrabEvaluationFormDescription();
      GrabParticipantLimitation();
      GrabObjective();
      GrabContents();
      GrabRemarks();
      GrabUrl();

      courseVersion.CourseVersion_ID.Value = Guid.NewGuid();
      courseVersion.Course_ID.Value = course.Course_ID.Value;

      courseVersion.Version.Value = 1;

      courseVersion.Log.Created.Value = dtmNow;
      courseVersion.Log.CreatedBy.Value = userID;
      courseVersion.Log.Updated.Value = dtmNow;
      courseVersion.Log.UpdatedBy.Value = userID;
```

```
      coursePoint.Course_Point_ID.Value = Guid.NewGuid();
      coursePoint.CourseVersion_ID.Value = courseVersion.CourseVersion_ID.Value;
      coursePoint.Part.Value = 1; //part is hard-coded to 1
      coursePoint.Log.Created.Value = dtmNow;
      coursePoint.Log.CreatedBy.Value = userID;
      coursePoint.Log.Updated.Value = dtmNow;
      coursePoint.Log.UpdatedBy.Value = userID;

      StudyPlanning.DAL.Point point =
        new StudyPlanning.DAL.Point();

      point.Point_ID.Value = coursePoint.Point_ID.Value;
      point.PointMin.Value = this.CreditPoints;
      point.PointMax.Value = this.CreditPoints;
      point.Log.Created.Value = dtmNow;
      point.Log.CreatedBy.Value = userID;
      point.Log.Updated.Value = dtmNow;
      point.Log.UpdatedBy.Value = userID;

      try
      {
        courseVersion.Create();
        coursePoint.Create();
        point.Create();
      }
      catch (DalException ex)
      {
        throw new BizException(ex);
      }

      SetDtuVersion();
      GrabSchedule();
      GrabDuration();
      GrabExaminationPlacement();
      GrabExaminationAids();
      GrabPointBlockingCourses();
      GrabPreviousCourseNumbers();
      GrabMandatoryPrerequisites();
      GrabTechnicalPrerequisites();
      GrabDesirablePrerequisites();
      GrabResponsibles();
      GrabExternalInstitutions();
      GrabLastUpdated();

      courseGrab.CourseGrab_ID.Value = Guid.NewGuid();
      courseGrab.CourseVersion_ID.Value = courseVersion.CourseVersion_ID.Value;
      courseGrab.Course_ID.Value = courseVersion.Course_ID.Value;

      courseGrab.Number.Value = courseVersion.Number.Value;

      courseGrab.Log.Created.Value = dtmNow;
      courseGrab.Log.CreatedBy.Value = userID;
      courseGrab.Log.Updated.Value = dtmNow;
      courseGrab.Log.UpdatedBy.Value = userID;

      try
      {
        courseGrab.Create();
      }
      catch (DalException ex)
      {
        throw new BizException(ex);
      }

      GrabStudyTypes();
      GrabRecommendedPlacements();
      GrabKeywords();
      GrabDepartments();
      GrabLecturers();
    }
  }
}

private void GetDocument(string cultureID, string documentID)
{
  string baseUrl = "http://www.kurser.dtu.dk/presentation/presentation.asp?";

  string documentUrl;
  documentUrl = baseUrl;
  documentUrl += "menulanguage=" + cultureID;
  documentUrl += "&coursecode=" + this.DtuCourseNumber;

  System.Net.WebRequest objWr = System.Net.WebRequest.Create(documentUrl);
  System.Net.WebResponse objWrs = objWr.GetResponse();
  System.IO.Stream objStream = objWrs.GetResponseStream();
  Encoding enc = Encoding.GetEncoding(1252);
  System.IO.StreamReader objReader = new System.IO.StreamReader(objStream, enc);
  String strDocument = objReader.ReadToEnd();

  int startPos = strDocument.IndexOf("<label class=\"header_white\">",0);
  strDocument = strDocument.Substring(startPos);

  int length = strDocument.Length;

  if (documentID.Equals("da-DK"))
  {
    startPos = strDocument.IndexOf("<td class=\"page\">Sidst opdateret:</td>");
    strDocument = strDocument.Remove(startPos + 100, 1100);
```

```
    }
    else if (documentID.Equals("en"))
    {
      startPos = strDocument.IndexOf("<td class=\"page\">Last updated:</td>");
      strDocument = strDocument.Remove(startPos + 100, 1100);
    }

    if (this.Document.Contains(documentID))
      this.Document[documentID] = strDocument;
    else
      this.Document.Add(documentID, strDocument);
  }

  #region CourseVersion methods

  private void GrabNameAndNumber()
  {
    String strPattern = @"<label class=""header_white"">(?<number>\d{5})(?<fill>[\n\t{6,7}\s{0,1}\n\t{6,7}])(?<name>(\w|\W|.|\s)*?)</label>";
    //String strPattern = @"<label class=""header_white"">(?<number>\d{5})(?<fill>[\n\t{6,7}\s{0,1}\n\t{6,7}])(?<name>(\w|\W|.|\s)*)</label
      >";
    MatchCollection objMatches = Regex.Matches(this.Document["da-DK"].ToString(), strPattern, RegexOptions.IgnoreCase);

    if (objMatches.Count > 0)
    {
      courseVersion.Number.Value = objMatches[0].Groups["number"].ToString();
      courseVersion.Name.Add("da-DK", objMatches[0].Groups["name"].ToString().Trim());
    }

    objMatches = null;
    objMatches = Regex.Matches(this.Document["en"].ToString(), strPattern, RegexOptions.IgnoreCase);

    if (objMatches.Count > 0)
      courseVersion.Name.Add("en", objMatches[0].Groups["name"].ToString().Trim());
  }

  private void GrabLanguage()
  {
    String strPattern;
    strPattern = @"<label class=""description"">Sprog:</label>  <label class=""value"">(?<language>[a-z]*)</label>";
    MatchCollection objMatches = Regex.Matches(this.Document["da-DK"].ToString(), strPattern, RegexOptions.IgnoreCase);

    string languageName, languageID;
    languageName = objMatches[0].Groups["language"].ToString();

    if (languageName.Equals("Engelsk"))
      courseVersion.Language_ID.Value = "en";
    else if (languageName.Equals("Dansk"))
      courseVersion.Language_ID.Value = "da";
  }

  private void GrabPoint()
  {
    String strPattern;
    strPattern = @"<label class=""description"">Point \(ECTS\):</label>[\D]*(?<point>\d[\d|.]*?)</td>";
    MatchCollection objMatches = Regex.Matches(this.Document["da-DK"].ToString(), strPattern, RegexOptions.IgnoreCase);

    this.CreditPoints = Convert.ToSingle(objMatches[0].Groups["point"].Value.ToString().Replace(".",","));
    coursePoint.Point_ID.Value = Guid.NewGuid();
    //courseVersion.Point_ID.Value = Guid.NewGuid();
  }

  private void GrabTeachingForm()
  {
    MatchCollection objMatches;

    String strBasePattern;
    strBasePattern = @"$$V1$$</label>(.|\n)*?";
    strBasePattern += @"<label class=""value"">(?<teachingform>(.|\n)*?)</label>";

    string strPattern;

    strPattern = strBasePattern;
    strPattern = strPattern.Replace("$$V1$$", "Undervisningsform:");
    objMatches = Regex.Matches(this.Document["da-DK"].ToString(), strPattern, RegexOptions.Singleline);
    if (objMatches.Count > 0)
    {
      string strContents = "";
      strContents = objMatches[0].Groups["teachingform"].ToString();
      strContents = strContents.Replace("<br xmlns=\"\">", "<br/>");
      courseVersion.TeachingForm.Add("da-DK", strContents);
    }

    strPattern = strBasePattern;
    strPattern = strPattern.Replace("$$V1$$", "Scope and form:");
    objMatches = null;
    objMatches = Regex.Matches(this.Document["en"].ToString(), strPattern, RegexOptions.Singleline);
    if (objMatches.Count > 0)
    {
      string strContents = "";
      strContents = objMatches[0].Groups["teachingform"].ToString();
      strContents = strContents.Replace("<br xmlns=\"\">", "<br/>");
      courseVersion.TeachingForm.Add("en", strContents);
    }
  }

  private void GrabAssessmentType()
  {
    String strPattern;
    strPattern = @"Bedømmelsesform:</label>(.|\n)*?";
```

```
    strPattern += @"<label class=""value"">(?<evaluationtype>(.|\n)*?)</label>";

    MatchCollection objMatches = Regex.Matches(this.Document["da-DK"].ToString(), strPattern, RegexOptions.Singleline);

    string assessmentType = "";

    if (objMatches.Count > 0)
      assessmentType = objMatches[0].Groups["evaluationtype"].ToString();
    else
      assessmentType = "13-skala, intern censur";

    switch(assessmentType)
    {
      case "13-skala, intern censur":
        courseVersion.AssessmentType_ID.Value = 1;
        break;
      case "13-skala, ekstern censur":
        courseVersion.AssessmentType_ID.Value = 2;
        break;
      case "bestået/ikke bestået, intern censur":
        courseVersion.AssessmentType_ID.Value = 3;
        break;
      case "bestået/ikke bestået, ekstern censur":
        courseVersion.AssessmentType_ID.Value = 4;
        break;
      default:
        courseVersion.AssessmentType_ID.Value = 1;
        break;
    }
  }

  private void GrabEvaluationFormDescription()
  {
    MatchCollection objMatches;

    String strBasePattern;
    strBasePattern = @"$$V1$$</label>(.|\n)*?";
    strBasePattern += @"</td>(.|\n)*?";
    strBasePattern += @"<td>(.|\n)*?";
    strBasePattern += @"<label class=""value"">(?<evaluationform>(.|\n)*?)</label>";

    String strPattern;

    strPattern = strBasePattern;
    strPattern = strPattern.Replace("$$V1$$", "Evalueringsform:");
    objMatches = Regex.Matches(this.Document["da-DK"].ToString(), strPattern, RegexOptions.Singleline);
    if (objMatches.Count > 0)
    {
      string strContents = "";
      strContents = objMatches[0].Groups["evaluationform"].ToString();
      strContents = strContents.Replace("<br xmlns=\"\">", "<br/>");
      courseVersion.EvaluationFormDescription.Add("da-DK", strContents);
    }

    strPattern = strBasePattern;
    strPattern = strPattern.Replace("$$V1$$", "Evaluation:");
    objMatches = null;
    objMatches = Regex.Matches(this.Document["en"].ToString(), strPattern, RegexOptions.Singleline);
    if (objMatches.Count > 0)
    {
      string strContents = "";
      strContents = objMatches[0].Groups["evaluationform"].ToString();
      strContents = strContents.Replace("<br xmlns=\"\">", "<br/>");
      courseVersion.EvaluationFormDescription.Add("en", strContents);
    }
  }

  private void GrabParticipantLimitation()
  {
    string strPattern;

    strPattern = @"Deltagerbegrænsning:</label>(.|\n)*?";
    strPattern += @"</td>(.|\n)*?";
    strPattern += @"<td class=""value"">(?<participantlimitation>(.|\n)*?)</td>";

    MatchCollection objMatches = Regex.Matches(this.Document["da-DK"].ToString(), strPattern, RegexOptions.Singleline);
    if (objMatches.Count > 0)
    {
      string strContent;
      strContent = objMatches[0].Groups["participantlimitation"].ToString();

      strPattern = @"Minimum:(\D)*(?<min>[0-9]*)";
      MatchCollection objMatchMin = Regex.Matches(strContent, strPattern, RegexOptions.Singleline);
      if (objMatchMin.Count > 0)
      {
        courseVersion.ParticipantLimitationMin.Value = Convert.ToInt32(objMatchMin[0].Groups["min"].ToString());
      }

      strPattern = @"Maksimum:(\D)*(?<max>[0-9]*)";
      MatchCollection objMatchMax = Regex.Matches(strContent, strPattern, RegexOptions.Singleline);
      if (objMatchMax.Count > 0)
      {
        string limitationMax = objMatchMax[0].Groups["max"].ToString();

        if (limitationMax.Trim().Length > 0)
          courseVersion.ParticipantLimitationMax.Value = Convert.ToInt32(limitationMax);
      }
    }
  }
```

```
private void GrabObjective()
{
  MatchCollection objMatches;

  string strBasePattern;
  strBasePattern = @"$$V1$$</label>(.|\n)*?";
  strBasePattern += @"<br/>(?<objective>(.|\n)*?)</td>";

  string strPattern;

  strPattern = strBasePattern;
  strPattern = strPattern.Replace("$$V1$$", "Kursusmål:");
  objMatches = Regex.Matches(this.Document["da-DK"].ToString(), strPattern, RegexOptions.Singleline);
  if (objMatches.Count > 0)
  {
    string strContents = "";
    strContents = objMatches[0].Groups["objective"].ToString();
    strContents = strContents.Replace("<br xmlns=\"\">", "<br/>");
    courseVersion.Objective.Add("da-DK", strContents);
  }

  strPattern = strBasePattern;
  strPattern = strPattern.Replace("$$V1$$", "Aim/objectives:");
  objMatches = null;
  objMatches = Regex.Matches(this.Document["en"].ToString(), strPattern, RegexOptions.Singleline);
  if (objMatches.Count > 0)
  {
    string strContents = "";
    strContents = objMatches[0].Groups["objective"].ToString();
    strContents = strContents.Replace("<br xmlns=\"\">", "<br/>");
    courseVersion.Objective.Add("en", strContents);
  }
}

private void GrabContents()
{
  MatchCollection objMatches;

  string strBasePattern;
  strBasePattern = @"$$V1$$</label>(.|\n)*?";
  strBasePattern += @"<br/>(?<contents>(.|\n)*?)</td>";

  string strPattern;

  strPattern = strBasePattern;
  strPattern = strPattern.Replace("$$V1$$", "Kursusindhold:");
  objMatches = Regex.Matches(this.Document["da-DK"].ToString(), strPattern, RegexOptions.Singleline);
  if (objMatches.Count > 0)
  {
    string strContents = "";
    strContents = objMatches[0].Groups["contents"].ToString();
    strContents = strContents.Replace("<br xmlns=\"\">", "<br/>");
    courseVersion.Contents.Add("da-DK", strContents);
  }

  strPattern = strBasePattern;
  strPattern = strPattern.Replace("$$V1$$", "Content:");
  objMatches = null;
  objMatches = Regex.Matches(this.Document["en"].ToString(), strPattern, RegexOptions.Singleline);
  if (objMatches.Count > 0)
  {
    string strContents = "";
    strContents = objMatches[0].Groups["contents"].ToString();
    strContents = strContents.Replace("<br xmlns=\"\">", "<br/>");
    courseVersion.Contents.Add("en", strContents);
  }
}

private void GrabRemarks()
{
  MatchCollection objMatches;

  string strBasePattern;
  strBasePattern = @"$$V1$$</label>(.|\n)*?";
  strBasePattern += @"<br/>(?<remarks>(.|\n)*?)</td>";

  string strPattern;

  strPattern = strBasePattern;
  strPattern = strPattern.Replace("$$V1$$", "Bemærkninger:");
  objMatches = Regex.Matches(this.Document["da-DK"].ToString(), strPattern, RegexOptions.Singleline);
  if (objMatches.Count > 0)
  {
    string strContents = "";
    strContents = objMatches[0].Groups["remarks"].ToString();
    strContents = strContents.Replace("<br xmlns=\"\">", "<br/>");
    courseVersion.Remark.Add("da-DK", strContents);
  }

  strPattern = strBasePattern;
  strPattern = strPattern.Replace("$$V1$$", "Remarks:");
  objMatches = null;
  objMatches = Regex.Matches(this.Document["en"].ToString(), strPattern, RegexOptions.Singleline);
  if (objMatches.Count > 0)
  {
    string strContents = "";
    strContents = objMatches[0].Groups["remarks"].ToString();
    strContents = strContents.Replace("<br xmlns=\"\">", "<br/>");
```

```
    courseVersion.Remark.Add("en", strContents);
  }
}

private void GrabUrl()
{
  MatchCollection objMatches;

  string strPattern;
  strPattern = @"Kursushjemmeside:</td>(.|\n)*?";
  strPattern += @"<td class=""value"">(?<urlcontent>(.|\n)*?)</td>";

  objMatches = Regex.Matches(this.Document["da-DK"].ToString(), strPattern, RegexOptions.Singleline);

  if (objMatches.Count > 0)
  {
    string strContent;
    strContent = objMatches[0].Groups["urlcontent"].ToString();

    strPattern = "";
    strPattern = @">http:([\W]{2})(?<url>([\w|\W]*?))</a>";

    MatchCollection objMatches2 = Regex.Matches(strContent, strPattern, RegexOptions.Singleline);
    if (objMatches2.Count > 0)
    {
      courseVersion.Url.Value = objMatches2[0].Groups["url"].ToString();
    }
  }
}

#endregion //CourseVersion methods

#region GrabCourse methods

private void SetDtuVersion()
{
  MatchCollection objMatches;

  string strPattern;
  strPattern = @"-(?<version>(\d{1,2}))";

  objMatches = Regex.Matches(this.DtuCourseNumber, strPattern, RegexOptions.Singleline);

  if (objMatches.Count > 0)
  {
    int version;
    version = Convert.ToInt32(objMatches[0].Groups["version"].ToString());
    if (version != 0)
      courseGrab.DtuVersion.Value = version;
  }
}

private void GrabSchedule()
{
  MatchCollection objMatches;
  string strPattern;

  string strBasePattern;
  strBasePattern = @"$$V1$$</a>(.|\n)*?";
  strBasePattern += @"</td>(.|\n)*?";
  strBasePattern += @"<td class=""value"" align=""left"">(?<schedule>(.|\n)*?)</td>";

  strPattern = strBasePattern;
  strPattern = strPattern.Replace("$$V1$$", "Skemaplacering:");
  objMatches = Regex.Matches(this.Document["da-DK"].ToString(), strPattern, RegexOptions.Singleline);
  if (objMatches.Count > 0)
  {
    string strContents = "";
    strContents = objMatches[0].Groups["schedule"].ToString();
    strContents = strContents.Replace("<br xmlns=\"\">", "<br/>");
    courseGrab.Schedule.Add("da-DK", strContents);
  }
  else
  {
    strPattern = "";
    strPattern = @"Skemaplacering:</td>(.|\n)*?";
    strPattern += @"<td class=""value"" align=""left"">(?<schedule>(.|\n)*?)</td>";

    objMatches = null;
    objMatches = Regex.Matches(this.Document["da-DK"].ToString(), strPattern, RegexOptions.Singleline);
    if (objMatches.Count > 0)
    {
      string strContents = "";
      strContents = objMatches[0].Groups["schedule"].ToString();
      strContents = strContents.Replace("<br xmlns=\"\">", "<br/>");
      courseGrab.Schedule.Add("da-DK", strContents);
    }
  }

  strPattern = strBasePattern;
  strPattern = strPattern.Replace("$$V1$$", "Schedule:");
  objMatches = null;
  objMatches = Regex.Matches(this.Document["en"].ToString(), strPattern, RegexOptions.Singleline);

  if (objMatches.Count > 0)
  {
    courseGrab.Schedule.Add("en", objMatches[0].Groups["schedule"].ToString());
  }
  else
```

```
    {
      strPattern = "";
      strPattern = @"Schedule:</td>(.|\n)*?";
      strPattern += @"<td class=""value"" align=""left"">(?<schedule>(.|\n)*?)</td>";

      objMatches = null;
      objMatches = Regex.Matches(this.Document["en"].ToString(), strPattern, RegexOptions.Singleline);
      if (objMatches.Count > 0)
      {
        courseGrab.Schedule.Add("en", objMatches[0].Groups["schedule"].ToString());
      }
    }
  }

  private void GrabDuration()
  {
    string strPattern;
    strPattern = @"Kursets varighed:</label>(.|\n)*?";
    strPattern += @"<label class=""value"">(?<duration>(.|\n)*?)</label>";

    MatchCollection objMatches;
    objMatches = Regex.Matches(this.Document["da-DK"].ToString(), strPattern, RegexOptions.Singleline);

    if (objMatches.Count > 0)
    {
      courseGrab.Duration.Value = objMatches[0].Groups["duration"].ToString();
    }
  }

  private void GrabExaminationPlacement()
  {
    MatchCollection objMatches;
    string strPattern;

    string strBasePattern;
    strBasePattern = @"$$V1$$</a>(.|\n)*?";
    strBasePattern += @"</td>(.|\n)*?";
    strBasePattern += @"<td class=""value"" align=""left"">(?<examinationplacement>(.|\n)*?)</td>";

    strPattern = strBasePattern;
    strPattern = strPattern.Replace("$$V1$$", "Eksamensplacering:");
    objMatches = Regex.Matches(this.Document["da-DK"].ToString(), strPattern, RegexOptions.Singleline);
    if (objMatches.Count > 0)
    {
      string strContents = "";
      strContents = objMatches[0].Groups["examinationplacement"].ToString();
      strContents = strContents.Replace("<br xmlns=\"\">", "<br/>");
      courseGrab.ExaminationPlacement.Add("da-DK", strContents);
    }

    strPattern = strBasePattern;
    strPattern = strPattern.Replace("$$V1$$", "Date of examination:");
    objMatches = null;
    objMatches = Regex.Matches(this.Document["en"].ToString(), strPattern, RegexOptions.Singleline);

    if (objMatches.Count > 0)
    {
      string strContents = "";
      strContents = objMatches[0].Groups["examinationplacement"].ToString();
      strContents = strContents.Replace("<br xmlns=\"\">", "<br/>");
      courseGrab.ExaminationPlacement.Add("en", strContents);
    }
  }

  private void GrabExaminationAids()
  {
    MatchCollection objMatches;
    string strPattern;

    string strBasePattern;
    strBasePattern = @"$$V1$$</label>(.|\n)*?";
    strBasePattern += @"<label class=""value"">(?<examinationaids>(.|\n)*?)</label>";

    strPattern = strBasePattern;
    strPattern = strPattern.Replace("$$V1$$", "Hjælpemidler:");
    objMatches = Regex.Matches(this.Document["da-DK"].ToString(), strPattern, RegexOptions.Singleline);
    if (objMatches.Count > 0)
    {
      string strContents = "";
      strContents = objMatches[0].Groups["examinationaids"].ToString();
      strContents = strContents.Replace("<br xmlns=\"\">", "<br/>");
      courseGrab.ExaminationAids.Add("da-DK", strContents);
    }

    strPattern = strBasePattern;
    strPattern = strPattern.Replace("$$V1$$", "Aid:");
    objMatches = null;
    objMatches = Regex.Matches(this.Document["en"].ToString(), strPattern, RegexOptions.Singleline);

    if (objMatches.Count > 0)
    {
      string strContents = "";
      strContents = objMatches[0].Groups["examinationaids"].ToString();
      strContents = strContents.Replace("<br xmlns=\"\">", "<br/>");
      courseGrab.ExaminationAids.Add("en", strContents);
    }
  }

  private void GrabPointBlockingCourses()
```

```
{
  MatchCollection objMatches;

  string strPattern;
  strPattern = @"Pointspærring:</label>(.|\n)*?";
  strPattern += @"</td>(.|\n)*?";
  strPattern += @"<td>(.|\n)*?";
  strPattern += @"<label class=""value"">(?<pointblocking>(.|\n)*?)</label>";

  objMatches = Regex.Matches(this.Document["da-DK"].ToString(), strPattern, RegexOptions.Singleline);

  if (objMatches.Count > 0)
  {
    courseGrab.PointBlockingCourses.Value = objMatches[0].Groups["pointblocking"].ToString().Trim();
  }
}

private void GrabPreviousCourseNumbers()
{
  MatchCollection objMatches;

  string strPattern;
  strPattern = @"Tidligere kursus:</label>(.|\n)*?";
  strPattern += @"<label class=""value"">(?<versions>(.|\n)*?)</label>";

  objMatches = Regex.Matches(this.Document["da-DK"].ToString(), strPattern, RegexOptions.Singleline);

  if (objMatches.Count > 0)
  {
    courseGrab.PreviousCourseNumbers.Value = objMatches[0].Groups["versions"].ToString();
  }
}

private void GrabMandatoryPrerequisites()
{
  MatchCollection objMatches;

  string strPattern;
  strPattern = @"Obligatoriske forudsætninger:</label>(.|\n)*?";
  strPattern += @"<label class=""value"">(?<mandatorypreq>(.|\n)*?)</label>";

  objMatches = Regex.Matches(this.Document["da-DK"].ToString(), strPattern, RegexOptions.Singleline);

  if (objMatches.Count > 0)
  {
    courseGrab.MandatoryPrerequisites.Value = objMatches[0].Groups["mandatorypreq"].ToString();
  }
}

private void GrabTechnicalPrerequisites()
{
  MatchCollection objMatches;

  string strPattern;
  strPattern = @"Faglige forudsætninger:</label>(.|\n)*?";
  strPattern += @"<label class=""value"">(?<technicalpreq>(.|\n)*?)</label>";

  objMatches = Regex.Matches(this.Document["da-DK"].ToString(), strPattern, RegexOptions.Singleline);

  if (objMatches.Count > 0)
  {
    courseGrab.TechnicalPrerequisites.Value = objMatches[0].Groups["technicalpreq"].ToString();
  }
}

private void GrabDesirablePrerequisites()
{
  MatchCollection objMatches;

  string strPattern;
  strPattern = @"Ønskelige forudsætninger:</label>(.|\n)*?";
  strPattern += @"<label class=""value"">(?<desirablepreq>(.|\n)*?)</label>";

  objMatches = Regex.Matches(this.Document["da-DK"].ToString(), strPattern, RegexOptions.Singleline);

  if (objMatches.Count > 0)
  {
    courseGrab.DesirablePrerequisites.Value = objMatches[0].Groups["desirablepreq"].ToString();
  }
}

private void GrabResponsibles()
{
  string strPattern;
  strPattern = @"Kursusansvarlig:</label>(.|\n)*?";
  strPattern += @"<br/>(?<responsibles>(.|\n)*?)";
  strPattern += @"</td>";

  MatchCollection objMatches = Regex.Matches(this.Document["da-DK"].ToString(), strPattern, RegexOptions.Singleline);

  if (objMatches.Count > 0)
  {
    courseGrab.Responsibles.Value = objMatches[0].Groups["responsibles"].ToString();
  }
}

private void GrabExternalInstitutions()
{
  string strPattern;
```

```
    strPattern = @"Ekstern samarbejdsinstitution:</td>(.|\n)*?";
    strPattern += @"<td class=""value"">(?<externalinstitution>(.|\n)*?)</td>";

    MatchCollection objMatches = Regex.Matches(this.Document["da-DK"].ToString(), strPattern, RegexOptions.Singleline);
    if (objMatches.Count > 0)
    {
      courseGrab.ExternalInstitutions.Value = objMatches[0].Groups["externalinstitution"].ToString();
    }
  }

  private void GrabLastUpdated()
  {
    MatchCollection objMatches;

    string strPattern;
    strPattern = @"Sidst opdateret:</td>(.|\n)*?";
    strPattern += @"<td class=""value"">(?<lastupdated>(.|\n)*?)</td>";

    objMatches = Regex.Matches(this.Document["da-DK"].ToString(), strPattern, RegexOptions.Singleline);

    if (objMatches.Count > 0)
    {
      string strInput;
      strInput = objMatches[0].Groups["lastupdated"].ToString();

      strPattern = "";
      strPattern = @"(?<day>(\d{1,2}))([\W]{2})(?<month>([\w])*?), (?<year>([\d]{4}))"; //>http:([\W]{2})(?<url>([\w|\W]*?))</a>";

      objMatches = null;
      objMatches = Regex.Matches(strInput, strPattern, RegexOptions.Singleline);

      if (objMatches.Count > 0)
      {
        int day;
        day = Convert.ToInt16(objMatches[0].Groups["day"].ToString());

        int month;

        string strMonth;
        strMonth = objMatches[0].Groups["month"].ToString();

        switch(strMonth)
        {
          case "januar":
            month = 1;
            break;
          case "februar":
            month = 2;
            break;
          case "marts":
            month = 3;
            break;
          case "april":
            month = 4;
            break;
          case "maj":
            month = 5;
            break;
          case "juni":
            month = 6;
            break;
          case "juli":
            month = 7;
            break;
          case "august":
            month = 8;
            break;
          case "september":
            month = 9;
            break;
          case "oktober":
            month = 10;
            break;
          case "november":
            month = 11;
            break;
          case "december":
            month = 12;
            break;
          default:
            month = 0;
            break;
        }

        int year;
        year = Convert.ToInt16(objMatches[0].Groups["year"].ToString());

        DateTime dtmLastUpdated = new DateTime(year, month, day);
        courseGrab.LastUpdated.Value = dtmLastUpdated;
      }
    }
  }

  #endregion //GrabCourse methods

  #region Related Information Methods

  private void GrabStudyTypes()
  {
```

```
string strPattern;
strPattern = @"<label class=""page"">Type:  </label>(.|\n)*?";
strPattern += @"<table cellpadding=""0"" cellspacing=""0"" border=""0"">(?<studytypes>(.|\n)*?)</table>";

MatchCollection objMatches = Regex.Matches(this.Document["da-DK"].ToString(), strPattern, RegexOptions.Singleline);

if (objMatches.Count > 0)
{
  string strContent;
  strContent = objMatches[0].Groups["studytypes"].ToString();

  strPattern = "";
  strPattern = @"<tr(.)*?>(?<noname>.|\n{1})";
  strPattern += @"(?<studytype>(.|\n)*?)</tr>";

  objMatches = null;
  objMatches = Regex.Matches(strContent, strPattern, RegexOptions.Singleline);

  if (objMatches.Count > 0)
  {
    foreach(Match objMatch in objMatches)
    {
      strContent = "";
      strContent = objMatch.Groups["studytype"].ToString();

      strPattern = "";
      strPattern = @"<td colspan=""2"" class=""value"">(?<studytype>[a-zA-Z|\w|.|\W|\s]*?)</td>";

      MatchCollection objMatches2 = Regex.Matches(strContent, strPattern, RegexOptions.Singleline);

      if (objMatches2.Count > 0)
      {
        string strType = objMatches2[0].Groups["studytype"].ToString();
        int StudyType_ID = 0;

        switch(strType)
        {
          case "Civil":
            StudyType_ID = 1;
            break;
          case "Diplom":
            StudyType_ID = 2;
            break;
          case "Ph.D.":
            StudyType_ID = 3;
            break;
          case "Levnedsmiddel":
            StudyType_ID = 4;
            break;
          case "Kurset udbydes under Åben Uddannelse":
            StudyType_ID = 5;
            break;
          default:
            break;
        }

        if (StudyType_ID != 0)
        {
          StudyPlanning.DAL.Courses.StudyType studyType =
            new StudyPlanning.DAL.Courses.StudyType();

          studyType.Course_StudyType_ID.Value = Guid.NewGuid();
          studyType.CourseVersion_ID.Value = courseVersion.CourseVersion_ID.Value;
          studyType.StudyType_ID.Value = StudyType_ID;

          studyType.Log.Created.Value = dtmNow;
          studyType.Log.CreatedBy.Value = userID;
          studyType.Log.Updated.Value = dtmNow;
          studyType.Log.UpdatedBy.Value = userID;

          try
          {
            studyType.Create();
          }
          catch (DalException ex)
          {
            throw new BizException(ex);
          }
        }
      }
      else
      {
        strPattern = "";
        strPattern = @"<td class=""value"" width=""75px"">";
        strPattern += @"(?<studytype>(.|\n)*?)</td>";
        strPattern += @"(?<noname>(.|\n)*?)<td>(?<categories>(.|\n)*?)</td>";

        objMatches2 = null;
        objMatches2 = Regex.Matches(strContent, strPattern, RegexOptions.Singleline);

        if (objMatches2.Count > 0)
        {
          string strType = objMatches2[0].Groups["studytype"].ToString();
          int StudyType_ID = 0;

          switch(strType)
          {
            case "Civil:":
              StudyType_ID = 1;
```

```
      break;
    case "Diplom:":
      StudyType_ID = 2;
      break;
    case "Ph.D.:":
      StudyType_ID = 3;
      break;
    case "Levnedsmiddel:":
      StudyType_ID = 4;
      break;
    case "Kurset udbydes under Åben Uddannelse:":
      StudyType_ID = 5;
      break;
    default:
      break;
  }

  if (StudyType_ID != 0)
  {
    StudyPlanning.DAL.Courses.StudyType studyType =
      new StudyPlanning.DAL.Courses.StudyType();

    studyType.Course_StudyType_ID.Value = Guid.NewGuid();
    studyType.CourseVersion_ID.Value = courseVersion.CourseVersion_ID.Value;
    studyType.StudyType_ID.Value = StudyType_ID;

    studyType.Log.Created.Value = dtmNow;
    studyType.Log.CreatedBy.Value = userID;
    studyType.Log.Updated.Value = dtmNow;
    studyType.Log.UpdatedBy.Value = userID;

    try
    {
      studyType.Create();
    }
    catch (DalException ex)
    {
      throw new BizException(ex);
    }

    string strCategories;
    strCategories = objMatches2[0].Groups["categories"].ToString().Trim();

    if (strCategories.Length > 0)
    {
      strPattern = "";
      strPattern += @"(?<category>[\w|\s]*?)([,|<])";

      objMatches2 = null;
      objMatches2 = Regex.Matches(strCategories, strPattern, RegexOptions.Singleline);

      if (objMatches2.Count > 0)
      {
        int Course_StudyTypeCategory_ID;

        foreach(Match objM in objMatches2)
        {
          string strCategory = "";
          strCategory = objM.Groups["category"].ToString().Trim().ToLower();

          Course_StudyTypeCategory_ID = 0;

          if (StudyType_ID == 1)
          {
            switch(strCategory)
            {
              case "initiativkursus":
                Course_StudyTypeCategory_ID = 1;
                break;
              case "basiskursus":
                Course_StudyTypeCategory_ID = 2;
                break;
              default:
                break;
            }
          }
          else if (StudyType_ID == 2)
          {
            switch(strCategory)
            {
              case "kemi":
                Course_StudyTypeCategory_ID = 20;
                break;
              case "bygning":
                Course_StudyTypeCategory_ID = 21;
                break;
              case "arktisk teknologi":
                Course_StudyTypeCategory_ID = 22;
                break;
              case "by- og byg. ing":
                Course_StudyTypeCategory_ID = 23;
                break;
              case "it":
                Course_StudyTypeCategory_ID = 24;
                break;
              case "elektro":
                Course_StudyTypeCategory_ID = 25;
                break;
              case "maskin":
```

```
                        Course_StudyTypeCategory_ID = 26;
                        break;
                      case "matematik":
                        Course_StudyTypeCategory_ID = 27;
                        break;
                      case "fysik og informatik":
                        Course_StudyTypeCategory_ID = 28;
                        break;
                      case "kemi og bioteknologi":
                        Course_StudyTypeCategory_ID = 29;
                        break;
                      case "elektronik og elteknik":
                        Course_StudyTypeCategory_ID = 30;
                        break;
                      default:
                        break;
                    }
                  }
                  else if (StudyType_ID == 3)
                  {
                    switch(strCategory)
                    {
                      case "kemi og bioteknologi":
                        Course_StudyTypeCategory_ID = 50;
                        break;
                      case "konstruktion":
                        Course_StudyTypeCategory_ID = 51;
                        break;
                      case "produktion":
                        Course_StudyTypeCategory_ID = 52;
                        break;
                      case "byggeri":
                        Course_StudyTypeCategory_ID = 53;
                        break;
                      case "miljø og transport":
                        Course_StudyTypeCategory_ID = 54;
                        break;
                      case "matematik":
                        Course_StudyTypeCategory_ID = 55;
                        break;
                      case "fysik og informatik":
                        Course_StudyTypeCategory_ID = 56;
                        break;
                      case "elektronik og elteknik":
                        Course_StudyTypeCategory_ID = 57;
                        break;
                      case "basiskursus":
                        Course_StudyTypeCategory_ID = 58;
                        break;
                      default:
                        break;
                    }
                  }

                  if (Course_StudyTypeCategory_ID != 0)
                  {
                    StudyPlanning.DAL.Courses.StudyType studyTypeCategory =
                      new StudyPlanning.DAL.Courses.StudyType();

                    studyTypeCategory.Course_StudyType_ID.Value = Guid.NewGuid();
                    studyTypeCategory.CourseVersion_ID.Value = courseVersion.CourseVersion_ID.Value;
                    studyTypeCategory.StudyType_ID.Value = StudyType_ID;
                    studyTypeCategory.Course_StudyTypeCategory_ID.Value =
                      Course_StudyTypeCategory_ID;

                    studyTypeCategory.Log.Created.Value = dtmNow;
                    studyTypeCategory.Log.CreatedBy.Value = userID;
                    studyTypeCategory.Log.Updated.Value = dtmNow;
                    studyTypeCategory.Log.UpdatedBy.Value = userID;

                    try
                    {
                      studyTypeCategory.Create();
                    }
                    catch (DalException ex)
                    {
                      throw new BizException(ex);
                    }
                  }
                }
              }
            }
          }
        }
      }
    }
  }
}

private void GrabRecommendedPlacements()
{
  string strPattern;
  strPattern = @"Anbefalet placering:</label>(.|\n)*?";
  strPattern += @"</td>(.|\n)*?";
  strPattern += @"<td class=""value"">(?<recommendedplacement>(.|\n)*?)</td>";

  MatchCollection objMatches = Regex.Matches(this.Document["da-DK"].ToString(), strPattern, RegexOptions.Singleline);
```

```
if (objMatches.Count > 0)
{
  string strContent;
  strContent = objMatches[0].Groups["recommendedplacement"].ToString();

  strPattern = "";
  strPattern = @"(?<recommendation>[a-zA-Z|\w|.|\W|\s]*?)<br/>";

  objMatches = null;
  objMatches = Regex.Matches(strContent, strPattern, RegexOptions.Singleline);

  if (objMatches.Count > 0)
  {
    foreach(Match objMatch in objMatches)
    {
      strContent = objMatch.Groups["recommendation"].ToString() + "<br/>";

      strPattern = "";
      strPattern = @"(?<studytype>[a-zA-Z|\s|.|\W]*?)(\s{1})(?<studytypecontent>[a-zA-Z|\s|\w|\W|.]*?)<br/>";

      MatchCollection objMatches2 = Regex.Matches(strContent, strPattern, RegexOptions.Singleline);

      if (objMatches2.Count > 0)
      {
        string strType = objMatches2[0].Groups["studytype"].ToString();
        int StudyType_ID = 0;

        switch(strType)
        {
          case "Civil:":
            StudyType_ID = 1;
            break;
          case "Diplom:":
            StudyType_ID = 2;
            break;
          case "Ph.D.:":
            StudyType_ID = 3;
            break;
          case "Levnedsmiddel":
            StudyType_ID = 5;
            break;
          default:
            break;
        }

        if (StudyType_ID != 0)
        {
          string strStudyTypeContent = "";
          strStudyTypeContent = objMatches2[0].Groups["studytypecontent"].ToString().Trim();

          if (strStudyTypeContent.Length > 0)
          {

            bool isConcept = false;
            int RecommendedPlacementConcept_ID = 0;

            if (strStudyTypeContent.Equals("først i studiet"))
            {
              isConcept = true;
              RecommendedPlacementConcept_ID = 1;
            }
            else if (strStudyTypeContent.Equals("midt i studiet"))
            {
              isConcept = true;
              RecommendedPlacementConcept_ID = 2;
            }
            else if (strStudyTypeContent.Equals("sidst i studiet"))
            {
              isConcept = true;
              RecommendedPlacementConcept_ID = 3;
            }
            else if (strStudyTypeContent.Equals("På ethvert tidspunkt"))
            {
              isConcept = true;
              RecommendedPlacementConcept_ID = 4;
            }

            if (isConcept)
            {
              StudyPlanning.DAL.Courses.RecommendedPlacement recP =
                new StudyPlanning.DAL.Courses.RecommendedPlacement();

              recP.Course_RecommendedPlacement_ID.Value = Guid.NewGuid();
              recP.CourseVersion_ID.Value = courseVersion.CourseVersion_ID.Value;
              recP.StudyType_ID.Value = StudyType_ID;
              recP.RecommendedPlacementConcept_ID.Value = RecommendedPlacementConcept_ID;

              recP.Log.Created.Value = dtmNow;
              recP.Log.CreatedBy.Value = userID;
              recP.Log.Updated.Value = dtmNow;
              recP.Log.UpdatedBy.Value = userID;

              try
              {
                recP.Create();
              }
              catch (DalException ex)
              {
                throw new BizException(ex);
```

```
                    }
                  }
                  else
                  {
                    strPattern = "";
                    strPattern = @"(?<from>[\d{1}]*?)-(?<to>[\d{1}]*?).(\s{1})semester";

                    MatchCollection objMatches3 = Regex.Matches(strStudyTypeContent, strPattern, RegexOptions.Singleline);

                    if (objMatches3.Count > 0)
                    {
                      StudyPlanning.DAL.Courses.RecommendedPlacement recP =
                        new StudyPlanning.DAL.Courses.RecommendedPlacement();

                      recP.Course_RecommendedPlacement_ID.Value = Guid.NewGuid();
                      recP.CourseVersion_ID.Value = courseVersion.CourseVersion_ID.Value;
                      recP.StudyType_ID.Value = StudyType_ID;
                      recP.Point_ID.Value = Guid.NewGuid();

                      recP.Log.Created.Value = dtmNow;
                      recP.Log.CreatedBy.Value = userID;
                      recP.Log.Updated.Value = dtmNow;
                      recP.Log.UpdatedBy.Value = userID;

                      StudyPlanning.DAL.Point point =
                        new StudyPlanning.DAL.Point();

                      point.Point_ID.Value = recP.Point_ID.Value;
                      point.PointMin.Value = Convert.ToSingle(objMatches3[0].Groups["from"].Value);
                      point.PointMax.Value = Convert.ToSingle(objMatches3[0].Groups["to"].Value);
                      point.Log.Created.Value = dtmNow;
                      point.Log.CreatedBy.Value = userID;
                      point.Log.Updated.Value = dtmNow;
                      point.Log.UpdatedBy.Value = userID;

                      try
                      {
                        recP.Create();
                        point.Create();
                      }
                      catch (DalException ex)
                      {
                        throw new BizException(ex);
                      }
                    }
                    else
                    {
                      strPattern = "";
                      strPattern = @"(?<termnumber>[\d{1}]*?).(\s{1})semester";

                      MatchCollection objMatches4 = Regex.Matches(strStudyTypeContent, strPattern, RegexOptions.Singleline);

                      if (objMatches4.Count > 0)
                      {
                        StudyPlanning.DAL.Courses.RecommendedPlacement recP =
                          new StudyPlanning.DAL.Courses.RecommendedPlacement();

                        recP.Course_RecommendedPlacement_ID.Value = Guid.NewGuid();
                        recP.CourseVersion_ID.Value = courseVersion.CourseVersion_ID.Value;
                        recP.StudyType_ID.Value = StudyType_ID;
                        recP.Point_ID.Value = Guid.NewGuid();

                        recP.Log.Created.Value = dtmNow;
                        recP.Log.CreatedBy.Value = userID;
                        recP.Log.Updated.Value = dtmNow;
                        recP.Log.UpdatedBy.Value = userID;

                        StudyPlanning.DAL.Point point =
                          new StudyPlanning.DAL.Point();

                        point.Point_ID.Value = recP.Point_ID.Value;
                        point.PointMin.Value = Convert.ToSingle(objMatches4[0].Groups["termnumber"].Value);
                        point.PointMax.Value = Convert.ToSingle(objMatches4[0].Groups["termnumber"].Value);
                        point.Log.Created.Value = dtmNow;
                        point.Log.CreatedBy.Value = userID;
                        point.Log.Updated.Value = dtmNow;
                        point.Log.UpdatedBy.Value = userID;

                        try
                        {
                          recP.Create();
                          point.Create();
                        }
                        catch (DalException ex)
                        {
                          throw new BizException(ex);
                        }
                      }
                    }
                  }
                }
              }
            }
          }
        }
      }
    }
  }
}

private void GrabKeywords()
```

```
{
  MatchCollection objMatches;
  string strPattern, keywords;

  string strBasePattern;
  strBasePattern = @"$$V1$$</td>(.|\n)*?";
  strBasePattern += @"<td class=""value"">(?<keyword>(.|\n)*?)</td>";

  strPattern = strBasePattern;
  strPattern = strPattern.Replace("$$V1$$", "Nøgleord:");
  objMatches = Regex.Matches(this.Document["da-DK"].ToString(), strPattern, RegexOptions.Singleline);
  if (objMatches.Count > 0)
  {
    keywords = objMatches[0].Groups["keyword"].ToString();
    keywords += ",";

    strPattern = @"(?<keyword>[\w|\s|\W]*?)([,|<])";

    objMatches = null;
    objMatches = Regex.Matches(keywords, strPattern, RegexOptions.Singleline);
    if (objMatches.Count > 0)
      AddKeyword(objMatches, "da-DK");
    else
    {
      keywords = objMatches[0].Groups["keyword"].ToString();
      keywords += ".";

      strPattern = @"(?<keyword>[\w|\s|\W]*?)([.|<])";

      objMatches = null;
      objMatches = Regex.Matches(keywords, strPattern, RegexOptions.Singleline);
      AddKeyword(objMatches, "da-DK");
    }
  }

  strPattern = strBasePattern;
  strPattern = strPattern.Replace("$$V1$$", "Key words:");
  objMatches = null;
  objMatches = Regex.Matches(this.Document["en"].ToString(), strPattern, RegexOptions.Singleline);

  if (objMatches.Count > 0)
  {
    keywords = "";
    keywords = objMatches[0].Groups["keyword"].ToString();
    keywords += ",";

    strPattern = @"(?<keyword>[\w|\s|\W]*?)([,|<])";

    objMatches = null;
    objMatches = Regex.Matches(keywords, strPattern, RegexOptions.Singleline);
    if (objMatches.Count > 0)
      AddKeyword(objMatches, "en");
    else
    {
      keywords = objMatches[0].Groups["keyword"].ToString();
      keywords += ".";

      strPattern = @"(?<keyword>[\w|\s|\W]*?)([.|<])";

      objMatches = null;
      objMatches = Regex.Matches(keywords, strPattern, RegexOptions.Singleline);
      AddKeyword(objMatches, "en");
    }
  }
}

private void AddKeyword(MatchCollection objMatches, string cultureID)
{
  if (objMatches.Count > 0)
  {
    foreach(Match objMatch in objMatches)
    {
      string strKeyword = "";
      strKeyword = objMatch.Groups["keyword"].ToString().Trim();

      if (strKeyword.Length > 0 && !CreatedKeywords.Contains(strKeyword))
      {

        //string[] arrkeywords = new string[1];
        //arrkeywords[0] = strKeyword;
        Guid[] keyIDs;

        try
        {
          keyIDs = StudyPlanning.DAL.Keyword.SearchForKeywords(strKeyword,cultureID);
        }
        catch (DalException dalEx)
        {
          throw new BizException(dalEx);
        }

        StudyPlanning.DAL.Courses.Keyword courseKeyword =
          new StudyPlanning.DAL.Courses.Keyword();

        if (keyIDs.Length > 0) //the keyword already exists in the database
        {
          courseKeyword.Keyword_ID.Value = keyIDs[0];
        }
        else //keyword does not exist in the database - create it!
```

```
        {
          StudyPlanning.DAL.Keyword keyword = new StudyPlanning.DAL.Keyword();

          keyword.Keyword_ID.Value = Guid.NewGuid();
          keyword.Name.Value = objMatch.Groups["keyword"].ToString().Trim();
          keyword.Culture_ID.Value = cultureID;

          keyword.Log.Created.Value = dtmNow;
          keyword.Log.CreatedBy.Value = userID;
          keyword.Log.Updated.Value = dtmNow;
          keyword.Log.UpdatedBy.Value = userID;

          try
          {
            keyword.Create();
          }
          catch (DalException ex)
          {
            throw new BizException(ex);
          }

          courseKeyword.Keyword_ID.Value = keyword.Keyword_ID.Value;
        }

        courseKeyword.Course_Keyword_ID.Value = Guid.NewGuid();
        courseKeyword.CourseVersion_ID.Value = courseVersion.CourseVersion_ID.Value;

        courseKeyword.Log.Created.Value = dtmNow;
        courseKeyword.Log.CreatedBy.Value = userID;
        courseKeyword.Log.Updated.Value = dtmNow;
        courseKeyword.Log.UpdatedBy.Value = userID;

        try
        {
          courseKeyword.Create();
          CreatedKeywords.Add(strKeyword);
        }
        catch (DalException ex)
        {
          throw new BizException(ex, "Keyword_ID / CourseVersion_ID", objMatch.Groups["keyword"].ToString().Trim() + " / " +
            courseKeyword.Keyword_ID.Value.ToString() + " / " + courseKeyword.Course_Keyword_ID.Value.ToString());
        }
      }
    }
  }
}

private void GrabDepartments()
{
  MatchCollection objMatches;

  string strPattern;
  strPattern = @"Institut:</td>(.|\n)*?";
  strPattern += @"<td class=""value"">(?<department>(.|\n)*?)</td>";

  objMatches = Regex.Matches(this.Document["da-DK"].ToString(), strPattern, RegexOptions.Singleline);
  AddDepartment(objMatches, true);

  strPattern = @"Deltagende institut:</td>(.|\n)*?";
  strPattern += @"<td class=""value"">(?<department>(.|\n)*?)</td>";

  objMatches = null;
  objMatches = Regex.Matches(this.Document["da-DK"].ToString(), strPattern, RegexOptions.Singleline);
  AddDepartment(objMatches, false);
}

private void AddDepartment(MatchCollection objMatches, bool isResponsible)
{
  if (objMatches.Count > 0)
  {
    string strInput = objMatches[0].Groups["department"].ToString();
    strInput += "<br/>";

    string strPattern;
    strPattern = @"(?<num>([0-9]{2}))([\s]{1})(?<name>[a-zA-Z|\s|\W|.]*?)<br/>";

    objMatches = null;
    objMatches = Regex.Matches(strInput, strPattern, RegexOptions.Singleline);
    if (objMatches.Count > 0)
    {
      foreach(Match objMatch in objMatches)
      {
        StudyPlanning.DAL.Courses.Department cdep = new StudyPlanning.DAL.Courses.Department();
        cdep.Course_Department_ID.Value = Guid.NewGuid();

        int depNumber = Convert.ToInt32(objMatch.Groups["num"].ToString());
        int departmentID;

        try
        {
          departmentID = StudyPlanning.DAL.Department.GetDepartmentID(depNumber);
          cdep.Department_ID.Value = departmentID;
        }
        catch (DalException excp)
        {
          throw new BizException(excp);
        }

        cdep.Course_ID.Value = courseVersion.Course_ID.Value;
```

```
            cdep.Responsible.Value = isResponsible;

            cdep.Log.Created.Value = dtmNow;
            cdep.Log.CreatedBy.Value = userID;
            cdep.Log.Updated.Value = dtmNow;
            cdep.Log.UpdatedBy.Value = userID;

            try
            {
              cdep.Create();
            }
            catch (DalException ex)
            {
              throw new BizException(ex);
            }
          }
        }
      }
    }

    private void GrabLecturers()
    {
      string strPattern;
      strPattern = @"Kursusansvarlig:</label>(.|\n)*?";
      strPattern += @"(?<lecturers>(.|\n)*?)";
      strPattern += "</td>";

      MatchCollection objMatches = Regex.Matches(this.Document["da-DK"].ToString(), strPattern, RegexOptions.Singleline);

      if (objMatches.Count > 0)
      {
        string strLecturers = objMatches[0].Groups["lecturers"].ToString();

        strPattern = "";
        strPattern = @"matrikel_id=(?<NUM>[0-9|\d]*?)\D>(?<name>(a-zA-Z|\W|\w|.|\s)*?)</a>, bygn. (?<building>[0-9|a-z|A-Z|.]*), rum (?<room
          >[0-9]*), ([\W]{2}[\d]{2}[\W]{1}[\s]{1})(?<phone>([\d|\s]{9})), <a ([\W|\w]*?)>(?<email>([\W|\w]*?))</a>";
        //strPattern = @"matrikel_id=(?<NUM>[0-9|\d]*?)\D>(?<name>[a-zA-Z|.|\s]*?)</a>, bygn. (?<building>[0-9|a-z|A-Z|.]*), rum (?<
          room>[0-9]*), ([\W]{2}[\d]{2}[\W]{1}[\s]{1})(?<phone>([\d|\s]{9})), <a ([\W|\w]*?)>(?<email>([\W|\w]*?))</a>";
        //strPattern = @"matrikel_id=(?<NUM>[0-9|\d]*?)\D>(?<name>[a-zA-Z|.|\s]*?)</a>, bygn. (?<building>[0-9]*), rum (?<room
          >[0-9]*), ([\W]{2}[\d]{2}[\W]{1}[\s]{1})(?<phone>([\d|\s]{9})), <a ([\W|\w]*?)>(?<email>([\W|\w]*?))</a>";

        objMatches = null;
        objMatches = Regex.Matches(strLecturers, strPattern, RegexOptions.Singleline);

        if (objMatches.Count > 0)
        {
          foreach(Match objMatch in objMatches)
          {
            bool lecturerExists = false;
            int cwisNumber = 0;
            Guid lecturerID = Guid.NewGuid();

            string strCwisNumber = objMatch.Groups["NUM"].ToString().Trim();

            if (strCwisNumber.Length > 0)
            {
              cwisNumber = Convert.ToInt32(strCwisNumber);

              try
              {
                lecturerID = StudyPlanning.Biz.Lecturer.GetLecturerID(cwisNumber);
                lecturerExists = true;
              }
              catch (BizException bizEx)
              {
                if (bizEx.Dal != null)
                {
                  if (bizEx.Dal.Number == 8)
                    lecturerExists = false;
                  else
                    throw bizEx;
                }
              }
            }

            if (lecturerExists)
            {
              StudyPlanning.DAL.Courses.Lecturer clec = new StudyPlanning.DAL.Courses.Lecturer();
              clec.Course_Lecturer_ID.Value = Guid.NewGuid();
              clec.CourseVersion_ID.Value = courseVersion.CourseVersion_ID.Value;
              clec.Lecturer_ID.Value = lecturerID;

              clec.Log.Created.Value = dtmNow;
              clec.Log.CreatedBy.Value = userID;
              clec.Log.Updated.Value = dtmNow;
              clec.Log.UpdatedBy.Value = userID;

              try
              {
                clec.Create();
              }
              catch (DalException ex)
              {
                throw new BizException(ex);
              }
            }
            else
            {
              string strEmail = objMatch.Groups["email"].ToString().Trim();
```

```
string strPhone = objMatch.Groups["room"].ToString().Trim();
string strBuilding = objMatch.Groups["building"].ToString().Trim();
string strRoom = objMatch.Groups["room"].ToString().Trim();
string strName = objMatch.Groups["name"].ToString() + "<br/>";
string givenName = "";
string middleName = "";
string familyName = "";
string initials = "";

strPattern = "";
strPattern = @"(?<givenname>(.)*?)\s";
strPattern += @"(?<middlenames>(.|\s)*?)\s";
strPattern += @"(?<familyname>(\w)+?)<br/>";

MatchCollection objMatches2 = Regex.Matches(strName, strPattern, RegexOptions.Singleline);

if (objMatches2.Count > 0)
{
  givenName = objMatches2[0].Groups["givenname"].ToString().Trim();
  middleName = objMatches2[0].Groups["middlenames"].ToString().Trim();
  familyName = objMatches2[0].Groups["familyname"].ToString().Trim();
}
else
{
  strPattern = @"(?<givenname>(.)*?)\s";
  strPattern += @"(?<familyname>(\w)+?)<br/>";

  objMatches2 = null;
  objMatches2 = Regex.Matches(strName, strPattern, RegexOptions.Singleline);

  if (objMatches2.Count > 0)
  {
    givenName = objMatches2[0].Groups["givenname"].ToString().Trim();
    familyName = objMatches2[0].Groups["familyname"].ToString().Trim();
  }
  else
  {
    givenName = strName;
    familyName = " ";
  }
}

strPattern = strPattern = @"(?<initials>(.)*?)@";
objMatches2 = null;
objMatches2 = Regex.Matches(strEmail, strPattern, RegexOptions.Singleline);

if (objMatches2.Count > 0)
{
  initials = objMatches2[0].Groups["initials"].ToString().ToLower();
}

//USER
StudyPlanning.DAL.Users.User user = new StudyPlanning.DAL.Users.User();

user.User_ID.Value = Guid.NewGuid();
user.Username.Value = strEmail;
user.Password.Value = System.Web.Security.FormsAuthentication.HashPasswordForStoringInCon↓gFile("1pass@word", "
  md5");
user.ChangePasswordAtNextLogon.Value = true;
user.UserCannotChangePassword.Value = false;
user.PasswordNeverExpires.Value = false;
user.PasswordLastChanged.Value = dtmNow;
user.Locked.Value = false;
user.Disabled.Value = false;

user.Log.Created.Value = dtmNow;
user.Log.CreatedBy.Value = userID;
user.Log.Updated.Value = dtmNow;
user.Log.UpdatedBy.Value = userID;

/*
StudyPlanning.Biz.User bizUser = new StudyPlanning.Biz.User();

bizUser.User_ID.Value = Guid.NewGuid();
bizUser.Username.Value = strEmail;
bizUser.Password.Value = System.Web.Security.FormsAuthentication.HashPasswordForStoringInConfigFile("1pass@word", "md5");
bizUser.ChangePasswordAtNextLogon.Value = true;
bizUser.UserCannotChangePassword.Value = false;
bizUser.PasswordNeverExpires.Value = false;
bizUser.PasswordLastChanged.Value = dtmNow;
bizUser.Locked.Value = false;
bizUser.Disabled.Value = false;

bizUser.Log.Created.Value = dtmNow;
bizUser.Log.CreatedBy.Value = userID;
bizUser.Log.Updated.Value = dtmNow;
bizUser.Log.UpdatedBy.Value = userID;
*/

//USER SECURITY ROLE
StudyPlanning.DAL.Users.SecurityRole securityRole =
  new StudyPlanning.DAL.Users.SecurityRole();

securityRole.User_SecurityRole_ID.Value = Guid.NewGuid();
securityRole.User_ID.Value = user.User_ID.Value;
securityRole.SecurityRole_ID.Value = 2;

securityRole.Log.Created.Value = dtmNow;
securityRole.Log.CreatedBy.Value = userID;
```

```
securityRole.Log.Updated.Value = dtmNow;
securityRole.Log.UpdatedBy.Value = userID;

//PERSON
StudyPlanning.DAL.Persons.Person person =
  new StudyPlanning.DAL.Persons.Person();

person.Person_ID.Value = Guid.NewGuid();

person.Log.Created.Value = dtmNow;
person.Log.CreatedBy.Value = userID;
person.Log.Updated.Value = dtmNow;
person.Log.UpdatedBy.Value = userID;

//PERSON VERSION
StudyPlanning.DAL.Persons.PersonVersion personVersion =
  new StudyPlanning.DAL.Persons.PersonVersion();

personVersion.PersonVersion_ID.Value = Guid.NewGuid();
personVersion.Person_ID.Value = person.Person_ID.Value;
personVersion.Version.Value = 1;

personVersion.GivenName.Value = givenName;
if (middleName.Length > 0)
  personVersion.MiddleName.Value = middleName;
if (familyName.Length > 0)
  personVersion.FamilyName.Value = familyName;
if (initials.Length > 0)
  personVersion.Initials.Value = initials;

personVersion.Gender_ID.Value = 0;
personVersion.User_ID.Value = user.User_ID.Value;

personVersion.Log.Created.Value = dtmNow;
personVersion.Log.CreatedBy.Value = userID;
personVersion.Log.Updated.Value = dtmNow;
personVersion.Log.UpdatedBy.Value = userID;

//CONTACT DATA
StudyPlanning.DAL.ContactData contactData =
  new StudyPlanning.DAL.ContactData();

contactData.ContactData_ID.Value = Guid.NewGuid();
contactData.ContactDataType_ID.Value = 2;

contactData.Email.Value = strEmail;
contactData.Phone.Value = strPhone;
contactData.Building.Value = strBuilding;
contactData.Room.Value = strRoom;

contactData.Log.Created.Value = dtmNow;
contactData.Log.CreatedBy.Value = userID;
contactData.Log.Updated.Value = dtmNow;
contactData.Log.UpdatedBy.Value = userID;

//PERSON CONTACT DATA

StudyPlanning.DAL.Persons.ContactData personContactData =
  new StudyPlanning.DAL.Persons.ContactData();

personContactData.Person_ContactData_ID.Value = Guid.NewGuid();
personContactData.ContactData_ID.Value = contactData.ContactData_ID.Value;
personContactData.PersonVersion_ID.Value = personVersion.PersonVersion_ID.Value;

personContactData.Log.Created.Value = dtmNow;
personContactData.Log.CreatedBy.Value = userID;
personContactData.Log.Updated.Value = dtmNow;
personContactData.Log.UpdatedBy.Value = userID;

//LECTURER
StudyPlanning.DAL.Lecturer lecturer =
  new StudyPlanning.DAL.Lecturer();

lecturer.Lecturer_ID.Value = Guid.NewGuid();

lecturer.CwisNumber.Value = cwisNumber;
lecturer.Person_ID.Value = person.Person_ID.Value;

lecturer.Log.Created.Value = dtmNow;
lecturer.Log.CreatedBy.Value = userID;
lecturer.Log.Updated.Value = dtmNow;
lecturer.Log.UpdatedBy.Value = userID;

//COURSE LECTURER
StudyPlanning.DAL.Courses.Lecturer courseLecturer =
  new StudyPlanning.DAL.Courses.Lecturer();

courseLecturer.Course_Lecturer_ID.Value = Guid.NewGuid();
courseLecturer.CourseVersion_ID.Value = courseVersion.CourseVersion_ID.Value;
courseLecturer.Lecturer_ID.Value = lecturer.Lecturer_ID.Value;

courseLecturer.Log.Created.Value = dtmNow;
courseLecturer.Log.CreatedBy.Value = userID;
courseLecturer.Log.Updated.Value = dtmNow;
courseLecturer.Log.UpdatedBy.Value = userID;

try
{
  user.Create();
```

```
                securityRole.Create();
                person.Create();
                personVersion.Create();
                contactData.Create();
                personContactData.Create();
                lecturer.Create();
                courseLecturer.Create();
              }
              catch (DalException ex)
              {
                throw new BizException(ex);
              }
            }
          }
        }
      }
    }

    #endregion //Related Information Methods
  }
}
```

3.3 Lecturer

```
using System;
using System.Collections;
using System.Text;
using System.Text.RegularExpressions;
using StudyPlanning.DAL;

namespace StudyPlanning.Biz
{
  /// <summary>
  /// Grabs courses from the DTU course catalogue.
  /// </summary>
  public class Lecturer : StudyPlanning.Biz.BizObject
  {

    /// <summary>
    /// Creates a new instance of the <see cref="StudyPlanning.Biz.Lecturer" />
    /// class.
    /// </summary>
    public Lecturer() {}

    public static Guid GetLecturerID(int cwisNumber)
    {
      try
      {
        Guid lecturerID =
          StudyPlanning.DAL.Lecturer.GetIDFromCwisNumber(cwisNumber);

        return lecturerID;
      }
      catch (DalException dalEx)
      {
        throw new BizException(dalEx);
      }
    }

  }
}
```

3.4 Project

```
using System;
using System.Collections;
using StudyPlanning.DAL;
using StudyPlanning.Biz;

namespace StudyPlanning.Biz
{
  /// <summary>
  /// Summary description for Project.
  /// </summary>
  public class Project : StudyPlanning.Biz.BizObject
  {
    private DalGuid _Project_ID             = new DalGuid(false);
    private DalString _Number               = new DalString(false);
    private BizStringLocalizable _Name      = new BizStringLocalizable(false);
    private DalInt _ProjectType_ID          = new DalInt(false);
    private BizStringLocalizable _ProjectType_Name  = new BizStringLocalizable(true);
    private DalInt _AssessmentType_ID       = new DalInt(false);
    private BizStringLocalizable _AssessmentType_Name = new BizStringLocalizable(true);
    private DalDateTime _StartDate          = new DalDateTime(false);
    private DalDateTime _EndDate            = new DalDateTime(true);
    private DalInt _Months                  = new DalInt(false);
    private DalFloat _PlacementMin          = new DalFloat(true);
    private DalFloat _PlacementMax          = new DalFloat(true);
    private DalFloat _Points                = new DalFloat(false);
```

```
private DalInt _PeriodType_ID           = new DalInt(false);
private BizStringLocalizable _PeriodType_Name   = new BizStringLocalizable(true);
private BizStringLocalizable _Period_Name     = new BizStringLocalizable(false);
private DataLog _Log                = new DataLog();

#region Public Properties

/// <summary>
/// Gets or sets the value of the Project_ID.
/// </summary>
public Guid Project_ID
{
  get { return _Project_ID.Value; }
  set { _Project_ID.Value = value; }
}

/// <summary>
/// Gets or sets the project number.
/// </summary>
public string Number
{
  get { return _Number.Value; }
  set { _Number.Value = value; }
}

/// <summary>
/// Gets the name of the project using the specified culture.
/// </summary>
public BizStringLocalizable Name
{
  get { return _Name; }
}

/// <summary>
/// Gets or sets the ProjectType_ID.
/// </summary>
public int ProjectType_ID
{
  get { return _ProjectType_ID.Value; }
  set { _ProjectType_ID.Value = value; }
}

/// <summary>
/// Gets the name of the project type using the specified culture.
/// </summary>
public BizStringLocalizable ProjectType_Name
{
  get { return _ProjectType_Name; }
}

/// <summary>
/// Gets or sets the AssessmentType_ID.
/// </summary>
public int AssessmentType_ID
{
  get { return _AssessmentType_ID.Value; }
  set { _AssessmentType_ID.Value = value; }
}

/// <summary>
/// Gets the name of the assessment type.
/// </summary>
public BizStringLocalizable AssessmentType_Name
{
  get { return _AssessmentType_Name; }
}

/// <summary>
/// Gets or sets the start date of the project
/// </summary>
public DateTime StartDate
{
  get { return _StartDate.Value; }
  set { _StartDate.Value = value;
        if (!_Months.IsNull)
          _EndDate.Value = _StartDate.Value.AddMonths(_Months.Value);
      }
}

/// <summary>
/// Gets the end date of the project.
/// </summary>
public DateTime EndDate
{
  get { return _EndDate.Value; }
}

/// <summary>
/// Gets or sets the duration of the project. The end date is calculated
/// automatically as the start date + the duration in months.
/// </summary>
public int Months
{
  get { return _Months.Value; }
  set { _Months.Value = value;
        if (!_StartDate.IsNull)
          _EndDate.Value = _StartDate.Value.AddMonths(_Months.Value);
      }
}
```

```
/// <summary>
/// Gets or sets the minimum number of credit points which should be obtained before
/// scheduling the project.
/// </summary>
public float PlacementMin
{
  get { return _PlacementMin.Value; }
  set { _PlacementMin.Value = value; }
}

/// <summary>
/// Gets or sets the maximum number of credit points which should be obtained before
/// scheduling the project.
/// </summary>
public float PlacementMax
{
  get { return _PlacementMax.Value; }
  set { _PlacementMax.Value = value; }
}

/// <summary>
/// Gets or sets the number of credit points for the project.
/// </summary>
public float Points
{
  get { return _Points.Value; }
  set { _Points.Value = value; }
}

/// <summary>
/// Gets or sets the PeriodType_ID of the project.
/// </summary>
public int PeriodType_ID
{
  get { return _PeriodType_ID.Value; }
  set { _PeriodType_ID.Value = value; }
}

/// <summary>
/// Gets the name of the period type.
/// </summary>
public BizStringLocalizable PeriodType_Name
{
  get { return _PeriodType_Name; }
}

/// <summary>
/// Gets the name of the period which is associated with the project.
/// </summary>
public BizStringLocalizable Period_Name
{
  get { return _Period_Name; }
}

/// <summary>
/// Gets or sets the data log information.
/// </summary>
public DataLog Log
{
  get { return _Log; }
  set { _Log = value; }
}
#endregion

/// <summary>
/// Initializes a new instance of a project.
/// </summary>
public Project() {}

/// <summary>
/// Initializes a new instance of a project using the specified Project_ID.
/// </summary>
/// <param name="Project_ID">Identification of the project.</param>
public Project(Guid Project_ID)
{
  this._Project_ID.Value = Project_ID;
}

/// <summary>
/// Retrieves the project using the Project_ID of the current instance.
/// </summary>
public void Retrieve()
{
  StudyPlanning.DAL.Project project = new StudyPlanning.DAL.Project();
  project.Project_ID.Value = this._Project_ID.Value;

  //Retrieve the project.
  try
  {
    project.Retrieve();
  }
  catch (DalException e)
  {
    throw new BizException(e, "Project_ID", this._Project_ID.Value.ToString());
  }

  StudyPlanning.DAL.ProjectType projecttype = new StudyPlanning.DAL.ProjectType();
  projecttype.ProjectType_ID.Value = project.ProjectType_ID.Value;
```

```
    //Retrieve the projecttype
    try
    {
      projecttype.Retrieve();
    }
    catch (DalException e)
    {
      throw new BizException(e, "ProjectType_ID", project.ProjectType_ID.Value.ToString());
    }

    StudyPlanning.DAL.Point point = new StudyPlanning.DAL.Point();
    point.Point_ID.Value = project.Point_ID.Value;

    //Retrieve the point
    try
    {
      point.Retrieve();
    }
    catch (DalException e)
    {
      throw new BizException(e, "Point_ID", project.Point_ID.Value.ToString());
    }

    StudyPlanning.DAL.AssessmentType assessmenttype =
      new StudyPlanning.DAL.AssessmentType();
    assessmenttype.AssessmentType_ID.Value = project.AssessmentType_ID.Value;

    //Retrieve the assessmenttype
    try
    {
      assessmenttype.Retrieve();
    }
    catch (DalException e)
    {
      throw new BizException(e, "AssessmentType_ID", assessmenttype.AssessmentType_ID.Value.ToString());
    }

    StudyPlanning.DAL.Period period = new StudyPlanning.DAL.Period();
    period.Period_ID.Value = project.Period_ID.Value;

    //Retrieve the period
    try
    {
      period.Retrieve();
    }
    catch (DalException e)
    {
      if (!e.Number.Equals(8))
        throw new BizException(e, "Period_ID", project.Period_ID.Value.ToString());
    }

    StudyPlanning.DAL.PeriodTypes.PeriodType periodtype =
      new StudyPlanning.DAL.PeriodTypes.PeriodType();

    if (!period.PeriodType_ID.IsNull)
    {
      periodtype.PeriodType_ID.Value = period.PeriodType_ID.Value;

      //Retrieve the periodtype
      try
      {
        periodtype.Retrieve();
      }
      catch (DalException e)
      {
        throw new BizException(e, "PeriodType_ID", period.PeriodType_ID.Value.ToString());
      }
    }

    this._AssessmentType_ID.Value = project.AssessmentType_ID.Value;
    this._AssessmentType_Name.LoadXml(assessmenttype.Name.Value);
    this._EndDate.Value = period.EndDate.Value;
    this._Name.LoadXml(project.Name.Value);
    this._Number.Value = project.Number.Value;
    this._Period_Name.LoadXml(period.Name.Value);
    this._PeriodType_ID.Value = period.PeriodType_ID.Value;
    this._PeriodType_Name.LoadXml(periodtype.Name.Value);
    this._Points.Value = point.PointMin.Value;
    this._ProjectType_ID.Value = project.ProjectType_ID.Value;
    this._ProjectType_Name.LoadXml(projecttype.Name.Value);
    this._StartDate.Value = period.StartDate.Value;

    TimeSpan span = period.EndDate.Value - period.StartDate.Value;
    this._Months.Value = span.Days / 30;
  }

  /// <summary>
  /// Creates a new project and its associated period and point objects using
  /// the values of the current instance.
  /// </summary>
  /// <returns>The identification of the newly created
  /// <see cref="StudyPlanning.DAL.Project"/>.</returns>
  public Guid Create()
  {
    this._Project_ID.Value = Guid.NewGuid();
    this.Validate();

    StudyPlanning.DAL.Period period = new StudyPlanning.DAL.Period();
```

```
      period.Period_ID.Value = Guid.NewGuid();
      period.PeriodType_ID.Value = this._PeriodType_ID.Value;
      period.StartDate.Value = this._StartDate.Value;
      period.EndDate.Value = this._StartDate.Value.AddMonths(this._Months.Value);
      period.Name.LoadXml(this._Period_Name.Value);

      period.Log.Created.Value = this._Log.Created.Value;
      period.Log.Updated.Value = this._Log.Updated.Value;
      period.Log.CreatedBy.Value = this._Log.CreatedBy.Value;
      period.Log.UpdatedBy.Value = this._Log.UpdatedBy.Value;

      try
      {
        period.Create();
      }
      catch (DalException e)
      {
        throw new BizException(e);
      }

      StudyPlanning.DAL.Point point = new StudyPlanning.DAL.Point();
      point.Point_ID.Value = Guid.NewGuid();
      point.PointMin.Value = this._Points.Value;
      point.PointMax.Value = this._Points.Value;

      point.Log.Created.Value = this._Log.Created.Value;
      point.Log.Updated.Value = this._Log.Updated.Value;
      point.Log.CreatedBy.Value = this._Log.CreatedBy.Value;
      point.Log.UpdatedBy.Value = this._Log.UpdatedBy.Value;

      try
      {
        point.Create();
      }
      catch (DalException e)
      {
        throw new BizException(e);
      }

      StudyPlanning.DAL.Project project = new StudyPlanning.DAL.Project();
      project.AssessmentType_ID.Value = this._AssessmentType_ID.Value;
      project.Project_ID.Value = this._Project_ID.Value;
      project.ProjectType_ID.Value = this._ProjectType_ID.Value;
      project.Name.LoadXml(this._Name.Value);
      project.Number.Value = this._Number.Value;
      project.Period_ID.Value = period.Period_ID.Value;
      project.Point_ID.Value = point.Point_ID.Value;

      project.Log.Created.Value = this._Log.Created.Value;
      project.Log.Updated.Value = this._Log.Updated.Value;
      project.Log.CreatedBy.Value = this._Log.CreatedBy.Value;
      project.Log.UpdatedBy.Value = this._Log.UpdatedBy.Value;

      try
      {
        project.Create();
      }
      catch (DalException e)
      {
        throw new BizException(e);
      }

      return project.Project_ID.Value;
    }

    /// <summary>
    /// Validates the properties of the project prior to creating or updating.
    /// </summary>
    public void Validate()
    {
      try
      {
        _Project_ID.Validate("Project_ID");
        _Number.Validate("Number");
        _Name.Validate("Name");
        _ProjectType_ID.Validate("ProjectType_ID");
        _ProjectType_Name.Validate("ProjectType_Name");
        _AssessmentType_ID.Validate("AssessmentType_ID");
        _AssessmentType_Name.Validate("AssessmentType_Name");
        _StartDate.Validate("StartDate");
        _EndDate.Validate("EndDate");
        _Months.Validate("Months");
        _Points.Validate("Points");
        _PeriodType_ID.Validate("PeriodType_ID");
        _PeriodType_Name.Validate("PeriodType_Name");
        _Period_Name.Validate("Period_Name");
        _Log.Validate();
      }
      catch (DalException e)
      {
        throw new BizException(e);
      }
    }

  }

}
```

3.5 Security

```
using System;
using StudyPlanning.DAL;

namespace StudyPlanning.Biz.Security
{
  /// <summary>
  /// Represents a password in the business services tier.
  /// </summary>
  public class Password : StudyPlanning.Biz.BizObject
  {
    #region Privat and Public Properties
    private string _value;

    /// <summary>
    /// Gets or sets the value of the password.
    /// </summary>
    public string Value
    {
      get { return _value; }
      set { _value = value; }
    }
    #endregion //Privat and Public Properties

    #region Constructors
    /// <summary>
    /// Initializes a new instance of the <see cref="StudyPlanning.Biz.Security.Password"/>
    /// class.
    /// </summary>
    public Password() {}

    /// <summary>
    /// Initializes a new instance of the <see cref="StudyPlanning.Biz.Security.Password"/>
    /// class with the specified value.
    /// </summary>
    public Password(string value) {}
    #endregion //Constructors

    /// <summary>
    /// Checks whether the password is valid i.e. that
    /// the password does not contain any illegal characters.
    /// </summary>
    /// <param name="password">The password to check whether is valid.</param>
    /// <returns><strong>true</strong>, if the password is valid; <strong>false</strong>, otherwise.</returns>
    public static bool IsValid(string password)
    {
      bool foundIllegalCharacter = false;

      for (int i=0; i < password.Length; i++)
      {
        if (IsIllegalCharacter(password[i]))
          foundIllegalCharacter = true;
      }

      return foundIllegalCharacter;
    }

    /// <summary>
    /// Checks whether the password meets complexity requirements i.e.
    /// that password
    /// </summary>
    /// <param name="password"></param>
    /// <returns><strong>true</strong>, if the password meets complexity requirements; <strong>false</strong>, otherwise.</returns>
    public static bool MeetsComplexityRequirements(string password)
    {
      bool result = false;

      if (ValidateLength(password))
      {
        bool foundLower, foundUpper, foundNumeric, foundSpecialCharacter;
        foundLower = ValidateLowerCase(password);
        foundUpper = ValidateUpperCase(password);
        foundNumeric = ValidateNumeric(password);
        foundSpecialCharacter = ValidateSpecialCharacter(password);

        if (foundLower && foundUpper && foundNumeric)
          result = true;
        else if (foundLower && foundUpper && foundSpecialCharacter)
          result = true;
        else if (foundLower && foundNumeric && foundSpecialCharacter)
          result = true;
        else if (foundUpper && foundNumeric && foundSpecialCharacter)
          result = true;
      }

      return result;
    }

    /// <summary>
    /// Checks whether the password is valid and meets complexity requirements.
    /// </summary>
    /// <param name="password">The password to validate.</param>
    /// <returns>
    ///  <strong>0</strong>, if the password is both valid and meets complexity
    ///  requirements.<br/>
    ///  <strong>1</strong>, if the password is not valid.<br/>
```

```
/// <strong>2</strong>, if the password does not meet complexity requirements.<br/>
/// <strong>3</strong>, if the password is not valid and does not meet
/// complexity requirements.
/// </returns>
public static int Validate(string password)
{
  int result = 0;

  if (IsValid(password))
  {
    if (MeetsComplexityRequirements(password))
      result = 0;
    else
      result = 2;
  }
  else
  {
    if (MeetsComplexityRequirements(password))
      result = 1;
    else
      result = 3;
  }

  return result;
}

/// <summary>
/// Checks if the given string is more than 7 characters long.
/// </summary>
/// <param name="strPassword"></param>
/// <returns></returns>
private static bool ValidateLength(string strPassword)
{
  if (strPassword.Length > 7)
    return true;
  else
    return false;
}

/// <summary>
/// Checks if the given string contains any lower case characters.
/// </summary>
/// <param name="strPassword"></param>
/// <returns></returns>
private static bool ValidateLowerCase(string strPassword)
{
  bool foundLower = false;

  for (int i=0; i < strPassword.Length; i++)
  {
    if (IsLower(strPassword[i]))
      foundLower = true;
  }

  return foundLower;
}

/// <summary>
/// Auxiliary function to validateLowerCase. Checks if the given character is a lower case character.
/// </summary>
/// <param name="character"></param>
/// <returns></returns>
private static bool IsLower(char character)
{
  if (character >= 'a' && character <= 'z')
    return true;
  else
    return false;
}

/// <summary>
/// Checks if the given string contains any upper case characters.
/// </summary>
/// <param name="strPassword"></param>
/// <returns></returns>
private static bool ValidateUpperCase(string strPassword)
{
  bool foundUpper = false;

  for (int i=0; i < strPassword.Length; i++)
  {
    if (IsUpper(strPassword[i]))
      foundUpper = true;
  }

  return foundUpper;
}

/// <summary>
/// Auxiliary function to validateUpperCase. Checks if the given character is an upper case character.
/// </summary>
/// <param name="character"></param>
/// <returns></returns>
private static bool IsUpper(char character)
{
  if (character >= 'A' && character <= 'Z')
    return true;
  else
    return false;
```

```
}

/// <summary>
/// Checks if the given string contains any numeric characters (0−9).
/// </summary>
/// <param name="strPassword"></param>
/// <returns></returns>
private static bool ValidateNumeric(String strPassword)
{
  bool foundNumeric = false;

  for (int i=0; i < strPassword.Length; i++)
  {
    if (IsNumeric(strPassword[i]))
      foundNumeric = true;
  }

  return foundNumeric;
}

/// <summary>
/// Auxiliary function to validateNumeric. Checks if the given character is a numeric character.
/// </summary>
/// <param name="character"></param>
/// <returns></returns>
private static bool IsNumeric(char character)
{
  if (character >= '0' && character <= '9')
    return true;
  else
    return false;
}

/// <summary>
/// Checks if the given string contains any special characters.
/// </summary>
/// <param name="strPassword"></param>
/// <returns></returns>
private static bool ValidateSpecialCharacter(String strPassword)
{
  bool foundSpecialCharacter = false;

  for (int i=0; i < strPassword.Length; i++)
  {
    if (IsSpecialCharacter(strPassword[i]))
      foundSpecialCharacter = true;
  }

  return foundSpecialCharacter;
}

/// <summary>
/// Auxiliary function to validateSpecialCharacter. Checks if the given character is a special character.
/// </summary>
/// <param name="character"></param>
/// <returns></returns>
private static bool IsSpecialCharacter(char character)
{
  if (character == '!')
    return true;
  else if (character == '@')
    return true;
  else if (character == '#')
    return true;
  else if (character == '$')
    return true;
  else if (character == '^')
    return true;
  else if (character == '%')
    return true;
  else if (character == '&')
    return true;
  else if (character == '(')
    return true;
  else if (character == ')')
    return true;
  else if (character == '?')
    return true;
  else if (character == '*')
    return true;
  else if (character == '/')
    return true;
  else if (character == '\\') // The character \ is normally used as an escaping character
    return true;
  else
    return false;
}

/// <summary>
/// Auxiliary function to the IsValid method. Checks if the given character is an
/// illegal character.
/// </summary>
/// <param name="character">The character which should be checked for illegality.</param>
/// <returns><strong>true</strong>, if the specified character is illegal;
/// <strong>false</strong>, otherwise.</returns>
private static bool IsIllegalCharacter(char character)
{
  bool result = false;
  char[] ichars = {'!', '#'};
```

```
      foreach(char ichar in ichars)
      {
        if (character.Equals(ichar))
          result = true;
      }

      return result;
    }

  }
}
```

3.5.1 Role

```
using System;
using StudyPlanning.DAL;

namespace StudyPlanning.Biz.Security
{
  /// <summary>
  /// Represents a user role in the business services tier.
  /// </summary>
  public class Role : StudyPlanning.Biz.BizObject
  {
    /// <summary>
    /// Initializes a new instance of the <see cref="StudyPlanning.Biz.Security.Role"/>
    /// class.
    /// </summary>
    public Role() {}

    /// <summary>
    /// Retrieves a list of permissions which the specified role has been assigned.
    /// </summary>
    /// <param name="role_ID">The ID of the role for which to retrieve a list of permissions.</param>
    /// <returns>An array of integer representing the permissions which the specified role
    /// has been assigned.</returns>
    public static int[] GetPermissions(int role_ID)
    {
      int[] permissions;

      try
      {
        permissions = StudyPlanning.DAL.SecurityRoles.SecurityPermission.GetPermissionsFromRoleID(role_ID);
      }
      catch (DalException dalEx)
      {
        throw new BizException(dalEx);
      }

      return permissions;
    }

    /// <summary>
    /// Checks whether the specified role has the specified permission.
    /// </summary>
    /// <param name="role_ID">The ID of the role.</param>
    /// <param name="permission_ID">The ID of the permission.</param>
    /// <returns><strong>true</strong>, if the specified role has the specified role assigned; <strong>false</strong>, otherwise.</returns>
    public static bool HasPermission(int role_ID, int permission_ID)
    {
      bool result = false;

      int[] permissions = StudyPlanning.Biz.Security.Role.GetPermissions(role_ID);

      foreach(int securityPermission_ID in permissions)
      {
        if (securityPermission_ID == permission_ID)
          result = true;
      }

      return result;
    }

  }
}
```

3.6 Specialization

```
using System;
using System.Collections;
using StudyPlanning.DAL;
using StudyPlanning.DAL.Courses;
using StudyPlanning.DAL.TechnicalLines;
using StudyPlanning.Biz;

namespace StudyPlanning.Biz
{
  /// <summary>
  /// Represents a specialization in the business services tier.
  /// </summary>
```

```
  public class Specialization
  {

    /// <summary>
    /// Gets a list of courses which constitute the specified specialization version.
    /// </summary>
    /// <param name="specializationVersion_ID">A numeric identifier of the specialization version.</param>
    /// <param name="supplementary">An indication of whether the list should be of supplementary courses or not.</param>
    /// <returns>An array of <see cref="StudyPlanning.Biz.CoursePart"/> objects.</returns>
    public static CoursePart[] GetCourses(int specializationVersion_ID, bool supplementary)
    {
      Guid[] courses;
      try
      {
        courses = StudyPlanning.DAL.Specializations.Course.GetCourses(specializationVersion_ID, supplementary);
      }
      catch (DalException e)
      {
        throw new BizException(e, "SpecializationVersion_ID / Supplementary", specializationVersion_ID.ToString() + " / " +
          supplementary);
      }

      CoursePart[] courseParts;

      try
      {
        courseParts = StudyPlanning.Biz.Course.GetCourseParts(courses, 1);
      }
      catch (BizException bizEx)
      {
        throw bizEx;
      }

      return courseParts;
    }

  }
}
```

3.7 Student

```
using System;
using System.Collections;
using System.Data;
using StudyPlanning.DAL;
using StudyPlanning.DAL.Students;
using StudyPlanning.DAL.Courses;

namespace StudyPlanning.Biz
{
  /// <summary>
  /// Represents a student in the business services tier.
  /// </summary>
  public class Student : StudyPlanning.Biz.BizObject
  {
    private DalGuid _StudentVersion_ID = new DalGuid(false);
    private DalGuid _Student_ID      = new DalGuid(false);
    private DalInt _Version         = new DalInt(false);
    private DalString _StudyNumber    = new DalString(false);
    private DalDateTime _DateOfSignUp  = new DalDateTime(false);

    /*
    private BizInt _StudyTypeVersion_ID = new BizInt(false);
    private BizGuid _Person_ID = new BizGuid(false);
    private BizInt _TechnicalPackageVersion_ID = new BizInt(false);
    private BizInt _TechnicalLineVersion_ID = new BizInt(false);
    */

    #region Public Properties

    /// <summary>
    ///
    /// </summary>
    public Guid StudentVersion_ID
    {
      get { return _StudentVersion_ID.Value; }
      set { _StudentVersion_ID.Value = value; }
    }

    /// <summary>
    ///
    /// </summary>
    public Guid Student_ID
    {
      get { return _Student_ID.Value; }
      set { _Student_ID.Value = value; }
    }

    /// <summary>
    ///
    /// </summary>
    public int Version
    {
      get { return _Version.Value; }
```

```
    set { _Version.Value = value; }
  }

  /// <summary>
  ///
  /// </summary>
  public string StudyNumber
  {
    get { return _StudyNumber.Value; }
    set { _StudyNumber.Value = value; }
  }

  /// <summary>
  ///
  /// </summary>
  public DateTime DateOfSignUp
  {
    get { return _DateOfSignUp.Value; }
    set { _DateOfSignUp.Value = value; }
  }

  #endregion

  /// <summary>
  ///
  /// </summary>
  public Student() {}

  /// <summary>
  ///
  /// </summary>
  /// <param name="studentVersion_ID"></param>
  public Student(Guid studentVersion_ID) {}

  /// <summary>
  ///
  /// </summary>
  public void Retrieve()
  {
    StudyPlanning.DAL.Students.StudentVersion objStudent = new StudyPlanning.DAL.Students.StudentVersion();
    objStudent.StudentVersion_ID.Value = _StudentVersion_ID.Value;

    try
    {
      objStudent.Retrieve();
      this._Student_ID.Value = objStudent.Student_ID.Value;
      this._Version.Value = objStudent.Version.Value;
      this._StudyNumber.Value = objStudent.StudyNumber.Value;
      this._DateOfSignUp.Value = objStudent.DateOfSignUp.Value;
    }
    catch (DalException dalEx)
    {
      throw new BizException(dalEx, "StudentVersion_ID", _StudentVersion_ID.Value.ToString());
    }
  }

  /// <summary>
  /// Gets the total number of credit points which the given StudentVersion
  /// has obtained.
  /// </summary>
  /// <param name="studentVersion_ID">The identification of the
  /// <see cref="StudyPlanning.DAL.Students.StudentVersion"/>.</param>
  /// <returns>The sum of credit points from courses which the student has passed.</returns>
  public static float GetTotalPoints(Guid studentVersion_ID)
  {
    Guid[] courses;
    float creditPoints = 0;

    try
    {
      courses = StudyPlanning.DAL.Students.Course.GetCourses(studentVersion_ID);
    }
    catch (DalException dalEx)
    {
      /* If a student has not taken any courses yet - no error */
      if (dalEx.Number == 8)
        courses = new Guid[0];
      else
        throw new BizException(dalEx, "StudentVersion_ID", studentVersion_ID.ToString());
    }

    foreach(Guid objGuid in courses)
    {
      StudyPlanning.DAL.Students.Course objCourse = new StudyPlanning.DAL.Students.Course();
      objCourse.Student_Course_ID.Value = objGuid;

      try
      {
        objCourse.Retrieve();
      }
      catch (DalException e)
      {
        throw new BizException(e, "Student_Course_ID", objGuid.ToString());
      }

      if (!objCourse.Assessment_ID.IsNull)
      {
        Assessment objAssessment = new Assessment();
        objAssessment.Assessment_ID.Value = objCourse.Assessment_ID.Value;
```

```
        try
        {
          objAssessment.Retrieve();
        }
        catch (DalException e)
        {
          throw new BizException(e, "Assessment_ID", objCourse.Assessment_ID.Value.ToString());
        }

        if (objAssessment.PointsCredited.Value)
        {
          CourseVersion objCV = new CourseVersion();
          objCV.CourseVersion_ID.Value = objCourse.CourseVersion_ID.Value;

          try
          {
            objCV.Retrieve();
          }
          catch (DalException e)
          {
            throw new BizException(e, "CourseVersion_ID", objCourse.CourseVersion_ID.Value.ToString());
          }

          int parts = 0;
          parts = objCV.Parts.Value;

          for(int i=1;i<=parts;i++)
          {
            Guid pointID;

            try
            {
              pointID = StudyPlanning.DAL.Courses.Point.GetPoint(objCV.CourseVersion_ID.Value, i);
            }
            catch (DalException objDalEx)
            {
              throw new BizException(objDalEx);
            }

            StudyPlanning.DAL.Point objPoint = new StudyPlanning.DAL.Point();
            objPoint.Point_ID.Value = pointID;

            try
            {
              objPoint.Retrieve();
            }
            catch (DalException e)
            {
              throw new BizException(e, "Point_ID", pointID.ToString());
            }

            creditPoints += objPoint.PointMin.Value;
          }
        }
      }
    }
    return creditPoints;
}

/// <summary>
///
/// </summary>
/// <param name="studentVersion_ID"></param>
/// <returns></returns>
public static int[] GetDepartments(Guid studentVersion_ID)
{
  return new int[0];
}

/// <summary>
/// Checks whether a given student has passed a given course.
/// </summary>
/// <param name="studentVersion_ID">The identification of the
/// <see cref="StudyPlanning.DAL.Students.StudentVersion"/>.</param>
/// <param name="course_ID">The identification of the
/// <see cref="StudyPlanning.DAL.Courses.Course"/>.</param>
/// <returns><strong>True</strong> if the student has passed the course
/// otherwise <strong>false</strong>.</returns>
public static bool HasPassedCourse(Guid studentVersion_ID, Guid course_ID)
{
  Guid[] versions = null;
  Guid[] ids = null;
  bool hasPassed = false;
  StudyPlanning.DAL.Students.Course course = null;
  Assessment assessment = null;

  try
  {
    versions = CourseVersion.GetVersions(course_ID);
  }
  catch (DalException e)
  {
    throw new BizException(e, "Course_ID", course_ID.ToString());
  }

  for(int i=0; i < versions.GetLength(0) && !hasPassed; i++)
  {
    try
```

```
    {
      ids = StudyPlanning.DAL.Students.Course.GetCourses(studentVersion_ID, versions[i]);
    }
    catch (DalException e)
    {
      if (e.Number == 8)
        ids = new Guid[0];
      else
        throw new BizException(e, "StudentVersion_ID / CourseVersion_ID",
          studentVersion_ID.ToString() + " / " + versions[i].ToString());
    }

    for(int j=0; j < ids.GetLength(0) && !hasPassed; j++)
    {
      try
      {
        course = StudyPlanning.DAL.Students.Course.Retrieve(ids[j]);
      }
      catch (DalException e)
      {
        throw new BizException(e, "Student_Course_ID", ids[j].ToString());
      }

      try
      {
        assessment = Assessment.Retrieve(course.Assessment_ID.Value);
      }
      catch (DalException e)
      {
        throw new BizException(e, "Assessment_ID", course.Assessment_ID.Value.ToString());
      }
      if (assessment.PointsCredited.Value)
        hasPassed = true;
    }
  }
  return hasPassed;
}

/// <summary>
/// Gets a list of courses which a given student has passed.
/// </summary>
/// <param name="studentVersion_ID">The identification of the
/// <see cref="StudyPlanning.DAL.Students.StudentVersion"/>.</param>
/// <returns>An array of <see cref="StudyPlanning.DAL.Courses.CourseVersion"/> IDs.</returns>
public static Guid[] GetPassedCourses(Guid studentVersion_ID)
{
  Guid[] courses = null;
  Guid[] passedCourses = null;
  ArrayList objPassedCourses = new ArrayList();

  try
  {
    courses = StudyPlanning.DAL.Students.Course.GetCourses(studentVersion_ID);
  }
  catch (DalException dalEx)
  {
    /* If a student has not taken any courses yet - no error */
    if (dalEx.Number == 8)
      courses = new Guid[0];
    else
      throw new BizException(dalEx, "StudentVersion_ID", studentVersion_ID.ToString());
  }

  foreach(Guid objGuid in courses)
  {
    StudyPlanning.DAL.Students.Course objCourse = new StudyPlanning.DAL.Students.Course();
    objCourse.Student_Course_ID.Value = objGuid;

    try
    {
      objCourse.Retrieve();
    }
    catch (DalException e)
    {
      throw new BizException(e, "Student_Course_ID", objGuid.ToString());
    }

    if (!objCourse.Assessment_ID.IsNull)
    {
      Assessment objAssessment = new Assessment();
      objAssessment.Assessment_ID.Value = objCourse.Assessment_ID.Value;

      try
      {
        objAssessment.Retrieve();
      }
      catch (DalException e)
      {
        throw new BizException(e, "Assessment_ID", objCourse.Assessment_ID.Value.ToString());
      }

      if (objAssessment.PointsCredited.Value)
      {
        objPassedCourses.Add(objCourse.CourseVersion_ID.Value);
      }
    }
  }
  passedCourses = new Guid[objPassedCourses.Count];
  for(int i=0; i < passedCourses.GetLength(0); i++)
```

```
    {
      passedCourses[i] = (Guid)objPassedCourses[i];
    }
    return passedCourses;
  }

  /// <summary>
  /// Retrives a list of which parts of a course a student has signed up for − regardless of
  /// whether the course part has been passed or not.
  /// </summary>
  /// <param name="studentVersion_ID">Identification of the <see cref="StudyPlanning.DAL.Students.StudentVersion"/>.</param>
  /// <returns>An array containing <see cref="StudyPlanning.Biz.CoursePart"/> objects.</returns>
  public static CoursePart[] GetCourseParts(Guid studentVersion_ID)
  {
    ArrayList courseParts = new ArrayList();
    Guid[] student_Course_IDs = null;
    try
    {
      student_Course_IDs = StudyPlanning.DAL.Students.Course.GetCourses(studentVersion_ID);
    }
    catch (DalException e)
    {
      if (e.Number == 8) //Not signed up − no error
        return new CoursePart[0];
      else
        throw new BizException(e, "StudentVersion_ID", studentVersion_ID.ToString());
    }

    StudyPlanning.DAL.Students.Course course = new StudyPlanning.DAL.Students.Course();
    IEnumerator scEnum = student_Course_IDs.GetEnumerator();
    while(scEnum.MoveNext())
    {
      course.Student_Course_ID.Value = (Guid)scEnum.Current;
      try
      {
        course.Retrieve();
      }
      catch (DalException e)
      {
        throw new BizException(e, "Student_Course_ID", course.Student_Course_ID.Value.ToString());
      }

      CourseVersion cv = new CourseVersion();
      cv.CourseVersion_ID.Value = course.CourseVersion_ID.Value;
      try
      {
        cv.Retrieve();
      }
      catch (DalException e)
      {
        throw new BizException(e, "CourseVersion_ID", cv.CourseVersion_ID.Value.ToString());
      }

      CoursePart cp = new CoursePart();
      cp.Course_ID = cv.Course_ID.Value;
      cp.CourseVersion_ID = course.CourseVersion_ID.Value;
      cp.Part = course.Part.Value;
      cp.TotalParts = cv.Parts.Value;
      cp.Workload = Course.GetWorkload(course.CourseVersion_ID.Value, course.Part.Value);
      courseParts.Add(cp);
    }
    return (CoursePart[])courseParts.ToArray(System.Type.GetType("StudyPlanning.Biz.CoursePart"));
  }

  /// <summary>
  /// Gets a list of study plans.
  /// </summary>
  /// <param name="student_ID">Identification of the student for which to get a list of study plans.</param>
  /// <returns>A <see cref="System.Data.DataTable"/> object containing the study plans. The
  /// <see cref="System.Data.DataTable"/> has the following columns:
  /// <ul>
  ///   <li>ID</li>
  ///   <li>Name</li>
  ///   <li>Created</li>
  /// </ul>
  /// </returns>
  public static DataTable GetStudyPlans(Guid student_ID)
  {
    Guid[] studyPlans;

    try
    {
      studyPlans = StudyPlanning.DAL.Students.StudyPlan.GetStudyPlans(student_ID);
    }
    catch(DalException dalEx)
    {
      throw new BizException(dalEx);
    }

    DataTable objDataTable = new DataTable();

    try
    {
      objDataTable.Columns.Add("ID", typeof(Guid));
      objDataTable.Columns.Add("Name", typeof(string));
      objDataTable.Columns.Add("Created", typeof(DateTime));
    }
    catch (InvalidExpressionException objEx)
    {
```

```
      throw new BizException(0, objEx.Message);
    }

    DataRow objDataRow;

    foreach(Guid studyPlan_ID in studyPlans)
    {
      StudyPlanning.DAL.StudyPlans.StudyPlan studyPlan =
        new StudyPlanning.DAL.StudyPlans.StudyPlan();
      studyPlan.StudyPlan_ID.Value = studyPlan_ID;

      try
      {
        studyPlan.Retrieve();

        objDataRow = objDataTable.NewRow();
        objDataRow["ID"] = studyPlan.StudyPlan_ID.Value;
        objDataRow["Name"] = studyPlan.Name.Value;
        objDataRow["Created"] = studyPlan.Log.Created.Value;

        try
        {
          objDataTable.Rows.Add(objDataRow);
        }
        catch (Exception)
        {
          //The current study plan is erroneous - continue adding possibly other study plan to the list.
        }
      }
      catch (DalException)
      {
        //throw new BizException(objEx);
      }
    }

    return objDataTable;
  }

  /// <summary>
  /// Gets a list of study plan criteria.
  /// </summary>
  /// <param name="student_ID">Identification of the student for which to get a list of study plan criteria.</param>
  /// <returns>A <see cref="System.Data.DataTable"/> object containing the study plan criteria. The
  /// <see cref="System.Data.DataTable"/> has the following columns:
  /// <ul>
  ///   <li>ID</li>
  ///   <li>Name</li>
  ///   <li>Created</li>
  /// </ul>
  /// </returns>
  public static DataTable GetStudyPlanCriteria(Guid student_ID)
  {
    Guid[] studyPlanCriteria;

    try
    {
      studyPlanCriteria = StudyPlanning.DAL.Students.StudyPlanCriterion.GetStudyPlanCriteria(student_ID);
    }
    catch(DalException dalEx)
    {
      throw new BizException(dalEx);
    }

    DataTable objDataTable = new DataTable();

    try
    {
      objDataTable.Columns.Add("ID", typeof(Guid));
      objDataTable.Columns.Add("Name", typeof(string));
      objDataTable.Columns.Add("Created", typeof(DateTime));
    }
    catch (InvalidExpressionException objEx)
    {
      throw new BizException(0, objEx.Message);
    }

    DataRow objDataRow;

    foreach(Guid studyPlanCriterion_ID in studyPlanCriteria)
    {
      StudyPlanning.DAL.StudyPlanCriteria.StudyPlanCriterion studyPlanCriterion =
        new StudyPlanning.DAL.StudyPlanCriteria.StudyPlanCriterion();
      studyPlanCriterion.StudyPlanCriterion_ID.Value = studyPlanCriterion_ID;

      try
      {
        studyPlanCriterion.Retrieve();

        objDataRow = objDataTable.NewRow();
        objDataRow["ID"] = studyPlanCriterion.StudyPlanCriterion_ID.Value;
        objDataRow["Name"] = studyPlanCriterion.Name.Value;
        objDataRow["Created"] = studyPlanCriterion.Log.Created.Value;

        try
        {
          objDataTable.Rows.Add(objDataRow);
        }
        catch (Exception)
        {
```

```
            //The current study plan criterion seems to be erroneous
            //...continue adding possibly other study plan criteria to the list.
          }
        }
        catch (DalException)
        {
          //Continue adding possibly other study plan criteria to the list.
        }
      }

      return objDataTable;
    }

  }
}
```

3.8 StudyPlan

```
using System;
using System.Collections;
using System.Data;
using StudyPlanning.Biz;
using StudyPlanning.DAL;
using System.Xml.Serialization;

namespace StudyPlanning.Biz
{
  /// <summary>
  /// Summary description for StudyPlan.
  /// </summary>
  public class StudyPlan
  {
    #region Private Properties

    private ArrayList _StudyPlanPeriods = new ArrayList();
    private ArrayList _Log = new ArrayList();

    #endregion //Private Properties

    #region Public Properties

    /// <summary>
    /// Gets or sets the StudyPlanPeriods property of the studyplan.
    /// </summary>
    public ArrayList StudyPlanPeriods
    {
      get { return _StudyPlanPeriods; }
      set { _StudyPlanPeriods = value; }
    }

    /// <summary>
    /// Gets or sets the Log property of the studyplan.
    /// </summary>
    public ArrayList Log
    {
      get { return _Log; }
      set { _Log = value; }
    }

    #endregion //Public Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the StudyPlan class using no values.
    /// </summary>
    public StudyPlan() {}

    #endregion //Constructors

    #region Public Methods

    /// <summary>
    /// Deletes the specified study plan.
    /// </summary>
    /// <param name="studyPlan_ID">Identification of the study plan which should be deleted.</param>
    /// <returns><strong>true</strong>, if the study plan is succesfully delete; <strong>false</strong>, otherwise.</returns>
    public static bool Delete(Guid studyPlan_ID)
    {
      bool result = false;

      StudyPlanning.DAL.StudyPlans.StudyPlan studyPlan =
        new StudyPlanning.DAL.StudyPlans.StudyPlan();

      studyPlan.StudyPlan_ID.Value = studyPlan_ID;

      try
      {
        studyPlan.Retrieve();
        studyPlan.Delete();
      }
      catch (DalException objEx)
      {
        throw new BizException(objEx, "Fejl i forbindelse med sletning på StudyPlan-tabellen", studyPlan_ID.ToString());
```

```
    }

    Guid studentStudyPlan_ID = Guid.Empty;

    try
    {
      studentStudyPlan_ID = StudyPlanning.DAL.Students.StudyPlan.GetID(studyPlan_ID);
    }
    catch (DalException objEx)
    {
      throw new BizException(objEx, "Fejl i forbindelse med hentning af ID på Student_StudyPlan-tabellen", studentStudyPlan_ID.
        ToString());
    }

    StudyPlanning.DAL.Students.StudyPlan studentStudyPlan =
      new StudyPlanning.DAL.Students.StudyPlan();

    studentStudyPlan.Student_StudyPlan_ID.Value = studentStudyPlan_ID;

    try
    {
      studentStudyPlan.Retrieve();
      result = studentStudyPlan.Delete();
    }
    catch (DalException objEx)
    {
      throw new BizException(objEx, "Fejl i forbindelse med sletning på Student_StudyPlan-tabellen", studentStudyPlan_ID.ToString());
    }

    return result;
  }

  /// <summary>
  /// Gets the specified study plan in the specified culture.
  /// </summary>
  /// <param name="studyPlan_ID">The identification of the study plan to return.</param>
  /// <param name="culture_ID">The identification of the culture in which the returned study plan should be.</param>
  /// <returns>A <see cref="System.Data.DataSet"/> object containing the study plan.</returns>
  public static DataSet GetStudyPlan(Guid studyPlan_ID, string culture_ID)
  {
    DataSet objDataSet = new DataSet();
    DataRow objDataRow;

    DataTable objStudyPlan = new DataTable("StudyPlan");

    try
    {
      objStudyPlan.Columns.Add("ID", typeof(Guid));
      objStudyPlan.Columns.Add("Name", typeof(string));
      objStudyPlan.Columns.Add("Created", typeof(DateTime));
    }
    catch (InvalidExpressionException objEx)
    {
      throw new BizException(0, objEx.Message);
    }

    objDataSet.Tables.Add(objStudyPlan);

    StudyPlanning.DAL.StudyPlans.StudyPlan objSp =
      new StudyPlanning.DAL.StudyPlans.StudyPlan();

    objSp.StudyPlan_ID.Value = studyPlan_ID;

    try
    {
      objSp.Retrieve();
    }
    catch (DalException objEx)
    {
      throw new BizException(objEx);
    }

    try
    {
      objDataRow = objStudyPlan.NewRow();
      objDataRow["ID"] = objSp.StudyPlan_ID.Value;
      objDataRow["Name"] = objSp.Name.Value;
      objDataRow["Created"] = objSp.Log.Created.Value;

      objStudyPlan.Rows.Add(objDataRow);
    }
    catch (Exception objEx)
    {
      throw objEx;
    }

    DataTable objPeriods = new DataTable("Periods");

    try
    {
      objPeriods.Columns.Add("StudyPlanPeriod_ID", typeof(Guid));
      objPeriods.Columns.Add("Period_ID", typeof(Guid));
      objPeriods.Columns.Add("Name", typeof(string));
      objPeriods.Columns.Add("StartDate", typeof(DateTime));
      objPeriods.Columns.Add("EndDate", typeof(DateTime));
    }
    catch (InvalidExpressionException objEx)
    {
      throw new BizException(objEx.Message);
```

```
}

objDataSet.Tables.Add(objPeriods);

DataTable objCourses = new DataTable("Courses");

try
{
  objCourses.Columns.Add("StudyPlanPeriod_ID", typeof(Guid));
  objCourses.Columns.Add("CourseVersion_ID", typeof(Guid));
  objCourses.Columns.Add("Part", typeof(int));
  objCourses.Columns.Add("Name", typeof(string));
  objCourses.Columns.Add("Points", typeof(float));
}
catch (InvalidExpressionException objEx)
{
  throw new BizException(objEx.Message);
}

objDataSet.Tables.Add(objCourses);
objDataSet.Relations.Add(
  "Period_Courses",
  objDataSet.Tables["Periods"].Columns["StudyPlanPeriod_ID"],
  objDataSet.Tables["Courses"].Columns["StudyPlanPeriod_ID"]
  );

DataTable objProjects = new DataTable("Projects");

try
{
  objProjects.Columns.Add("StudyPlanPeriod_ID", typeof(Guid));
  objProjects.Columns.Add("Project_ID", typeof(Guid));
  objProjects.Columns.Add("Name", typeof(string));
  objProjects.Columns.Add("Points", typeof(float));
}
catch (InvalidExpressionException objEx)
{
  throw new BizException(objEx.Message);
}

objDataSet.Tables.Add(objProjects);
objDataSet.Relations.Add(
  "Period_Projects",
  objDataSet.Tables["Periods"].Columns["StudyPlanPeriod_ID"],
  objDataSet.Tables["Projects"].Columns["StudyPlanPeriod_ID"]
  );

Guid[] periods;

try
{
  periods = StudyPlanning.DAL.StudyPlans.Period.GetPeriods(studyPlan_ID);
}
catch (DalException objEx)
{
  throw new BizException(objEx);
}

StudyPlanning.DAL.StudyPlans.Period objPlanPeriod =
  new StudyPlanning.DAL.StudyPlans.Period();

#region Iterate through periods
foreach(Guid studyPlanPeriod_ID in periods)
{
  objPlanPeriod.StudyPlan_Period_ID.Value = studyPlanPeriod_ID;

  try
  {
    objPlanPeriod.Retrieve();
  }
  catch(DalException objEx)
  {
    throw new BizException(objEx);
  }

  StudyPlanning.DAL.Project objProject =
    new StudyPlanning.DAL.Project();

  #region Add project
  if (!objPlanPeriod.Project_ID.IsNull)
  {

    objProject.Project_ID.Value = objPlanPeriod.Project_ID.Value;

    try
    {
      objProject.Retrieve();
    }
    catch (DalException objEx)
    {
      throw new BizException(objEx);
    }

    objDataRow = objProjects.NewRow();
    objDataRow["StudyPlanPeriod_ID"] = objPlanPeriod.StudyPlan_Period_ID.Value;
    objDataRow["Project_ID"] = objProject.Project_ID.Value;
    objDataRow["Name"] = objProject.Name[culture_ID];

    StudyPlanning.DAL.Point objPoint =
```

```
      new StudyPlanning.DAL.Point();
    objPoint.Point_ID.Value = objProject.Point_ID.Value;

    try
    {
      objPoint.Retrieve();
    }
    catch(DalException objEx)
    {
      throw new BizException(objEx);
    }

    objDataRow["Points"] = objPoint.PointMin.Value;

    try
    {
      objProjects.Rows.Add(objDataRow);
    }
    catch (Exception objEx)
    {
      throw objEx;
    }

}
#endregion

#region Add period

objDataRow = objPeriods.NewRow();
objDataRow["StudyPlanPeriod_ID"] = studyPlanPeriod_ID;

if (!objPlanPeriod.Period_ID.IsNull)
{
  objDataRow["Period_ID"] = objPlanPeriod.Period_ID.Value;
}
else if (!objPlanPeriod.Project_ID.IsNull)
{
  objDataRow["Period_ID"] = objProject.Project_ID.Value;
}
else
{
  throw new BizException("No period set for study plan period.");
}

StudyPlanning.DAL.Period objPeriod =
  new StudyPlanning.DAL.Period();

objPeriod.Period_ID.Value = (Guid)objDataRow["Period_ID"];

try
{
  objPeriod.Retrieve();

}
catch (DalException objEx)
{
  throw new BizException(objEx);
}

objDataRow["StartDate"] = objPeriod.StartDate.Value;
objDataRow["EndDate"] = objPeriod.EndDate.Value;

try
{
  objPeriods.Rows.Add(objDataRow);
}
catch (Exception objEx)
{
  throw objEx;
}

#endregion //Add period

#region Add courses
Guid[] courses;

try
{
  courses = StudyPlanning.DAL.StudyPlans.PeriodCourse.GetCourses(studyPlanPeriod_ID);
}
catch (DalException objEx)
{
  throw new BizException(objEx);
}

StudyPlanning.DAL.StudyPlans.PeriodCourse objCourse =
  new StudyPlanning.DAL.StudyPlans.PeriodCourse();

foreach(Guid periodCourse_ID in courses)
{
  objCourse.StudyPlan_PeriodCourse_ID.Value = periodCourse_ID;

  try
  {
    objCourse.Retrieve();
  }
  catch (DalException objEx)
  {
    throw new BizException(objEx);
```

```
        }

        objDataRow = objCourses.NewRow();
        objDataRow["StudyPlanPeriod_ID"] = studyPlanPeriod_ID;
        objDataRow["CourseVersion_ID"] = objCourse.CourseVersion_ID.Value;
        objDataRow["Part"] = objCourse.Part.Value;

        StudyPlanning.DAL.Courses.CourseVersion objCourseVersion =
          new StudyPlanning.DAL.Courses.CourseVersion();

        objCourseVersion.CourseVersion_ID.Value = objCourse.CourseVersion_ID.Value;

        try
        {
          objCourseVersion.Retrieve();
        }
        catch (DalException objEx)
        {
          throw new BizException(objEx);
        }

        objDataRow["Name"] = objCourseVersion.Name[culture_ID];

        float points;
        try
        {
          points = StudyPlanning.Biz.Course.GetWorkload(
            objCourse.CourseVersion_ID.Value,
            objCourse.Part.Value
            );
        }
        catch (BizException objEx)
        {
          throw objEx;
        }

        objDataRow["Points"] = points;

        try
        {
          objCourses.Rows.Add(objDataRow);
        }
        catch (Exception objEx)
        {
          throw objEx;
        }
      }
      #endregion //Add courses
    }
    #endregion //Iterate through periods

    return objDataSet;
  }

  #endregion //Public Methods
}

/// <summary>
/// Summary description for StudyPlanPeriod.
/// </summary>
public class StudyPlanPeriod
{
  #region Private Properties

  private ArrayList _CourseParts = new ArrayList();
  private ArrayList _AvailableModules = new ArrayList();
  private Guid _Period;
  private float[] _DesiredWorkload = new float[2];
  private float[] _RemainingWorkload = new float[2];
  private float _TotalWorkload = 0;
  private int _PeriodNumber;

  #endregion //Private Properties

  #region Public Properties

  /// <summary>
  /// Gets or sets the Periods property of the studyplanperiod.
  /// </summary>
  public Guid Period
  {
    get { return _Period; }
    set { _Period = value; }
  }

  /// <summary>
  /// Gets or sets the DesiredWorkload property of the studyplanperiod.
  /// </summary>
  public float[] DesiredWorkload
  {
    get { return _DesiredWorkload; }
    set
    {
      _DesiredWorkload = value;
      _RemainingWorkload[0] += _DesiredWorkload[0];
      _RemainingWorkload[1] += _DesiredWorkload[1]; }
  }

  /// <summary>
```

```
/// Gets the RemainingWorkload of the studyplanperiod.
/// </summary>
public float[] RemainingWorkload
{
  get { return _RemainingWorkload; }
}

/// <summary>
/// Gets the total workload of the studyplanperiod.
/// </summary>
public float TotalWorkload
{
  get { return _TotalWorkload; }
}

/// <summary>
/// Gets or sets the AvailableModules property of the studyplanperiod.
/// </summary>
public ArrayList AvailableModules
{
  get { return _AvailableModules; }
  set { _AvailableModules = value; }
}

/// <summary>
/// Gets or sets the PeriodNumber property of the studyplanperiod.
/// </summary>
public int PeriodNumber
{
  get { return _PeriodNumber; }
  set { _PeriodNumber = value; }
}

/// <summary>
/// Gets the number of CourseParts currently in the studyplanperiod.
/// </summary>
public int NumberOfCourseParts
{
  get { return _CourseParts.Count; }
}

#endregion //Public Properties

#region Constructors

/// <summary>
/// Creates a new instance of the StudyPlanPeriod class using no values.
/// </summary>
public StudyPlanPeriod() {}

/// <summary>
/// Creates a new instance of the StudyPlanPeriod class using the
/// supplied values.
/// </summary>
/// <param name="period">Identification of the <see cref="StudyPlanning.DAL.Period"/>.</param>
/// <param name="courseParts">An arraylist containing the <see cref="StudyPlanning.DAL.Courses.CoursePart"/>
/// objects which have been scheduled for this period.</param>
public StudyPlanPeriod(Guid period, ArrayList courseParts)
{
  this._Period   = period;
  this._CourseParts = courseParts;
}

#endregion //Constructors

#region Public Methods

/// <summary>
/// Adds a CoursePart object to the set of CourseParts in the StudyPlanPeriod.
/// </summary>
/// <param name="cp">The CoursePart object to add.</param>
public void AddCoursePart(CoursePart cp)
{
  _CourseParts.Add(cp);
  _RemainingWorkload[0] = _RemainingWorkload[0] - cp.Workload;
  _RemainingWorkload[1] = _RemainingWorkload[1] - cp.Workload;
  _TotalWorkload += cp.Workload;

  foreach(int module in cp.SelectedModules)
    _AvailableModules.Remove(module);
}

/// <summary>
/// Removes a CoursePart object from the set of CourseParts in the StudyPlanPeriod.
/// </summary>
/// <param name="cp">The CoursePart object to remove.</param>
public void RemoveCoursePart(CoursePart cp)
{
  _CourseParts.Remove(cp);
  _RemainingWorkload[0] = _RemainingWorkload[0] + cp.Workload;
  _RemainingWorkload[1] = _RemainingWorkload[1] + cp.Workload;
  _TotalWorkload -= cp.Workload;

  _AvailableModules.AddRange(cp.SelectedModules);
}

/// <summary>
///
/// </summary>
```

```
    /// <param name="cp"></param>
    /// <returns></returns>
    public bool ContainsCoursePart(CoursePart cp)
    {
      return _CourseParts.Contains(cp);
    }

    /// <summary>
    ///
    /// </summary>
    /// <param name="index"></param>
    /// <returns></returns>
    public CoursePart GetCoursePart(int index)
    {
      return (CoursePart)_CourseParts[index];
    }
    #endregion //Public Methods
  }
}
```

3.9 StudyPlanCriterion

```
using System;
using System.Collections;
using StudyPlanning.DAL;
using StudyPlanning.DAL.Courses;
using StudyPlanning.DAL.StudyPlanCriteria;
using StudyPlanning.DAL.PeriodTypes;
using StudyPlanning.DAL.StudyPlanCriteria.WorkloadPeriods;
using StudyPlanning.DAL.StudyPlanCriteria.WorkloadPeriodTypes;
using StudyPlanning.Biz;

namespace StudyPlanning.Biz
{
  /// <summary>
  /// Summary description for StudyPlanCriterion.
  /// </summary>
  public class StudyPlanCriterion
  {

    /// <summary>
    /// Retrieves a list of the languages selected in a study plan criterion.
    /// </summary>
    /// <param name="studyPlanCriterion_ID">Identification of the
    /// <see cref="StudyPlanning.DAL.StudyPlanCriteria.StudyPlanCriterion"/>.</param>
    /// <returns>An array of <see cref="StudyPlanning.DAL.Language"/> identifications.</returns>
    public static string[] GetLanguages(Guid studyPlanCriterion_ID)
    {
      string[] languages = null;

      try
      {
        languages = StudyPlanning.DAL.StudyPlanCriteria.Language.GetLanguages(studyPlanCriterion_ID);
      }
      catch (DalException e)
      {
        throw new BizException(e, "StudyPlanCriterion_ID", studyPlanCriterion_ID.ToString());
      }
      return languages;
    }

    /// <summary>
    /// Retrieves a list of the lecturers which have been deselected in a study plan criterion
    /// </summary>
    /// <param name="studyPlanCriterion_ID">Identification of the
    /// <see cref="StudyPlanning.DAL.StudyPlanCriteria.StudyPlanCriterion"/>.</param>
    /// <returns>An array of <see cref="StudyPlanning.DAL.Lecturer"/> identifications.</returns>
    public static Guid[] GetLecturers(Guid studyPlanCriterion_ID)
    {
      Guid[] lecturers = null;

      try
      {
        lecturers = StudyPlanning.DAL.StudyPlanCriteria.Lecturer.GetLecturers(studyPlanCriterion_ID);
      }
      catch (DalException e)
      {
        if (e.Number == 8) //No lecturers deselected - no error
          lecturers = new Guid[0];
        else
          throw new BizException(e, "StudyPlanCriterion_ID", studyPlanCriterion_ID.ToString());
      }
      return lecturers;
    }

    /// <summary>
    /// Retrieves a list of study types in a study plan criterion
    /// </summary>
    /// <param name="studyPlanCriterion_ID">Identification of the
    /// <see cref="StudyPlanning.DAL.StudyPlanCriteria.StudyPlanCriterion"/>.</param>
    /// <returns>An array of <see cref="StudyPlanning.DAL.StudyTypes.StudyType"/> identifications.</returns>
    public static int[] GetStudyTypes(Guid studyPlanCriterion_ID)
    {
      int[] studytypes = null;
```

```
  try
  {
    studytypes = StudyPlanning.DAL.StudyPlanCriteria.StudyType.GetStudyTypes(studyPlanCriterion_ID);
  }
  catch (DalException e)
  {
    throw new BizException(e, "StudyPlanCriterion_ID", studyPlanCriterion_ID.ToString());
  }
  return studytypes;
}

/// <summary>
/// Retrieves a list of keywords associated with a study plan criterion.
/// </summary>
/// <param name="studyPlanCriterion_ID">Identification of the
/// <see cref="StudyPlanning.DAL.StudyPlanCriteria.StudyPlanCriterion"/>.</param>
/// <returns>An array of <see cref="StudyPlanning.DAL.Keyword"/> identifications.</returns>
public static Guid[] GetKeywords(Guid studyPlanCriterion_ID)
{
  Guid[] keywords = null;

  try
  {
    keywords = StudyPlanning.DAL.StudyPlanCriteria.Keyword.GetKeywords(studyPlanCriterion_ID);
  }
  catch (DalException e)
  {
    if (e.Number == 8) //No keywords - no error
      keywords = new Guid[0];
    else
      throw new BizException(e, "StudyPlanCriterion_ID", studyPlanCriterion_ID.ToString());
  }
  return keywords;
}

/// <summary>
/// Retrieves the desired workload for a given period in study plan criterion.
/// If no desired workload exists for the given period, the general workload
/// for the period type is returned.
/// </summary>
/// <param name="studyPlanCriterion_ID">Identification of the
/// <see cref="StudyPlanning.DAL.StudyPlanCriteria.StudyPlanCriterion"/>.</param>
/// <param name="period_ID">Identification of the
/// <see cref="StudyPlanning.DAL.Period"/>.</param>
/// <returns>An array of two elements - upper and lower bounds for the desired workload.</returns>
public static float[] GetWorkload(Guid studyPlanCriterion_ID, Guid period_ID)
{
  float[] points = new float[2];
  Guid point_ID = new Guid();
  bool workloadperiodFound = false;

  //Retrieve WorkloadPeriod - need not exist!
  try
  {
    point_ID = WorkloadPeriod.GetPoint(studyPlanCriterion_ID, period_ID);
    workloadperiodFound = true;
  }
  catch (DalException e)
  {
    if (e.Number == 8) //No workloadperiod - no error
      workloadperiodFound = false;
    else
      throw new BizException(e, "StudyPlanCriterion_ID / Period_ID",
        studyPlanCriterion_ID.ToString() + " / " + period_ID.ToString());
  }

  if (workloadperiodFound)
  {
    StudyPlanning.DAL.Point point = new StudyPlanning.DAL.Point();
    point.Point_ID.Value = point_ID;

    //Retrieve Point - must exist!
    try
    {
      point.Retrieve();
    }
    catch (DalException e)
    {
      throw new BizException(e, "Point_ID", point_ID.ToString());
    }
    points[0] = point.PointMin.Value;
    points[1] = point.PointMax.Value;
    return points;
  }
  else
  {
    StudyPlanning.DAL.Period period = new StudyPlanning.DAL.Period();
    period.Period_ID.Value = period_ID;

    //Retrieve Period - must exist!
    try
    {
      period.Retrieve();
    }
    catch (DalException e)
    {
      throw new BizException(e, "Period_ID", period_ID.ToString());
    }
```

```
      //Retrieve WorkloadPeriodType – must exist!
      try
      {
        point_ID = WorkloadPeriodType.GetPoint(studyPlanCriterion_ID, period.PeriodType_ID.Value);
      }
      catch (DalException e)
      {
        throw new BizException(e, "StudyPlanCriterion_ID / PeriodType_ID",
          studyPlanCriterion_ID.ToString() + " / " + period.PeriodType_ID.Value.ToString());
      }

      StudyPlanning.DAL.Point point = new StudyPlanning.DAL.Point();
      point.Point_ID.Value = point_ID;

      //Retrieve Point – must exist!
      try
      {
        point.Retrieve();
      }
      catch (DalException e)
      {
        throw new BizException(e, "Point_ID", point_ID.ToString());
      }
      points[0] = point.PointMin.Value;
      points[1] = point.PointMax.Value;
      return points;
    }
  }

  /// <summary>
  /// Retrieves a list of modules for a given period in a study plan criterion.
  /// </summary>
  /// <param name="studyPlanCriterion_ID">Identification of the
  /// <see cref="StudyPlanning.DAL.StudyPlanCriteria.StudyPlanCriterion"/>.</param>
  /// <param name="period_ID">Identification of the
  /// <see cref="StudyPlanning.DAL.Period"/>.</param>
  /// <returns>An array of <see cref="StudyPlanning.DAL.Module"/> identifications.</returns>
  public static int[] GetModules(Guid studyPlanCriterion_ID, Guid period_ID)
  {
    int[] modules = null;
    int[] generalModules = null;
    int[] speci↓cModules = null;

    bool generalFound = false;
    bool speci↓cFound = false;

    Guid generalID = new Guid();
    Guid speci↓cID = new Guid();

    StudyPlanning.DAL.Period period = new StudyPlanning.DAL.Period();
    period.Period_ID.Value = period_ID;

    //Retrieve period – must exist!!
    try
    {
      period.Retrieve();
    }
    catch (DalException e)
    {
      throw new BizException(e, "Period_ID", period_ID.ToString());
    }

    //Retrieve modules for the given period type – need not exist!!
    try
    {
      modules = StudyPlanning.DAL.PeriodTypes.Module.GetModules(period.PeriodType_ID.Value);
    }
    catch (DalException e)
    {
      if (e.Number == 8)
        modules = new int[0];
      else
        throw new BizException(e, "PeriodType_ID", period.PeriodType_ID.Value.ToString());
    }

    if (modules.GetLength(0) > 0)
    {
      try
      {
        speci↓cID = WorkloadPeriod.GetID(studyPlanCriterion_ID, period_ID);
        speci↓cFound = true;
      }
      catch (DalException e)
      {
        if (e.Number == 8)
          speci↓cFound = false;
        else
          throw new BizException(e, "StudyPlanCriterion_ID / Period_ID",
            studyPlanCriterion_ID.ToString() + " / " + period_ID.ToString());
      }

      if (speci↓cFound)
      {
        //Retrieve specifically deselected modules – need not exist!!
        try
        {
          speci↓cModules = StudyPlanning.DAL.StudyPlanCriteria.WorkloadPeriods.Module.GetDeselectedModules(speci↓cID);
        }
```

```
      catch (DalException e)
      {
        if (e.Number == 8)
          speci↓cModules = new int[0];
        else
          throw new BizException(e, "StudyPlanCriterion_WorkloadPeriod_ID", speci↓cID.ToString());
      }
    }

    //If no specifically deselected modules exist − try generally deselected modules
    if (!speci↓cFound)
    {
      try
      {
        generalID = WorkloadPeriodType.GetID(studyPlanCriterion_ID, period.PeriodType_ID.Value);
        generalFound = true;
      }
      catch (DalException e)
      {
        if (e.Number == 8)
          generalFound = false;
        else
          throw new BizException(e, "StudyPlanCriterion_ID / PeriodType_ID",
            studyPlanCriterion_ID.ToString() + " / " + period.PeriodType_ID.Value.ToString());
      }

      if (generalFound)
      {
        //Retrieve generally deselected modules − need not exist!!
        try
        {
          generalModules = StudyPlanning.DAL.StudyPlanCriteria.WorkloadPeriodTypes.Module.GetDeselectedModules(
            generalID);
        }
        catch (DalException e)
        {
          if (e.Number == 8)
            generalModules = new int[0];
          else
            throw new BizException(e, "StudyPlanCriterion_WorkloadPeriodType_ID", generalID.ToString());
        }
      }
    }

    //Copy modules to temporary arraylist
    ArrayList objModules = new ArrayList(modules);

    //Remove specifically deselected modules
    if (speci↓cFound)
    {
      for (int i=0; i < speci↓cModules.GetLength(0); i++)
      {
        objModules.Remove(speci↓cModules[i]);
      }
    }
      //Remove generally deselected modules
    else if (generalFound)
    {
      for (int i=0; i < generalModules.GetLength(0); i++)
      {
        objModules.Remove(generalModules[i]);
      }
    }
    objModules.TrimToSize();
    modules = (int[])objModules.ToArray(System.Type.GetType("System.Int32"));
    return modules;
  }

  /// <summary>
  /// Retrieves a list of specifically selected courses in a study plan criterion.
  /// </summary>
  /// <param name="studyPlanCriterion_ID">Identification of the
  /// <see cref="StudyPlanning.DAL.StudyPlanCriteria.StudyPlanCriterion"/>.</param>
  /// <returns>An array of <see cref="StudyPlanning.DAL.Courses.Course"/> identifications.</returns>
  public static CoursePart[] GetSelectedCourses(Guid studyPlanCriterion_ID)
  {
    Guid[] courses = null;

    try
    {
      courses = StudyPlanning.DAL.StudyPlanCriteria.Course.GetCourses(studyPlanCriterion_ID, true);
    }
    catch (DalException e)
    {
      if (e.Number == 8) //No selected courses − no error
        courses = new Guid[0];
      else
        throw new BizException(e, "StudyPlanCriterion_ID / AdditionalChoice",
          studyPlanCriterion_ID.ToString() + " / " + true.ToString());
    }
    return Course.GetCourseParts(courses, 1);
  }

  /// <summary>
  /// Retrieves a list of specifically selected courses for a given period in a study plan criterion.
  /// </summary>
  /// <param name="studyPlanCriterion_ID">Identification of the
  /// <see cref="StudyPlanning.DAL.StudyPlanCriteria.StudyPlanCriterion"/>.</param>
```

```
/// <param name="period_ID">Identification of the
/// <see cref="StudyPlanning.DAL.Period"/>.</param>
/// <returns>An array of <see cref="StudyPlanning.DAL.Courses.Course"/> identifications.</returns>
public static CoursePart[] GetSelectedCourses(Guid studyPlanCriterion_ID, Guid period_ID)
{
  Guid[] courses = null;

  try
  {
    courses = StudyPlanning.DAL.StudyPlanCriteria.CoursePeriod.GetCourses(studyPlanCriterion_ID, period_ID, true);
  }
  catch (DalException e)
  {
    if (e.Number == 8) //No selected courses - no error
      return new CoursePart[0];
    else
      throw new BizException(e, "StudyPlanCriterion_ID / Period_ID / AdditionalChoice",
        studyPlanCriterion_ID.ToString() + " / " + period_ID.ToString() + " / " + true.ToString());
  }
  return Course.GetCourseParts(courses, 1);
}

/// <summary>
/// Retrieves a list of specifically deselected courses in a study plan criterion.
/// </summary>
/// <param name="studyPlanCriterion_ID">Identification of the
/// <see cref="StudyPlanning.DAL.StudyPlanCriteria.StudyPlanCriterion"/>.</param>
/// <returns>An array of <see cref="StudyPlanning.DAL.Courses.Course"/> identifications.</returns>
public static Guid[] GetDeSelectedCourses(Guid studyPlanCriterion_ID)
{
  Guid[] courses = null;

  try
  {
    courses = StudyPlanning.DAL.StudyPlanCriteria.Course.GetCourses(studyPlanCriterion_ID, false);
  }
  catch (DalException e)
  {
    if (e.Number == 8) //No deselected courses - no error
      courses = new Guid[0];
    else
      throw new BizException(e, "StudyPlanCriterion_ID / AdditionalChoice",
        studyPlanCriterion_ID.ToString() + " / " + false.ToString());
  }
  return courses;
}

/// <summary>
/// Retrieves a list of specifically deselected courses for a given period in a study plan criterion.
/// </summary>
/// <param name="studyPlanCriterion_ID">Identification of the
/// <see cref="StudyPlanning.DAL.StudyPlanCriteria.StudyPlanCriterion"/>.</param>
/// <param name="period_ID">Identification of the
/// <see cref="StudyPlanning.DAL.Period"/>.</param>
/// <returns>An array of <see cref="StudyPlanning.DAL.Courses.Course"/> identifications.</returns>
public static Guid[] GetDeSelectedCourses(Guid studyPlanCriterion_ID, Guid period_ID)
{
  Guid[] courses = null;

  try
  {
    courses = StudyPlanning.DAL.StudyPlanCriteria.CoursePeriod.GetCourses(studyPlanCriterion_ID, period_ID, false);
  }
  catch (DalException e)
  {
    if (e.Number == 8) //No deselected courses - no error
      courses = new Guid[0];
    else
      throw new BizException(e, "StudyPlanCriterion_ID / Period_ID / AdditionalChoice",
        studyPlanCriterion_ID.ToString() + " / " + period_ID.ToString() + " / " + false.ToString());
  }
  return courses;
}

/// <summary>
/// Retrieves the specified workload for a project and a given period in study plan criterion.
/// </summary>
/// <param name="studyPlanCriterion_ID">Identification of the
/// <see cref="StudyPlanning.DAL.StudyPlanCriteria.StudyPlanCriterion"/>.</param>
/// <returns>An array of <see cref="StudyPlanning.Biz.Project"/> objects.</returns>
public static Project[] GetProjects(Guid studyPlanCriterion_ID)
{
  Guid[] ids = new Guid[0];
  ArrayList projects = new ArrayList();

  try
  {
    ids = StudyPlanning.DAL.StudyPlanCriteria.ProjectWorkload.GetID(studyPlanCriterion_ID);
  }
  catch (DalException e)
  {
    throw new BizException(e, "StudyPlanCriterion_ID", studyPlanCriterion_ID.ToString());
  }

  //Loop through all projects in the SPC
  foreach(Guid id in ids)
  {
    StudyPlanning.DAL.StudyPlanCriteria.ProjectWorkload pw =
      new StudyPlanning.DAL.StudyPlanCriteria.ProjectWorkload();
```

```
pw.StudyPlanCriterion_ProjectWorkload_ID.Value = id;

//Retrieve the project workload
try
{
  pw.Retrieve();
}
catch (DalException e)
{
  throw new BizException(e, "StudyPlanCriterion_ProjectWorkload_ID", id.ToString());
}

//Use only the first instance of each project.
if (!pw.FromPercent.Value.Equals(0))
  continue;

if (pw.TechnicalPackage_Project_ID.IsNull)
{
  StudyPlanning.DAL.StudyTypes.ProjectType projectType =
    new StudyPlanning.DAL.StudyTypes.ProjectType();
  projectType.StudyType_ProjectType_ID.Value = pw.StudyType_ProjectType_ID.Value;

  //Retrieve the StudyType_ProjectType.
  try
  {
    projectType.Retrieve();
  }
  catch (DalException e)
  {
    throw new BizException(e,
      "StudyType_ProjectType_ID",
      pw.StudyType_ProjectType_ID.Value.ToString());
  }

  StudyPlanning.DAL.Point wlPoint = new StudyPlanning.DAL.Point();
  wlPoint.Point_ID.Value = projectType.WorkloadPoint_ID.Value;

  //Retrieve the workload point object
  try
  {
    wlPoint.Retrieve();
  }
  catch (DalException e)
  {
    throw new BizException(e,
      "Point_ID",
      projectType.WorkloadPoint_ID.Value.ToString());
  }

  Project project = new Project();
  project.Points = wlPoint.PointMin.Value;
  project.AssessmentType_ID = 1;
  project.Months = projectType.Months.Value;
  project.PeriodType_ID = 3; //individual period
  project.ProjectType_ID = projectType.ProjectType_ID.Value;
  project.Name["da-DK"] = "";
  project.Number = "";
  project.Period_Name["da-DK"] = "";

  projects.Add(project);

}
else
{
  StudyPlanning.DAL.TechnicalPackages.Project tpProject =
    new StudyPlanning.DAL.TechnicalPackages.Project();
  tpProject.TechnicalPackage_Project_ID.Value = pw.TechnicalPackage_Project_ID.Value;

  //Retrieve the technical package project
  try
  {
    tpProject.Retrieve();
  }
  catch (DalException e)
  {
    throw new BizException(e,
      "TechnicalPackage_Project_ID",
      pw.TechnicalPackage_Project_ID.Value.ToString());
  }

  StudyPlanning.DAL.Point point = new StudyPlanning.DAL.Point();
  point.Point_ID.Value = tpProject.Point_ID.Value;

  //Retrieve the placement of the project
  try
  {
    point.Retrieve();
  }
  catch (DalException e)
  {
    throw new BizException(e, "Point_ID", tpProject.Point_ID.Value.ToString());
  }

  StudyPlanning.Biz.Project project = new StudyPlanning.Biz.Project();
  project.Project_ID = tpProject.Project_ID.Value;

  //Retrieve the project into a biz project.
  project.Retrieve();
```

```
            project.AssessmentType_ID = 1;
            project.Months = tpProject.Months.Value;
            project.Period_Name["da-DK"] = "";
            project.PeriodType_ID = 3; //individual period.
            project.PlacementMin = point.PointMin.Value;
            project.PlacementMax = point.PointMax.Value;

            projects.Add(project);

          }
        }
        return (Project[])projects.ToArray(Type.GetType("StudyPlanning.Biz.Project"));
      }
    }
}
```

3.10 StudyPlanElaborator

```
using System;
using System.Collections;
using System.Collections.Specialized;
using StudyPlanning.DAL;
using StudyPlanning.DAL.Courses;
using StudyPlanning.DAL.TechnicalPackages;
using StudyPlanning.Biz;

namespace StudyPlanning.Biz
{
  /// <summary>
  /// Summary description for StudyPlanElaborator.
  /// </summary>
  public class StudyPlanElaborator
  {
    #region Private Properties

    private Guid _StudentVersion_ID;
    private Guid _StudyPlanCriterion_ID;

    #endregion //Private Properties

    #region Public Properties

    /// <summary>
    /// Gets or set the StudentVersion_ID.
    /// </summary>
    public Guid StudentVersion_ID
    {
      get { return _StudentVersion_ID; }
      set { _StudentVersion_ID = value; }
    }

    /// <summary>
    /// Gets or sets the StudyPlanCriterion_ID.
    /// </summary>
    public Guid StudyPlanCriterion_ID
    {
      get { return _StudyPlanCriterion_ID; }
      set { _StudyPlanCriterion_ID = value; }
    }

    #endregion //Public Properties

    #region Constructors

    /// <summary>
    /// Creates a new instance of the StudyPlanElaborator class using no values.
    /// </summary>
    public StudyPlanElaborator() {}

    /// <summary>
    /// Creates a new instance of the StudyPlanElaborator class using the values
    /// of the supplied properties.
    /// </summary>
    /// <param name="studentVersion_ID">Identification of the
    /// <see cref="StudyPlanning.DAL.Students.StudentVersion"/> for whom the studyplan should be
    /// elaborated.</param>
    /// <param name="studyPlanCriterion_ID">Identification of the
    /// <see cref="StudyPlanning.DAL.StudyPlanCriteria.StudyPlanCriterion"/> which should be
    /// elaborated into a studyplan.</param>
    public StudyPlanElaborator(Guid studentVersion_ID, Guid studyPlanCriterion_ID)
    {
      _StudentVersion_ID   = studentVersion_ID;
      _StudyPlanCriterion_ID = studyPlanCriterion_ID;
    }

    #endregion //Constructors

    #region Public Methods

    /// <summary>
    /// Performs the actual elaboration of a studyplan.
    /// </summary>
    /// <returns>A <see cref="StudyPlanning.Biz.StudyPlan"/>.</returns>
    public StudyPlan Run()
```

```
{
  if (_StudentVersion_ID.Equals(Guid.Empty) || _StudyPlanCriterion_ID.Equals(Guid.Empty))
    throw new BizException("StudentVersion_ID and StudyPlanCriterion_ID must be supplied");

  GetSCDetails();
  GetProjects();
  GetTLCourses();
  GetKeywordMatchingCourses();
  GetSignedUpCourses();

  currentPeriod = GetNextPeriod();
  Guid period = currentPeriod;
  GetTPPreviousCourses(currentPeriod);

  for(int i=0; i < 20; i++)
  {
    InitializeSchedule(period);
    GetTPCourses(period);
    GetTPOptionalCourses(period);

    SchedulePeriod(period);

    //Handle rescheduling
    period = Guid.Empty;
    while(reschedulePeriods.Count > 0)
    {
      period = (Guid)reschedulePeriods[0];
      if (PerformRescheduling(period))
      {
        reschedulePeriods.Remove(period);
        break;
      }
      else
      {
        reschedulePeriods.Remove(period);
        period = Guid.Empty;
      }
    }

    if (period.Equals(Guid.Empty))
    {
      currentPeriod = GetNextPeriod(currentPeriod);
      period = currentPeriod;
    }
  }

  //Create studyplan
  currentPeriod = GetNextPeriod();
  while(!currentPeriod.Equals(Guid.Empty))
  {
    StudyPlanPeriod spp = (StudyPlanPeriod)schedule[currentPeriod];

    if (spp != null)
      studyPlan.StudyPlanPeriods.Add(spp);

    currentPeriod = GetNextPeriod(currentPeriod);
  }
  return studyPlan;
}

#endregion //Public Methods

#region Global Variables

private StudyPlan studyPlan = new StudyPlan();

private Hashtable schedule = new Hashtable();
private Hashtable reSchedule = new Hashtable();

private ArrayList modules = new ArrayList();
private ArrayList languages = new ArrayList();
private ArrayList lecturers = new ArrayList();

private ArrayList reschedulePeriods = new ArrayList();

private ArrayList tpAllParts = new ArrayList();
private ArrayList tpCourseParts = new ArrayList();
private ArrayList tpOptionalCourses = new ArrayList();
private ArrayList tpPreviousCourses = new ArrayList();
private ArrayList tpFundamentalCourses = new ArrayList();
private ArrayList tpPeriods = new ArrayList();

private ArrayList tfCourseParts = new ArrayList();

private ArrayList spSupplementary = new ArrayList();
private ArrayList spPointGiving = new ArrayList();

private ArrayList tlCourseParts = new ArrayList();

private ArrayList deSelectedCourses = new ArrayList();
private ArrayList selectedCourses = new ArrayList();

//Redundant data for faster access.
private Hashtable signedUpCourses = new Hashtable();
private Hashtable scheduledCourses = new Hashtable();

private Guid[][] keywordMatchingCourses;

private int tpVersion = 0;
```

```
private int tlVersion = 0;
private int tfVersion = 0;
private int spVersion = 0;

private Project[] projects;

private int studyType = 0;
private ArrayList studyTypes = new ArrayList();

private Guid currentPeriod = Guid.Empty;
private Hashtable periods = new Hashtable();

private float signedUpTotal = 0;

private enum PeriodTypes { Thirteen = 1, Three, Other };

#endregion

#region Private Methods

/// <summary>
/// Retrieves StudyPlanCriterion details such as chosen technicalpackage,
/// specialization or technical field.
/// </summary>
private void GetSCDetails()
{
  //Retrieve the studyplan criterion
  StudyPlanning.DAL.StudyPlanCriteria.StudyPlanCriterion spc = new StudyPlanning.DAL.StudyPlanCriteria.
    StudyPlanCriterion();
  spc.StudyPlanCriterion_ID.Value = this.StudyPlanCriterion_ID;

  try
  {
    spc.Retrieve();
  }
  catch (DalException e)
  {
    throw new BizException(e, "StudyPlanCriterion_ID", this.StudyPlanCriterion_ID.ToString());
  }

  //Retrieve the student version
  StudyPlanning.DAL.Students.StudentVersion sv = new StudyPlanning.DAL.Students.StudentVersion();
  sv.StudentVersion_ID.Value = this.StudentVersion_ID;

  try
  {
    sv.Retrieve();
  }
  catch (DalException e)
  {
    throw new BizException(e, "StudentVersion_ID", this.StudentVersion_ID.ToString());
  }

  //Retrieve the studytype version
  StudyPlanning.DAL.StudyTypes.StudyTypeVersion stv = new StudyPlanning.DAL.StudyTypes.StudyTypeVersion();
  stv.StudyTypeVersion_ID.Value = sv.StudyTypeVersion_ID.Value;
  try
  {
    stv.Retrieve();
  }
  catch (DalException e)
  {
    throw new BizException(e, "StudyTypeVersion_ID", sv.StudyTypeVersion_ID.Value.ToString());
  }

  //Studytype
  studyType = stv.StudyType_ID.Value;

  //Technical package
  if (spc.TechnicalPackageVersion_ID.IsNull)
    tpVersion = sv.TechnicalPackageVersion_ID.Value;
  else
    tpVersion = spc.TechnicalPackageVersion_ID.Value;

  //Technical line
  if (spc.TechnicalLineVersion_ID.IsNull)
    if (sv.TechnicalLineVersion_ID.IsNull)
      tlVersion = 0;
    else
      tlVersion = sv.TechnicalLineVersion_ID.Value;
  else
    tlVersion = spc.TechnicalLineVersion_ID.Value;

  //Technical field
  if (spc.TechnicalFieldVersion_ID.IsNull)
    tfVersion = 0;
  else
    tfVersion = spc.TechnicalFieldVersion_ID.Value;

  //Specialization
  if (spc.SpecializationVersion_ID.IsNull)
    spVersion = 0;
  else
    spVersion = spc.SpecializationVersion_ID.Value;

  //Get courses for the technical field
  if (tfVersion != 0)
    tfCourseParts.AddRange(TechnicalField.GetCourses(tfVersion));
```

```
    //Get courses for the specialization
    if (spVersion != 0)
    {
      spSupplementary.AddRange(Specialization.GetCourses(spVersion, true));
      spPointGiving.AddRange(Specialization.GetCourses(spVersion, false));
    }

    //Get all courses in a technicalpackage
    CoursePart[][][] tpCourses = TechnicalPackage.GetCourses(tpVersion);
    foreach(CoursePart[][] cpx in tpCourses)
      tpAllParts.AddRange(cpx);

    //Get all periods in a technicalpackage
    tpPeriods.AddRange(TechnicalPackage.GetPeriods(tpVersion));

    //Get all fundamental courses in a technicalpackage
    tpFundamentalCourses.AddRange(TechnicalPackage.GetFundamentalCourses(this.tpVersion));

    //Get deselected courses
    deSelectedCourses.AddRange(StudyPlanCriterion.GetDeSelectedCourses(this.StudyPlanCriterion_ID));

    //Get selected courses
    selectedCourses.AddRange(StudyPlanCriterion.GetSelectedCourses(this.StudyPlanCriterion_ID));

    //Get languages
    languages.AddRange(StudyPlanCriterion.GetLanguages(this.StudyPlanCriterion_ID));

    //Get deselected lecturers
    lecturers.AddRange(StudyPlanCriterion.GetLecturers(this.StudyPlanCriterion_ID));

    //Get studytypes
    studyTypes.AddRange(StudyPlanCriterion.GetStudyTypes(this.StudyPlanCriterion_ID));

  }

  /// <summary>
  /// Retrieves the projects which must be scheduled in the
  /// study plan.
  /// </summary>
  private void GetProjects()
  {
    projects = StudyPlanCriterion.GetProjects(this.StudyPlanCriterion_ID);
  }

  /// <summary>
  ///
  /// </summary>
  /// <param name="period"></param>
  private void GetTPCourses(Guid period)
  {
    tpCourseParts.Clear();
    if (tpPeriods.Contains(period))
    {
      CoursePart[][] tpCourses = null;
      try
      {
        tpCourses = TechnicalPackage.GetCoursesInAPeriod(
          tpVersion,
          period);
      }
      catch (BizException e)
      {
        //Error handling here??
        throw e;
      }
      CoursePart[] cp;
      IEnumerator tpCoursesEnum = tpCourses.GetEnumerator();
      while(tpCoursesEnum.MoveNext())
      {
        //always use 1st alternative course for now!
        cp = ((CoursePart[])tpCoursesEnum.Current);
        if (cp != null)
          tpCourseParts.Add(cp[0]);
      }
    }
  }

  /// <summary>
  /// Retrieves the technicalpackage courses from previous periods up to this period
  /// which have not yet been signed up for.
  /// </summary>
  /// <param name="period"></param>
  private void GetTPPreviousCourses(Guid period)
  {
    foreach(Guid per in tpPeriods)
    {
      if (per.Equals(period))
        break;

      CoursePart[][] tpCourses = null;
      try
      {
        tpCourses = TechnicalPackage.GetCoursesInAPeriod(tpVersion, per);
      }
      catch (BizException e)
      {
        throw e;
      }
      CoursePart[] cp;
```

```
      IEnumerator tpCoursesEnum = tpCourses.GetEnumerator();
      while(tpCoursesEnum.MoveNext())
      {
        //always use 1st alternative course for now!
        cp = ((CoursePart[])tpCoursesEnum.Current);
        if (cp != null)
          tpPreviousCourses.Add(cp[0]);
      }
    }
  }

  /// <summary>
  /// Retrieves the technicalpackage optional courses for the given period.
  /// </summary>
  /// <param name="period"></param>
  private void GetTPOptionalCourses(Guid period)
  {
    tpOptionalCourses.Clear();
    if (tpPeriods.Contains(period))
    {
      try
      {
        tpOptionalCourses.AddRange(TechnicalPackage.GetOptCoursesInAPeriod(this.tpVersion, period));
      }
      catch (BizException e)
      {
        throw e;
      }
    }
  }

  /// <summary>
  /// Retrieves the prerequisite courses for the technical line
  /// </summary>
  private void GetTLCourses()
  {
    if (!tlVersion.Equals(0))
    {
      TechnicalPackageVersion tpv = new TechnicalPackageVersion();
      tpv.TechnicalPackageVersion_ID.Value = tpVersion;

      try
      {
        tpv.Retrieve();
      }
      catch (DalException e)
      {
        throw new BizException(e, "TechnicalPackageVersion_ID", tpVersion.ToString());
      }

      tlCourseParts.AddRange(TechnicalLine.GetPrerequisiteCourses(tlVersion, tpv.TechnicalPackage_ID.Value));
    }
  }

  /// <summary>
  /// Retrieves the courses that have been deselected in the given period - courses that should
  /// not be scheduled in the given period.
  /// </summary>
  /// <param name="period"></param>
  /// <returns></returns>
  private ArrayList GetDeselectedCourses(Guid period)
  {
    ArrayList list = new ArrayList();
    list.AddRange(StudyPlanCriterion.GetDeSelectedCourses(this.StudyPlanCriterion_ID, period));
    return list;
  }

  /// <summary>
  /// Retrieved the specifically selected courses - courses which should be scheduled in the
  /// specified period.
  /// </summary>
  /// <param name="period"></param>
  /// <returns></returns>
  private ArrayList GetSelectedCourses(Guid period)
  {
    ArrayList list = new ArrayList();
    list.AddRange(StudyPlanCriterion.GetSelectedCourses(this.StudyPlanCriterion_ID, period));
    return list;
  }

  /*
  private Guid[][] GetPrerequisiteCourses(Guid courseVersion, int type)
  {
    return Course.GetPrerequisiteCourseCombinations(courseVersion, type, true);
  }
  */
  /// <summary>
  /// Retrieves the type of the specified period.
  /// </summary>
  /// <param name="period"></param>
  /// <returns></returns>
  private int GetPeriodType(Guid period)
  {
    StudyPlanning.DAL.Period p = new StudyPlanning.DAL.Period();
    p.Period_ID.Value = period;

    try
    {
      p.Retrieve();
```

```
  }
  catch (DalException e)
  {
    throw new BizException(e, "Period_ID", period.ToString());
  }
  return p.PeriodType_ID.Value;
}

/// <summary>
/// Retrieves a list of the available modules in the specified period.
/// </summary>
/// <param name="period"></param>
/// <returns></returns>

private ArrayList GetModules(Guid period)
{
  ArrayList modules = new ArrayList();
  try
  {
    modules.AddRange(StudyPlanCriterion.GetModules(this.StudyPlanCriterion_ID, period));
  }
  catch (BizException e)
  {
    //Error handling here??
    throw e;
  }
  return modules;
}

/// <summary>
/// Retrieves the courses which match 1 or more of the (optionally) specified keywords in
/// the studyplan criterion.
/// </summary>
private void GetKeywordMatchingCourses()
{
  Guid[] ids = StudyPlanCriterion.GetKeywords(this._StudyPlanCriterion_ID);
  string[] keywords = new string[ids.GetLength(0)];

  for(int i=0; i < ids.GetLength(0); i++)
  {
    StudyPlanning.DAL.Keyword keyword = new StudyPlanning.DAL.Keyword();
    keyword.Keyword_ID.Value = ids[i];
    try
    {
      keyword.Retrieve();
    }
    catch (DalException e)
    {
      throw new BizException(e, "Keyword_ID", ids[i].ToString());
    }
    keywords[i] = keyword.Name.Value;
  }
  keywordMatchingCourses = Course.GetKeywordMatchingCourses(keywords, "en");
}

/// <summary>
/// Retrieves the identification of the next period relative to the current date – ie. the next
/// period which has a startdate > current date.
/// </summary>
/// <returns></returns>
private Guid GetNextPeriod()
{
  Object obj = periods[Guid.Empty];
  Guid nextPeriod = Guid.Empty;

  if (obj == null)
  {
    try
    {
      nextPeriod = StudyPlanning.DAL.Period.GetNextPeriod(Guid.Empty);
    }
    catch (DalException e)
    {
      if (e.Number == 8)
        return Guid.Empty;
      else
        throw new BizException(e, "Period_ID", Guid.Empty.ToString());
    }
    periods.Add(Guid.Empty, nextPeriod);
  }
  else
  {
    nextPeriod = (Guid)obj;
  }
  return nextPeriod;
}

/// <summary>
/// Retrieves the identification of the next period relative to the specified period – ie.
/// the next period in the chronological sequence.
/// </summary>
/// <param name="period"></param>
/// <returns></returns>

private Guid GetNextPeriod(Guid period)
{
  Object obj = periods[period];
  Guid nextPeriod = Guid.Empty;
```

```
  if (obj == null)
  {
    try
    {
      nextPeriod = StudyPlanning.DAL.Period.GetNextPeriod(period);
    }
    catch (DalException e)
    {
      if (e.Number == 8)
        return Guid.Empty;
      else
        throw new BizException(e, "Period_ID", period.ToString());
    }
    periods.Add(period, nextPeriod);
  }
  else
  {
    nextPeriod = (Guid)obj;
  }
  return nextPeriod;
}

/// <summary>
/// Retrieves the desired workload (min and max points) for the specified period.
/// </summary>
/// <param name="period"></param>
/// <returns></returns>
private float[] GetDesiredWorkload(Guid period)
{
  float[] desiredWorkload = null;
  try
  {
    desiredWorkload = StudyPlanCriterion.GetWorkload(this.StudyPlanCriterion_ID, period);
  }
  catch (BizException e)
  {
    //Error handling here???
    throw e;
  }
  return desiredWorkload;
}

/// <summary>
/// Initializes the schedule by creating a hashtable with an entry for each period from now on
/// and forward. Each entry consists of a studyplan period which is initialized with the
/// desired workload and the available modules for the period, to which it belongs.
/// </summary>
private void InitializeSchedule(Guid period)
{
  if (!schedule.ContainsKey(period))
  {
    StudyPlanPeriod spp = new StudyPlanPeriod();
    spp.Period = period;
    spp.DesiredWorkload = GetDesiredWorkload(period);
    spp.AvailableModules = GetModules(period);
    spp.PeriodNumber = 0;
    schedule.Add(period, spp);
  }
}

/// <summary>
/// Performs the actual scheduling of a period. Courses are scheduled in the sequence shown in
/// the body of the method. When the desired workload has been reached or no more courses are
/// available for the specified period, the method returns.
/// </summary>
/// <param name="period"></param>
private void SchedulePeriod(Guid period)
{
  StudyPlanPeriod spp = null;
  CoursePart cp = null;

  /* Schedule specifically selected courses in this period
   * - move is not allowed
   * - language is checked
   * - lecturer is checked
   * - studytype is checked */
  ArrayList selected = GetSelectedCourses(period);
  for(int i=0; i < selected.Count; i++)
  {
    cp = (CoursePart)selected[i];
    cp.Reason = CoursePartReason.ManuallySelectedPeriod;
    if (LanguageSelected(cp.CourseVersion_ID)
      && !LecturerDeSelected(cp.CourseVersion_ID)
      && StudyTypeSelected(cp.CourseVersion_ID))
      ScheduleChain(cp, period, false, false, false);
  }

  /* Schedule the recommended courses for this period
   * - move is allowed */
  for(int i=0; i < tpCourseParts.Count; i++)
  {
    cp = (CoursePart)tpCourseParts[i];
    cp.Reason = CoursePartReason.TechnicalPackagePeriod;
    ScheduleChain(cp, period, true, true, false);
  }

  /* Schedule the previous courses that have not been completed
   * - move is allowed */
  for(int i=0; i < tpPreviousCourses.Count; i++)
```

```
{
  cp = (CoursePart)tpPreviousCourses[i];
  cp.Reason = CoursePartReason.TechnicalPackage;
  ScheduleChain(cp, period, true, true, false);
}

spp = (StudyPlanPeriod)schedule[period];
if (spp.RemainingWorkload[0] <= 0)
  return;

/* Schedule the fundamental courses
 * - move is not allowed
 * - recommended placement is checked */
for(int i=0; i < tpFundamentalCourses.Count; i++)
{
  CoursePart[] cpx = (CoursePart[])tpFundamentalCourses[i];
  foreach(CoursePart coursePart in cpx)
  {
    coursePart.Reason = CoursePartReason.Fundamental;
    if (ScheduleChain(coursePart, period, false, true, true))
      break;
  }
}

spp = (StudyPlanPeriod)schedule[period];
if (spp.RemainingWorkload[0] <= 0)
  return;

/* Schedule specifically selected courses
 * - move is not allowed
 * - language is checked
 * - lecturer is checked
 * - recommend placement is checked
 * - studytype is checked */
for(int i=0; i < selectedCourses.Count; i++)
{
  cp = (CoursePart)selectedCourses[i];
  cp.Reason = CoursePartReason.ManuallySelected;
  if (LanguageSelected(cp.CourseVersion_ID)
    && !LecturerDeSelected(cp.CourseVersion_ID)
    && StudyTypeSelected(cp.CourseVersion_ID))
    ScheduleChain(cp, period, false, true, true);
}

spp = (StudyPlanPeriod)schedule[period];
if (spp.RemainingWorkload[0] <= 0)
  return;

if (TechnicalPackagePassed(period))
{
  /* Schedule the prerequisite courses for the technical line
   * - move is not allowed
   * - recommended placement is checked */
  for(int i=0; i < tlCourseParts.Count; i++)
  {
    cp = (CoursePart)tlCourseParts[i];
    cp.Reason = CoursePartReason.TechnicalLinePrerequisite;
    ScheduleChain(cp, period, false, true, true);
  }

  spp = (StudyPlanPeriod)schedule[period];
  if (spp.RemainingWorkload[0] <= 0)
    return;

  /* Schedule the courses for the technical field
   * - move is not allowed
   * - recommended placement is checked
   * - deselected courses are checked
   * - languages are checked
   * - lecturer is checked
   * - studytype is checked */
  for(int i=0; i < tfCourseParts.Count; i++)
  {
    cp = (CoursePart)tfCourseParts[i];
    cp.Reason = CoursePartReason.TechnicalField;
    if (!deSelectedCourses.Contains(cp.Course_ID)
      && LanguageSelected(cp.CourseVersion_ID)
      && !LecturerDeSelected(cp.CourseVersion_ID)
      && StudyTypeSelected(cp.CourseVersion_ID))
      ScheduleChain(cp, period, false, true, true);
  }

  spp = (StudyPlanPeriod)schedule[period];
  if (spp.RemainingWorkload[0] <= 0)
    return;

  /* Schedule the pointgiving courses for the specialization
   * - move is not allowed
   * - recommended placement is checked
   * - deselected courses are checked
   * - languages are checked
   * - lecturer is checked
   * - studytype is checked */
  for(int i=0; i < spPointGiving.Count; i++)
  {
    cp = (CoursePart)spPointGiving[i];
    cp.Reason = CoursePartReason.SpecializationPointGiving;
    if (!deSelectedCourses.Contains(cp.Course_ID)
      && LanguageSelected(cp.CourseVersion_ID)
```

```
            && !LecturerDeSelected(cp.CourseVersion_ID)
            && StudyTypeSelected(cp.CourseVersion_ID))
            ScheduleChain(cp, period, false, true, true);
        }

        spp = (StudyPlanPeriod)schedule[period];
        if (spp.RemainingWorkload[0] <= 0)
          return;

        /* Schedule the supplementary courses for the specialization
         * - move is not allowed
         * - recommended placement is checked
         * - deselected courses are checked
         * - languages are checked
         * - lecturer is checked
         * - studytype is checked */
        for(int i=0; i < spSupplementary.Count; i++)
        {
          cp = (CoursePart)spSupplementary[i];
          cp.Reason = CoursePartReason.SpecializationSupplementary;
          if (!deSelectedCourses.Contains(cp.Course_ID)
            && LanguageSelected(cp.CourseVersion_ID)
            && !LecturerDeSelected(cp.CourseVersion_ID)
            && StudyTypeSelected(cp.CourseVersion_ID))
            ScheduleChain(cp, period, false, false, true);
        }

        spp = (StudyPlanPeriod)schedule[period];
        if (spp.RemainingWorkload[0] <= 0)
          return;
      }

      /* Schedule the optional courses for this period
         * - move is not allowed
         * - recommended placement is checked
         * - deselected courses are checked
         * - languages are checked
         * - lecturer is checked
         * - studytype is checked */
      for (int i=0; i < tpOptionalCourses.Count; i++)
      {
        cp = (CoursePart)tpOptionalCourses[i];
        cp.Reason = CoursePartReason.TechnicalPackageOptional;
        if (!deSelectedCourses.Contains(cp.Course_ID)
          && LanguageSelected(cp.CourseVersion_ID)
          && !LecturerDeSelected(cp.CourseVersion_ID)
          && StudyTypeSelected(cp.CourseVersion_ID))
          ScheduleChain(cp, period, false, false, true);
      }

      spp = (StudyPlanPeriod)schedule[period];
      if (spp.RemainingWorkload[0] <= 0)
        return;

      /* Schedule keyword matching courses
         * - move is not allowed
         * - recommended placement is checked
         * - deselected courses are checked
         * - languages are checked
         * - lecturer is checked
         * - studytype is checked */
      foreach(Guid[] cvs in keywordMatchingCourses)
      {
        foreach(Guid cv in cvs)
        {
          CoursePart[] courseParts = Course.GetCourseParts(new Guid[1] {cv}, 2);
          foreach(CoursePart coursePart in courseParts)
          {
            cp = coursePart;
            cp.Reason = CoursePartReason.KeywordMatching;
            if (!deSelectedCourses.Contains(cp.Course_ID)
              && LanguageSelected(cp.CourseVersion_ID)
              && !LecturerDeSelected(cp.CourseVersion_ID)
              && StudyTypeSelected(cp.CourseVersion_ID))
              ScheduleChain(cp, period, false, false, true);
          }
        }
      }
    }

    /// <summary>
    /// Schedules a course including mandatory and technical prerequisites.
    /// </summary>
    /// <param name="cp"></param>
    /// <param name="period"></param>
    /// <param name="allowMove"></param>
    /// <param name="schedulePrereq"></param>
    /// <param name="checkRecommendedPlacement"></param>
    /// <returns></returns>
    private bool ScheduleChain(CoursePart cp, Guid period, bool allowMove, bool schedulePrereq, bool
      checkRecommendedPlacement)
    {
      /*
      Guid[][] chains = null;

      //Get chains for mandatory prerequisites
      chains = PrerequisiteChains(cp.CourseVersion_ID, period, PrerequisiteCourseType.Mandatory);

      if (chains.Length > 0)
```

```
      {
        if (chains[0].Length > 0)
        {
          //If scheduling of (mandatory) prerequisites is not allowed, the course cannot be scheduled at all.
          if (!schedulePrereq)
            return false;

          //Otherwise start scheduling prerequisites
          CoursePart[] cps = Course.GetCourseParts(chains[0], 1);
          foreach(CoursePart cpx in cps)
          {
            studyPlan.Log.Add("CoursePart " + cp + " has mandatory prerequisite " + cpx);
            cpx.Reason = CoursePartReason.MandatoryPrerequisite;
            ScheduleCourse(cpx, period, allowMove, checkRecommendedPlacement);
          }
        }
      }

      //Get chains for technical prerequisites
      chains = PrerequisiteChains(cp.CourseVersion_ID, period, PrerequisiteCourseType.Technical);

      if (chains.Length > 0)
      {
        if (chains[0].Length > 0)
        {
          if (schedulePrereq)
          {
            CoursePart[] cps = Course.GetCourseParts(chains[0], 1);
            foreach(CoursePart cpx in cps)
            {
              studyPlan.Log.Add("CoursePart " + cp + " has technical prerequisite " + cpx);
              cpx.Reason = CoursePartReason.TechnicalPrerequisite;
              ScheduleCourse(cpx, period, allowMove, checkRecommendedPlacement);
            }
          }
        }
      }
*/
      //Schedule the course itself
      return ScheduleCourse(cp, period, allowMove, checkRecommendedPlacement);
    }

    /// <summary>
    /// Schedules all parts of a course in the specified period. If the course has multiple parts
    /// part 1 is placed in the specified period and the remaining parts are scheduled as soon as
    /// possible herafter. If a coursepart cannot be scheduled in the specified period, all parts of
    /// the course are unscheduled and rescheduled at a later time. However, if the period is restricted
    /// no rescheduling will be done, which means that if all parts of the course cannot be scheduled
    /// starting with part 1 in the specified period, the course will not be scheduled at all!
    /// </summary>
    /// <param name="cp">The course part to schedule</param>
    /// <param name="period">The period in which part 1 of the course should be scheduled</param>
    /// <param name="allowMove">If <strong>true</strong> the course will be moved to
    /// another period if scheduling in the specified period fails. If <strong>false</strong>
    /// the course will not be moved if scheduling in the specified period fails.</param>
    /// <param name="checkRecommendedPlacement">If <strong>true</strong> the recommended placement
    /// of the course will be checked and the course will only be scheduled at the the recommended time.
    /// If <strong>false</strong> the recommended placement of the course is not checked.</param>
    /// <returns><strong>True</strong> if the course has been scheduled, otherwise <strong>false</strong>.</returns>
    private bool ScheduleCourse(CoursePart cp, Guid period, bool allowMove, bool checkRecommendedPlacement)
    {
      Guid per = Guid.Empty;
      Guid partOnePeriod = Guid.Empty;
      Guid previousPeriod = period;
      CoursePart cpx = cp.GetPart(1);
      while(cpx != null)
      {
        if (!SignedUp(cpx))
        {
          //Find the first period where the part is taught.
          per = previousPeriod;
          while(!per.Equals(Guid.Empty))
          {
            if (TaughtInPeriod(cpx, per))
            {
              //Check recommended placement - allow exceeding of upper limit.
              if (!RecommendedPlacement(cpx.CourseVersion_ID, per, true) && checkRecommendedPlacement)
                return false;

              //Can the course part be scheduled in this period?
              if (ScheduleCoursePart(cpx, per))
              {
                if (cpx.Part.Equals(1))
                  partOnePeriod = per;
                previousPeriod = GetNextPeriod(per);
                break;
              }
              else
              {
                //Unschedule parts with lower priority (higher reason) in the period.
                bool removed = false;
                StudyPlanPeriod spp = (StudyPlanPeriod)schedule[per];
                for(int i=0; i < spp.NumberOfCourseParts; i++)
                {
                  CoursePart coursePart = spp.GetCoursePart(i);
                  if (coursePart.Reason > cpx.Reason)
                  {
                    UnScheduleCourse(coursePart.CourseVersion_ID);
                    removed = true;
```

```
            }
          }
          //If lower priority parts have been removed − retry.
          if (removed)
          {
            if (cpx.Part > 1)
              per = partOnePeriod;
            continue;
          }
          else if (allowMove)
          {
            //Moving is allowed − unschedule previous parts and restart from part 1 in a new period.
            if (cpx.Part > 1)
            {
              UnScheduleCourse(cp.CourseVersion_ID);
              cpx = cp.GetPart(1);

              while(SignedUp(cpx))
              {
                cpx = cpx.GetNextPart();
                if (cpx == null)
                  return false;
              }

              if (!partOnePeriod.Equals(Guid.Empty))
                per = GetNextPeriod(partOnePeriod);
              else
                per = GetNextPeriod(per);

              continue;
            }
              //Moving is allowed − try scheduling (part 1) in the next period
            else
            {
              per = GetNextPeriod(per);
              continue;
            }
          }
            //Moving is not allowed − remove any parts and give up
          else
          {
            if (cpx.Part > 1)
              UnScheduleCourse(cp.CourseVersion_ID);
            return false;
          }
        }
      }
      else if (allowMove)
        per = GetNextPeriod(per);
      else
        return false;
    }
    if (per.Equals(Guid.Empty))
      throw new BizException("CourseVersion " + cp.CourseVersion_ID +
        " (" + cpx.Part + ") cannot be scheduled");
  }
  cpx = cpx.GetNextPart();
 }
 return true;
}

/// <summary>
/// Checks whether a coursepart can be scheduled in a given period and if possible, the
/// coursepart is scheduled.
/// </summary>
/// <param name="cp">The coursepart which should be scheduled.</param>
/// <param name="period">The period in which to schedule the coursepart.</param>
/// <returns><strong>True</strong> if scheduling was successful, otherwise <strong>false</strong>.</returns>
private bool ScheduleCoursePart(CoursePart cp, Guid period)
{
 //Make sure the period has been initialized in the schedule.
 InitializeSchedule(period);

 //Has the course been deselected in this period?
 if (GetDeselectedCourses(period).Contains(cp.Course_ID))
 {
  studyPlan.Log.Add("Course " + cp.CourseVersion_ID + " has been deselected in period " + period);
  return false;
 }

 //Is the course pointblocking?
 if (PointBlocking(cp.CourseVersion_ID))
  return false;

 //Is the course taught in available module(s)?
 int[] selectedSchedule = new int[0];
 int periodtype = GetPeriodType(period);
 if (periodtype.Equals(1))
 {
  int[][] coursePartModules = Course.GetModules(cp, period);
  selectedSchedule = SelectSchedule(coursePartModules, period);
  if (selectedSchedule.GetLength(0).Equals(0))
  {
   studyPlan.Log.Add("No possible schedule for " + cp + " in period " + period);
   return false;
  }
 }

 //Does the remaining workload allow the addition of this course part?
```

```
    StudyPlanPeriod spp = (StudyPlanPeriod)schedule[period];
    if (spp.RemainingWorkload[1] < cp.Workload)
    {
      studyPlan.Log.Add("Remaining workload of " + spp.RemainingWorkload[1] + " does not allow the addition of course part " + cp + "
        with a workload of " + cp.Workload + " in period " + period);
      return false;
    }

    //Add the course part to the schedule
    cp.SelectedModules.Clear();
    cp.SelectedModules.AddRange(selectedSchedule);
    spp.AddCoursePart(cp);
    scheduledCourses.Add(cp.MD5, cp);
    studyPlan.Log.Add("Coursepart " + cp + " has been scheduled in period " + period);
    return true;
  }

  /// <summary>
  /// Unschedules a course, by removing all its parts from the schedule.
  /// </summary>
  /// <param name="courseVersion">The course to unschedule.</param>
  private void UnScheduleCourse(Guid courseVersion)
  {
    foreach(Guid period in schedule.Keys)
    {
      StudyPlanPeriod spp = (StudyPlanPeriod)schedule[period];
      for(int i=0; i < spp.NumberOfCourseParts; i++)
      {
        CoursePart cp = spp.GetCoursePart(i);
        if (cp.CourseVersion_ID.Equals(courseVersion))
        {
          studyPlan.Log.Add("Course part " + cp + " is unscheduled from period " + period);
          spp.RemoveCoursePart(cp);
          scheduledCourses.Remove(cp.MD5);
          if (!reschedulePeriods.Contains(period))
            reschedulePeriods.Add(period);
        }
      }
    }
  }

  /// <summary>
  /// Detmines whether rescheduling of a given period should be performed or not.
  /// The state of the period is saved each time it is rescheduled and if a state
  /// reccurs rescheduling is cancelled. Otherwise the elaborator might be trapped
  /// in an infinite loop of unscheduling/rescheduling.
  /// </summary>
  /// <param name="period">Identification of the period.</param>
  /// <returns><strong>True</strong> if the period should be rescheduled, otherwise <strong>false
  /// </strong>is returned.</returns>
  private bool PerformRescheduling(Guid period)
  {
    ArrayList outer = null;
    ArrayList inner = null;
    bool found = false;
    StudyPlanPeriod spp = (StudyPlanPeriod)schedule[period];

    //Has this period been rescheduled before?
    if (reSchedule.ContainsKey(period))
    {
      //Retrieve the previous states
      outer = (ArrayList)reSchedule[period];
      IEnumerator outerEnum = outer.GetEnumerator();

      //Iterate through the states
    next:
      while(outerEnum.MoveNext())
      {
        inner = (ArrayList)outerEnum.Current;
        if (spp.NumberOfCourseParts.Equals(inner.Count))
        {
          //Iterate through one state
          for(int i=0; i < inner.Count; i++)
          {
            CoursePart cp = spp.GetCoursePart(i);
            foreach(CoursePart cpi in inner)
            {
              if (cp.Equals(cpi))
              {
                found = true;
                break;
              }
              else
                found = false;
            }
            if (!found)
              goto next;
          }
        }
      }

      //The state has been previously saved, so no rescheduling again.
      if (found)
        return false;
      else
      {
        //A new state - save the new state and allow rescheduling.
        inner = new ArrayList();
        for(int i=0; i < spp.NumberOfCourseParts; i++)
```

```
          inner.Add(spp.GetCoursePart(i));
        outer.Add(inner);
        return true;
      }
    }
    else
    {
      //The period has not been rescheduled before – create key and save state.
      outer = new ArrayList();
      inner = new ArrayList();
      for(int i=0; i < spp.NumberOfCourseParts; i++)
        inner.Add(spp.GetCoursePart(i));
      outer.Add(inner);
      reSchedule.Add(period, outer);
      return true;
    }
  }

  /// <summary>
  /// Gets chains of prerequisite courses sorted by complexity
  /// </summary>
  /// <param name="courseVersion"></param>
  /// <param name="period"></param>
  /// <param name="prerequisiteCourseType"></param>
  /// <returns></returns>
  private Guid[][] PrerequisiteChains(Guid courseVersion, Guid period, PrerequisiteCourseType prerequisiteCourseType)
  {
    Guid[][] prereq = Course.GetPrerequisiteCourseCombinations(courseVersion, prerequisiteCourseType, true);
    Guid[][] chains = new Guid[0][];

    if (prereq.Length > 0)
    {
      Hashtable courses = GetSignedUpAndScheduledCourses(period);
      chains = Course.SortCombinationsByComplexity(courses, prereq);
    }
    return chains;
  }

  /// <summary>
  /// Retrieves a list of courseparts which have been signed up for outside the schedule
  /// </summary>
  private void GetSignedUpCourses()
  {
    signedUpTotal = 0;

    CoursePart[] courseParts = Student.GetCourseParts(this.StudentVersion_ID);
    foreach(CoursePart cp in courseParts)
    {
      signedUpTotal += cp.Workload;
      signedUpCourses.Add(cp.MD5, cp);
    }
  }

  /// <summary>
  /// Retrieves a list of the courses which have been signed up for either outside
  /// the schedule or scheduled up to (and including) the specified period.
  /// </summary>
  /// <param name="period">Identification of the period.</param>
  /// <returns>A hashtable of <see cref="StudyPlanning.DAL.Courses.Course"/> identifications.</returns>
  private Hashtable GetSignedUpAndScheduledCourses(Guid period)
  {
    Hashtable courseParts = new Hashtable();

    //Iterate through all the periods up to the specified period
    Guid per = GetNextPeriod();
    while(!per.Equals(Guid.Empty))
    {
      StudyPlanPeriod spp = (StudyPlanPeriod)schedule[per];
      if (spp == null)
      {
        per = GetNextPeriod(per);
        continue;
      }

      for(int i=0; i < spp.NumberOfCourseParts; i++)
      {
        CoursePart cp = spp.GetCoursePart(i);
        if (cp.Part.Equals(cp.TotalParts))
          courseParts.Add(cp.Course_ID, null);
      }

      if (per.Equals(period))
        break;
      else
        per = GetNextPeriod(per);
    }
    return courseParts;
  }

  /// <summary>
  /// Determines whether a coursepart has already been signed up for – either on the Student_Course
  /// table or any other period in the schedule.
  /// </summary>
  /// <param name="coursePart"></param>
  /// <returns><strong>True</strong> if the coursepart has already been signed up for,
  /// otherwise <strong>false</strong>.</returns>
  private bool SignedUp(CoursePart coursePart)
  {
    if (signedUpCourses.Contains(coursePart.MD5))
```

```
      return true;

    if (scheduledCourses.Contains(coursePart.MD5))
      return true;

    return false;
  }

  /// <summary>
  /// Determines whether a coursepart has already been signed up for − either on the Student_Course
  /// table or in this period or previous periods.
  /// </summary>
  /// <param name="coursePart"></param>
  /// <param name="period"></param>
  /// <returns><strong>True</strong> if the coursepart has already been signed up for,
  /// otherwise <strong>false</strong>.</returns>
  private bool SignedUp(CoursePart coursePart, Guid period)
  {
    if (signedUpCourses.Contains(coursePart.MD5))
      return true;

    Guid per = GetNextPeriod();
    while(!per.Equals(Guid.Empty))
    {
      StudyPlanPeriod spp = (StudyPlanPeriod)schedule[per];
      if (spp == null)
        return false;

      for(int i=0; i < spp.NumberOfCourseParts; i++)
      {
        CoursePart cp = spp.GetCoursePart(i);
        if (cp.Equals(coursePart))
          return true;
      }

      if (per.Equals(period))
        return false;
      else
        per = GetNextPeriod(per);
    }
    return false;
  }

  /// <summary>
  /// Determines whether a course has already been signed up for − either on the Student_Course
  /// table or any other period in the schedule.
  /// </summary>
  /// <param name="course"></param>
  /// <returns><strong>True</strong> if the course has already been signed up for,
  /// otherwise <strong>false</strong>.</returns>
  private bool SignedUp(Guid course)
  {
    foreach(CoursePart cp in signedUpCourses.Values)
    {
      if (cp.Course_ID.Equals(course))
        return true;
    }

    foreach(CoursePart cp in scheduledCourses.Values)
    {
      if (cp.Course_ID.Equals(course))
        return true;
    }
    return false;
  }

  /// <summary>
  /// Determines whether a coursepart is taught in a given period.
  /// </summary>
  /// <param name="cp"></param>
  /// <param name="period"></param>
  /// <returns><strong>True</strong> if the coursepart is taught in the specified period,
  /// otherwise false.</returns>
  private bool TaughtInPeriod(CoursePart cp, Guid period)
  {
    bool taught = Course.TaughtInPeriod(cp, period);
    if (!taught)
      studyPlan.Log.Add("Coursepart " + cp + " is not taught in period " + period);
    return taught;
  }

  /// <summary>
  /// Determines whether a technical package has been passed.
  /// </summary>
  /// <returns><strong>True</strong> if the technical package has been passed, otherwise
  /// <strong>false</strong> is returned.</returns>
  private bool TechnicalPackagePassed(Guid period)
  {
    for(int i=0; i < tpAllParts.Count; i++)
    {
      CoursePart[] cpx = (CoursePart[])tpAllParts[i];
      if (cpx != null)
        foreach(CoursePart cp in cpx)
        {
          if (SignedUp(cp, period))
            break;
          else
          {
            studyPlan.Log.Add("Coursepart " + cp + " has not been passed in the technical package periods before or in " + period);
```

```
            return false;
          }
        }
      else
        tpAllParts.Remove(cpx);
    }
    //Clear list of courses − because technical package has now been passed.
    tpAllParts.Clear();
    return true;
  }

  /// <summary>
  /// Selects a schedule from the one or more possible combinations of modules.
  /// </summary>
  /// <param name="coursePartModules">A two−dimensional array containing modules. There is
  /// implicit logical AND between the inner arrays and implicit logical OR between the
  /// elements in the inner arrays.</param>
  /// <param name="period">The period for which to select a schedule.</param>
  /// <returns>An array of modules. If the coursepart cannot be scheduled, an empty array
  /// is returned.</returns>
  private int[] SelectSchedule(int[][] coursePartModules, Guid period)
  {
    bool available = true;
    ArrayList availableModules = null;

    if (schedule.ContainsKey(period))
      availableModules = ((StudyPlanPeriod)schedule[period]).AvailableModules;
    else
      return new int[0];

    if (coursePartModules.GetLength(0) == 0)
      return new int[0];
    else if (coursePartModules.GetLength(0) == 1)
    {
      for(int i=0; i < coursePartModules[0].GetLength(0); i++)
      {
        if (availableModules.Contains(coursePartModules[0][i]))
          return new int[1] {coursePartModules[0][i]};
      }
      return new int[0];
    }
    else
    {
      ArrayList res = new ArrayList();
      for(int i=0; i < coursePartModules.GetLength(0); i++)
      {
        int[][] resx = new int[1][];
        resx[0] = coursePartModules[i];
        res = Combinations(resx, res);
      }
      IEnumerator resEnum = res.GetEnumerator();
      while(resEnum.MoveNext())
      {
        int[][] current = (int[][])resEnum.Current;
        available = true;
        for(int i=0; i < current.GetLength(0); i++)
          available = available && availableModules.Contains(current[i][0]);

        if (available)
        {
          int[] result = new int[current.GetLength(0)];
          for(int i=0; i < current.GetLength(0); i++)
            result[i] = current[i][0];
          return result;
        }
      }
    }
    return new int[0];
  }

  /// <summary>
  /// Combines the elements of 2 two−dimensional arrays. The input arrays must have the following
  /// forms:
  /// <ul>
  /// <li>Logical AND − [[A], [B], [C]] means A and B and C</li>
  /// <li>Logical OR − [[A, B, C]] means A or B or C</li>
  /// </ul>
  /// </summary>
  /// <param name="one">Array number one.</param>
  /// <param name="two">Array number two.</param>
  /// <returns>An arraylist of two−dimensional arrays containing the combinations of the input arrays.</returns>
  private ArrayList Combine(int[][] one, int[][] two)
  {
    ArrayList res = new ArrayList();
    int[][] result = null;

    int col = 0;

    if ((one.GetLength(0) > 1) && (two.GetLength(0) > 1))
    {
      result = new int[one.GetLength(0) + two.GetLength(0)][];
      for(int j=0; j < result.GetLength(0); j++)
        result[j] = new int[1];

      for(int i=0; i < one.GetLength(0); i++)
      {
        result[col++][0] = one[i][0];
      }
      for(int i=0; i < two.GetLength(0); i++)
```

```
      {
        result[col++][0] = two[i][0];
      }
      res.Add(result);
    }
    else if ((one.GetLength(0) == 1) && (two.GetLength(0) > 1))
    {
      for(int i=0; i < one[0].GetLength(0); i++)
      {
        result = new int[one.GetLength(0) + two.GetLength(0)][];
        result[col] = new int[1];
        result[col++][0] = one[0][i];
        for(int j=0; j < two.GetLength(0); j++)
        {
          result[col] = new int[1];
          result[col++][0] = two[j][0];
        }
        res.Add(result);
        col = 0;
      }
    }
    else if ((one.GetLength(0) > 1) && (two.GetLength(0) == 1))
    {
      for(int i=0; i < two[0].GetLength(0); i++)
      {
        result = new int[one.GetLength(0) + two.GetLength(0)][];
        result[col] = new int[1];
        result[col++][0] = two[0][i];
        for(int j=0; j < one.GetLength(0); j++)
        {
          result[col] = new int[1];
          result[col++][0] = one[j][0];
        }
        res.Add(result);
        col = 0;
      }
    }
    else if ((one.GetLength(0) == 1) && (two.GetLength(0) == 1))
    {
      for(int i=0; i < one[0].GetLength(0); i++)
      {
        for(int j=0; j < two[0].GetLength(0); j++)
        {
          result = new int[2][];
          result[0] = new int[1];
          result[0][0] = one[0][i];
          result[1] = new int[1];
          result[1][0] = two[0][j];
          res.Add(result);
        }
      }
    }
    return res;
  }

  /// <summary>
  /// Combines the elements of a two-dimensional with the elements of an arraylist containing
  /// two-dimensional arrays.
  /// </summary>
  /// <param name="elem">A two-dimensional array - see form under the method Combine.</param>
  /// <param name="list">An arraylist containing zero, one or more two-dimensional arrays
  /// of the form described under the method Combine.</param>
  /// <returns>An arraylist of two-dimensional arrays containing the combinations.</returns>
  private ArrayList Combinations(int[][] elem, ArrayList list)
  {
    //If the list is empty the single element is returned.
    if (list.Count == 0)
    {
      ArrayList resx = new ArrayList(1);
      resx.Add(elem);
      return resx;
    }
    //If the list contains one element the element is extracted and combined with the other element.
    else if (list.Count == 1)
    {
      return Combine(elem, (int[][])list[0]);
    }
    //If the list contains multiple elements, the first element is extracted and combined with the
    //other element and then a recursive call to Combinations is performed with the remaining part
    //of the list.
    else
    {
      ArrayList resx = Combine(elem, (int[][])list[0]);
      resx.AddRange(Combinations(elem, list.GetRange(1, list.Count - 1)));
      return resx;
    }
  }

  /// <summary>
  /// Determines whether a given courseversion is taught in one of the preferred languages specified
  /// in the studyplan criterion.
  /// </summary>
  /// <param name="courseVersion">The unique identification of the courseversion.</param>
  /// <returns><strong>True</strong> if the courseversion is taught in one of the preferrede
  /// languages, otherwise <strong>false</strong>.</returns>
  private bool LanguageSelected(Guid courseVersion)
  {
    CourseVersion cv = new CourseVersion();
    cv.CourseVersion_ID.Value = courseVersion;
```

```
    try
    {
      cv.Retrieve();
    }
    catch (DalException e)
    {
      throw new BizException(e, "CourseVersion_ID", courseVersion.ToString());
    }

    if (languages.Contains(cv.Language_ID.Value))
    {
      return true;
    }
    else
    {
      studyPlan.Log.Add("Courseversion " + courseVersion + "has language " + cv.Language_ID.Value + " which has not been selected");
      return false;
    }
  }

  /// <summary>
  /// Determines whether at least one of the lecturers, who teaches a course
  /// has been deselected.
  /// </summary>
  /// <param name="courseVersion">The identification of the courseversion.</param>
  /// <returns><strong>True</strong> if at least one lecturer has been deselected, otherwise
  /// <strong>false</strong> is returned.</returns>
  private bool LecturerDeSelected(Guid courseVersion)
  {
    Guid[] lect = null;

    try
    {
      lect = Course.GetLecturers(courseVersion);
    }
    catch (BizException)
    {
      return false;
    }

    foreach(Guid lecturer in lect)
      if (lecturers.Contains(lecturer))
        return true;

    return false;
  }

  /// <summary>
  /// Determines whether a course is pointblocking in relation to other courses which have been
  /// signed up for already.
  /// </summary>
  /// <param name="courseVersion">Identification of the courseversion.</param>
  /// <returns><strong>True</strong> if the course is pointblocking, otherwise <strong>false</strong>.</returns>
  private bool PointBlocking(Guid courseVersion)
  {
    bool found = false;

    Guid[][] pointBlocking = Course.GetPointBlockingCourses(courseVersion);
    foreach(Guid[] pblist in pointBlocking)
    {
      foreach(Guid course in pblist)
      {
        if(SignedUp(course))
        {
          found = true;
          break;
        }
        else
          found = false;
      }
      if (!found)
        return false;
    }
    return found;
  }

  /// <summary>
  /// Determines whether the recommended placement for the course has been satisfied
  /// in for the specified period.
  /// </summary>
  /// <param name="courseVersion">Identification of the courseversion.</param>
  /// <param name="period">Identification of the period.</param>
  /// <param name="allowExceed">If <strong>true</strong> it allowed to exceed the upper limit
  /// of the recommended placement, if <strong>false</strong> the course must be placed within
  /// the lower and upper bounds of the recommended placement.</param>
  /// <returns><strong>True</strong> if the recommended placement is satisfied, otherwise
  /// <strong>false</strong> is returned.</returns>
  private bool RecommendedPlacement(Guid courseVersion, Guid period, bool allowExceed)
  {
    Guid per = GetNextPeriod();
    float total = signedUpTotal;

    float[] rp = Course.GetRecommendedPlacement(courseVersion, studyType);

    if (rp == null)
      return true;

    while(!per.Equals(Guid.Empty))
```

```
      {
        if (per.Equals(period))
          break;

        Object obj = schedule[per];
        if (obj != null)
        {
          StudyPlanPeriod spp = (StudyPlanPeriod)obj;
          total += spp.TotalWorkload;
        }
        per = GetNextPeriod(per);
      }

      if (allowExceed)
        return (total >= rp[0]);
      else
        return ((total >= rp[0]) && (total <= rp[1]));
    }

    /// <summary>
    /// Determines whether the study type of a course is among the selected
    /// study types.
    /// </summary>
    /// <param name="courseVersion">Identification of the course version.</param>
    /// <returns><strong>True</strong> if the study type of the course has been
    /// selected, otherwise <strong>false</strong>.</returns>
    private bool StudyTypeSelected(Guid courseVersion)
    {
      int[] studytypes = Course.GetStudyTypes(courseVersion);
      if (studytypes.Length.Equals(0))
        return true;
      else
      {
        foreach(int st in studytypes)
        {
          if (studyTypes.Contains(st))
            return true;
        }
        return false;
      }
    }
    #endregion //Private Methods
  }
}
```

3.11 TechnicalField

```
using System;
using System.Collections;
using StudyPlanning.DAL;
using StudyPlanning.DAL.Courses;
using StudyPlanning.DAL.TechnicalFields;
using StudyPlanning.Biz;

namespace StudyPlanning.Biz
{
  /// <summary>
  /// Summary description for TechnicalField.
  /// </summary>
  public class TechnicalField
  {
    /// <summary>
    /// Retrieves a list of course parts associated with the technical field.
    /// </summary>
    /// <param name="technicalFieldVersion_ID">Identification of the
    /// <see cref="StudyPlanning.DAL.TechnicalFields.TechnicalFieldVersion"/>.</param>
    /// <returns>An array of <see cref="StudyPlanning.Biz.CoursePart"/> objects.</returns>
    public static CoursePart[] GetCourses(int technicalFieldVersion_ID)
    {
      int[] ids = null;
      try
      {
        ids = StudyPlanning.DAL.TechnicalFields.Course.GetIDFromVersion(technicalFieldVersion_ID);
      }
      catch (DalException e)
      {
        throw new BizException(e, "TechnicalFieldVersion_ID", technicalFieldVersion_ID.ToString());
      }

      Guid[] courses = new Guid[ids.Length];

      for(int i=0; i<ids.Length; i++)
      {
        StudyPlanning.DAL.TechnicalFields.Course tfc = new StudyPlanning.DAL.TechnicalFields.Course();
        tfc.TechnicalField_Course_ID.Value = ids[i];
        try
        {
          tfc.Retrieve();
        }
        catch (DalException e)
        {
          throw new BizException(e, "TechnicalField_Course_ID", ids[i].ToString());
        }
```

```
        courses[i] = tfc.Course_ID.Value;
      }

      CoursePart[] courseParts;

      try
      {
        courseParts = StudyPlanning.Biz.Course.GetCourseParts(courses, 1);
      }
      catch (BizException bizEx)
      {
        throw bizEx;
      }

      return courseParts;
    }
  }
}
```

3.12 TechnicalLine

```
using System;
using System.Collections;
using StudyPlanning.DAL;
using StudyPlanning.DAL.Courses;
using StudyPlanning.DAL.TechnicalLines;
using StudyPlanning.Biz;

namespace StudyPlanning.Biz
{
  /// <summary>
  /// Represents a technical line in the business services tier.
  /// </summary>
  public class TechnicalLine
  {

    /// <summary>
    /// Gets a list of courses which are prerequisite for the specified technical line version.
    /// </summary>
    /// <param name="technicalLineVersion_ID">The ID of the technical line version.</param>
    /// <returns>An array of <see cref="StudyPlanning.Biz.CoursePart"/> objects.</returns>
    public static CoursePart[] GetPrerequisiteCourses(int technicalLineVersion_ID)
    {
      Guid[] courses;
      try
      {
        courses = StudyPlanning.DAL.TechnicalLines.PrerequisiteCourse.GetCourses(technicalLineVersion_ID);
      }
      catch (DalException e)
      {
        if (e.Number == 8)
          return new CoursePart[0];
        else
          throw new BizException(e, "TechnicalLineVersion_ID", technicalLineVersion_ID.ToString());
      }

      CoursePart[] courseParts;

      try
      {
        courseParts = StudyPlanning.Biz.Course.GetCourseParts(courses, 1);
      }
      catch (BizException bizEx)
      {
        throw bizEx;
      }

      return courseParts;
    }

    /// <summary>
    /// Gets a list of courses which - in conjunction with the specified technical package -
    /// are prerequisite for a technical line (the one which the specified technical
    /// package is associated with).
    /// </summary>
    /// <param name="technicalPackage_ID">The numeric identifier of the technical line version.</param>
    /// <param name="technicalLineVersion_ID">The numeric identifier of the technical package.</param>
    /// <returns>An array of <see cref="StudyPlanning.Biz.CoursePart"/> objects.</returns>
    public static CoursePart[] GetPrerequisiteCourses(int technicalLineVersion_ID, int technicalPackage_ID)
    {
      int PrerequisiteTechnicalPackage_ID;

      try
      {
        PrerequisiteTechnicalPackage_ID = StudyPlanning.DAL.TechnicalLines.PrerequisiteTechnicalPackage.GetID(
          technicalLineVersion_ID,
          technicalPackage_ID);
      }
      catch (DalException e)
      {
        if (e.Number == 8)
          return new CoursePart[0];
        else
          throw new BizException(e, "TechnicalLineVersion_ID / TechnicalPackage_ID", technicalLineVersion_ID.ToString() + " / " +
            technicalPackage_ID.ToString());
```

```
      }

      Guid[] courses;
      try
      {
        courses = StudyPlanning.DAL.TechnicalLines.PrerequisiteTechnicalPackageCourse.GetCourses(
          PrerequisiteTechnicalPackage_ID);
      }
      catch (DalException e)
      {
        if (e.Number.Equals(8))
          return new CoursePart[0];
        else
          throw new BizException(e, "PrerequisiteTechnicalPackage_ID", PrerequisiteTechnicalPackage_ID.ToString());
      }

      CoursePart[] courseParts;

      try
      {
        courseParts = StudyPlanning.Biz.Course.GetCourseParts(courses, 1);
      }
      catch (BizException bizEx)
      {
        throw bizEx;
      }

      return courseParts;
    } //GetTpPrerequisiteCourses

    /// <summary>
    /// Gets a list of technical packages which are prerequisite for the specified
    /// technical line version.
    /// </summary>
    /// <param name="technicalLineVersion_ID">The ID of the technical line version.</param>
    /// <returns>An array containing numeric identifiers of the technical packages.</returns>
    public static int[] GetPrerequisiteTechnicalPackages(int technicalLineVersion_ID)
    {
      int[] technicalPackages;

      try
      {
        technicalPackages = StudyPlanning.DAL.TechnicalLines.PrerequisiteTechnicalPackage.GetTechnicalPackages(
          technicalLineVersion_ID);
      }
      catch (DalException dalEx)
      {
        throw new BizException(dalEx, "TechnicalLineVersion_ID", technicalLineVersion_ID.ToString());
      }

      return technicalPackages;
    }

  }
}
```

3.13 TechnicalPackage

```
using System;
using System.Collections;
using StudyPlanning.DAL;
using StudyPlanning.DAL.Courses;
using StudyPlanning.DAL.TechnicalPackages;
using StudyPlanning.Biz;

namespace StudyPlanning.Biz
{
  /// <summary>
  /// Summary description for TechnicalPackage.
  /// </summary>
  public class TechnicalPackage
  {

    /// <summary>
    /// Retrieves a list of periods represented by Period_IDs in a technical package.
    /// </summary>
    /// <param name="technicalPackageVersion_ID"> The identification of the
    /// <see cref="StudyPlanning.DAL.TechnicalPackages.TechnicalPackageVersion"/>.</param>
    public static Guid[] GetPeriods(int technicalPackageVersion_ID)
    {
      Guid[] periods = null;
      DateTime[] startdates = null;

      try
      {
        periods = StudyPlanning.DAL.TechnicalPackages.Period.GetPeriods(technicalPackageVersion_ID);
      }
      catch (DalException e)
      {
        throw new BizException(e);
      }

      startdates = new DateTime[periods.GetLength(0)];
```

```
    for (int i=0; i < periods.GetLength(0); i++)
    {
      StudyPlanning.DAL.Period objPeriod = new StudyPlanning.DAL.Period();
      objPeriod.Period_ID.Value = periods[i];

      try
      {
        objPeriod.Retrieve();
      }
      catch (DalException e)
      {
        throw new BizException(e, "Period_ID", periods[i].ToString());
      }

      startdates[i] = objPeriod.StartDate.Value;
    }
    Array.Sort(startdates, periods);
    return periods;
  }

  /// <summary>
  /// Retrieves a list of fundamental courses represented by Course_IDs in a technical package.
  /// </summary>
  /// <param name="technicalPackageVersion_ID"> The identification of the
  /// <see cref="StudyPlanning.DAL.TechnicalPackages.TechnicalPackageVersion"/>.</param>
  public static CoursePart[][] GetFundamentalCourses(int technicalPackageVersion_ID)
  {
    int[] ids = null;
    Guid[][] courses = null;

    try
    {
      ids = FundamentalCourse.GetIDFromVersion(technicalPackageVersion_ID);
    }
    catch (DalException e)
    {
      if (e.Number.Equals(8)) //No fundamental courses found (no error)
        ids = new int[0];
      else
        throw new BizException(e, "TechnicalPackageVersion_ID", technicalPackageVersion_ID.ToString());
    }

    courses = new Guid[ids.GetLength(0)][];

    for (int i=0; i < ids.GetLength(0); i++)
    {
      try
      {
        courses[i] = FundamentalCourseItem.GetCourses(ids[i]);
      }
      catch (DalException e)
      {
        if (e.Number == 8) //No fundamental courses found (no error)
          courses = new Guid[0][];
        else
          throw new BizException(e, "TechnicalPackage_FundamentalCourseItem_ID", ids[i].ToString());
      }
    }

    //Split up in courseparts
    CoursePart[][] courseParts = new CoursePart[courses.GetLength(0)][];
    for(int i=0; i < courses.GetLength(0); i++)
      courseParts[i] = Course.GetCourseParts(courses[i], 1);
    return courseParts;
  }

  /// <summary>
  /// Retrieves a list of courses represented by <see cref="StudyPlanning.Biz.CoursePart"/> objects
  /// for a given period in a technical package.
  /// </summary>
  /// <param name="technicalPackageVersion_ID"> Identification of the
  /// <see cref="StudyPlanning.DAL.TechnicalPackages.TechnicalPackageVersion"/>.</param>
  /// <param name="period_ID"> Identification of the
  /// <see cref="StudyPlanning.DAL.Period"/>.</param>
  public static CoursePart[][] GetCoursesInAPeriod(int technicalPackageVersion_ID, Guid period_ID)
  {
    int id = 0;
    int[] ids = null;
    StudyPlanning.Biz.CoursePart[][] coursesBiz = null;
    StudyPlanning.DAL.Courses.CoursePart[][] coursesDal = null;

    try
    {
      id = StudyPlanning.DAL.TechnicalPackages.Period.GetIDFromVersion(technicalPackageVersion_ID, period_ID);
    }
    catch (DalException e)
    {
      throw new BizException(e, "TechnicalPackageVersion_ID", technicalPackageVersion_ID.ToString());
    }

    try
    {
      ids = PeriodCourse.GetIDFromPeriod(id);
    }
    catch (DalException e)
    {
      throw new BizException(e, "TechnicalPackage_Period_ID", id.ToString());
    }
```

```
    coursesDal = new StudyPlanning.DAL.Courses.CoursePart[ids.GetLength(0)][];

    for (int i=0; i < ids.GetLength(0); i++)
    {
      try
      {
        coursesDal[i] = PeriodCourseItem.GetCourses(ids[i]);
      }
      catch (DalException e)
      {
        if (e.Number != 8)
          throw new BizException(e, "TechnicalPackage_PeriodCourse_ID", ids[i].ToString());
      }
    }

    //Convert to Biz
    coursesBiz = new CoursePart[coursesDal.GetLength(0)][];

    for (int i=0; i < coursesBiz.GetLength(0); i++)
    {
      if (coursesDal[i] == null)
      {
        coursesBiz[i] = null;
      }
      else
      {
        coursesBiz[i] = new CoursePart[coursesDal[i].GetLength(0)];

        for (int j=0; j < coursesBiz[i].GetLength(0); j++)
        {
          coursesBiz[i][j] = new CoursePart();
          coursesBiz[i][j].Part = coursesDal[i][j].Part;
          coursesBiz[i][j].Course_ID = coursesDal[i][j].Course_ID;
          coursesBiz[i][j].CourseVersion_ID = Course.GetNewestVersion(coursesDal[i][j].Course_ID);
          coursesBiz[i][j].Workload = Course.GetWorkload(
            coursesBiz[i][j].CourseVersion_ID,
            coursesBiz[i][j].Part);

          CourseVersion cv = new CourseVersion();
          cv.CourseVersion_ID.Value = coursesBiz[i][j].CourseVersion_ID;
          try
          {
            cv.Retrieve();
          }
          catch (DalException e)
          {
            throw new BizException(e, "CourseVersion_ID", cv.CourseVersion_ID.Value.ToString());
          }
          coursesBiz[i][j].TotalParts = cv.Parts.Value;
        }
      }
    }
    return coursesBiz;
  }

  /// <summary>
  /// Retrieves a list of optional courses represented by <see cref="StudyPlanning.Biz.CoursePart"/> objects
  /// for a given period in a technical package.
  /// </summary>
  /// <param name="technicalPackageVersion_ID"> Identification of the
  /// <see cref="StudyPlanning.DAL.TechnicalPackages.TechnicalPackageVersion"/>.</param>
  /// <param name="period_ID"> Identification of the
  /// <see cref="StudyPlanning.DAL.Period"/>.</param>
  public static CoursePart[] GetOptCoursesInAPeriod(int technicalPackageVersion_ID, Guid period_ID)
  {
    int id = 0;
    StudyPlanning.Biz.CoursePart[] coursesBiz = null;
    StudyPlanning.DAL.Courses.CoursePart[] coursesDal = null;

    try
    {
      id = StudyPlanning.DAL.TechnicalPackages.Period.GetIDFromVersion(technicalPackageVersion_ID, period_ID);
    }
    catch (DalException e)
    {
      throw new BizException(e, "TechnicalPackageVersion_ID", technicalPackageVersion_ID.ToString());
    }

    try
    {
      coursesDal = PeriodOptionalCourse.GetCourses(id);
    }
    catch (DalException e)
    {
      if (e.Number == 8) //No optional courses - no error!
        return new CoursePart[0];
      else
        throw new BizException(e, "TechnicalPackage_Period_ID", id.ToString());
    }

    coursesBiz = new CoursePart[coursesDal.GetLength(0)];

    for (int i=0; i < coursesBiz.GetLength(0); i++)
    {
      coursesBiz[i] = new CoursePart();
      coursesBiz[i].Part = coursesDal[i].Part;
      coursesBiz[i].Course_ID = coursesDal[i].Course_ID;
      coursesBiz[i].CourseVersion_ID = Course.GetNewestVersion(coursesDal[i].Course_ID);
      coursesBiz[i].Workload = Course.GetWorkload(
```

```
            coursesBiz[i].CourseVersion_ID,
            coursesBiz[i].Part);

          CourseVersion cv = new CourseVersion();
          cv.CourseVersion_ID.Value = coursesBiz[i].CourseVersion_ID;
          try
          {
            cv.Retrieve();
          }
          catch (DalException e)
          {
            throw new BizException(e, "CourseVersion_ID", cv.CourseVersion_ID.Value.ToString());
          }
          coursesBiz[i].TotalParts = cv.Parts.Value;
        }
        return coursesBiz;
      }

      /// <summary>
      /// Retrieves a list of courses represented by <see cref="StudyPlanning.Biz.CoursePart"/>
      /// objects grouped by periods in a technical package.
      /// </summary>
      /// <param name="technicalPackageVersion_ID"> Identification of the
      /// <see cref="StudyPlanning.DAL.TechnicalPackages.TechnicalPackageVersion"/>.</param>
      public static CoursePart[][][] GetCourses(int technicalPackageVersion_ID)
      {
        Guid[] periods      = null;
        CoursePart[][][] courses = null;

        try
        {
          periods = GetPeriods(technicalPackageVersion_ID);
        }
        catch (BizException e)
        {
          throw e;
        }

        courses = new CoursePart[periods.GetLength(0)][][];

        try
        {
          for (int i=0; i < periods.GetLength(0); i++)
          {
            courses[i] = GetCoursesInAPeriod(
              technicalPackageVersion_ID,
              periods[i]);
          }
        }
        catch (BizException e)
        {
          throw e;
        }
        return courses;
      }

    }
}
```

3.14 Text

```
using System;
using System.Data;
using System.Data.SqlClient;
using StudyPlanning.DAL;
using System.Collections;

namespace StudyPlanning.Biz
{
  /// <summary>
  /// Represents a text string in the business services tier.
  /// </summary>
  public class TextItem : StudyPlanning.Biz.BizObject
  {
    #region Private Properties

    private DalInt _Text_ID      = new DalInt(false);
    private DalInt _TextGroup_ID   = new DalInt(false);
    private DalString _NumberInGroup = new DalString(false);
    private DalString _Culture_ID   = new DalString(false);
    private DalString _Text      = new DalString(true);
    private DalString _Description  = new DalString(true);

    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Initializes a new instance of the
    /// <see cref="StudyPlanning.Biz.TextItem" />
    /// class.
    /// </summary>
    public TextItem()
    {
```

```
}

/// <summary>
/// Initializes a new instance of the <see cref="StudyPlanning.DAL.TextItem" />
/// class with the a text group ID, the number of the text in the group, and
/// a culture ID.
/// </summary>
/// <param name="textGroup_ID">The ID of the text group.</param>
/// <param name="numberInGroup">The number of the text in the group.</param>
/// <param name="culture_ID">The ID of the culture.</param>
public TextItem(
  int textGroup_ID,
  string numberInGroup,
  string culture_ID)
{
  this._TextGroup_ID.Value = textGroup_ID;
  this._NumberInGroup.Value = numberInGroup;
  this._Culture_ID.Value = culture_ID;
}

#endregion //Constructors

#region Public Properties

/// <summary>
/// Gets the ID of the text item.
/// </summary>
public int Text_ID
{
  get { return _Text_ID.Value; }
  set { _Text_ID.Value = value;}
}

/// <summary>
/// Gets the text group ID of the text item.
/// </summary>
public int TextGroup_ID
{
  get { return _TextGroup_ID.Value; }
  set { _TextGroup_ID.Value = value;}
}

/// <summary>
/// Gets or sets the <see cref="StudyPlanning.DAL.DalString">DalString</see>
/// object for the text number in the group.
/// </summary>
public string NumberInGroup
{
  get { return _NumberInGroup.Value; }
  set { _NumberInGroup.Value = value;}
}

/// <summary>
/// Gets the culture ID of the text item.
/// </summary>
public string Culture_ID
{
  get { return _Culture_ID.Value; }
  set { _Culture_ID.Value = value;}
}

/// <summary>
/// Gets the text of the text item.
/// </summary>
public string Text
{
  get { return _Text.Value; }
  set { _Text.Value = value;}
}

/// <summary>
/// Gets the description of the text item.
/// </summary>
public string Description
{
  get { return _Description.Value; }
  set { _Description.Value = value;}
}

#endregion //Public Properties

#region Retrieve Methods

/// <summary>
/// Retrieves the text from the database using the value of the
/// TextGroup_ID, NumberInGroup and Culture_ID property of the
/// current instance.
/// </summary>
public void Retrieve()
{
  try
  {
    _TextGroup_ID.Validate();
    _NumberInGroup.Validate();
    _Culture_ID.Validate();

    RetrieveExecute(
      _TextGroup_ID.Value,
      _NumberInGroup.Value,
```

```
            _Culture_ID.Value,
            this);
        }
        catch (BizException objEx)
        {
          throw objEx;
        }
      }

      /// <summary>
      /// Retrieves the database the text with the specified text group ID, number in
      /// the text group and culture ID.
      /// </summary>
      /// <param name="textGroup_ID">The ID of the text group.</param>
      /// <param name="textNumberInGroup">The number of the text in the group.</param>
      /// <param name="culture_ID">The ID of the culture.</param>
      public static TextItem Retrieve(int textGroup_ID, string textNumberInGroup, string culture_ID)
      {
        StudyPlanning.Biz.TextItem objText = new StudyPlanning.Biz.TextItem();

        try
        {
          RetrieveExecute(
            textGroup_ID,
            textNumberInGroup,
            culture_ID,
            objText
          );

          return objText;
        }
        catch (BizException objExc)
        {
          throw objExc;
        }
      }

      private static void RetrieveExecute(
        int textGroup_ID,
        string textNumberInGroup,
        string culture_ID,
        StudyPlanning.Biz.TextItem textItem)
      {
        StudyPlanning.DAL.TextItem dalTextItem = new StudyPlanning.DAL.TextItem();
        dalTextItem.TextGroup_ID.Value = textGroup_ID;
        dalTextItem.NumberInGroup.Value = textNumberInGroup;
        dalTextItem.Culture_ID.Value = culture_ID;

        try
        {
          dalTextItem.Retrieve();
          textItem.Text_ID = dalTextItem.Text_ID.Value;
          textItem.TextGroup_ID = dalTextItem.TextGroup_ID.Value;
          textItem.NumberInGroup = dalTextItem.NumberInGroup.Value;
          textItem.Culture_ID = dalTextItem.Culture_ID.Value;

          if (!dalTextItem.Text.IsNull)
            textItem.Text = dalTextItem.Text.Value;
        }
        catch (DalException excp)
        {
          throw new BizException(excp);
        }
      }

      public static Hashtable GetTextGroup(int textGroupID, string cultureID)
      {
        Hashtable texts;

        try
        {
          texts = StudyPlanning.DAL.TextItem.GetTextsInGroup(textGroupID, cultureID);
        }
        catch (DalException dalEx)
        {
          throw new BizException(dalEx);
        }

        return texts;
      }

      #endregion //Retrive Methods
    }
}
```

3.15 User

3.15.1 AuthenticationResult

```
using System;
using StudyPlanning.DAL;
using StudyPlanning.DAL.Users;
```

```
using System.Collections;

namespace StudyPlanning.Biz
{
  /// <summary>
  /// Represents the result of a user authentication.
  /// </summary>
  /// <remarks>
  /// The following security cases leads to rejection of authentication:
  /// 2, 3, 10, 11, 15, 17, 19, 20, 26, 27, 30, 31, 34.<br/>
  /// <br/>
  /// The following security cases leads to approval of authentication:
  /// 13, 16, 33
  /// </remarks>
  public class AuthenticationResult
  {
    /// <summary>
    /// Initializes a new instance of the <see cref="StudyPlanning.Biz.AuthenticationResult"/>
    /// class.
    /// </summary>
    public AuthenticationResult() {}

    /// <summary>
    /// Gets the authentication case number.
    /// </summary>
    public int AuthenticationCase;

    /// <summary>
    /// Gets an indication of whether the user should be approved or rejected authentication.
    /// </summary>
    public bool Authenticate = false;

    /// <summary>
    /// Gets execution details about the authentication process.
    /// </summary>
    public ArrayList Trace = new ArrayList(20);

    /// <summary>
    /// Approves authentication of the user.
    /// </summary>
    public void ApproveAuthentication()
    {
      this.Authenticate = true;
    }

    /// <summary>
    /// Approves authentication of the user using the specified security
    /// case.
    /// </summary>
    /// <param name="authenticationCase">The security case which approves the
    /// authentication.</param>
    public void ApproveAuthentication(int authenticationCase)
    {
      this.Authenticate = true;
      this.AuthenticationCase = authenticationCase;
    }

    /// <summary>
    /// Rejects authentication of the user.
    /// </summary>
    public void RejectAuthentication()
    {
      this.Authenticate = false;
    }

    /// <summary>
    /// Rejects authentication of the user using the specified security
    /// case.
    /// </summary>
    /// <param name="authenticationCase">The security case which rejects the
    /// authentication.</param>
    public void RejectAuthentication(int authenticationCase)
    {
      this.Authenticate = false;
      this.AuthenticationCase = authenticationCase;
    }
  }

}
```

3.15.2 User

```
using System;
using StudyPlanning.DAL;
using StudyPlanning.DAL.Users;
using System.Collections;

namespace StudyPlanning.Biz
{
  /// <summary>
  /// Represents a user in the business services tier.
  /// </summary>
  public class User : StudyPlanning.Biz.BizObject
  {
    private StudyPlanning.DAL.Users.User user =
      new StudyPlanning.DAL.Users.User();
```

```
private StudyPlanning.DAL.Users.User origUser;

#region Public Properties
/// <summary>
/// Gets or sets the ID of the user.
/// </summary>
public Guid User_ID
{
  get { return user.User_ID.Value; }
  set { user.User_ID.Value = value; }
}

/// <summary>
/// Gets or sets the username of the user.
/// </summary>
public string Username
{
  get { return user.Username.Value; }
  set { user.Username.Value = value; }
}

/// <summary>
/// Gets or sets the password of the user.
/// </summary>
public string Password
{
  get { return user.Password.Value; }
  set { user.Password.Value = value; }
}

/// <summary>
/// Gets or sets an indication of whether the user should change his
/// password at next logon.
/// </summary>
public bool ChangePasswordAtNextLogon
{
  get { return user.ChangePasswordAtNextLogon.Value; }
  set { user.ChangePasswordAtNextLogon.Value = value; }
}

/// <summary>
/// Gets or sets an indication of whether the user is allowed to change his password.
/// </summary>
public bool UserCannotChangePassword
{
  get { return user.UserCannotChangePassword.Value; }
  set { user.UserCannotChangePassword.Value = value; }
}

/// <summary>
/// Gets or sets an indication of whether the user's password does ever expire.
/// </summary>
public bool PasswordNeverExpires
{
  get { return user.PasswordNeverExpires.Value; }
  set { user.PasswordNeverExpires.Value = value; }
}

/// <summary>
/// Gets or sets the date and time of the user's last change of password.
/// </summary>
public DateTime PasswordLastChanged
{
  get { return user.PasswordLastChanged.Value; }
  set { user.PasswordLastChanged.Value = value; }
}

/// <summary>
/// Gets or sets an indication of whether the user (account) has been locked.
/// </summary>
public bool Locked
{
  get { return user.Locked.Value; }
  set { user.Locked.Value = value; }
}

/// <summary>
/// Gets or sets the date and time at which the user (account) was locked.
/// </summary>
public DateTime LockedAt
{
  get { return user.LockedAt.Value; }
  set { user.LockedAt.Value = value; }
}

/// <summary>
/// Gets or sets an indication of whether the user (account) is disabled.
/// </summary>
public bool Disabled
{
  get { return user.Disabled.Value; }
  set { user.Disabled.Value = value; }
}
#endregion //Public Properties

#region Constructors

/// <summary>
```

```
/// Initializes a new instance of the <see cref="StudyPlanning.Biz.User"/> class.
/// </summary>
public User() {}

/// <summary>
/// Initializes a new instance of the <see cref="StudyPlanning.Biz.User"/> class using
/// the specified user ID.
/// </summary>
/// <param name="userID">The ID of the user.</param>
public User(Guid userID)
{
  this.User_ID = userID;
}

#endregion //Constructors

/// <summary>
/// Retrieves the user from the data base using the value of the
/// <see cref="StudyPlanning.Biz.User.User_ID"/> property of the current
/// instance.
/// </summary>
public void Retrieve()
{
  try
  {
    user.Retrieve();
  }
  catch (DalException dalEx)
  {
    throw new BizException(dalEx);
  }

  this.User_ID = user.User_ID.Value;
  this.Username = user.Username.Value;
  this.Password = user.Password.Value;
  this.ChangePasswordAtNextLogon = user.ChangePasswordAtNextLogon.Value;
  this.UserCannotChangePassword = user.UserCannotChangePassword.Value;
  this.PasswordNeverExpires = user.PasswordNeverExpires.Value;
  this.PasswordLastChanged = user.PasswordLastChanged.Value;
  this.Locked = user.Locked.Value;

  if (!user.LockedAt.IsNull)
    this.LockedAt = user.LockedAt.Value;

  this.Disabled = user.Disabled.Value;

  origUser = user.Clone();
}

/// <summary>
/// Updates the user in the data base using the values of the properties
/// of the current instance.
/// </summary>
public void Update()
{
  if (user.Locked.Value && user.LockedAt.IsNull)
  {
    user.LockedAt.Value = System.DateTime.Now;
  }
  else if (!user.Locked.Value)
  {
    user.LockedAt.IsNull = true;
  }

  try
  {
    user.Update(origUser);
  }
  catch (DalException dalEx)
  {
    throw new BizException(dalEx);
  }

  /*
  origUser.User_ID.Value = originalUser.User_ID;
  origUser.Username.Value = originalUser.Username;
  origUser.Password.Value = originalUser.Password;
  origUser.ChangePasswordAtNextLogon.Value = originalUser.ChangePasswordAtNextLogon;
  origUser.UserCannotChangePassword.Value = originalUser.UserCannotChangePassword;
  origUser.PasswordNeverExpires.Value = originalUser.PasswordNeverExpires;
  origUser.PasswordLastChanged.Value = originalUser.PasswordLastChanged;
  origUser.Locked.Value = originalUser.Locked;
  origUser.LockedAt.Value = originalUser.LockedAt;
  */
}

/// <summary>
/// Get the ID of the user having the specified username.
/// </summary>
/// <param name="username">The username for which to get a user ID.</param>
/// <returns>The ID of the user having the specified username.</returns>
public static Guid GetID(string username)
{
  Guid userID;

  try
  {
    userID = StudyPlanning.DAL.Users.User.GetIDFromUsername(username);
  }
```

```
  catch (DalException dalEx)
  {
    throw new BizException(dalEx);
  }

  return userID;
}

#region Misc Security Methods
/// <summary>
/// Retrieves the roles which the specified user has been assigned.
/// </summary>
/// <param name="user_ID">The ID of the user for which to get a list of roles.</param>
/// <returns>An array of integers representing the roles which are assigned to
/// the specified user.</returns>
public static int[] GetRoles(Guid user_ID)
{
  int[] roles;

  try
  {
    roles = StudyPlanning.DAL.Users.SecurityRole.GetRolesFromUserID(user_ID);
  }
  catch (DalException dalEx)
  {
    throw new BizException(dalEx);
  }

  return roles;
}

/// <summary>
/// Checks whether the specified user has the specified permission.
/// </summary>
/// <param name="user_ID">The ID of the user.</param>
/// <param name="permission_ID">The ID of the permission.</param>
/// <returns><strong>true</strong>, if the specified user has the specified permission; <strong>false</strong>, otherwise.</returns>
public static bool HasPermission(Guid user_ID, int permission_ID)
{
  bool result = false;
  int[] roles;

  try
  {
    roles = StudyPlanning.Biz.User.GetRoles(user_ID);
  }
  catch (BizException bizEx)
  {
    throw bizEx;
  }

  bool temp_result;

  foreach(int role_ID in roles)
  {
    temp_result = false;

    try
    {
      temp_result = StudyPlanning.Biz.Security.Role.HasPermission(role_ID, permission_ID);
    }
    catch (BizException bizEx)
    {
      throw bizEx;
    }

    if (temp_result)
      result = true;
  }

  return result;
}

/// <summary>
/// Checks whether the specified user has the specified role.
/// </summary>
/// <param name="user_ID">The ID of the user.</param>
/// <param name="role_ID">The ID of the role.</param>
/// <returns><strong>true</strong>, if the specified user has the specified role; <strong>false</strong>, otherwise.</returns>
public static bool HasRole(Guid user_ID, int role_ID)
{
  bool result = false;
  int[] roles;

  try
  {
    roles = StudyPlanning.Biz.User.GetRoles(user_ID);
  }
  catch (BizException bizEx)
  {
    throw bizEx;
  }

  foreach(int securityRole_ID in roles)
  {
    if (securityRole_ID == role_ID)
      result = true;
  }
```

```
    return result;
  }

  #endregion

  #region Authentication Methods

  /// <summary>
  /// Authenticates the user having the specified username using the specified
  /// password.
  /// </summary>
  /// <param name="username">The username of the user to authenticate.</param>
  /// <param name="password">The password specified by the user.</param>
  /// <returns>An <see cref="StudyPlanning.Biz.AuthenticationResult"/> object
  /// holding the result of the authentication.</returns>
  public static AuthenticationResult Authenticate(string username, string password)
  {
    UserAuthenticator objAuthenticator = new UserAuthenticator(username, password);
    AuthenticationResult result = new AuthenticationResult();

    try
    {
      result = objAuthenticator.Authenticate();
    }
    catch (BizException bizEx)
    {
      throw bizEx;
    }

    return result;
  }

  /// <summary>
  /// Authenticates the user having the specified username using the specified
  /// password.
  /// </summary>
  /// <param name="username">The username of the user to authenticate.</param>
  /// <param name="password">The password specified by the user.</param>
  /// <param name="newPassword">The new password specified by the user.</param>
  /// <param name="newPasswordRepeat">The new password repeated by the user.</param>
  /// <returns>An <see cref="StudyPlanning.Biz.AuthenticationResult"/> object
  /// holding the result of the authentication.</returns>
  public static AuthenticationResult Authenticate(string username, string password, string newPassword, string
    newPasswordRepeat)
  {
    UserAuthenticator objAuthenticator = new UserAuthenticator(username, password, newPassword, newPasswordRepeat);
    AuthenticationResult result = new AuthenticationResult();

    try
    {
      result = objAuthenticator.Authenticate();
    }
    catch (BizException bizEx)
    {
      throw bizEx;
    }

    return result;
  }

  #endregion //Authentication Methods

  /// <summary>
  /// Checks whether the specified password differs from the specified
  /// number of previous password of the specified user.
  /// </summary>
  /// <param name="userID">The ID if the user for which to make the check.</param>
  /// <param name="password">The password </param>
  /// <param name="numberOfPreviousPasswords">The number of previous passwords which to make the check for. The check will be made for the "
    newest" previous passwords.</param>
  /// <returns><strong>true</strong>, if the password does differ from the user's previous passwords; <strong>false</strong>, otherwise.</returns
    >
  public static bool DoesPasswordDi↑erFromPreviousPasswords(Guid userID, string password, int numberOfPreviousPasswords)
  {
    bool result = true;
    Guid[] passwords;

    try
    {
      passwords = StudyPlanning.DAL.Users.PasswordHistory.GetPreviousPasswords(userID);
    }
    catch (DalException dalEx)
    {
      throw new BizException(dalEx);
    }

    for(int i=0;i<numberOfPreviousPasswords;i++)
    {
      StudyPlanning.DAL.Users.PasswordHistory objPwHist =
        new StudyPlanning.DAL.Users.PasswordHistory();

      objPwHist.User_PasswordHistory_ID.Value = passwords[i];

      try
      {
        objPwHist.Retrieve();
      }
      catch (DalException dalEx)
      {
```

```
          throw new BizException(dalEx);
        }

        if (password.Equals(objPwHist.Password.Value.ToString()))
          result = false;
      }

      return result;
    }
  }

  /// <summary>
  /// Specifies the different types of authentication.
  /// </summary>
  ///
  public enum AuthenticationType
  {
    /// <summary>
    /// Indicates that the type of authentication is standard i.e. some user
    /// specifies his username and password and requests to be logged into the
    /// system (authenticated).
    /// </summary>
    Standard=0,

    /// <summary>
    /// Indicates that the type of authentication is extended i.e. the user requests to
    /// change his password as part of the authentication.
    /// </summary>
    ChangePassword=1};
}
```

3.15.3 UserAuthenticator

```
using System;
using StudyPlanning.DAL;
using System.Collections;

namespace StudyPlanning.Biz
{
  /// <summary>
  /// Supports authentication of a user.
  /// </summary>
  public class UserAuthenticator : StudyPlanning.Biz.BizObject
  {
    #region Private Properties
    private string _Username;
    private string _Password;
    private string _NewPassword;
    private string _NewPasswordRepeat;
    private AuthenticationType _Type;
    private Guid _UserID;
    private DateTime _CurrentDateTime; // The current date and time
    private StudyPlanning.Biz.User _User = new StudyPlanning.Biz.User();
    #endregion //Private Properties

    #region Constructors

    /// <summary>
    /// Initializes a new instance of the <see cref="StudyPlanning.Biz.UserAuthenticator"/>
    /// class.
    /// </summary>
    public UserAuthenticator()
    {
      _CurrentDateTime = System.DateTime.Now;
    }

    /// <summary>
    /// Initializes a new instance of the <see cref="StudyPlanning.Biz.UserAuthenticator"/>
    /// class with the specified username and password.
    /// </summary>
    /// <remarks>Type of authentication is set to
    /// <see cref="StudyPlanning.Biz.AuthenticationType.Standard"/> as default.
    /// </remarks>
    /// <param name="username">The username specified by the user.</param>
    /// <param name="password">The password specified by the user.</param>
    public UserAuthenticator(string username, string password)
    {
      _CurrentDateTime = System.DateTime.Now;
      Username = username;
      Password = password;
      Type = AuthenticationType.Standard;
    }

    /// <summary>
    /// Initializes a new instance of the <see cref="StudyPlanning.Biz.UserAuthenticator"/>
    /// class with the specified username, password, new password and repetition of the new password.
    /// </summary>
    /// <remarks>Type of authentication is set to
    /// <see cref="StudyPlanning.Biz.AuthenticationType.ChangePassword"/> as default.
    /// </remarks>
    /// <param name="username">The username specified by the user.</param>
    /// <param name="password">The password specified by the user.</param>
    /// <param name="newPassword">The new password specified by the user.</param>
    /// <param name="newPasswordRepeat">The repetition of the new password specified by the user.</param>
    public UserAuthenticator(string username, string password, string newPassword, string newPasswordRepeat)
    {
```

```
    _CurrentDateTime = System.DateTime.Now;
    Username = username;
    Password = password;
    NewPassword = newPassword;
    NewPasswordRepeat = newPasswordRepeat;
    Type = AuthenticationType.ChangePassword;
  }

  #endregion //Constructors

  #region Public Properties

  /// <summary>
  /// Gets or sets the username specified by the user.
  /// </summary>
  public string Username
  {
    get { return _Username; }
    set { _Username = value; }
  }

  /// <summary>
  /// Gets or sets the password specified by the user.
  /// </summary>
  public string Password
  {
    get { return _Password; }
    set { _Password = value; }
  }

  /// <summary>
  /// Gets a hashed version (using the MD5 algorithm) of the password specified by the user.
  /// </summary>
  public string PasswordHashed
  {
    get { return System.Web.Security.FormsAuthentication.HashPasswordForStoringInCon↓gFile(this.Password, "md5"); }
  }

  /// <summary>
  /// Gets or sets the new password specified by the user.
  /// </summary>
  public string NewPassword
  {
    get { return _NewPassword; }
    set { _NewPassword = value; }
  }

  /// <summary>
  /// Gets or sets the new password repeated by the user.
  /// </summary>
  public string NewPasswordRepeat
  {
    get { return _NewPasswordRepeat; }
    set { _NewPasswordRepeat = value; }
  }

  /// <summary>
  /// Gets a hashed version (using the MD5 algorithm) of the new password specified by the user.
  /// </summary>
  public string NewPasswordHashed
  {
    get { return System.Web.Security.FormsAuthentication.HashPasswordForStoringInCon↓gFile(this.NewPassword, "md5"); }
  }

  /// <summary>
  /// Gets or sets the type of authentication.
  /// </summary>
  public AuthenticationType Type
  {
    get { return _Type; }
    set { _Type = value; }
  }
  #endregion //Public Properties

  /// <summary>
  /// Authenticates the user of the current instance.
  /// </summary>
  /// <returns>
  /// An <see cref="StudyPlanning.Biz.AuthenticationResult"/> object holding
  /// the result of the authentication.
  /// </returns>
  public AuthenticationResult Authenticate()
  {
    AuthenticationResult result = new StudyPlanning.Biz.AuthenticationResult();

    bool userAccountExists = true;

    try
    {
      _UserID = StudyPlanning.Biz.User.GetID(_Username);
    }
    catch (BizException bizEx)
    {
      if (bizEx.Dal != null)
      {
        //CASE 2
        if (bizEx.Dal.Number == 8)
        {
          //The specified user account does NOT exist
```

```
        userAccountExists = false;
        result.Trace.Add("Case 2. The specified user account does NOT exist.");

        //Reject authentication
        result.RejectAuthentication(2);
      }
      else
        throw bizEx;
    }
    else
      throw bizEx;
  }

  if (userAccountExists) //CASE 1
  {
    // The specified user account exists, now proceed.
    result.Trace.Add("Case 1. The specified user account exists.");

    _User.User_ID = _UserID;

    try
    {
      _User.Retrieve();
    }
    catch (DalException dalEx)
    {
      throw new BizException(dalEx);
    }

    if (_User.Disabled) //CASE 3
    {
      // The user account is disabled
      result.Trace.Add("Case 3. The user account is DISABLED.");

      // Reject authentication
      result.RejectAuthentication(3);
    }
    else //CASE 4
    {
      // The user account is enabled
      result.Trace.Add("Case 4. The user account is ENABLED.");
      // Now, check if the specified password is correct

      result.Trace.Add("Case 4. Password in DB (hash value): " + _User.Password.ToString());
      result.Trace.Add("Case 4. Specified password (hash value): " + this.PasswordHashed);

      if (_User.Password.Equals(this.PasswordHashed)) //CASE 5
      {
        // The specified password is correct
        result.Trace.Add("Case 5. The specified password is correct.");

        // Now, check if the user account is locked

        if (_User.Locked) //CASE 7
        {
          //The user account is locked
          result.Trace.Add("Case 7. The user account is locked.");
          //Now, check if it is more than 20 minutes ago since the account was locked

          //The difference (time interval) between the current date/time and the locked date/time
          TimeSpan dtmDi↑ = this._CurrentDateTime.Subtract(_User.LockedAt);

          if (dtmDi↑.TotalMinutes > 20) //CASE 9
          {
            //It is MORE than 20 minutes ago since the user account was locked
            result.Trace.Add("Case 9. It is MORE than 20 minutes ago since the user account was locked.");

            //Now, unlock acccount and proceed
            _User.Locked = false;

            try
            {
              _User.Update();
            }
            catch (DalException dalEx)
            {
              throw new BizException(dalEx);
            }

            // User account was succesfully unlocked
            result.Trace.Add("Case 9. User account was succesfully unlocked.");

            // Now, proceed performing other security checks
            CheckSystemState(result);
          }
          else //CASE 10
          {
            //It is LESS than 20 minutes ago since the user account was locked
            result.Trace.Add("Case 10. It is LESS than 20 minutes ago that the user account was locked.");

            //Reject authentication
            result.RejectAuthentication(10);
          }
        }
        else //CASE 8
        {
          // The user account is NOT locked

          // Now, proceed performing other security checks
```

```
        CheckSystemState(result);
      }
    }
    else //CASE 6
    {
      // The specified password is INCORRECT
      result.Trace.Add("Case 6. The specified password is INCORRECT.");

      // Now, check if the user account is locked

      if (_User.Locked) //CASE 17
      {
        //The user account is locked
        result.Trace.Add("Case 17. The user account is locked.");

        //Reject authentication
        result.RejectAuthentication(17);
      }
      else //CASE 18
      {
        // The user account is NOT locked
        result.Trace.Add("Case 18. The user account is NOT locked.");

        // Now, perform database operations and reject authentication.

        // We need to log that an authentication has been rejected for the current user account.
        // We do so by adding an entry to DB table tb_User_Login (LoginType=0).
        // Why do all this work? To prevent bruteforce attacks.

        StudyPlanning.DAL.Users.Login userLogin =
          new StudyPlanning.DAL.Users.Login();

        userLogin.User_Login_ID.Value = Guid.NewGuid();
        userLogin.User_ID.Value = _User.User_ID;
        userLogin.User_LoginType_ID.Value = 1;

        userLogin.Log.Created.Value = this._CurrentDateTime;
        userLogin.Log.CreatedBy.Value = _User.User_ID;
        userLogin.Log.Updated.Value = this._CurrentDateTime;
        userLogin.Log.UpdatedBy.Value = _User.User_ID;

        try
        {
          userLogin.Create();
        }
        catch (DalException dalEx)
        {
          throw new BizException(dalEx);
        }

        // Now, we have to check if the user account should be locked.
        // When should a user account be locked? If we can detect that authentication has been rejected three times in
        // a row within the last five minutes, the user account has to be locked (pure system policy).

        int numberOfRejectedLogins =
          StudyPlanning.DAL.Users.Login.GetNumberOfLogins(
          _User.User_ID, 1, this._CurrentDateTime, 5);

        // Remark, if less than 3 rejected logins are returned, we do not
        // take any action.
        // It should not be possible to get more than 3 rejected logins
        // according to the way the process of authentication has been designed
        // however, just be sure we check for 3 or more rejected logins.

        if (numberOfRejectedLogins >= 3) //CASE 19
        {
          //The user account should be locked (dealt with in the lines to come).
          result.Trace.Add("User account should be locked. Case 19.");

          //Now, lock the acccount and proceed
          _User.Locked = true;

          try
          {
            _User.Update();
          }
          catch (DalException dalEx)
          {
            throw new BizException(dalEx);
          }

          // User account was succesfully locked
          result.Trace.Add("Case 19. User account was succesfully locked.");

          //Reject authentication
          result.RejectAuthentication(19);
        }
        else //CASE 20
        {
          //Do not take any further measures (danish: forholdsregler).
          result.Trace.Add("Case 20. No further measures are taken.");

          //Reject authentication
          result.RejectAuthentication(20);
        }
      }
    }
  }
```

```
      result.Trace.Add("Authentication approved. Security case " + result.AuthenticationCase + ".");

      StudyPlanning.DAL.Users.Login login =
        new StudyPlanning.DAL.Users.Login();

      login.User_Login_ID.Value = Guid.NewGuid();
      login.User_ID.Value = this._UserID;
      login.User_LoginType_ID.Value = 2; //Approval
      login.Log.Created.Value = this._CurrentDateTime;
      login.Log.CreatedBy.Value = this._UserID;
      login.Log.Updated.Value = this._CurrentDateTime;
      login.Log.UpdatedBy.Value = this._UserID;

      try
      {
        login.Create();
      }
      catch (DalException dalEx)
      {
        throw new BizException(dalEx);
      }
    } // user account exists
    return result;
  }

  /// <summary>
  /// Auxiliary method to the
  /// <see cref="StudyPlanning.Biz.UserAuthenticator.Authenticate()"/> method.
  /// Checks what state the system is in. Subsequently, other security checks
  /// are performed. This method is invoked by security cases 8 and 9.
  /// </summary>
  private void CheckSystemState(AuthenticationResult result)
  {
    if (Type.Equals(AuthenticationType.Standard)) //CASE 21
    {
      // System is in state A
      result.Trace.Add("Case 21. System is in state A.");
      //Now, check if user should be forced to renew password

      if (_User.ChangePasswordAtNextLogon) //CASE 11
      {
        //User should be forced to renew password
        result.Trace.Add("Case 11. User should be forced to renew password (ChangePasswordAtNextLogon is true).");

        //Reject authentication
        result.RejectAuthentication(11);
        //Response.Redirect("login.aspx?State=B&Msg=Du skal forny dit kodeord for at kunne logge ind.");
      }
      else //CASE 12
      {
        //User should NOT be forced to renew password
        result.Trace.Add("Case 12. User should NOT be forced to renew password (ChangePasswordAtNextLogon is false).");

        //Now, check if user's password does ever expire

        if (_User.PasswordNeverExpires) //CASE 13
        {
          // The user's password never expires
          result.Trace.Add("Case 13. The user's password never expires (PasswordNeverExpires is true).");

          // Approve authentication
          result.ApproveAuthentication(13);
        }
        else //CASE 14
        {
          //The user's password may have expired
          result.Trace.Add("Case 14. The user's password may have expired. Check if password has expired...");

          //Now, check if password has expired

          //TimeSpan dtmDiff = dtmLastChangeOfPassword.Subtract(dtmNow);
          TimeSpan dtmDi↑ = this._CurrentDateTime.Subtract(_User.PasswordLastChanged);

          if (dtmDi↑.TotalDays > 30) //CASE 15
          {
            //The user's password has expired (it is more than 30 days ago since the password was changed)
            result.Trace.Add("Case 15. The user's password has expired.");

            //Now, user should renew his or hers password

            //Reject authentication
            result.RejectAuthentication(15);
            //Response.Redirect("login.aspx?State=B&Msg=Dit kodeord er udløbet, og du skal forny dette for at kunne logge ind.");
          }
          else //CASE 16
          {
            //The user's password has NOT expired
            result.Trace.Add("Case 16. The user's password has NOT expired.");

            //Approve authentication
            result.ApproveAuthentication(16);
          }
        }
      }
    }
    else if (this.Type.Equals(AuthenticationType.ChangePassword)) //CASE 22
    {
      // System is in state B
      result.Trace.Add("Case 22. System is in state B.");
```

```
      //Now, check if the user may change his or hers password

      if (_User.UserCannotChangePassword) //CASE 24
      {
        // The user may NOT change the password him- or herself
        result.Trace.Add("Case 24. The user may not change the password him- or herself.");

        // Now, check if the user has been forced to renew password

        if (_User.ChangePasswordAtNextLogon) //CASE 25
        {
          // The user has been forced to change password
          result.Trace.Add("Case 25. The user has been forced to change password");

          // Now, proceed performing other security checks
          //CheckPasswordRepetition();
        }
        else //CASE 26
        {
          // The user has NOT been forced to change password

          //Reject authentication
          result.RejectAuthentication(26);
          //RejectAuthentication(26, "Du har ikke lov til selv at ændre kodeordet for denne brugerkonto. Foretag <a href=\"login.aspx\">almindeligt
            login</a> uden at ændre kodeord.");
        }
      }
      else //CASE 23
      {
        // The user may change password him- or herself
        result.Trace.Add("Case 23. The user may change password him- or herself.");

        // Now, proceed performing other security checks
        //CheckPasswordRepetition();
      }
    }
  } //method CheckSystemState

  /// <summary>
  /// Auxiliary method to the
  /// <see cref="StudyPlanning.Biz.UserAuthenticator.CheckSystemState"/> method.
  /// Checks if the repetition of the new password is equal to the specified
  /// new password. Subsequently, other security checks are performed. This
  /// method is invoked by security cases 23 and 25.
  /// </summary>
  private void CheckPasswordRepetition(AuthenticationResult result)
  {
    if (this.NewPassword.Equals(this.NewPasswordRepeat)) //CASE 29
    {
      // Password has been repeated correctly
      result.Trace.Add("Case 29. Password has been repeated correctly.");

      // Now, check if the specified new password is equal to the user's current password

      if (this.NewPassword.Equals(this.Password)) //CASE 27
      {
        // The specified new password is equal to the user's current password
        result.Trace.Add("Case 27. The specified new password is equal to the user's current password.");

        //Reject authentication
        result.RejectAuthentication(27);
        //RejectAuthentication(27, "Det anførte nye kodeord er det samme som dit nuværende kodeord. Dette er ikke tilladt. Forsøg venligst igen.");
      }
      else //CASE 28
      {
        // The specified new password is NOT equal to the user's current password
        result.Trace.Add("Case 28. The specified new password is NOT equal to the user's current password");

        // Now, check if the specified new password is equal to one of the user's previous 5 passwords

        if (!StudyPlanning.Biz.User.DoesPasswordDi↑erFromPreviousPasswords(_User.User_ID, this.NewPasswordHashed, 5)) //
          CASE 31
        {
          // The specified new password is NOT valid (i.e. it is equal to one of the user's previous two passwords)
          result.Trace.Add("Case 31. The specified new password is NOT valid (i.e. it is equal to one of the user's previous two passwords).");

          // Reject authentication
          result.RejectAuthentication(31);
          //RejectAuthentication(31, "Det anførte nye kodeord er det samme som et af dine to tidligere kodeord. Dette er ikke tilladt af
            sikkerhedsårsager. Forsøg venligst igen.");
        }
        else //CASE 32
        {
          // The specified new password is valid (i.e. not equal to any of the user's previous two passwords)
          result.Trace.Add("Case 32. The specified new password is valid (i.e. not equal to any of the user's previous two passwords).");

          // Now, check if the specified new password meets complexity requirements

          if (StudyPlanning.Biz.Security.Password.MeetsComplexityRequirements(this.Password)) //CASE 33
          {
            // New password meets complexity requirements
            result.Trace.Add("Case 33. New password meets complexity requirements");

            // Now, add the user's current password to the user's password history

            StudyPlanning.DAL.Users.PasswordHistory pwHist =
              new StudyPlanning.DAL.Users.PasswordHistory();
```

```
            pwHist.User_PasswordHistory_ID.Value = Guid.NewGuid();
            pwHist.User_ID.Value = this._UserID;
            pwHist.Password.Value = this._User.Password;
            pwHist.Log.Created.Value = this._CurrentDateTime;
            pwHist.Log.CreatedBy.Value = this._UserID;
            pwHist.Log.Updated.Value = this._CurrentDateTime;
            pwHist.Log.UpdatedBy.Value = this._UserID;

            try
            {
              pwHist.Create();
            }
            catch (DalException dalEx)
            {
              throw new BizException(dalEx);
            }

            this._User.Password = this.NewPasswordHashed;
            this._User.PasswordLastChanged = this._CurrentDateTime;

            try
            {
              this._User.Update();
            }
            catch (DalException dalEx)
            {
              throw new BizException(dalEx);
            }

            // Approve authentication
            result.ApproveAuthentication(33);
          }
          else //CASE 34
          {
            // New password DOES NOT meet complexity requirements
            result.Trace.Add("Case 34. New password DOES NOT meet complexity requirements");

            //Reject authentication
            result.RejectAuthentication(34);
            //RejectAuthentication(34, "Det anførte nye kodeord opfylder ikke de krav, der er til et kodeord. Forsøg venligst igen. ");
          }
        }
      }
    }
    else //CASE 30
    {
      // Password has NOT been repeated correctly
      result.Trace.Add("Case 30. Password has NOT been repeated correctly.");

      //Reject authentication
      result.RejectAuthentication(30);
      //RejectAuthentication(30, "Din gentagelse af det nye kodeord stemte ikke overens med det valgte nye kodeord. Forsøg venligst igen.");
    }
  }
 }
}
```

Chapter 4

Business Tier Test

4.1 StudyPlanElaborator

4.1.1 StudyPlanElaborator

studyplanelaborator.aspx

```
<%@ Page language="c#" src="studyplanelaborator.aspx.cs" CodeBehind="StudyPlanElaborator.aspx.cs" AutoEventWireup="false" Inherits
  ="StudyPlanning.Biz.StudyPlanElaboratorTest" trace="false" warningLevel="4" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" >
<HTML>
  <HEAD>
    <title>StudyPlanElaborator</title>
    <meta name="vs_showGrid" content="True">
    <meta name="GENERATOR" Content="Microsoft Visual Studio 7.0">
    <meta name="CODE_LANGUAGE" Content="C#">
    <meta name="vs_defaultClientScript" content="JavaScript">
    <meta name="vs_targetSchema" content="http://schemas.microsoft.com/intellisense/ie5">
    <link rel="stylesheet" href="../../../misc/style.css">
    <style>
    .textbox { WIDTH: 400px }
    .button { WIDTH: 120px }
    </style>
  </HEAD>
  <body>
    <P align="center"><STRONG><FONT size="4">StudyPlanElaborator</FONT></STRONG></P>
    <form id="form" method="post" runat="server">
      <TABLE id="Table2" cellSpacing="1" cellPadding="1" border="1" align="center">
        <TR>
          <TD>
            <P>
              <asp:Button id="btnRetrieve" onclick="btnRetrieve_Click" runat="server" Text="Elaborate" CssClass="button"></asp:
                Button></P>
            <P>
              <asp:Button id="btnReset" onclick="btnReset_Click" runat="server" CssClass="button" Text="Reset"></asp:Button></P
                >
          </TD>
          <TD>
            <TABLE id="Table1" cellSpacing="1" cellPadding="1" width="500" border="1">
              <TR>
                <TD style="WIDTH: 138px">
                  <P><STRONG>StudentVersion_id</STRONG></P>
                </TD>
                <TD>
                  <asp:TextBox id="tbxStudentVersion_ID" runat="server" CssClass="textbox"></asp:TextBox></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>StudyPlanCriterion_id</STRONG></TD>
                <TD>
                  <asp:TextBox id="tbxStudyPlanCriterion_ID" runat="server" CssClass="textbox"></asp:TextBox></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Show ids</STRONG></TD>
                <TD>
                  <asp:CheckBox id="chkShowIDs" runat="server"></asp:CheckBox></TD>
              </TR>
              <TR>
                <TD style="WIDTH: 138px"><STRONG>Execution time</STRONG></TD>
                <TD>
                  <asp:Label id="lblExecutionTime" runat="server"></asp:Label></TD>
```

```
            </TR>
          </TABLE>
        </TD>
      </TR>
    </TABLE>
  </form>
  <P></P>
  <P>
    <asp:Label id="lblError" Runat="server" ForeColor="MediumBlue"></asp:Label></P>
 </body>
</HTML>
```

studyplanelaborator.aspx.cs

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data.SqlClient;
using System.Data;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using System.Text;
using System.Web.Security;
using StudyPlanning.DAL;
using StudyPlanning.DAL.Courses;

namespace StudyPlanning.Biz
{
  /// <summary>
  /// Summary description for forgotPassword.
  /// </summary>
  public class StudyPlanElaboratorTest : System.Web.UI.Page
  {
    protected Button btnRetrieve;
    protected System.Web.UI.HtmlControls.HtmlForm form;
    protected System.Web.UI.WebControls.Button btnReset;
    protected System.Web.UI.WebControls.TextBox tbxStudentVersion_ID;
    protected System.Web.UI.WebControls.TextBox tbxStudyPlanCriterion_ID;
    protected System.Web.UI.WebControls.CheckBox chkShowIDs;
    protected System.Web.UI.WebControls.Label lblExecutionTime;
    protected Label lblError;

    private void Page_Load(object sender, System.EventArgs e)
    {
      tbxStudentVersion_ID.Text   = "{D8D64E9D-BFF1-4352-9A77-E1BF7FB5B300}";
      tbxStudyPlanCriterion_ID.Text = "{EFEDC7CE-FB47-48F6-8240-CC359D73DD31}";
    }

    private string writeError(BizException x)
    {
      string strError = "";
      strError += "<h1>Av for søren....</h1>";
      strError += x.ToString();
      if (x.Dal != null)
      {
        strError += "<br><br>" + x.Dal.ToString();
        if (x.Dal.Sql != null)
          strError += "<br><br>" + x.Dal.Sql.ToString();
      }

      return strError;

    }

    protected void btnRetrieve_Click(object sender, System.EventArgs e)
    {
      lblError.Text = "";
      lblExecutionTime.Text = "";
      StudyPlan studyPlan = null;
      StudyPlanElaborator spe = new StudyPlanElaborator();

      spe.StudentVersion_ID   = new Guid(tbxStudentVersion_ID.Text);
      spe.StudyPlanCriterion_ID = new Guid(tbxStudyPlanCriterion_ID.Text);

      DateTime before = DateTime.MinValue;
      DateTime after = DateTime.MinValue;

      try
      {
        before = DateTime.Now;
        studyPlan = spe.Run();
        after = DateTime.Now;
      }
      catch (BizException x)
      {
        lblError.Text = writeError(x);
        return;
      }

      TimeSpan ts = new TimeSpan(after.Ticks - before.Ticks);
      lblExecutionTime.Text = ts.ToString();

      IEnumerator spEnum = studyPlan.StudyPlanPeriods.GetEnumerator();
```

```
IEnumerator sppEnum = null;
StudyPlanPeriod spp = null;
Guid period = new Guid();
CoursePart cp = null;
float workload = 0;
while(spEnum.MoveNext())
{
  spp = ((StudyPlanPeriod)spEnum.Current);
  period = spp.Period;

  if (chkShowIDs.Checked)
  {
    lblError.Text += "<b>" + period.ToString() + "  [" + spp.PeriodNumber + "]</b>  " +
      "(" + spp.DesiredWorkload[0] + " - " + spp.DesiredWorkload[1] + " points)<br><br>";
  }
  else
  {
    StudyPlanning.DAL.Period per = new StudyPlanning.DAL.Period();
    per.Period_ID.Value = period;

    try
    {
      per.Retrieve();
    }
    catch (DalException ex)
    {
      lblError.Text = writeError(new BizException(ex));
      return;
    }
    lblError.Text += "<b>" + per.Name["da-DK"] + "  [" + spp.PeriodNumber + "]</b>  "
      + "(" + spp.DesiredWorkload[0] + " - " + spp.DesiredWorkload[1] + " points)<br><br>";
  }

  lblError.Text += "<table>";

  if (chkShowIDs.Checked)
  {
    lblError.Text += "<tr>";
    lblError.Text += "<td><b>Course_ID</b></td>";
    lblError.Text += "<td><b>CourseVersion_ID</b></td>";
    lblError.Text += "<td><b>Part</b></td>";
    lblError.Text += "<td><b>Workload</b></td>";
    lblError.Text += "</tr>";
  }

  for(int j=0; j < spp.NumberOfCourseParts; j++)
  {
    cp = spp.GetCoursePart(j);
    CourseVersion cv = new CourseVersion();
    cv.CourseVersion_ID.Value = cp.CourseVersion_ID;

    try
    {
      cv.Retrieve();
    }
    catch (DalException ex)
    {
      lblError.Text = writeError(new BizException(ex));
      return;
    }

    lblError.Text += "<tr>";
    if (chkShowIDs.Checked)
    {
      lblError.Text += "<td>" + cp.Course_ID + "</td>";
      lblError.Text += "<td>" + cp.CourseVersion_ID + "</td>";
      if (cv.Parts.Value > 1)
        lblError.Text += "<td style=\"width: 40px;\">" + "(" + cp.Part + "/" + cv.Parts.Value + ")" + "</td>";
      else
        lblError.Text += "<td style=\"width: 40px;\"></td>";
      lblError.Text += "<td>" + cp.Workload + "</td>";
      lblError.Text += "<td>" + cp.ReasonText + "</td>";
      lblError.Text += "<td>";
      for(int i=0; i < cp.SelectedModules.Count; i++)
      {
        Module mod = new Module();
        mod.Module_ID.Value = (int)cp.SelectedModules[i];
        try
        {
          mod.Retrieve();
        }
        catch (DalException ex)
        {
          lblError.Text = writeError(new BizException(ex));
          return;
        }
        if (i.Equals(cp.SelectedModules.Count - 1))
          lblError.Text += mod.Name.Value;
        else
          lblError.Text += mod.Name.Value + ", ";
      }
      lblError.Text += "</td>";
    }
    else
    {
      lblError.Text += "<td>" + cv.Number.Value + "</td>";
      if (cv.Parts.Value > 1)
        lblError.Text += "<td style=\"width: 40px;\">" + "(" + cp.Part + "/" + cv.Parts.Value + ")" + "</td>";
      else
```

```
            lblError.Text += "<td style=\"width: 40px;\"></td>";
          lblError.Text += "<td></td>";
          lblError.Text += "<td>" + cp.Workload;
          lblError.Text += "<td>" + cp.ReasonText + "</td>";
          lblError.Text += "<td>";
          for(int i=0; i < cp.SelectedModules.Count; i++)
          {
            Module mod = new Module();
            mod.Module_ID.Value = (int)cp.SelectedModules[i];
            try
            {
              mod.Retrieve();
            }
            catch (DalException ex)
            {
              lblError.Text = writeError(new BizException(ex));
              return;
            }
            if (i.Equals(cp.SelectedModules.Count - 1))
              lblError.Text += mod.Name.Value;
            else
              lblError.Text += mod.Name.Value + ", ";
          }
          lblError.Text += "</td>";
        }
        lblError.Text += "</tr>";
      }
      workload += spp.TotalWorkload;
      lblError.Text += "<tr>";
      lblError.Text += "<td></td><td></td><td></td><td><b>" + spp.TotalWorkload + "</b></td>";
      lblError.Text += "</tr>";
      lblError.Text += "</table>";
      lblError.Text += "<br>";

      lblError.Text += "<table>";
      lblError.Text += "<tr><td width=500px></td><td><b>" + workload + "</b></td></tr>";
      lblError.Text += "</table>";
    }

    lblError.Text += "<br><br><b>StudyPlan Log</b><br>";
    foreach(string s in studyPlan.Log)
    {
      lblError.Text += s + "<br>";
    }
  }

  protected void btnReset_Click(object sender, System.EventArgs e)
  {
    lblError.Text = "";
    tbxStudentVersion_ID.Text = "";
    tbxStudyPlanCriterion_ID.Text = "";
  }

  #region Web Form Designer generated code
  override protected void OnInit(EventArgs e)
  {
    //
    // CODEGEN: This call is required by the ASP.NET Web Form Designer.
    //
    InitializeComponent();
    base.OnInit(e);
  }

  /// <summary>
  /// Required method for Designer support - do not modify
  /// the contents of this method with the code editor.
  /// </summary>
  private void InitializeComponent()
  {
    this.Load += new System.EventHandler(this.Page_Load);

  }
  #endregion

 }
}
```

Chapter 5

Presentation Tier

5.1 Control Panel

5.1.1 changepassword.aspx

```
<%@ Page language="c#" codebehind="changePassword.aspx.cs" AutoEventWireup="false" Inherits="StudyPlanning.UI.ControlPanel.
 ChangePassword" %>
<%@ Register TagPre↓x="UserControl" TagName="Header" Src="../header.ascx" %>
<%@ Register TagPre↓x="UserControl" TagName="Footer" Src="../footer.ascx" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" >
<html>
  <head>
    <title><asp:Literal id="lblPageTitle" Runat="server" /></title>
    <meta name="GENERATOR" content="Microsoft Visual Studio 7.0">
    <meta name="CODE_LANGUAGE" content="C#">
    <meta name="vs_defaultClientScript" content="JavaScript">
    <meta name="vs_targetSchema" content="http://schemas.microsoft.com/intellisense/ie5">
    <link rel="stylesheet" href="../common/style.css" />
  </head>
  <body>
    <form runat="server" ID="Form1">
    <usercontrol:header id="MyHeader" runat="server" />

    <h2><asp:Label ID="lblPageName" Runat="server" /></h2>

    <br/>
    <br/>
    <br/>

    <usercontrol:footer id="MyFooter" runat="server" />
    </form>
  </body>
</html>
```

5.1.2 changepassword.aspx.cs

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using StudyPlanning.Biz;
using System.Text;

namespace StudyPlanning.UI.ControlPanel
{
  /// <summary>
  /// Supports the "Change Password" page i.e. the page where users can change
  /// their password.
  /// </summary>
  public class ChangePassword : System.Web.UI.Page
  {
    protected System.Web.UI.WebControls.Label lblPageName;
```

```
    protected System.Web.UI.WebControls.Literal lblPageTitle;
    protected StudyPlanning.UI.header MyHeader;
    private Hashtable Texts = new Hashtable();
    private string Culture_ID;
    private int TextGroup_ID = 430;

    private void Page_Load(object sender, System.EventArgs e)
    {
      Culture_ID = Request.Cookies["UserData"]["CultureID"];
      try
      {
        Texts = StudyPlanning.Biz.TextItem.GetTextGroup(TextGroup_ID, Culture_ID);
      }
      catch (BizException bizEx)
      {
        Response.Write("An error occured.");
      }

      lblPageName.Text = Texts["1"].ToString();
      lblPageTitle.Text = Texts["2"].ToString();

      #region Breadcrumb
      //The breadcrumb is built in the following
      string Seperator = " <img src=\"" + System.Con↓guration.Con↓gurationSettings.AppSettings["breadcrumbSeperator"] + "\"/> ";
      StringBuilder Breadcrumb = new StringBuilder();
      Breadcrumb.Append("<a href=\"/\">" + Texts["100"].ToString() + "</a>"); //Home
      Breadcrumb.Append(Seperator);
      Breadcrumb.Append("<a href=\"/controlpanel/\">" + Texts["101"].ToString() + "</a>"); //Control Panel
      Breadcrumb.Append(Seperator);
      Breadcrumb.Append("<strong>" + Texts["102"].ToString() + "</strong>"); //Change Your Password

      MyHeader.Text = Breadcrumb.ToString();
      //<a href='/'>Home</a> <img src='../common/gfx/arrow−right.gif' /> <a href='../controlpanel/'>Control Panel</a> <img src='../common
        /gfx/arrow−right.gif' /> <strong>Change Your Password</strong>
      #endregion
    }

    #region Web Form Designer generated code
    override protected void OnInit(EventArgs e)
    {
      //
      // CODEGEN: This call is required by the ASP.NET Web Form Designer.
      //
      InitializeComponent();
      base.OnInit(e);
    }

    /// <summary>
    /// Required method for Designer support − do not modify
    /// the contents of this method with the code editor.
    /// </summary>
    private void InitializeComponent()
    {
      this.Load += new System.EventHandler(this.Page_Load);
    }
    #endregion
  }
}
```

5.1.3 default.aspx

```
<%@ Page language="c#" codebehind="default.aspx.cs" AutoEventWireup="false" Inherits="StudyPlanning.UI.ControlPanel.Main" %>
<%@ Register TagPre↓x="UserControl" TagName="Header" Src="../header.ascx" %>
<%@ Register TagPre↓x="UserControl" TagName="Footer" Src="../footer.ascx" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" >
<html>
  <head>
    <title><asp:Literal id="lblPageTitle" Runat="server" /></title>
    <meta name="GENERATOR" content="Microsoft Visual Studio 7.0">
    <meta name="CODE_LANGUAGE" content="C#">
    <meta name="vs_defaultClientScript" content="JavaScript">
    <meta name="vs_targetSchema" content="http://schemas.microsoft.com/intellisense/ie5">
    <link rel="stylesheet" href="../common/style.css" />
  </head>
  <body>
    <form runat="server" ID="Form1">
    <usercontrol:header id="MyHeader" runat="server" />

    <h2><asp:Label ID="lblPageName" Runat="server" /></h2>

    <usercontrol:footer id="MyFooter" runat="server" />
    </form>
  </body>
</html>
```

5.1.4 default.aspx.cs

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
```

```
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using StudyPlanning.Biz;
using System.Text;

namespace StudyPlanning.UI.ControlPanel
{
  /// <summary>
  /// Supports the default page of the "Control Panel" section.
  /// </summary>
  public class Main : System.Web.UI.Page
  {
    protected System.Web.UI.WebControls.Label lblPageName;
    protected System.Web.UI.WebControls.Literal lblPageTitle;
    protected StudyPlanning.UI.header MyHeader;
    private Hashtable Texts = new Hashtable();
    private string Culture_ID;
    private int TextGroup_ID = 425;

    private void Page_Load(object sender, System.EventArgs e)
    {
      Culture_ID = Request.Cookies["UserData"]["CultureID"];
      try
      {
        Texts = StudyPlanning.Biz.TextItem.GetTextGroup(TextGroup_ID, Culture_ID);
      }
      catch (BizException bizEx)
      {
        Response.Write("An error occured.");
      }

      lblPageName.Text = Texts["1"].ToString();
      lblPageTitle.Text = Texts["2"].ToString();

      #region Breadcrumb
      //The breadcrumb is built in the following
      string Seperator = " <img src=\"" + System.Con|guration.Con|gurationSettings.AppSettings["breadcrumbSeperator"] + "\"/> ";
      StringBuilder Breadcrumb = new StringBuilder();
      Breadcrumb.Append("<a href=\"/\">" + Texts["100"].ToString() + "</a>"); //Home
      Breadcrumb.Append(Seperator);
      Breadcrumb.Append("<strong>" + Texts["101"].ToString() + "</strong>"); //Your Control Panel
      MyHeader.Text = Breadcrumb.ToString();
      //<a href='/'>Forside</a> <img src='../common/gfx/arrow-right.gif' /> <strong>Your Control Panel</strong>
      #endregion
    }

    #region Web Form Designer generated code
    override protected void OnInit(EventArgs e)
    {
      //
      // CODEGEN: This call is required by the ASP.NET Web Form Designer.
      //
      InitializeComponent();
      base.OnInit(e);
    }

    /// <summary>
    /// Required method for Designer support - do not modify
    /// the contents of this method with the code editor.
    /// </summary>
    private void InitializeComponent()
    {
      this.Load += new System.EventHandler(this.Page_Load);
    }
    #endregion
  }
}
```

5.1.5 userprofile.aspx

```
<%@ Page language="c#" codebehind="userProfile.aspx.cs" AutoEventWireup="false" Inherits="StudyPlanning.UI.ControlPanel.UserProfile"
  %>
<%@ Register TagPre|x="UserControl" TagName="Header" Src="../header.ascx" %>
<%@ Register TagPre|x="UserControl" TagName="Footer" Src="../footer.ascx" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" >
<html>
  <head>
    <title><asp:Literal id="lblPageTitle" Runat="server" /></title>
    <meta name="GENERATOR" content="Microsoft Visual Studio 7.0">
    <meta name="CODE_LANGUAGE" content="C#">
    <meta name="vs_defaultClientScript" content="JavaScript">
    <meta name="vs_targetSchema" content="http://schemas.microsoft.com/intellisense/ie5">
    <link rel="stylesheet" href="../common/style.css" />
  </head>
  <body>
    <form runat="server" ID="Form1">
    <usercontrol:header id="MyHeader" runat="server" />

    <h2><asp:Label ID="lblPageName" Runat="server" /></h2>

    <br/>
    <br/>
```

```
    <br/>

    <usercontrol:footer id="MyFooter" runat="server" />
    </form>
  </body>
</html>
```

5.1.6 userprofile.aspx.cs

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using StudyPlanning.Biz;
using System.Text;

namespace StudyPlanning.UI.ControlPanel
{
  /// <summary>
  /// Supports the "User Profile" page in the "Control Panel" section i.e. the page
  /// where users can view and manage their user profile.
  /// </summary>
  public class UserPro↓le : System.Web.UI.Page
  {
    protected System.Web.UI.WebControls.Label lblPageName;
    protected System.Web.UI.WebControls.Literal lblPageTitle;
    protected StudyPlanning.UI.header MyHeader;
    private Hashtable Texts = new Hashtable();
    private string Culture_ID;
    private int TextGroup_ID = 435;

    private void Page_Load(object sender, System.EventArgs e)
    {
      Culture_ID = Request.Cookies["UserData"]["CultureID"];
      try
      {
        Texts = StudyPlanning.Biz.TextItem.GetTextGroup(TextGroup_ID, Culture_ID);
      }
      catch (BizException bizEx)
      {
        Response.Write("An error occured.");
      }

      lblPageName.Text = Texts["1"].ToString();
      lblPageTitle.Text = Texts["2"].ToString();

      #region Breadcrumb
      //The breadcrumb is built in the following
      string Seperator = " <img src=\"" + System.Con↓guration.Con↓gurationSettings.AppSettings["breadcrumbSeperator"] + "\"/> ";
      StringBuilder Breadcrumb = new StringBuilder();
      Breadcrumb.Append("<a href=\"/\">" + Texts["100"].ToString() + "</a>"); //Home
      Breadcrumb.Append(Seperator);
      Breadcrumb.Append("<a href=\"/controlpanel/\">" + Texts["101"].ToString() + "</a>"); //Control Panel
      Breadcrumb.Append(Seperator);
      Breadcrumb.Append("<strong>" + Texts["102"].ToString() + "</strong>"); //Your User Profile

      MyHeader.Text = Breadcrumb.ToString();
      //<a href='/'>Home</a> <img src='../common/gfx/arrow−right.gif' /> <a href='../controlpanel/'>Control Panel</a> <img src='../common
        /gfx/arrow−right.gif' /> <strong>Your User Profile</strong>
      #endregion
    }

    #region Web Form Designer generated code
    override protected void OnInit(EventArgs e)
    {
      //
      // CODEGEN: This call is required by the ASP.NET Web Form Designer.
      //
      InitializeComponent();
      base.OnInit(e);
    }

    /// <summary>
    /// Required method for Designer support − do not modify
    /// the contents of this method with the code editor.
    /// </summary>
    private void InitializeComponent()
    {
      this.Load += new System.EventHandler(this.Page_Load);
    }
    #endregion
  }
}
```

5.2 Course Base

5.2.1 alfabetical.aspx

```
<%@ Page language="c#" codebehind="alfabetical.aspx.cs" AutoEventWireup="false" Inherits="StudyPlanning.UI.CourseBase.Alfabetical" %
  >
<%@ Register TagPre↓x="UserControl" TagName="Header" Src="../header.ascx" %>
<%@ Register TagPre↓x="UserControl" TagName="Footer" Src="../footer.ascx" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" >
<html>
  <head>
    <title>Alle kurser i alfabetisk rækkefølge</title>
    <meta name="GENERATOR" content="Microsoft Visual Studio 7.0">
    <meta name="CODE_LANGUAGE" content="C#">
    <meta name="vs_defaultClientScript" content="JavaScript">
    <meta name="vs_targetSchema" content="http://schemas.microsoft.com/intellisense/ie5">
    <link rel="stylesheet" href="../common/style.css" />
  </head>
  <body>
    <form runat="server" ID="Form1">
    <usercontrol:header id="MyHeader" runat="server" Text="<a href='/'>Forside</a> <img src='../common/gfx/arrow-right.gif' /> <a href='../
      coursebase/'>Kursusbasen</a> <img src='../common/gfx/arrow-right.gif' /> <strong>Alle kurser i alfabetisk rækkefølge</strong>" />

    <h2>Alle kurser i alfabetisk rækkefølge</h2>

    <br/>
    <br/>
    <br/>
    <br/>
    <br/>
    <br/>

    <usercontrol:footer id="MyFooter" runat="server" />
    </form>
  </body>
</html>
```

5.2.2 alfabetical.aspx.cs

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;

namespace StudyPlanning.UI.CourseBase
{
  /// <summary>
  /// Summary description for WebForm1.
  /// </summary>
  public class Alfabetical : System.Web.UI.Page
  {
    private void Page_Load(object sender, System.EventArgs e)
    {
    }

    #region Web Form Designer generated code
    override protected void OnInit(EventArgs e)
    {
      //
      // CODEGEN: This call is required by the ASP.NET Web Form Designer.
      //
      InitializeComponent();
      base.OnInit(e);
    }

    /// <summary>
    /// Required method for Designer support - do not modify
    /// the contents of this method with the code editor.
    /// </summary>
    private void InitializeComponent()
    {
      this.Load += new System.EventHandler(this.Page_Load);
    }
    #endregion
  }
}
```

5.2.3 default.aspx

```
<%@ Page Language="c#" Codebehind="default.aspx.cs" AutoEventWireup="false" Inherits="StudyPlanning.UI.CourseBase.Main" %>
<%@ Register TagPre↓x="UserControl" TagName="Header" Src="../header.ascx" %>
```

```
<%@ Register TagPre↓x="UserControl" TagName="Footer" Src="../footer.ascx" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" >
<html>
  <head>
    <title><asp:Literal id="lblPageTitle" Runat="server" /></title>
    <meta name="GENERATOR" content="Microsoft Visual Studio 7.0">
    <meta name="CODE_LANGUAGE" content="C#">
    <meta name="vs_defaultClientScript" content="JavaScript">
    <meta name="vs_targetSchema" content="http://schemas.microsoft.com/intellisense/ie5">
    <link rel="stylesheet" href="../common/style.css" />
  </head>
  <body>
    <form runat="server" ID="Form1">
    <usercontrol:header id="MyHeader" runat="server" />

    <h2><asp:Label ID="lblPageName" Runat="server" /></h2>

    <usercontrol:footer id="MyFooter" runat="server" />
    </form>
  </body>
</html>
```

5.2.4 default.aspx.cs

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using StudyPlanning.Biz;
using System.Text;

namespace StudyPlanning.UI.CourseBase
{
  /// <summary>
  /// Supports the default page in the "Course Base" section.
  /// </summary>
  public class Main : System.Web.UI.Page
  {
    protected System.Web.UI.WebControls.Label lblPageName;
    protected System.Web.UI.WebControls.Literal lblPageTitle;
    protected StudyPlanning.UI.header MyHeader;
    private Hashtable Texts = new Hashtable();
    private string Culture_ID;
    private int TextGroup_ID = 335;

    private void Page_Load(object sender, System.EventArgs e)
    {
      Culture_ID = Request.Cookies["UserData"]["CultureID"];
      try
      {
        Texts = StudyPlanning.Biz.TextItem.GetTextGroup(TextGroup_ID, Culture_ID);
      }
      catch (BizException bizEx)
      {
        Response.Write("An error occured.");
      }

      lblPageName.Text = Texts["1"].ToString();
      lblPageTitle.Text = Texts["2"].ToString();

      #region Breadcrumb
      //The breadcrumb is built in the following
      string Seperator = " <img src=\"" + System.Con↓guration.Con↓gurationSettings.AppSettings["breadcrumbSeperator"] + "\"/> ";
      StringBuilder Breadcrumb = new StringBuilder();
      Breadcrumb.Append("<a href=\"/\">" + Texts["100"].ToString() + "</a>"); //Home
      Breadcrumb.Append(Seperator);
      Breadcrumb.Append("<strong>" + Texts["101"].ToString() + "</strong>"); //Your Control Panel
      MyHeader.Text = Breadcrumb.ToString();
      //<a href='/'>Home</a> <img src='../common/gfx/arrow-right.gif' /> <strong>Course Base</strong>
      #endregion
    }

    #region Web Form Designer generated code
    override protected void OnInit(EventArgs e)
    {
      //
      // CODEGEN: This call is required by the ASP.NET Web Form Designer.
      //
      InitializeComponent();
      base.OnInit(e);
    }

    /// <summary>
    /// Required method for Designer support - do not modify
    /// the contents of this method with the code editor.
    /// </summary>
    private void InitializeComponent()
    {
      this.Load += new System.EventHandler(this.Page_Load);
```

```
    }
    #endregion
  }
}
```

5.2.5 institute.aspx

```
<%@ Page language="c#" codebehind="institute.aspx.cs" AutoEventWireup="false" Inherits="StudyPlanning.UI.CourseBase.Institute" %>
<%@ Register TagPrefix="UserControl" TagName="Header" Src="../header.ascx" %>
<%@ Register TagPrefix="UserControl" TagName="Footer" Src="../footer.ascx" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" >
<html>
  <head>
    <title>Kurser i forhold til institut</title>
    <meta name="GENERATOR" content="Microsoft Visual Studio 7.0">
    <meta name="CODE_LANGUAGE" content="C#">
    <meta name="vs_defaultClientScript" content="JavaScript">
    <meta name="vs_targetSchema" content="http://schemas.microsoft.com/intellisense/ie5">
    <link rel="stylesheet" href="../common/style.css" />
  </head>
  <body>
    <form runat="server" ID="Form1">
    <usercontrol:header id="MyHeader" runat="server" Text="<a href='/'>Forside</a> <img src='../common/gfx/arrow-right.gif' /> <a href='../
      coursebase/'>Kursusbasen</a> <img src='../common/gfx/arrow-right.gif' /> <strong>Kurser i forhold til institut</strong>" />

    <h2>Kurser i forhold til institut</h2>

    <a href="../studyplanning/">Studieplanlægning</a><br/>
    <a href="../studyinfo/">Studieinfo</a>

    <usercontrol:footer id="MyFooter" runat="server" />
    </form>
  </body>
</html>
```

5.2.6 institute.aspx.cs

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;

namespace StudyPlanning.UI.CourseBase
{
  /// <summary>
  /// Summary description for WebForm1.
  /// </summary>
  public class Institute : System.Web.UI.Page
  {
    private void Page_Load(object sender, System.EventArgs e)
    {
    }

    #region Web Form Designer generated code
    override protected void OnInit(EventArgs e)
    {
      //
      // CODEGEN: This call is required by the ASP.NET Web Form Designer.
      //
      InitializeComponent();
      base.OnInit(e);
    }

    /// <summary>
    /// Required method for Designer support - do not modify
    /// the contents of this method with the code editor.
    /// </summary>
    private void InitializeComponent()
    {
      this.Load += new System.EventHandler(this.Page_Load);
    }
    #endregion
  }
}
```

5.2.7 search.aspx

```
<%@ Page language="c#" codebehind="search.aspx.cs" AutoEventWireup="false" Inherits="StudyPlanning.UI.CourseBase.Search" %>
<%@ Register TagPrefix="UserControl" TagName="Header" Src="../header.ascx" %>
<%@ Register TagPrefix="UserControl" TagName="Footer" Src="../footer.ascx" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" >
<html>
```

```
  <head>
    <title>Søg efter kurser</title>
    <meta name="GENERATOR" content="Microsoft Visual Studio 7.0">
    <meta name="CODE_LANGUAGE" content="C#">
    <meta name="vs_defaultClientScript" content="JavaScript">
    <meta name="vs_targetSchema" content="http://schemas.microsoft.com/intellisense/ie5">
    <link rel="stylesheet" href="../common/style.css" />
  </head>
  <body>
    <form runat="server" ID="Form1">
    <usercontrol:header id="MyHeader" runat="server" Text="<a href='/'>Forside</a> <img src='../common/gfx/arrow-right.gif' /> <a href='../
      coursebase/'>Kursusbasen</a> <img src='../common/gfx/arrow-right.gif' /> <strong>Søg efter kurser</strong>" />

    <h2>Søg efter kurser</h2>

    <a href="../studyplanning/">Studieplanlægning</a><br/>
    <a href="../studyinfo/">Studieinfo</a>

    <usercontrol:footer id="MyFooter" runat="server" />
    </form>
  </body>
</html>
```

5.2.8 search.aspx.cs

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;

namespace StudyPlanning.UI.CourseBase
{
  /// <summary>
  /// Summary description for WebForm1.
  /// </summary>
  public class Search : System.Web.UI.Page
  {
    private void Page_Load(object sender, System.EventArgs e)
    {
    }

    #region Web Form Designer generated code
    override protected void OnInit(EventArgs e)
    {
      //
      // CODEGEN: This call is required by the ASP.NET Web Form Designer.
      //
      InitializeComponent();
      base.OnInit(e);
    }

    /// <summary>
    /// Required method for Designer support - do not modify
    /// the contents of this method with the code editor.
    /// </summary>
    private void InitializeComponent()
    {
      this.Load += new System.EventHandler(this.Page_Load);
    }
    #endregion
  }
}
```

5.3 Planning

5.3.1 default.aspx

```
<%@ Page language="c#" codebehind="default.aspx.cs" AutoEventWireup="false" Inherits="StudyPlanning.UI.Planning.Main" %>
<%@ Register TagPre⌊x="UserControl" TagName="Header" Src="../header.ascx" %>
<%@ Register TagPre⌊x="UserControl" TagName="Footer" Src="../footer.ascx" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" >
<html>
  <head>
    <title><asp:Literal id="lblPageTitle" Runat="server" /></title>
    <meta name="GENERATOR" content="Microsoft Visual Studio 7.0">
    <meta name="CODE_LANGUAGE" content="C#">
    <meta name="vs_defaultClientScript" content="JavaScript">
    <meta name="vs_targetSchema" content="http://schemas.microsoft.com/intellisense/ie5">
    <link rel="stylesheet" href="../common/style.css" />
  </head>
  <body>
    <form runat="server" ID="Form1">
```

```
      <usercontrol:header id="MyHeader" runat="server" />

      <h2><asp:Label ID="lblPageName" Runat="server" /></h2>

      <usercontrol:footer id="MyFooter" runat="server" />
    </form>
  </body>
</html>
```

5.3.2 default.aspx.cs

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using StudyPlanning.Biz;
using System.Text;

namespace StudyPlanning.UI.Planning
{
  /// <summary>
  /// Supports the default page of the "Study Planning" section.
  /// </summary>
  public class Main : System.Web.UI.Page
  {
    protected System.Web.UI.WebControls.Label lblPageName;
    protected System.Web.UI.WebControls.Literal lblPageTitle;
    protected StudyPlanning.UI.header MyHeader;
    private Hashtable Texts = new Hashtable();
    private string Culture_ID;
    private int TextGroup_ID = 235;

    private void Page_Load(object sender, System.EventArgs e)
    {
      Culture_ID = Request.Cookies["UserData"]["CultureID"];
      try
      {
        Texts = StudyPlanning.Biz.TextItem.GetTextGroup(TextGroup_ID, Culture_ID);
      }
      catch (BizException bizEx)
      {
        Response.Write("An error occured.");
      }

      lblPageName.Text = Texts["1"].ToString();
      lblPageTitle.Text = Texts["2"].ToString();

      #region Breadcrumb
      //The breadcrumb is built in the following
      string Seperator = " <img src=\"" + System.Con↓guration.Con↓gurationSettings.AppSettings["breadcrumbSeperator"] + "\"/> ";
      StringBuilder Breadcrumb = new StringBuilder();
      Breadcrumb.Append("<a href=\"/\">" + Texts["100"].ToString() + "</a>"); //Home
      Breadcrumb.Append(Seperator);
      Breadcrumb.Append("<strong>" + Texts["101"].ToString() + "</strong>"); //Your Control Panel
      MyHeader.Text = Breadcrumb.ToString();
      //<a href='/'>Forside</a> <img src='../common/gfx/arrow-right.gif' /> <strong>Study Planning</strong>
      #endregion
    }

    #region Web Form Designer generated code
    override protected void OnInit(EventArgs e)
    {
      //
      // CODEGEN: This call is required by the ASP.NET Web Form Designer.
      //
      InitializeComponent();
      base.OnInit(e);
    }

    /// <summary>
    /// Required method for Designer support - do not modify
    /// the contents of this method with the code editor.
    /// </summary>
    private void InitializeComponent()
    {
      this.Load += new System.EventHandler(this.Page_Load);
    }
    #endregion
  }
}
```

5.3.3 studyplan.aspx

```
<%@ Page language="c#" codebehind="studyplan.aspx.cs" AutoEventWireup="false" Inherits="StudyPlanning.UI.Planning.StudyPlan" %>
<%@ Register TagPre↓x="UserControl" TagName="Header" Src="../header.ascx" %>
<%@ Register TagPre↓x="UserControl" TagName="Footer" Src="../footer.ascx" %>
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" >
<html>
  <head>
    <title><asp:Literal id="lblPageTitle" Runat="server" /></title>
    <meta name="GENERATOR" content="Microsoft Visual Studio 7.0">
    <meta name="CODE_LANGUAGE" content="C#">
    <meta name="vs_defaultClientScript" content="JavaScript">
    <meta name="vs_targetSchema" content="http://schemas.microsoft.com/intellisense/ie5">
    <link rel="stylesheet" href="../common/style.css" />
    <style>
      .datagrid
      {
        margin-left: 30px;
      }

      .button
      {
        width: 100px;
      }
    </style>
  </head>
  <body>
    <form runat="server" ID="Form1">
      <usercontrol:header id="MyHeader" runat="server" />

      <h2><asp:Label ID="lblPageName" Runat="server" /></h2>

      <asp:DataGrid ID="studyPlan" Runat="server"
         GridLines="Both"
         BorderWidth="1"
         BorderColor="#000000"
         CellPadding="5"
         HeaderStyle-Font-Bold="True"
         HeaderStyle-Font-Size="110%"
         HeaderStyle-BackColor="#d3d3d3"
         CssClass="datagrid"
         AlternatingItemStyle-BackColor="#e0dfe3"
         AutoGenerateColumns="True"
      >
      </asp:DataGrid>

      <usercontrol:footer id="MyFooter" runat="server" />
    </form>
  </body>
</html>
```

5.3.4 studyplan.aspx.cs

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using StudyPlanning.Biz;
using System.Text;

namespace StudyPlanning.UI.Planning
{
  /// <summary>
  /// Supports the "Study Plans" page in the "Study Planning" section.
  /// </summary>
  public class StudyPlan : System.Web.UI.Page
  {
    protected System.Web.UI.WebControls.Label lblPageName;
    protected System.Web.UI.WebControls.Literal lblPageTitle;
    protected StudyPlanning.UI.header MyHeader;
    private Hashtable Texts = new Hashtable();
    private string Culture_ID;
    private int TextGroup_ID = 245;

    protected DataGrid studyPlan;

    private void Page_Load(object sender, System.EventArgs e)
    {
      Culture_ID = Request.Cookies["UserData"]["CultureID"];
      try
      {
        Texts = StudyPlanning.Biz.TextItem.GetTextGroup(TextGroup_ID, Culture_ID);
      }
      catch (BizException bizEx)
      {
        Response.Write("An error occured.");
      }

      lblPageName.Text = "Vis dittelidutplan"; //Texts["1"].ToString();
      lblPageTitle.Text = Texts["2"].ToString();

      #region Breadcrumb
      //The breadcrumb is built in the following
      string Seperator = " <img src=\"" + System.Con↓guration.Con↓gurationSettings.AppSettings["breadcrumbSeperator"] + "\"/> ";
```

```
        StringBuilder Breadcrumb = new StringBuilder();
        Breadcrumb.Append("<a href=\"/\">" + Texts["100"].ToString() + "</a>"); //Home
        Breadcrumb.Append(Seperator);
        Breadcrumb.Append("<a href=\"/planning/\">" + Texts["101"].ToString() + "</a>"); //Study Planning
        Breadcrumb.Append(Seperator);
        Breadcrumb.Append("<strong>" + Texts["102"].ToString() + "</strong>"); //Study Plans

        MyHeader.Text = Breadcrumb.ToString();
        //<a href='/'>Home</a> <img src='../common/gfx/arrow-right.gif' /> <a href='../planning/'>Study Planning</a> <img src='../common/
          gfx/arrow-right.gif' /> <strong>Study Plans</strong>
        #endregion

        System.Collections.Specialized.NameValueCollection objQueryString = this.Request.QueryString;
        Guid studyPlan_ID = new Guid(objQueryString["studyPlan_ID"]);

        /*
        string Xml = "";
        try
        {
          Xml = StudyPlanning.Biz.StudyPlan.GetStudyPlan(studyPlan_ID, Culture_ID).GetXmlSchema(); //.GetXml();
        }
        catch (BizException objEx)
        {
          Response.Write(objEx.Message);
        }

        Response.Write(Server.HtmlEncode(Xml));
        */
        BindStudyPlan(studyPlan_ID, Culture_ID);
      }

      private void BindStudyPlan(Guid studyPlan_ID, string culture_ID)
      {
        DataSet objStudyPlan = new DataSet();

        try
        {
          objStudyPlan = StudyPlanning.Biz.StudyPlan.GetStudyPlan(studyPlan_ID, culture_ID);
        }
        catch (BizException objEx)
        {
          Response.Write(objEx.Message + "<br/><br/>");
        }

        studyPlan.DataSource = objStudyPlan.Tables["Periods"].DefaultView;
        studyPlan.DataBind();
      }

      #region Web Form Designer generated code
      override protected void OnInit(EventArgs e)
      {
        //
        // CODEGEN: This call is required by the ASP.NET Web Form Designer.
        //
        InitializeComponent();
        base.OnInit(e);
      }

      /// <summary>
      /// Required method for Designer support - do not modify
      /// the contents of this method with the code editor.
      /// </summary>
      private void InitializeComponent()
      {
        this.Load += new System.EventHandler(this.Page_Load);
      }
      #endregion
    }
}
```

5.3.5 studyplancriteria.aspx

```
<%@ Page language="c#" codebehind="studyplancriteria.aspx.cs" AutoEventWireup="false" Inherits="StudyPlanning.UI.Planning.
  StudyPlanCriteria" %>
<%@ Register TagPrefix="UserControl" TagName="Header" Src="../header.ascx" %>
<%@ Register TagPrefix="UserControl" TagName="Footer" Src="../footer.ascx" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" >
<html>
  <head>
    <title><asp:Literal id="lblPageTitle" Runat="server" /></title>
    <meta name="GENERATOR" content="Microsoft Visual Studio 7.0">
    <meta name="CODE_LANGUAGE" content="C#">
    <meta name="vs_defaultClientScript" content="JavaScript">
    <meta name="vs_targetSchema" content="http://schemas.microsoft.com/intellisense/ie5">
    <link rel="stylesheet" href="../common/style.css" />
    <style>
      .datagrid
      {
        margin-left: 30px;
      }

      .button
      {
        width: 100px;
      }
```

```
    </style>
  </head>
  <body>
    <form runat="server" ID="Form1">
      <usercontrol:header id="MyHeader" runat="server" />

      <h2><asp:Label ID="lblPageName" Runat="server" /></h2>

      <asp:DataGrid ID="studyPlanCriteria" Runat="server"
        GridLines="Both"
        BorderWidth="1"
        BorderColor="#000000"
        CellPadding="5"
        HeaderStyle-Font-Bold="True"
        HeaderStyle-Font-Size="110%"
        HeaderStyle-BackColor="#d3d3d3"
        CssClass="datagrid"
        AlternatingItemStyle-BackColor="#e0dfe3"
        AutoGenerateColumns="False"
        OnItemCommand="DoAction"
      >
        <Columns>
          <asp:BoundColumn HeaderText="Name" DataField="Name"></asp:BoundColumn>
          <asp:BoundColumn HeaderText="Created" DataField="Created" DataFormatString="{0:g}"></asp:BoundColumn>
          <asp:TemplateColumn HeaderText="Actions" HeaderStyle-HorizontalAlign="Center">
            <ItemTemplate>
              <input type="hidden" value='<%# DataBinder.Eval(Container.DataItem, "ID")%>' id="id" name="id" runat="server"
                />
              <asp:Button ID="view" Runat="server" Text="Edit" CssClass="button" CommandName="View" />
              <asp:Button ID="delete" Runat="server" Text="Delete" CssClass="button" CommandName="Delete" />
              <asp:Button ID="elaborate" Runat="server" Text="Elaborate Study Plan" CssClass="button" Width="160px"
                CommandName="Elaborate" />
            </ItemTemplate>
          </asp:TemplateColumn>
        </Columns>

      </asp:DataGrid>

      <br/><br/><a href="studyplancriterion.aspx">studyplancriterion.aspx</a>

      <usercontrol:footer id="MyFooter" runat="server" />
    </form>
  </body>
</html>
```

5.3.6 studyplancriteria.aspx.cs

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using StudyPlanning.Biz;
using System.Text;

namespace StudyPlanning.UI.Planning
{
  /// <summary>
  /// Supports the "Study Plan Criteria" page in the "Study Planning" section.
  /// </summary>
  public class StudyPlanCriteria : System.Web.UI.Page
  {
    protected System.Web.UI.WebControls.Label lblPageName;
    protected System.Web.UI.WebControls.Literal lblPageTitle;
    protected StudyPlanning.UI.header MyHeader;
    private Hashtable Texts = new Hashtable();
    private string Culture_ID;
    private int TextGroup_ID = 240;

    protected DataGrid studyPlanCriteria;

    private void Page_Load(object sender, System.EventArgs e)
    {
      Culture_ID = Request.Cookies["UserData"]["CultureID"];
      try
      {
        Texts = StudyPlanning.Biz.TextItem.GetTextGroup(TextGroup_ID, Culture_ID);
      }
      catch (BizException bizEx)
      {
        Response.Write("An error occured.");
      }

      lblPageName.Text = Texts["1"].ToString();
      lblPageTitle.Text = Texts["2"].ToString();

      #region Breadcrumb
      //The breadcrumb is built in the following
      string Seperator = " <img src=\"" + System.Con↓guration.Con↓gurationSettings.AppSettings["breadcrumbSeperator"] + "\"/> ";
      StringBuilder Breadcrumb = new StringBuilder();
```

```
        Breadcrumb.Append("<a href=\"/\">" + Texts["100"].ToString() + "</a>"); //Home
        Breadcrumb.Append(Seperator);
        Breadcrumb.Append("<a href=\"/planning/\">" + Texts["101"].ToString() + "</a>"); //Study Planning
        Breadcrumb.Append(Seperator);
        Breadcrumb.Append("<strong>" + Texts["102"].ToString() + "</strong>"); //Study Plan Criteria

        MyHeader.Text = Breadcrumb.ToString();
        //<a href='/'>Home</a> <img src='../common/gfx/arrow-right.gif' /> <a href='../planning/'>Study Planning</a> <img src='../common/
          gfx/arrow-right.gif' /> <strong>Study Plan Criteria</strong>
        #endregion

        if (!Page.IsPostBack)
        {
          BindStudyPlanCriteria(new Guid("{68C9F6D6-633E-480c-A610-F3DB5A9734DA}"));
        }
      }

      private void BindStudyPlanCriteria(Guid student_ID)
      {
        System.Data.DataTable objStudyPlanCriteria =
          new System.Data.DataTable();

        try
        {
          objStudyPlanCriteria = StudyPlanning.Biz.Student.GetStudyPlanCriteria(student_ID);
        }
        catch (BizException objEx)
        {
          Response.Write(objEx.Message + "<br/><br/>");
        }

        studyPlanCriteria.DataSource = objStudyPlanCriteria.DefaultView;
        studyPlanCriteria.DataBind();

      }

      protected void DoAction(object objSender, DataGridCommandEventArgs objArgs)
      {
        //Guid studyPlan_ID = new Guid(((System.Web.UI.HtmlControls.HtmlInputHidden)objArgs.Item.Cells[2].FindControl("id")).Value);

        if (objArgs.CommandName == "View")
        {
        }
        else if (objArgs.CommandName == "Delete")
        {
        }
      }

      #region Web Form Designer generated code
      override protected void OnInit(EventArgs e)
      {
        //
        // CODEGEN: This call is required by the ASP.NET Web Form Designer.
        //
        InitializeComponent();
        base.OnInit(e);
      }

      /// <summary>
      /// Required method for Designer support - do not modify
      /// the contents of this method with the code editor.
      /// </summary>
      private void InitializeComponent()
      {
        this.Load += new System.EventHandler(this.Page_Load);
      }
      #endregion
    }
}
```

5.3.7 studyplancriterion.aspx

```
<%@ Page language="c#" codebehind="studyplancriterion.aspx.cs" AutoEventWireup="false" Inherits="StudyPlanning.UI.Planning.
  StudyPlanCriterion" %>
<%@ Import Namespace="Microsoft.Web.UI.WebControls" %>
<%@ Register TagPre|x="UserControl" TagName="Footer" Src="../footer.ascx" %>
<%@ Register TagPre|x="UserControl" TagName="Header" Src="../header.ascx" %>
<%@ Register TagPre|x="cyberakt" Namespace="CYBERAKT.WebControls.Navigation" Assembly="ASPnetMenu" %>
<%@ Register TagPre|x="Microsoft" Namespace="Microsoft.Web.UI.WebControls" Assembly="Microsoft.Web.UI.WebControls, Version=1.0.2.226,
  Culture=neutral, PublicKeyToken=31bf3856ad364e35" %>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" >
<HTML>
  <HEAD>
    <title>
      <asp:Literal id="lblPageTitle" Runat="server" /></title>
    <meta name="GENERATOR" content="Microsoft Visual Studio 7.0">
    <meta name="CODE_LANGUAGE" content="C#">
    <meta name="vs_defaultClientScript" content="JavaScript">
    <meta name="vs_targetSchema" content="http://schemas.microsoft.com/intellisense/ie5">
    <link rel="stylesheet" href="../common/style.css">
    <link rel="stylesheet" href="studyplancriterion_style.css">
    <style>
      .datagrid { MARGIN-LEFT: 30px }
      .button { WIDTH: 100px }
```

```
      .buttonsection { BORDER−RIGHT: #999999 1px solid; BORDER−TOP: #999999 1px solid; BORDER−LEFT: #f1f1f1 1px
        solid; BORDER−BOTTOM: #f1f1f1 1px solid; BACKGROUND−COLOR: #f1f1f1 }
      .leftsection { BORDER−RIGHT: #999999 1px solid; BORDER−TOP: #f1f1f1 1px solid; BORDER−LEFT: #f1f1f1 1px solid;
        BORDER−BOTTOM: #f1f1f1 1px solid; BACKGROUND−COLOR: #f1f1f1 }
      .contentsection { BORDER−TOP: #999999 1px solid; PADDING−LEFT: 15px }
    </style>
  </HEAD>
  <body>
    <form runat="server" ID="Form1">
      <usercontrol:header id="MyHeader" runat="server" />
      <h2><asp:Label ID="lblPageName" Runat="server" /></h2>
      <table border="0" cellpadding="5" cellspacing="0" width="100%" style="MARGIN-LEFT:-10px"
        class="buttonsection">
        <tr>
          <td>
            <asp:Button ID="save" Runat="server" Text="Save Criterion" style="MARGIN-LEFT:20px" OnCommand="DoAction"
              CommandName="save" />
            <asp:Button ID="cancel" Runat="server" Text="Cancel" OnCommand="DoAction" CommandName="cancel" />
            <asp:Button ID="elaborate" Runat="server" Text="Elaborate Study Plan" OnCommand="DoAction" CommandName="elaborate
              " />
          </td>
        </tr>
      </table>
      <table border="0" cellpadding="0" cellspacing="0" height="100%" style="MARGIN-LEFT:-10px">
        <tr>
          <td width="200" nowrap valign="top" class="-leftsection">
            <cyberakt:ASPnetMenu
              id="critMenu"
              runat="server"
              MenuStyle="ClassicVertical"
              Height="100%"
            />
          </td>
          <td width="100%" valign="top" class="contentsection">

            <Microsoft:multipage id="criteria" runat="server" selectedindex="0">
              <Microsoft:pageview id="general">
                <h2>Generelle indstillinger</h2>
              </Microsoft:pageview>
              <Microsoft:pageview id="interests">
                <h2>Generelle indstillinger</h2>
              </Microsoft:pageview>
              <Microsoft:pageview id="technicalPackage">
                <h2>Fagpakke, retningsbetegnelse m.m.</h2>
              </Microsoft:pageview>
              <Microsoft:pageview id="courses">
                <h2>Til−/fravalg af kurser</h2>
              </Microsoft:pageview>
              <Microsoft:pageview id="workload">
                <h2>Arbejdsbyrde</h2>
              </Microsoft:pageview>
              <Microsoft:pageview id="projects">
                <h2>Projekter</h2>
              </Microsoft:pageview>
            </Microsoft:multipage>
          </td>
        </tr>
      </table>
      <usercontrol:footer id="MyFooter" runat="server" />
    </form>
  </body>
</HTML>
```

5.3.8 studyplancriterion.aspx.cs

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using StudyPlanning.Biz;
using System.Text;
using Microsoft.Web.UI.WebControls;
using CYBERAKT.WebControls.Navigation;

namespace StudyPlanning.UI.Planning
{
  /// <summary>
  /// Supports the "Study Plan Criterion" page in the "Study Planning" section.
  /// </summary>
  public class StudyPlanCriterion : System.Web.UI.Page
  {
    protected System.Web.UI.WebControls.Label lblPageName;
    protected System.Web.UI.WebControls.Literal lblPageTitle;
    protected StudyPlanning.UI.header MyHeader;
    private Hashtable Texts = new Hashtable();
    private string Culture_ID;
    private int TextGroup_ID = 240;
    protected System.Web.UI.WebControls.Button save;
    protected System.Web.UI.WebControls.Button cancel;
```

```
protected System.Web.UI.WebControls.Button elaborate;
protected Microsoft.Web.UI.WebControls.MultiPage criteria;

protected CYBERAKT.WebControls.Navigation.ASPnetMenu critMenu;
protected CYBERAKT.WebControls.Navigation.MenuItem menuItem;

private void Page_Load(object sender, System.EventArgs e)
{
  Culture_ID = Request.Cookies["UserData"]["CultureID"];
  try
  {
    Texts = StudyPlanning.Biz.TextItem.GetTextGroup(TextGroup_ID, Culture_ID);
  }
  catch (BizException bizEx)
  {
    Response.Write("An error occured.");
  }

  lblPageName.Text = "Creation of Study Plan Criterion"; //Texts["1"].ToString();
  lblPageTitle.Text = Texts["2"].ToString();

  #region Breadcrumb
  //The breadcrumb is built in the following
  string Seperator = " <img src=\"" + System.Con|guration.Con|gurationSettings.AppSettings["breadcrumbSeperator"] + "\"/> ";
  StringBuilder Breadcrumb = new StringBuilder();
  Breadcrumb.Append("<a href=\"/\">" + Texts["100"].ToString() + "</a>"); //Home
  Breadcrumb.Append(Seperator);
  Breadcrumb.Append("<a href=\"/planning/\">" + Texts["101"].ToString() + "</a>"); //Study Planning
  Breadcrumb.Append(Seperator);
  Breadcrumb.Append("<strong>" + Texts["102"].ToString() + "</strong>"); //Study Plan Criteria

  MyHeader.Text = Breadcrumb.ToString();
  //<a href='/'>Home</a> <img src='../common/gfx/arrow-right.gif' /> <a href='../planning/'>Study Planning</a> <img src='../common/
    gfx/arrow-right.gif' /> <strong>Study Plan Criteria</strong>
  #endregion

  if (!this.IsPostBack)
  {
    BuildMenu();
  }

  criteria.SelectedIndex = 2;
}

private void BuildMenu()
{
  critMenu.CssClass = "CritMenu";

  menuItem = critMenu.TopGroup.Items.Add();
  menuItem.CssClass = "CritMenu-Break";
  menuItem.Label = " ";
  menuItem.Height = "30px";
  //menuItem.Height = "50px";

  menuItem = critMenu.TopGroup.Items.Add();
  menuItem.CssClass = "CritMenu-MenuItem";
  menuItem.CssClassOver = "CritMenu-MenuItemOver";
  menuItem.Label = "Generelle indstillinger";
  menuItem.ID = "critMenuGeneral";
  menuItem.Height = "20";

  menuItem = critMenu.TopGroup.Items.Add();
  menuItem.CssClass = "CritMenu-Break";
  menuItem.Label = " ";
  menuItem.Height = "5px";

  menuItem = critMenu.TopGroup.Items.Add();
  menuItem.CssClass = "CritMenu-MenuItem";
  menuItem.CssClassOver = "CritMenu-MenuItemOver";
  menuItem.Label = "Interesser m.m.";
  menuItem.ID = "critMenuInterests";
  menuItem.Height = "20";

  menuItem = critMenu.TopGroup.Items.Add();
  menuItem.CssClass = "CritMenu-Break";
  menuItem.Label = " ";
  menuItem.Height = "5px";

  menuItem = critMenu.TopGroup.Items.Add();
  menuItem.CssClass = "CritMenu-MenuItem";
  menuItem.CssClassOver = "CritMenu-MenuItemOver";
  menuItem.Label = "Fagpakke, retningsbetegnelse m.m.";
  menuItem.ID = "critMenuTechnicalPackage";
  menuItem.Height = "20";

  menuItem = critMenu.TopGroup.Items.Add();
  menuItem.CssClass = "CritMenu-Break";
  menuItem.Label = " ";
  menuItem.Height = "5px";

  menuItem = critMenu.TopGroup.Items.Add();
  menuItem.CssClass = "CritMenu-MenuItem";
  menuItem.CssClassOver = "CritMenu-MenuItemOver";
  menuItem.Label = "Til-/fravalg af kurser";
  menuItem.ID = "critMenuCourses";
  menuItem.Height = "20";

  menuItem = critMenu.TopGroup.Items.Add();
  menuItem.CssClass = "CritMenu-Break";
```

```
    menuItem.Label = " ";
    menuItem.Height = "5px";

    menuItem = critMenu.TopGroup.Items.Add();
    menuItem.CssClass = "CritMenu-MenuItem";
    menuItem.CssClassOver = "CritMenu-MenuItemOver";
    menuItem.Label = "Arbejdsbyrde";
    menuItem.ID = "critMenuWorkload";
    menuItem.Height = "20";

    menuItem = critMenu.TopGroup.Items.Add();
    menuItem.CssClass = "CritMenu-Break";
    menuItem.Label = " ";
    menuItem.Height = "5px";

    menuItem = critMenu.TopGroup.Items.Add();
    menuItem.CssClass = "CritMenu-MenuItem";
    menuItem.CssClassOver = "CritMenu-MenuItemOver";
    menuItem.Label = "Projekter";
    menuItem.ID = "critMenuProjects";
    menuItem.Height = "20";

    menuItem = critMenu.TopGroup.Items.Add();
    menuItem.CssClass = "CritMenu-Break";
    menuItem.Label = " ";
    menuItem.Height = "100%";
}

public void CritMenuItemSelected(object sender, CYBERAKT.WebControls.Navigation.MenuItemSelectedEventArgs objArgs)
{
    switch(objArgs.ItemID)
    {

      case "critMenuGeneral":
      {
        criteria.SelectedIndex = 0;
        break;
      }
      case "critMenuInterests":
      {
        criteria.SelectedIndex = 1;
        break;
      }
      case "critMenuTechnicalPackage":
      {
        criteria.SelectedIndex = 2;
        break;
      }
      default:
        break;
    }
}

protected void DoAction(object objSender, CommandEventArgs objArgs)
{
    switch(objArgs.CommandName)
    {

      case "save":
      {
        criteria.SelectedIndex = 0;
        break;
      }
      case "cancel":
      {
        criteria.SelectedIndex = 1;
        break;
      }
      case "elaborate":
      {
        criteria.SelectedIndex = 2;
        break;
      }
      default:
        break;
    }
}

#region Web Form Designer generated code
override protected void OnInit(EventArgs e)
{
    //
    // CODEGEN: This call is required by the ASP.NET Web Form Designer.
    //
    InitializeComponent();
    base.OnInit(e);
}

/// <summary>
/// Required method for Designer support - do not modify
/// the contents of this method with the code editor.
/// </summary>
private void InitializeComponent()
{
    this.critMenu.MenuItemSelected += new CYBERAKT.WebControls.Navigation.ASPnetMenu.MenuItemSelectedEvent(this.
      CritMenuItemSelected);
    //this.critMenu.MenuItemSelected += new CYBERAKT.WebControls.Navigation.ASPnetMenu.MenuItemSelectedEvent(this.
      critMenu_MenuItemSelected);
    this.Load += new System.EventHandler(this.Page_Load);
```

```
    }

    #endregion
  }
}
```

5.3.9 studyplans.aspx

```
<%@ Page language="c#" codebehind="studyplans.aspx.cs" AutoEventWireup="false" Inherits="StudyPlanning.UI.Planning.StudyPlans" %>
<%@ Register TagPre↓x="UserControl" TagName="Header" Src="../header.ascx" %>
<%@ Register TagPre↓x="UserControl" TagName="Footer" Src="../footer.ascx" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" >
<html>
  <head>
    <title><asp:Literal id="lblPageTitle" Runat="server" /></title>
    <meta name="GENERATOR" content="Microsoft Visual Studio 7.0">
    <meta name="CODE_LANGUAGE" content="C#">
    <meta name="vs_defaultClientScript" content="JavaScript">
    <meta name="vs_targetSchema" content="http://schemas.microsoft.com/intellisense/ie5">
    <link rel="stylesheet" href="../common/style.css" />
    <style>
      .datagrid
      {
        margin-left: 30px;
      }

      .button
      {
        width: 100px;
      }
    </style>
  </head>
  <body>
    <form runat="server" ID="Form1">
      <usercontrol:header id="MyHeader" runat="server" />

      <h2><asp:Label ID="lblPageName" Runat="server" /></h2>

      <asp:DataGrid ID="studyPlans" Runat="server"
        GridLines="Both"
        BorderWidth="1"
        BorderColor="#000000"
        CellPadding="5"
        HeaderStyle-Font-Bold="True"
        HeaderStyle-Font-Size="110%"
        HeaderStyle-BackColor="#d3d3d3"
        CssClass="datagrid"
        AlternatingItemStyle-BackColor="#e0dfe3"
        AutoGenerateColumns="False"
        OnItemCommand="DoAction"
      >
        <Columns>
          <asp:BoundColumn HeaderText="Name" DataField="Name"></asp:BoundColumn>
          <asp:BoundColumn HeaderText="Created" DataField="Created" DataFormatString="{0:g}"></asp:BoundColumn>
          <asp:TemplateColumn HeaderText="Actions" HeaderStyle-HorizontalAlign="Center">
            <ItemTemplate>
              <input type="hidden" value='<%# DataBinder.Eval(Container.DataItem, "ID")%>' id="id" name="id" runat="server"
                />
              <asp:Button ID="view" Runat="server" Text="View" CssClass="button" CommandName="View" />
              <asp:Button ID="delete" Runat="server" Text="Delete" CssClass="button" CommandName="Delete" />
            </ItemTemplate>
          </asp:TemplateColumn>
        </Columns>

      </asp:DataGrid>

      <usercontrol:footer id="MyFooter" runat="server" />
    </form>
  </body>
</html>
```

5.3.10 studyplans.aspx.cs

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using StudyPlanning.Biz;
using System.Text;

namespace StudyPlanning.UI.Planning
{
  /// <summary>
  /// Supports the "Study Plans" page in the "Study Planning" section.
  /// </summary>
  public class StudyPlans : System.Web.UI.Page
```

```
{
  protected System.Web.UI.WebControls.Label lblPageName;
  protected System.Web.UI.WebControls.Literal lblPageTitle;
  protected StudyPlanning.UI.header MyHeader;
  private Hashtable Texts = new Hashtable();
  private string Culture_ID;
  private int TextGroup_ID = 245;

  protected DataGrid studyPlans;

  private void Page_Load(object sender, System.EventArgs e)
  {
    Culture_ID = Request.Cookies["UserData"]["CultureID"];
    try
    {
      Texts = StudyPlanning.Biz.TextItem.GetTextGroup(TextGroup_ID, Culture_ID);
    }
    catch (BizException bizEx)
    {
      Response.Write("An error occured.");
    }

    lblPageName.Text = Texts["1"].ToString();
    lblPageTitle.Text = Texts["2"].ToString();

    #region Breadcrumb
    //The breadcrumb is built in the following
    string Seperator = " <img src=\"" + System.Con↓guration.Con↓gurationSettings.AppSettings["breadcrumbSeperator"] + "\"/> ";
    StringBuilder Breadcrumb = new StringBuilder();
    Breadcrumb.Append("<a href=\"/\">" + Texts["100"].ToString() + "</a>"); //Home
    Breadcrumb.Append(Seperator);
    Breadcrumb.Append("<a href=\"/planning/\">" + Texts["101"].ToString() + "</a>"); //Study Planning
    Breadcrumb.Append(Seperator);
    Breadcrumb.Append("<strong>" + Texts["102"].ToString() + "</strong>"); //Study Plans

    MyHeader.Text = Breadcrumb.ToString();
    //<a href='/'>Home</a> <img src='../common/gfx/arrow−right.gif' /> <a href='../planning/'>Study Planning</a> <img src='../common/
      gfx/arrow−right.gif' /> <strong>Study Plans</strong>
    #endregion

    if (!Page.IsPostBack)
    {
      BindStudyPlans(new Guid("{68C9F6D6-633E-480c-A610-F3DB5A9734DA}"));
    }
  }

  protected void DoAction(object objSender, DataGridCommandEventArgs objArgs)
  {
    Guid studyPlan_ID = new Guid(((System.Web.UI.HtmlControls.HtmlInputHidden)objArgs.Item.Cells[2].FindControl("id")).
      Value);

    if (objArgs.CommandName == "View")
    {
      Response.Redirect("studyplan.aspx?StudyPlan_ID=" + studyPlan_ID);
      //Response.Write(objArgs.Item.Cells[2].FindControl("id"));
      //Response.Write("Vis mig så den elendige studieplan!");
    }
    else if (objArgs.CommandName == "Delete")
    {
      bool result = false;

      try
      {
        result = StudyPlanning.Biz.StudyPlan.Delete(studyPlan_ID);
      }
      catch (BizException objEx)
      {
        Response.Write(objEx.Message);

        if (objEx.Dal != null)
        {
          Response.Write("<br/>" + objEx.Dal.Message + "<br/><br/>");

          if (objEx.Dal.Sql != null)
          {
            Response.Write("<br/>" + objEx.Dal.Sql.Message + "<br/><br/>");
          }
        }
      }

      if (result)
        BindStudyPlans(new Guid("{68C9F6D6-633E-480c-A610-F3DB5A9734DA}"));
    }
  }

  private void BindStudyPlans(Guid student_ID)
  {
    System.Data.DataTable objStudyPlans =
      new System.Data.DataTable();

    try
    {
      objStudyPlans = StudyPlanning.Biz.Student.GetStudyPlans(student_ID);
    }
    catch (BizException objEx)
    {
      Response.Write(objEx.Message + "<br/><br/>");

    }
```

```
      studyPlans.DataSource = objStudyPlans.DefaultView;
      studyPlans.DataBind();

    }
    #region Web Form Designer generated code
    override protected void OnInit(EventArgs e)
    {
      //
      // CODEGEN: This call is required by the ASP.NET Web Form Designer.
      //
      InitializeComponent();
      base.OnInit(e);
    }

    /// <summary>
    /// Required method for Designer support - do not modify
    /// the contents of this method with the code editor.
    /// </summary>
    private void InitializeComponent()
    {
      this.Load += new System.EventHandler(this.Page_Load);
    }
    #endregion
  }
}
```

5.4 Study Info

5.4.1 default.aspx

```
<%@ Page language="c#" codebehind="default.aspx.cs" AutoEventWireup="false" Inherits="StudyPlanning.UI.StudyInfo.Main" %>
<%@ Register TagPre↓x="UserControl" TagName="Header" Src="../header.ascx" %>
<%@ Register TagPre↓x="UserControl" TagName="Footer" Src="../footer.ascx" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" >
<html>
  <head>
    <title><asp:Literal id="lblPageTitle" Runat="server" /></title>
    <meta name="GENERATOR" content="Microsoft Visual Studio 7.0">
    <meta name="CODE_LANGUAGE" content="C#">
    <meta name="vs_defaultClientScript" content="JavaScript">
    <meta name="vs_targetSchema" content="http://schemas.microsoft.com/intellisense/ie5">
    <link rel="stylesheet" href="../common/style.css" />
  </head>
  <body>
    <form runat="server" ID="Form1">
    <usercontrol:header id="MyHeader" runat="server" />

    <h2><asp:Label ID="lblPageName" Runat="server" /></h2>

    <usercontrol:footer id="MyFooter" runat="server" />
    </form>
  </body>
</html>
```

5.4.2 default.aspx.cs

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using StudyPlanning.Biz;
using System.Text;

namespace StudyPlanning.UI.StudyInfo
{
  /// <summary>
  /// Supports the default page in the "Study Info" section.
  /// </summary>
  public class Main : System.Web.UI.Page
  {
    protected System.Web.UI.WebControls.Label lblPageName;
    protected System.Web.UI.WebControls.Literal lblPageTitle;
    protected StudyPlanning.UI.header MyHeader;
    private Hashtable Texts = new Hashtable();
    private string Culture_ID;
    private int TextGroup_ID = 285;

    private void Page_Load(object sender, System.EventArgs e)
    {
      Culture_ID = Request.Cookies["UserData"]["CultureID"];
      try
```

```
      {
        Texts = StudyPlanning.Biz.TextItem.GetTextGroup(TextGroup_ID, Culture_ID);
      }
      catch (BizException bizEx)
      {
        Response.Write("An error occured.");
      }

      lblPageName.Text = Texts["1"].ToString();
      lblPageTitle.Text = Texts["2"].ToString();

      #region Breadcrumb
      //The breadcrumb is built in the following
      string Seperator = " <img src=\"" + System.Con↓guration.Con↓gurationSettings.AppSettings["breadcrumbSeperator"] + "\"/> ";
      StringBuilder Breadcrumb = new StringBuilder();
      Breadcrumb.Append("<a href=\"/\">" + Texts["100"].ToString() + "</a>"); //Home
      Breadcrumb.Append(Seperator);
      Breadcrumb.Append("<strong>" + Texts["101"].ToString() + "</strong>"); //Your Control Panel
      MyHeader.Text = Breadcrumb.ToString();
      //<a href='/'>Forside</a> <img src='../common/gfx/arrow−right.gif' /> <strong>Your Control Panel</strong>
      #endregion
    }

    #region Web Form Designer generated code
    override protected void OnInit(EventArgs e)
    {
      //
      // CODEGEN: This call is required by the ASP.NET Web Form Designer.
      //
      InitializeComponent();
      base.OnInit(e);
    }

    /// <summary>
    /// Required method for Designer support − do not modify
    /// the contents of this method with the code editor.
    /// </summary>
    private void InitializeComponent()
    {
      this.Load += new System.EventHandler(this.Page_Load);
    }
    #endregion
  }
}
```

5.4.3 lines.aspx

```
<%@ Page language="c#" codebehind="lines.aspx.cs" AutoEventWireup="false" Inherits="StudyPlanning.UI.StudyInfo.Lines" %>
<%@ Register TagPre↓x="UserControl" TagName="Header" Src="../header.ascx" %>
<%@ Register TagPre↓x="UserControl" TagName="Footer" Src="../footer.ascx" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" >
<html>
  <head>
    <title><asp:Literal id="lblPageTitle" Runat="server" /></title>
    <meta name="GENERATOR" content="Microsoft Visual Studio 7.0">
    <meta name="CODE_LANGUAGE" content="C#">
    <meta name="vs_defaultClientScript" content="JavaScript">
    <meta name="vs_targetSchema" content="http://schemas.microsoft.com/intellisense/ie5">
    <link rel="stylesheet" href="../common/style.css" />
  </head>
  <body>
    <form runat="server" ID="Form1">
    <usercontrol:header id="MyHeader" runat="server" />

    <h2><asp:Label ID="lblPageName" Runat="server" /></h2>

    <br/>
    <br/>
    <br/>
    <br/>
    <br/>

    <usercontrol:footer id="MyFooter" runat="server" />
    </form>
  </body>
</html>
```

5.4.4 lines.aspx.cs

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using StudyPlanning.Biz;
using System.Text;
```

```
namespace StudyPlanning.UI.StudyInfo
{
  /// <summary>
  /// Summary description for WebForm1.
  /// </summary>
  public class Lines : System.Web.UI.Page
  {
    protected System.Web.UI.WebControls.Label lblPageName;
    protected System.Web.UI.WebControls.Literal lblPageTitle;
    protected StudyPlanning.UI.header MyHeader;
    private Hashtable Texts = new Hashtable();
    private string Culture_ID;
    private int TextGroup_ID = 290;

    private void Page_Load(object sender, System.EventArgs e)
    {
      Culture_ID = Request.Cookies["UserData"]["CultureID"];
      try
      {
        Texts = StudyPlanning.Biz.TextItem.GetTextGroup(TextGroup_ID, Culture_ID);
      }
      catch (BizException bizEx)
      {
        Response.Write("An error occured.");
      }

      lblPageName.Text = Texts["1"].ToString();
      lblPageTitle.Text = Texts["2"].ToString();

      #region Breadcrumb
      //The breadcrumb is built in the following
      string Seperator = " <img src=\"" + System.Con|guration.Con|gurationSettings.AppSettings["breadcrumbSeperator"] + "\"/> ";
      StringBuilder Breadcrumb = new StringBuilder();
      Breadcrumb.Append("<a href=\"/\">" + Texts["100"].ToString() + "</a>"); //Home
      Breadcrumb.Append(Seperator);
      Breadcrumb.Append("<a href=\"/studyinfo/\">" + Texts["101"].ToString() + "</a>"); //Study Info
      Breadcrumb.Append(Seperator);
      Breadcrumb.Append("<strong>" + Texts["102"].ToString() + "</strong>"); //Technical Packages

      MyHeader.Text = Breadcrumb.ToString();
      //<a href='/'>Home</a> <img src='../common/gfx/arrow-right.gif' /> <a href='../studyinfo/'>Study Info</a> <img src='../common/gfx/
        arrow-right.gif' /> <strong>Technical Lines</strong>
      #endregion
    }

    #region Web Form Designer generated code
    override protected void OnInit(EventArgs e)
    {
      //
      // CODEGEN: This call is required by the ASP.NET Web Form Designer.
      //
      InitializeComponent();
      base.OnInit(e);
    }

    /// <summary>
    /// Required method for Designer support - do not modify
    /// the contents of this method with the code editor.
    /// </summary>
    private void InitializeComponent()
    {
      this.Load += new System.EventHandler(this.Page_Load);
    }
    #endregion
  }
}
```

5.4.5 periods.aspx

```
<%@ Page language="c#" codebehind="periods.aspx.cs" AutoEventWireup="false" Inherits="StudyPlanning.UI.StudyInfo.Periods" %>
<%@ Register TagPre|x="UserControl" TagName="Header" Src="../header.ascx" %>
<%@ Register TagPre|x="UserControl" TagName="Footer" Src="../footer.ascx" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" >
<html>
  <head>
    <title><asp:Literal id="lblPageTitle" Runat="server" /></title>
    <meta name="GENERATOR" content="Microsoft Visual Studio 7.0">
    <meta name="CODE_LANGUAGE" content="C#">
    <meta name="vs_defaultClientScript" content="JavaScript">
    <meta name="vs_targetSchema" content="http://schemas.microsoft.com/intellisense/ie5">
    <link rel="stylesheet" href="../common/style.css" />
  </head>
  <body>
    <form runat="server" ID="Form1">
    <usercontrol:header id="MyHeader" runat="server" />

    <h2><asp:Label ID="lblPageName" Runat="server" /></h2>

    <br/>
    <br/>
    <br/>
    <br/>
    <br/>

    <usercontrol:footer id="MyFooter" runat="server" />
```

```
    </form>
  </body>
</html>
```

5.4.6 periods.aspx.cs

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using StudyPlanning.Biz;
using System.Text;

namespace StudyPlanning.UI.StudyInfo
{
  /// <summary>
  /// Supports the page that provides an overview of the different teaching periods.
  /// </summary>
  public class Periods : System.Web.UI.Page
  {
    protected System.Web.UI.WebControls.Label lblPageName;
    protected System.Web.UI.WebControls.Literal lblPageTitle;
    protected StudyPlanning.UI.header MyHeader;
    private Hashtable Texts = new Hashtable();
    private string Culture_ID;
    private int TextGroup_ID = 295;

    private void Page_Load(object sender, System.EventArgs e)
    {
      Culture_ID = Request.Cookies["UserData"]["CultureID"];
      try
      {
        Texts = StudyPlanning.Biz.TextItem.GetTextGroup(TextGroup_ID, Culture_ID);
      }
      catch (BizException bizEx)
      {
        Response.Write("An error occured.");
      }

      lblPageName.Text = Texts["1"].ToString();
      lblPageTitle.Text = Texts["2"].ToString();

      #region Breadcrumb
      //The breadcrumb is built in the following
      string Seperator = " <img src=\"" + System.Con↓guration.Con↓gurationSettings.AppSettings["breadcrumbSeperator"] + "\"/> ";
      StringBuilder Breadcrumb = new StringBuilder();
      Breadcrumb.Append("<a href=\"/\">" + Texts["100"].ToString() + "</a>"); //Home
      Breadcrumb.Append(Seperator);
      Breadcrumb.Append("<a href=\"/studyinfo/\">" + Texts["101"].ToString() + "</a>"); //Study Info
      Breadcrumb.Append(Seperator);
      Breadcrumb.Append("<strong>" + Texts["102"].ToString() + "</strong>"); //Periods

      MyHeader.Text = Breadcrumb.ToString();
      //<a href='/'>Home</a> <img src='../common/gfx/arrow−right.gif' /> <a href='../studyinfo/'>Study Info</a> <img src='../common/gfx/
        arrow−right.gif' /> <strong>Periods</strong>
      #endregion
    }

    #region Web Form Designer generated code
    override protected void OnInit(EventArgs e)
    {
      //
      // CODEGEN: This call is required by the ASP.NET Web Form Designer.
      //
      InitializeComponent();
      base.OnInit(e);
    }

    /// <summary>
    /// Required method for Designer support − do not modify
    /// the contents of this method with the code editor.
    /// </summary>
    private void InitializeComponent()
    {
      this.Load += new System.EventHandler(this.Page_Load);
    }
    #endregion
  }
}
```

5.4.7 technicalpackages.aspx

```
<%@ Page language="c#" codebehind="technicalpackages.aspx.cs" AutoEventWireup="false" Inherits="StudyPlanning.UI.StudyInfo.
  TechnicalPackages" %>
<%@ Register TagPre↓x="UserControl" TagName="Header" Src="../header.ascx" %>
<%@ Register TagPre↓x="UserControl" TagName="Footer" Src="../footer.ascx" %>
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" >
<html>
  <head>
    <title><asp:Literal id="lblPageTitle" Runat="server" /></title>
    <meta name="GENERATOR" content="Microsoft Visual Studio 7.0">
    <meta name="CODE_LANGUAGE" content="C#">
    <meta name="vs_defaultClientScript" content="JavaScript">
    <meta name="vs_targetSchema" content="http://schemas.microsoft.com/intellisense/ie5">
    <link rel="stylesheet" href="../common/style.css" />
  </head>
  <body>
    <form runat="server" ID="Form1">
    <usercontrol:header id="MyHeader" runat="server" />

    <h2><asp:Label ID="lblPageName" Runat="server" /></h2>

    <br/>
    <br/>
    <br/>
    <br/>
    <br/>

    <usercontrol:footer id="MyFooter" runat="server" />
    </form>
  </body>
</html>
```

5.4.8 technicalpackages.aspx.cs

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
using StudyPlanning.Biz;
using System.Text;

namespace StudyPlanning.UI.StudyInfo
{
  /// <summary>
  /// Supports the page presenting the different technical packages.
  /// </summary>
  public class TechnicalPackages : System.Web.UI.Page
  {
    protected System.Web.UI.WebControls.Label lblPageName;
    protected System.Web.UI.WebControls.Literal lblPageTitle;
    protected StudyPlanning.UI.header MyHeader;
    private Hashtable Texts = new Hashtable();
    private string Culture_ID;
    private int TextGroup_ID = 300;

    private void Page_Load(object sender, System.EventArgs e)
    {
      Culture_ID = Request.Cookies["UserData"]["CultureID"];
      try
      {
        Texts = StudyPlanning.Biz.TextItem.GetTextGroup(TextGroup_ID, Culture_ID);
      }
      catch (BizException bizEx)
      {
        Response.Write("An error occured.");
      }

      lblPageName.Text = Texts["1"].ToString();
      lblPageTitle.Text = Texts["2"].ToString();

      #region Breadcrumb
      //The breadcrumb is built in the following
      string Seperator = " <img src=\"" + System.Con↓guration.Con↓gurationSettings.AppSettings["breadcrumbSeperator"] + "\"/> ";
      StringBuilder Breadcrumb = new StringBuilder();
      Breadcrumb.Append("<a href=\"/\">" + Texts["100"].ToString() + "</a>"); //Home
      Breadcrumb.Append(Seperator);
      Breadcrumb.Append("<a href=\"/studyinfo/\">" + Texts["101"].ToString() + "</a>"); //Study Info
      Breadcrumb.Append(Seperator);
      Breadcrumb.Append("<strong>" + Texts["102"].ToString() + "</strong>"); //Technical Packages

      MyHeader.Text = Breadcrumb.ToString();
      //<a href='/'>Home</a> <img src='../common/gfx/arrow−right.gif' /> <a href='../studyinfo/'>Study Info</a> <img src='../common/gfx/
        arrow−right.gif' /> <strong>Technical Packages</strong>
      #endregion
    }

    #region Web Form Designer generated code
    override protected void OnInit(EventArgs e)
    {
      //
      // CODEGEN: This call is required by the ASP.NET Web Form Designer.
      //
      InitializeComponent();
      base.OnInit(e);
```

```
    }

    /// <summary>
    /// Required method for Designer support - do not modify
    /// the contents of this method with the code editor.
    /// </summary>
    private void InitializeComponent()
    {
      this.Load += new System.EventHandler(this.Page_Load);
    }
    #endregion
  }
}
```

A STUDY PLANNING SYSTEM

VOLUME III

DATABASE SOURCE FILES

Morten Milling Jensen
Teddy Kaarløv Nielsen

LYNGBY 2003
EKSAMENSPROJEKT
NR. 70/03

IMM

Trykt af IMM, DTU

Contents

Chapter 1

Database

```
CREATE TABLE [dbo].[Assessment] (
    [Assessment_ID] [uniqueidenti↓er] NOT NULL ,
    [Passed] [bit] NULL ,
    [Grade_ID] [int] NULL ,
    [PointsCredited] [bit] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[AssessmentType] (
    [AssessmentType_ID] [int] NOT NULL ,
    [Name] [text] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] uniqueidenti↓er ROWGUIDCOL NOT NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

CREATE TABLE [dbo].[ContactData] (
    [ContactData_ID] [uniqueidenti↓er] NOT NULL ,
    [ContactDataType_ID] [int] NOT NULL ,
    [Email] [varchar] (100) NULL ,
    [Mobile] [varchar] (30) NULL ,
    [Phone] [varchar] (30) NULL ,
    [Facsimile] [varchar] (30) NULL ,
    [Homepage] [varchar] (100) NULL ,
    [Address1] [varchar] (100) NULL ,
    [Address2] [varchar] (100) NULL ,
    [PostalCode] [varchar] (10) NULL ,
    [City] [varchar] (50) NULL ,
    [Building] [varchar] (10) NULL ,
    [Room] [varchar] (5) NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[ContactDataType] (
    [ContactDataType_ID] [int] NOT NULL ,
    [Name] [text] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

CREATE TABLE [dbo].[ContentConcept] (
    [ContentConcept_ID] [int] NOT NULL ,
    [Name] [text] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [int] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [int] NOT NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

CREATE TABLE [dbo].[Course] (
    [Course_ID] [uniqueidenti↓er] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[CourseGrab] (
    [CourseGrab_ID] [uniqueidenti↓er] NOT NULL ,
    [CourseVersion_ID] [uniqueidenti↓er] NOT NULL ,
    [Course_ID] [uniqueidenti↓er] NOT NULL ,
    [Number] [varchar] (20) NOT NULL ,
    [DtuVersion] [int] NULL ,
    [Schedule] [text] NULL ,
    [Duration] [text] NULL ,
    [ExaminationPlacement] [text] NULL ,
    [ExaminationAids] [text] NULL ,
    [PointBlockingCourses] [text] NULL ,
    [PreviousCourseNumbers] [text] NULL ,
    [MandatoryPrerequisites] [text] NULL ,
    [TechnicalPrerequisites] [text] NULL ,
    [DesirablePrerequisites] [text] NULL ,
    [Responsibles] [text] NULL ,
    [ExternalInstitutions] [text] NULL ,
    [LastUpdated] [datetime] NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

CREATE TABLE [dbo].[CourseVersion] (
    [CourseVersion_ID] [uniqueidenti↓er] NOT NULL ,
```

```
    [Course_ID] [uniqueidenti↓er] NOT NULL ,
    [Version] [int] NOT NULL ,
    [Number] [varchar] (20) NOT NULL ,
    [Name] [text] NOT NULL ,
    [Language_ID] [char] (2) NOT NULL ,
    [TeachingForm] [text] NULL ,
    [Parts] [int] NOT NULL ,
    [AssessmentType_ID] [int] NOT NULL ,
    [EvaluationFormDescription] [text] NULL ,
    [ParticipantLimitationMin] [int] NULL ,
    [ParticipantLimitationMax] [int] NULL ,
    [Objective] [text] NULL ,
    [Contents] [text] NOT NULL ,
    [Remark] [text] NULL ,
    [Url] [varchar] (100) NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

CREATE TABLE [dbo].[Course_Department] (
    [Course_Department_ID] [uniqueidenti↓er] NOT NULL ,
    [Course_ID] [uniqueidenti↓er] NOT NULL ,
    [Department_ID] [int] NOT NULL ,
    [Responsible] [bit] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Course_EvaluationForm] (
    [Course_EvaluationForm_ID] [uniqueidenti↓er] NOT NULL ,
    [CourseVersion_ID] [uniqueidenti↓er] NOT NULL ,
    [EvaluationForm_ID] [int] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Course_Keyword] (
    [Course_Keyword_ID] [uniqueidenti↓er] NOT NULL ,
    [CourseVersion_ID] [uniqueidenti↓er] NOT NULL ,
    [Keyword_ID] [uniqueidenti↓er] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Course_Language] (
    [Course_Language_ID] [int] NOT NULL ,
    [CourseVersion_ID] [int] NOT NULL ,
    [Language_ID] [int] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [int] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [int] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Course_Lecturer] (
    [Course_Lecturer_ID] [uniqueidenti↓er] NOT NULL ,
    [CourseVersion_ID] [uniqueidenti↓er] NOT NULL ,
    [Lecturer_ID] [uniqueidenti↓er] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Course_Period] (
    [Course_Period_ID] [uniqueidenti↓er] NOT NULL ,
    [CourseVersion_ID] [uniqueidenti↓er] NOT NULL ,
    [Period_ID] [uniqueidenti↓er] NOT NULL ,
    [Part] [int] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Course_Period_Module] (
    [Course_Period_Module_ID] [uniqueidenti↓er] NOT NULL ,
    [Course_Period_ID] [uniqueidenti↓er] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO
```

```
CREATE TABLE [dbo].[Course_Period_ModuleItem] (
    [Course_Period_ModuleItem_ID] [uniqueidenti↓er] NOT NULL ,
    [Course_Period_Module_ID] [uniqueidenti↓er] NOT NULL ,
    [Module_ID] [int] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Course_Point] (
    [Course_Point_ID] [uniqueidenti↓er] NOT NULL ,
    [CourseVersion_ID] [uniqueidenti↓er] NOT NULL ,
    [Part] [int] NOT NULL ,
    [Point_ID] [uniqueidenti↓er] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Course_RecommendedPlacement] (
    [Course_RecommendedPlacement_ID] [uniqueidenti↓er] NOT NULL ,
    [CourseVersion_ID] [uniqueidenti↓er] NOT NULL ,
    [StudyType_ID] [int] NOT NULL ,
    [Point_ID] [uniqueidenti↓er] NULL ,
    [RecommendedPlacementConcept_ID] [int] NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Course_RelationCourse] (
    [Course_RelationCourse_ID] [uniqueidenti↓er] NOT NULL ,
    [CourseVersion_ID] [uniqueidenti↓er] NOT NULL ,
    [Course_RelationCourseType_ID] [int] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Course_RelationCourseItem] (
    [Course_RelationCourseItem_ID] [uniqueidenti↓er] NOT NULL ,
    [Course_RelationCourse_ID] [uniqueidenti↓er] NOT NULL ,
    [Course_ID] [uniqueidenti↓er] NULL ,
    [CourseNumber] [varchar] (20) NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Course_RelationCourseType] (
    [Course_RelationCourseType_ID] [int] NOT NULL ,
    [Name] [varchar] (200) NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Course_StudyType] (
    [Course_StudyType_ID] [uniqueidenti↓er] NOT NULL ,
    [CourseVersion_ID] [uniqueidenti↓er] NOT NULL ,
    [StudyType_ID] [int] NULL ,
    [Course_StudyTypeCategory_ID] [int] NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Course_StudyTypeCategory] (
    [Course_StudyTypeCategory_ID] [int] NOT NULL ,
    [StudyType_ID] [int] NOT NULL ,
    [Name] [text] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

CREATE TABLE [dbo].[Department] (
    [Department_ID] [int] NOT NULL ,
    [ContactData_ID] [uniqueidenti↓er] NULL ,
    [Name] [text] NOT NULL ,
    [Number] [int] NOT NULL ,
    [Created] [datetime] NOT NULL ,
```

```
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

CREATE TABLE [dbo].[Dummy] (
    [dummy] [varchar] (10) NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[EvaluationForm] (
    [EvaluationForm_ID] [int] NOT NULL ,
    [Name] [text] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

CREATE TABLE [dbo].[Gender] (
    [Gender_ID] [int] NOT NULL ,
    [Name] [text] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

CREATE TABLE [dbo].[Grade] (
    [Grade_ID] [int] NOT NULL ,
    [Number] [int] NOT NULL ,
    [Name] [varchar] (10) NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Keyword] (
    [Keyword_ID] uniqueidenti↓er ROWGUIDCOL NOT NULL ,
    [Name] [varchar] (100) NOT NULL ,
    [Culture_ID] [varchar] (10) NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Language] (
    [Language_ID] [char] (2) NOT NULL ,
    [Name] [text] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

CREATE TABLE [dbo].[Lecturer] (
    [Lecturer_ID] uniqueidenti↓er ROWGUIDCOL NOT NULL ,
    [CwisNumber] [int] NULL ,
    [Person_ID] [uniqueidenti↓er] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Module] (
    [Module_ID] [int] NOT NULL ,
    [Name] [varchar] (10) NOT NULL ,
    [StartTime] [datetime] NOT NULL ,
    [EndTime] [datetime] NOT NULL ,
    [Weekday_ID] [int] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Period] (
    [Period_ID] [uniqueidenti↓er] NOT NULL ,
    [PeriodType_ID] [int] NOT NULL ,
    [Name] [text] NOT NULL ,
    [StartDate] [datetime] NOT NULL ,
    [EndDate] [datetime] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
```

```
CREATE TABLE [dbo].[PeriodType] (
    [PeriodType_ID] [int] NOT NULL ,
    [Name] [text] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

CREATE TABLE [dbo].[PeriodType_Module] (
    [PeriodType_Module_ID] [int] NOT NULL ,
    [PeriodType_ID] [int] NOT NULL ,
    [Module_ID] [int] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Person] (
    [Person_ID] [uniqueidenti↓er] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[PersonVersion] (
    [PersonVersion_ID] [uniqueidenti↓er] NOT NULL ,
    [Person_ID] [uniqueidenti↓er] NOT NULL ,
    [Version] [int] NOT NULL ,
    [GivenName] [varchar] (200) NOT NULL ,
    [MiddleName] [varchar] (200) NULL ,
    [FamilyName] [varchar] (200) NOT NULL ,
    [CivilRegistrationNumber] [varchar] (20) NULL ,
    [Initials] [varchar] (20) NULL ,
    [Title] [varchar] (100) NULL ,
    [Birthday] [datetime] NULL ,
    [Gender_ID] [int] NOT NULL ,
    [User_ID] [uniqueidenti↓er] NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Person_ContactData] (
    [Person_ContactData_ID] [uniqueidenti↓er] NOT NULL ,
    [PersonVersion_ID] [uniqueidenti↓er] NOT NULL ,
    [ContactData_ID] [uniqueidenti↓er] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Point] (
    [Point_ID] [uniqueidenti↓er] NOT NULL ,
    [PointMin] [real] NULL ,
    [PointMax] [real] NULL ,
    [AmsGiving] [bit] NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Project] (
    [Project_ID] uniqueidenti↓er ROWGUIDCOL NOT NULL ,
    [Number] [varchar] (20) NOT NULL ,
    [Name] [text] NOT NULL ,
    [ProjectType_ID] [int] NOT NULL ,
    [AssessmentType_ID] [int] NOT NULL ,
    [Period_ID] [uniqueidenti↓er] NULL ,
    [Point_ID] [uniqueidenti↓er] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

CREATE TABLE [dbo].[ProjectType] (
    [ProjectType_ID] [int] NOT NULL ,
    [Name] [text] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
```

```
CREATE TABLE [dbo].[RecommendedPlacementConcept] (
    [RecommendedPlacementConcept_ID] [int] NOT NULL ,
    [Name] [text] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

CREATE TABLE [dbo].[RecommendedPlacementConcept_StudyType] (
    [RecommendedPlacementConcept_StudyType_ID] [int] NOT NULL ,
    [RecommendedPlacementConcept_ID] [int] NOT NULL ,
    [StudyType_ID] [int] NOT NULL ,
    [Point_ID] [uniqueidenti↓er] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[SecurityPermission] (
    [SecurityPermission_ID] [int] NOT NULL ,
    [SecurityPermissionCategory_ID] [int] NOT NULL ,
    [Description] [char] (200) NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[SecurityPermissionCategory] (
    [SecurityPermissionCategory_ID] [int] NOT NULL ,
    [Description] [varchar] (200) NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[SecurityRole] (
    [SecurityRole_ID] [int] NOT NULL ,
    [Description] [varchar] (200) NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[SecurityRole_SecurityPermission] (
    [SecurityRole_SecurityPermission_ID] [int] NOT NULL ,
    [SecurityRole_ID] [int] NOT NULL ,
    [SecurityPermission_ID] [int] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Specialization] (
    [Specialization_ID] [int] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[SpecializationVersion] (
    [SpecializationVersion_ID] [int] NOT NULL ,
    [Specialization_ID] [int] NOT NULL ,
    [Version] [int] NOT NULL ,
    [Name] [text] NOT NULL ,
    [Point_ID] [uniqueidenti↓er] NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

CREATE TABLE [dbo].[Specialization_Course] (
    [Specialization_Course_ID] [int] NOT NULL ,
    [SpecializationVersion_ID] [int] NOT NULL ,
    [Course_ID] [uniqueidenti↓er] NULL ,
    [Number] [varchar] (20) NULL ,
    [PointGiving] [bit] NOT NULL ,
    [Supplementary] [bit] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO
```

```
CREATE TABLE [dbo].[Student] (
    [Student_ID] [uniqueidenti↓er] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[StudentVersion] (
    [StudentVersion_ID] [uniqueidenti↓er] NOT NULL ,
    [Student_ID] [uniqueidenti↓er] NOT NULL ,
    [Version] [int] NOT NULL ,
    [StudyNumber] [varchar] (20) NOT NULL ,
    [DateOfSignUp] [datetime] NOT NULL ,
    [Person_ID] [uniqueidenti↓er] NOT NULL ,
    [StudyTypeVersion_ID] [int] NOT NULL ,
    [TechnicalPackageVersion_ID] [int] NULL ,
    [TechnicalLineVersion_ID] [int] NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Student_Course] (
    [Student_Course_ID] [uniqueidenti↓er] NOT NULL ,
    [StudentVersion_ID] [uniqueidenti↓er] NOT NULL ,
    [CourseVersion_ID] [uniqueidenti↓er] NOT NULL ,
    [Part] [int] NOT NULL ,
    [DateOfSignUp] [datetime] NOT NULL ,
    [DateOfCancel] [datetime] NULL ,
    [DateOfExaminationSignUp] [datetime] NULL ,
    [DateOfExaminationCancel] [datetime] NULL ,
    [Assessment_ID] [uniqueidenti↓er] NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Student_Department] (
    [Student_Department_ID] [uniqueidenti↓er] NOT NULL ,
    [StudentVersion_ID] [uniqueidenti↓er] NOT NULL ,
    [Department_ID] [int] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Student_Project] (
    [Student_Project_ID] [uniqueidenti↓er] NOT NULL ,
    [StudentVersion_ID] [uniqueidenti↓er] NOT NULL ,
    [Project_ID] [uniqueidenti↓er] NOT NULL ,
    [Assessment_ID] [uniqueidenti↓er] NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Student_StudyPlan] (
    [Student_StudyPlan_ID] [uniqueidenti↓er] NOT NULL ,
    [Student_ID] [uniqueidenti↓er] NOT NULL ,
    [StudyPlan_ID] [uniqueidenti↓er] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Student_StudyPlanCriterion] (
    [Student_StudyPlanCriterion_ID] [uniqueidenti↓er] NOT NULL ,
    [Student_ID] [uniqueidenti↓er] NOT NULL ,
    [StudyPlanCriterion_ID] [uniqueidenti↓er] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[StudyPlan] (
    [StudyPlan_ID] uniqueidenti↓er ROWGUIDCOL NOT NULL ,
    [Name] [varchar] (200) NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO
```

```
CREATE TABLE [dbo].[StudyPlanCriterion] (
    [StudyPlanCriterion_ID] uniqueidenti↓er ROWGUIDCOL NOT NULL ,
    [Name] [varchar] (200) NOT NULL ,
    [AllowPointBlockCourses] [bit] NOT NULL ,
    [RecommendedPlacementCogent] [bit] NOT NULL ,
    [AllowForTechnicalPrerequisites] [bit] NOT NULL ,
    [TechnicalPackageVersion_ID] [int] NULL ,
    [TechnicalLineVersion_ID] [int] NULL ,
    [TechnicalFieldVersion_ID] [int] NULL ,
    [SpecializationVersion_ID] [int] NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[StudyPlanCriterion_ContentConcept] (
    [StudyPlanCriterion_ContentConcept_ID] uniqueidenti↓er ROWGUIDCOL NOT NULL ,
    [StudyPlanCriterion_ID] [uniqueidenti↓er] NOT NULL ,
    [ContentConcept_ID] [int] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[StudyPlanCriterion_Course] (
    [StudyPlanCriterion_Course_ID] uniqueidenti↓er ROWGUIDCOL NOT NULL ,
    [StudyPlanCriterion_ID] [uniqueidenti↓er] NOT NULL ,
    [Course_ID] [uniqueidenti↓er] NOT NULL ,
    [AdditionalChoice] [bit] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[StudyPlanCriterion_CoursePeriod] (
    [StudyPlanCriterion_CoursePeriod_ID] uniqueidenti↓er ROWGUIDCOL NOT NULL ,
    [StudyPlanCriterion_ID] [uniqueidenti↓er] NOT NULL ,
    [Period_ID] [uniqueidenti↓er] NOT NULL ,
    [Course_ID] [uniqueidenti↓er] NOT NULL ,
    [AdditionalChoice] [bit] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[StudyPlanCriterion_EvaluationForm] (
    [StudyPlanCriterion_EvaluationForm_ID] uniqueidenti↓er ROWGUIDCOL NOT NULL ,
    [StudyPlanCriterion_ID] [uniqueidenti↓er] NOT NULL ,
    [EvaluationForm_ID] [int] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[StudyPlanCriterion_Keyword] (
    [StudyPlanCriterion_Keyword_ID] [uniqueidenti↓er] NOT NULL ,
    [StudyPlanCriterion_ID] [uniqueidenti↓er] NOT NULL ,
    [Keyword_ID] [uniqueidenti↓er] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[StudyPlanCriterion_Language] (
    [StudyPlanCriterion_Language_ID] uniqueidenti↓er ROWGUIDCOL NOT NULL ,
    [StudyPlanCriterion_ID] [uniqueidenti↓er] NOT NULL ,
    [Language_ID] [char] (2) NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[StudyPlanCriterion_Lecturer] (
    [StudyPlanCriterion_Lecturer_ID] [uniqueidenti↓er] NOT NULL ,
    [StudyPlanCriterion_ID] [uniqueidenti↓er] NOT NULL ,
    [Lecturer_ID] [uniqueidenti↓er] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[StudyPlanCriterion_ProjectPeriodPoint] (
    [StudyPlanCriterion_ProjectPeriodPoint_ID] [uniqueidenti↓er] NOT NULL ,
```

```
    [StudyPlanCriterion_ID] [uniqueidenti↓er] NOT NULL ,
    [Project_ID] [uniqueidenti↓er] NOT NULL ,
    [Period_ID] [uniqueidenti↓er] NOT NULL ,
    [Point_ID] [uniqueidenti↓er] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[StudyPlanCriterion_StudyType] (
    [StudyPlanCriterion_StudyType_ID] [uniqueidenti↓er] NOT NULL ,
    [StudyPlanCriterion_ID] [uniqueidenti↓er] NOT NULL ,
    [StudyType_ID] [int] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[StudyPlanCriterion_TechnicalPrerequisiteCourse] (
    [StudyPlanCriterion_TechnicalPrerequisiteCourse_ID] [uniqueidenti↓er] NOT NULL ,
    [StudyPlanCriterion_ID] [uniqueidenti↓er] NOT NULL ,
    [Course_ID] [uniqueidenti↓er] NOT NULL ,
    [AdditionalChoice] [bit] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[StudyPlanCriterion_WorkloadPeriod] (
    [StudyPlanCriterion_WorkloadPeriod_ID] [uniqueidenti↓er] NOT NULL ,
    [StudyPlanCriterion_ID] [uniqueidenti↓er] NOT NULL ,
    [Period_ID] [uniqueidenti↓er] NOT NULL ,
    [Point_ID] [uniqueidenti↓er] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[StudyPlanCriterion_WorkloadPeriodType] (
    [StudyPlanCriterion_WorkloadPeriodType_ID] [uniqueidenti↓er] NOT NULL ,
    [StudyPlanCriterion_ID] [uniqueidenti↓er] NOT NULL ,
    [PeriodType_ID] [int] NOT NULL ,
    [Point_ID] [uniqueidenti↓er] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[StudyPlanCriterion_WorkloadPeriodType_Module] (
    [StudyPlanCriterion_WorkloadPeriodType_Module_ID] [uniqueidenti↓er] NOT NULL ,
    [StudyPlanCriterion_WorkloadPeriodType_ID] [uniqueidenti↓er] NOT NULL ,
    [Module_ID] [int] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[StudyPlanCriterion_WorkloadPeriod_Module] (
    [StudyPlanCriterion_WorkloadPeriod_Module_ID] [uniqueidenti↓er] NOT NULL ,
    [StudyPlanCriterion_WorkloadPeriod_ID] [uniqueidenti↓er] NOT NULL ,
    [Module_ID] [int] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[StudyPlan_Period] (
    [StudyPlan_Period_ID] uniqueidenti↓er ROWGUIDCOL NOT NULL ,
    [StudyPlan_ID] [uniqueidenti↓er] NOT NULL ,
    [Period_ID] [uniqueidenti↓er] NOT NULL ,
    [Project_ID] [uniqueidenti↓er] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[StudyPlan_PeriodCourse] (
    [StudyPlan_PeriodCourse_ID] uniqueidenti↓er ROWGUIDCOL NOT NULL ,
    [StudyPlan_Period_ID] [uniqueidenti↓er] NOT NULL ,
    [CourseVersion_ID] [uniqueidenti↓er] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
```

```
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[StudyType] (
    [StudyType_ID] [int] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[StudyTypeVersion] (
    [StudyTypeVersion_ID] [int] NOT NULL ,
    [StudyType_ID] [int] NOT NULL ,
    [Version] [int] NOT NULL ,
    [Name] [text] NOT NULL ,
    [Point_ID] [uniqueidenti↓er] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

CREATE TABLE [dbo].[StudyType_ProjectType] (
    [StudyType_ProjectType_ID] [int] NOT NULL ,
    [StudyTypeVersion_ID] [int] NOT NULL ,
    [ProjectType_ID] [int] NOT NULL ,
    [SequenceNumber] [int] NOT NULL ,
    [Point_ID] [uniqueidenti↓er] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[StudyType_TechnicalLine] (
    [StudyType_TechnicalLine_ID] [int] NOT NULL ,
    [StudyTypeVersion_ID] [int] NOT NULL ,
    [TechnicalLine_ID] [int] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[StudyType_TechnicalPackage] (
    [StudyType_TechnicalPackage_ID] [int] NOT NULL ,
    [StudyTypeVersion_ID] [int] NOT NULL ,
    [TechnicalPackage_ID] [int] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Table] (
    [Table_ID] [varchar] (250) NOT NULL ,
    [Responsible_ID] [char] (3) NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Table_StoredProcedure] (
    [StoredProcedure_ID] [varchar] (250) NOT NULL ,
    [Table_ID] [varchar] (250) NOT NULL ,
    [Responsible_ID] [char] (3) NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[TechnicalField] (
    [TechnicalField_ID] [int] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[TechnicalFieldVersion] (
    [TechnicalFieldVersion_ID] [int] NOT NULL ,
    [TechnicalField_ID] [int] NOT NULL ,
    [Version] [int] NOT NULL ,
    [Name] [text] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

CREATE TABLE [dbo].[TechnicalField_Course] (
    [TechnicalField_Course_ID] [int] NOT NULL ,
    [TechnicalFieldVersion_ID] [int] NOT NULL ,
    [Course_ID] [uniqueidenti↓er] NULL ,
```

```
    [Number] [varchar] (20) NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[TechnicalLine] (
    [TechnicalLine_ID] [int] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[TechnicalLineVersion] (
    [TechnicalLineVersion_ID] [int] NOT NULL ,
    [TechnicalLine_ID] [int] NOT NULL ,
    [Version] [int] NOT NULL ,
    [Name] [text] NOT NULL ,
    [Point_ID] [uniqueidenti↓er] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

CREATE TABLE [dbo].[TechnicalLine_PrerequisiteCourse] (
    [TechnicalLine_PrerequisiteCourse_ID] [int] NOT NULL ,
    [TechnicalLineVersion_ID] [int] NOT NULL ,
    [Course_ID] [uniqueidenti↓er] NULL ,
    [Number] [varchar] (20) NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[TechnicalLine_PrerequisiteTechnicalPackage] (
    [TechnicalLine_PrerequisiteTechnicalPackage_ID] [int] NOT NULL ,
    [TechnicalLineVersion_ID] [int] NOT NULL ,
    [TechnicalPackage_ID] [int] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[TechnicalLine_PrerequisiteTechnicalPackageCourse] (
    [TechnicalLine_PrerequisiteTechnicalPackageCourse_ID] [int] NOT NULL ,
    [TechnicalLine_PrerequisiteTechnicalPackage_ID] [int] NOT NULL ,
    [Course_ID] [uniqueidenti↓er] NULL ,
    [Number] [varchar] (20) NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[TechnicalLine_Specialization] (
    [TechnicalLine_Specialization_ID] [int] NOT NULL ,
    [TechnicalLineVersion_ID] [int] NOT NULL ,
    [Specialization_ID] [int] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[TechnicalLine_TechnicalField] (
    [TechnicalLine_TechnicalField_ID] [int] NOT NULL ,
    [TechnicalLineVersion_ID] [int] NOT NULL ,
    [TechnicalField_ID] [int] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[TechnicalPackage] (
    [TechnicalPackage_ID] [int] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[TechnicalPackageVersion] (
    [TechnicalPackageVersion_ID] [int] NOT NULL ,
    [TechnicalPackage_ID] [int] NOT NULL ,
```

```
    [Version] [int] NOT NULL ,
    [Number] [varchar] (20) NOT NULL ,
    [Name] [text] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

CREATE TABLE [dbo].[TechnicalPackage_FundamentalCourse] (
    [TechnicalPackage_FundamentalCourse_ID] [int] NOT NULL ,
    [TechnicalPackageVersion_ID] [int] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[TechnicalPackage_FundamentalCourseItem] (
    [TechnicalPackage_FundamentalCourseItem_ID] [int] NOT NULL ,
    [TechnicalPackage_FundamentalCourse_ID] [int] NOT NULL ,
    [Course_ID] [uniqueidenti↓er] NULL ,
    [Number] [varchar] (20) NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[TechnicalPackage_Period] (
    [TechnicalPackage_Period_ID] [int] NOT NULL ,
    [TechnicalPackageVersion_ID] [int] NOT NULL ,
    [Period_ID] [uniqueidenti↓er] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[TechnicalPackage_PeriodCourse] (
    [TechnicalPackage_PeriodCourse_ID] [int] NOT NULL ,
    [TechnicalPackage_Period_ID] [int] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[TechnicalPackage_PeriodCourseItem] (
    [TechnicalPackage_PeriodCourseItem_ID] [int] NOT NULL ,
    [TechnicalPackage_PeriodCourse_ID] [int] NOT NULL ,
    [Course_ID] [uniqueidenti↓er] NULL ,
    [Part] [int] NOT NULL ,
    [Number] [varchar] (20) NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[TechnicalPackage_PeriodOptionalCourse] (
    [TechnicalPackage_PeriodOptionalCourse_ID] [int] NOT NULL ,
    [TechnicalPackage_Period_ID] [int] NOT NULL ,
    [Course_ID] [uniqueidenti↓er] NULL ,
    [Part] [int] NOT NULL ,
    [Number] [varchar] (20) NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[TechnicalPackage_Project] (
    [TechnicalPackage_Project_ID] [int] NOT NULL ,
    [TechnicalPackageVersion_ID] [int] NOT NULL ,
    [Project_ID] [uniqueidenti↓er] NOT NULL ,
    [Period_ID] [uniqueidenti↓er] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Text] (
    [Text_ID] [int] NOT NULL ,
    [TextGroup_ID] [int] NOT NULL ,
    [NumberInGroup] [varchar] (25) NOT NULL ,
    [Culture_ID] [varchar] (10) NOT NULL ,
    [Text] [text] NULL ,
    [Description] [varchar] (200) NULL ,
    [Created] [datetime] NOT NULL ,
```

```
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

CREATE TABLE [dbo].[TextGroup] (
    [TextGroup_ID] [int] NOT NULL ,
    [Name] [varchar] (200) NOT NULL ,
    [Description] [varchar] (600) NULL ,
    [ParentGroup_ID] [int] NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[User] (
    [User_ID] uniqueidenti↓er ROWGUIDCOL NOT NULL ,
    [Username] [varchar] (50) NOT NULL ,
    [Password] [varchar] (50) NOT NULL ,
    [ChangePasswordAtnextLogon] [bit] NOT NULL ,
    [UserCannotChangePassword] [bit] NOT NULL ,
    [PasswordNeverExpires] [bit] NOT NULL ,
    [PasswordLastChanged] [datetime] NOT NULL ,
    [Locked] [bit] NOT NULL ,
    [LockedAt] [datetime] NULL ,
    [Disabled] [bit] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[User_Login] (
    [User_Login_ID] [uniqueidenti↓er] NOT NULL ,
    [User_ID] [uniqueidenti↓er] NOT NULL ,
    [User_LoginType_ID] [int] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[User_LoginType] (
    [User_LoginType_ID] [int] NOT NULL ,
    [Name] [varchar] (100) NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[User_PasswordHistory] (
    [User_PasswordHistory_ID] [uniqueidenti↓er] NOT NULL ,
    [User_ID] [uniqueidenti↓er] NOT NULL ,
    [Password] [varchar] (50) NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[User_SecurityRole] (
    [User_SecurityRole_ID] [uniqueidenti↓er] NOT NULL ,
    [User_ID] [uniqueidenti↓er] NOT NULL ,
    [SecurityRole_ID] [int] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Weekday] (
    [Weekday_ID] [int] NOT NULL ,
    [Name] [text] NOT NULL ,
    [Created] [datetime] NOT NULL ,
    [CreatedBy] [uniqueidenti↓er] NOT NULL ,
    [Updated] [datetime] NOT NULL ,
    [UpdatedBy] [uniqueidenti↓er] NOT NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

ALTER TABLE [dbo].[Assessment] WITH NOCHECK ADD
    CONSTRAINT [PK_Assessment_1] PRIMARY KEY CLUSTERED
    (
        [Assessment_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[AssessmentType] WITH NOCHECK ADD
    CONSTRAINT [PK_Assessment] PRIMARY KEY CLUSTERED
    (
```

```
        [AssessmentType_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[ContactData] WITH NOCHECK ADD
    CONSTRAINT [PK_ContactData] PRIMARY KEY CLUSTERED
    (
        [ContactData_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[ContactDataType] WITH NOCHECK ADD
    CONSTRAINT [PK_ContactDataType] PRIMARY KEY CLUSTERED
    (
        [ContactDataType_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[ContentConcept] WITH NOCHECK ADD
    CONSTRAINT [PK_ContentConcept] PRIMARY KEY CLUSTERED
    (
        [ContentConcept_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[Course] WITH NOCHECK ADD
    CONSTRAINT [PK_Course] PRIMARY KEY CLUSTERED
    (
        [Course_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[CourseGrab] WITH NOCHECK ADD
    CONSTRAINT [PK_CourseGrab] PRIMARY KEY CLUSTERED
    (
        [CourseGrab_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[CourseVersion] WITH NOCHECK ADD
    CONSTRAINT [PK_CourseVersion] PRIMARY KEY CLUSTERED
    (
        [CourseVersion_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[Course_Department] WITH NOCHECK ADD
    CONSTRAINT [PK_Course_Department] PRIMARY KEY CLUSTERED
    (
        [Course_Department_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[Course_EvaluationForm] WITH NOCHECK ADD
    CONSTRAINT [PK_Course_EvaluationForm] PRIMARY KEY CLUSTERED
    (
        [Course_EvaluationForm_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[Course_Keyword] WITH NOCHECK ADD
    CONSTRAINT [PK_Course_Keyword] PRIMARY KEY CLUSTERED
    (
        [Course_Keyword_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[Course_Language] WITH NOCHECK ADD
    CONSTRAINT [PK_Course_Language] PRIMARY KEY CLUSTERED
    (
        [Course_Language_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[Course_Lecturer] WITH NOCHECK ADD
    CONSTRAINT [PK_Course_Lecturer] PRIMARY KEY CLUSTERED
    (
        [Course_Lecturer_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[Course_Period] WITH NOCHECK ADD
    CONSTRAINT [PK_Course_Period] PRIMARY KEY CLUSTERED
    (
        [Course_Period_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[Course_Period_Module] WITH NOCHECK ADD
    CONSTRAINT [PK_Course_Module] PRIMARY KEY CLUSTERED
    (
        [Course_Period_Module_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[Course_Period_ModuleItem] WITH NOCHECK ADD
    CONSTRAINT [PK_Course_Period_ModuleItem] PRIMARY KEY CLUSTERED
    (
        [Course_Period_ModuleItem_ID]
```

```
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[Course_Point] WITH NOCHECK ADD
    CONSTRAINT [PK_Course_Point] PRIMARY KEY CLUSTERED
    (
        [Course_Point_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[Course_RecommendedPlacement] WITH NOCHECK ADD
    CONSTRAINT [PK_Course_RecommendedPlacementSpeci↓c] PRIMARY KEY CLUSTERED
    (
        [Course_RecommendedPlacement_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[Course_RelationCourse] WITH NOCHECK ADD
    CONSTRAINT [PK_Course_RelationCourse] PRIMARY KEY CLUSTERED
    (
        [Course_RelationCourse_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[Course_RelationCourseItem] WITH NOCHECK ADD
    CONSTRAINT [PK_Course_RelationCourseItem] PRIMARY KEY CLUSTERED
    (
        [Course_RelationCourseItem_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[Course_RelationCourseType] WITH NOCHECK ADD
    CONSTRAINT [PK_Course_RelationCourseType] PRIMARY KEY CLUSTERED
    (
        [Course_RelationCourseType_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[Course_StudyType] WITH NOCHECK ADD
    CONSTRAINT [PK_Course_StudyType] PRIMARY KEY CLUSTERED
    (
        [Course_StudyType_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[Course_StudyTypeCategory] WITH NOCHECK ADD
    CONSTRAINT [PK_Course_StudyTypeCategory] PRIMARY KEY CLUSTERED
    (
        [Course_StudyTypeCategory_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[Department] WITH NOCHECK ADD
    CONSTRAINT [PK_Department] PRIMARY KEY CLUSTERED
    (
        [Department_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[EvaluationForm] WITH NOCHECK ADD
    CONSTRAINT [PK_EvaluationForm] PRIMARY KEY CLUSTERED
    (
        [EvaluationForm_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[Gender] WITH NOCHECK ADD
    CONSTRAINT [PK_Gender] PRIMARY KEY CLUSTERED
    (
        [Gender_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[Grade] WITH NOCHECK ADD
    CONSTRAINT [PK_Grade] PRIMARY KEY CLUSTERED
    (
        [Grade_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[Keyword] WITH NOCHECK ADD
    CONSTRAINT [PK_Keyword] PRIMARY KEY CLUSTERED
    (
        [Keyword_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[Language] WITH NOCHECK ADD
    CONSTRAINT [PK_Language] PRIMARY KEY CLUSTERED
    (
        [Language_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[Lecturer] WITH NOCHECK ADD
    CONSTRAINT [PK_Lecturer] PRIMARY KEY CLUSTERED
    (
        [Lecturer_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
```

```
GO

ALTER TABLE [dbo].[Module] WITH NOCHECK ADD
    CONSTRAINT [PK_Module] PRIMARY KEY CLUSTERED
    (
        [Module_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[Period] WITH NOCHECK ADD
    CONSTRAINT [PK_Periode] PRIMARY KEY CLUSTERED
    (
        [Period_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[PeriodType] WITH NOCHECK ADD
    CONSTRAINT [PK_PeriodType] PRIMARY KEY CLUSTERED
    (
        [PeriodType_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[PeriodType_Module] WITH NOCHECK ADD
    CONSTRAINT [PK_PeriodType_Module] PRIMARY KEY CLUSTERED
    (
        [PeriodType_Module_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[Person] WITH NOCHECK ADD
    CONSTRAINT [PK_Person] PRIMARY KEY CLUSTERED
    (
        [Person_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[PersonVersion] WITH NOCHECK ADD
    CONSTRAINT [PK_PersonVersion] PRIMARY KEY CLUSTERED
    (
        [PersonVersion_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[Person_ContactData] WITH NOCHECK ADD
    CONSTRAINT [PK_Person_ContactData] PRIMARY KEY CLUSTERED
    (
        [Person_ContactData_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[Point] WITH NOCHECK ADD
    CONSTRAINT [PK_Point] PRIMARY KEY CLUSTERED
    (
        [Point_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[Project] WITH NOCHECK ADD
    CONSTRAINT [PK_Project] PRIMARY KEY CLUSTERED
    (
        [Project_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[ProjectType] WITH NOCHECK ADD
    CONSTRAINT [PK_ProjectType] PRIMARY KEY CLUSTERED
    (
        [ProjectType_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[RecommendedPlacementConcept] WITH NOCHECK ADD
    CONSTRAINT [PK_RecommendedPlacementConcept] PRIMARY KEY CLUSTERED
    (
        [RecommendedPlacementConcept_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[RecommendedPlacementConcept_StudyType] WITH NOCHECK ADD
    CONSTRAINT [PK_RecommendedPlacementConcept_StudyType] PRIMARY KEY CLUSTERED
    (
        [RecommendedPlacementConcept_StudyType_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[SecurityPermission] WITH NOCHECK ADD
    CONSTRAINT [PK_SecurityPermission] PRIMARY KEY CLUSTERED
    (
        [SecurityPermission_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[SecurityPermissionCategory] WITH NOCHECK ADD
    CONSTRAINT [PK_SecurityPermissionCategory_ID] PRIMARY KEY CLUSTERED
    (
        [SecurityPermissionCategory_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO
```

```
ALTER TABLE [dbo].[SecurityRole] WITH NOCHECK ADD
    CONSTRAINT [PK_SecurityRole] PRIMARY KEY CLUSTERED
    (
        [SecurityRole_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[SecurityRole_SecurityPermission] WITH NOCHECK ADD
    CONSTRAINT [PK_SecurityRole_SecurityPermission] PRIMARY KEY CLUSTERED
    (
        [SecurityRole_SecurityPermission_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[Specialization] WITH NOCHECK ADD
    CONSTRAINT [PK_Specialization] PRIMARY KEY CLUSTERED
    (
        [Specialization_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[SpecializationVersion] WITH NOCHECK ADD
    CONSTRAINT [PK_SpecializationVersion] PRIMARY KEY CLUSTERED
    (
        [SpecializationVersion_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[Specialization_Course] WITH NOCHECK ADD
    CONSTRAINT [PK_Specialization_Course] PRIMARY KEY CLUSTERED
    (
        [Specialization_Course_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[Student] WITH NOCHECK ADD
    CONSTRAINT [PK_Student] PRIMARY KEY CLUSTERED
    (
        [Student_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[StudentVersion] WITH NOCHECK ADD
    CONSTRAINT [PK_StudentVersion] PRIMARY KEY CLUSTERED
    (
        [StudentVersion_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[Student_Course] WITH NOCHECK ADD
    CONSTRAINT [PK_Student_Course] PRIMARY KEY CLUSTERED
    (
        [Student_Course_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[Student_Department] WITH NOCHECK ADD
    CONSTRAINT [PK_Student_Department] PRIMARY KEY CLUSTERED
    (
        [Student_Department_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[Student_Project] WITH NOCHECK ADD
    CONSTRAINT [PK_Student_Project] PRIMARY KEY CLUSTERED
    (
        [Student_Project_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[Student_StudyPlan] WITH NOCHECK ADD
    CONSTRAINT [PK_Student_StudyPlan] PRIMARY KEY CLUSTERED
    (
        [Student_StudyPlan_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[Student_StudyPlanCriterion] WITH NOCHECK ADD
    CONSTRAINT [PK_Student_StudyPlanCriterion] PRIMARY KEY CLUSTERED
    (
        [Student_StudyPlanCriterion_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[StudyPlan] WITH NOCHECK ADD
    CONSTRAINT [PK_StudyPlan] PRIMARY KEY CLUSTERED
    (
        [StudyPlan_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[StudyPlanCriterion] WITH NOCHECK ADD
    CONSTRAINT [PK_StudyPlanCriterion] PRIMARY KEY CLUSTERED
    (
        [StudyPlanCriterion_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO
```

```
ALTER TABLE [dbo].[StudyPlanCriterion_ContentConcept] WITH NOCHECK ADD
    CONSTRAINT [PK_StudyPlanCriterion_ContentConcept] PRIMARY KEY CLUSTERED
    (
        [StudyPlanCriterion_ContentConcept_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[StudyPlanCriterion_Course] WITH NOCHECK ADD
    CONSTRAINT [PK_StudyPlanCriterion_Course] PRIMARY KEY CLUSTERED
    (
        [StudyPlanCriterion_Course_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[StudyPlanCriterion_CoursePeriod] WITH NOCHECK ADD
    CONSTRAINT [PK_StudyPlanCriterion_CoursePeriod] PRIMARY KEY CLUSTERED
    (
        [StudyPlanCriterion_CoursePeriod_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[StudyPlanCriterion_EvaluationForm] WITH NOCHECK ADD
    CONSTRAINT [PK_StudyPlanCriterion_EvaluationForm] PRIMARY KEY CLUSTERED
    (
        [StudyPlanCriterion_EvaluationForm_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[StudyPlanCriterion_Keyword] WITH NOCHECK ADD
    CONSTRAINT [PK_StudyPlanCriterion_Keyword] PRIMARY KEY CLUSTERED
    (
        [StudyPlanCriterion_Keyword_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[StudyPlanCriterion_Language] WITH NOCHECK ADD
    CONSTRAINT [PK_StudyPlanCriterion_Language] PRIMARY KEY CLUSTERED
    (
        [StudyPlanCriterion_Language_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[StudyPlanCriterion_Lecturer] WITH NOCHECK ADD
    CONSTRAINT [PK_StudyPlanCriterion_Lecturer] PRIMARY KEY CLUSTERED
    (
        [StudyPlanCriterion_Lecturer_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[StudyPlanCriterion_ProjectPeriodPoint] WITH NOCHECK ADD
    CONSTRAINT [PK_StudyPlanCriterion_ProjectPeriodPoint] PRIMARY KEY CLUSTERED
    (
        [StudyPlanCriterion_ProjectPeriodPoint_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[StudyPlanCriterion_StudyType] WITH NOCHECK ADD
    CONSTRAINT [PK_StudyPlanCriterion_StudyType] PRIMARY KEY CLUSTERED
    (
        [StudyPlanCriterion_StudyType_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[StudyPlanCriterion_TechnicalPrerequisiteCourse] WITH NOCHECK ADD
    CONSTRAINT [PK_StudyPlanCriterion_TechnicalPrerequisiteCourse] PRIMARY KEY CLUSTERED
    (
        [StudyPlanCriterion_TechnicalPrerequisiteCourse_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[StudyPlanCriterion_WorkloadPeriod] WITH NOCHECK ADD
    CONSTRAINT [PK_StudyPlanCriterion_WorkloadPeriod] PRIMARY KEY CLUSTERED
    (
        [StudyPlanCriterion_WorkloadPeriod_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[StudyPlanCriterion_WorkloadPeriodType] WITH NOCHECK ADD
    CONSTRAINT [PK_StudyPlanCriterion_WorkloadPeriodType] PRIMARY KEY CLUSTERED
    (
        [StudyPlanCriterion_WorkloadPeriodType_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[StudyPlanCriterion_WorkloadPeriodType_Module] WITH NOCHECK ADD
    CONSTRAINT [PK_StudyPlanCriterion_WorkloadPeriodType_Module] PRIMARY KEY CLUSTERED
    (
        [StudyPlanCriterion_WorkloadPeriodType_Module_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[StudyPlanCriterion_WorkloadPeriod_Module] WITH NOCHECK ADD
    CONSTRAINT [PK_StudyPlanCriterion_WorkloadPeriod_Module] PRIMARY KEY CLUSTERED
    (
        [StudyPlanCriterion_WorkloadPeriod_Module_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[StudyPlan_Period] WITH NOCHECK ADD
```

```
    CONSTRAINT [PK_StudyPlan_Period] PRIMARY KEY CLUSTERED
    (
        [StudyPlan_Period_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[StudyPlan_PeriodCourse] WITH NOCHECK ADD
    CONSTRAINT [PK_StudyPlan_PeriodCourse] PRIMARY KEY CLUSTERED
    (
        [StudyPlan_PeriodCourse_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[StudyType] WITH NOCHECK ADD
    CONSTRAINT [PK_StudyType] PRIMARY KEY CLUSTERED
    (
        [StudyType_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[StudyTypeVersion] WITH NOCHECK ADD
    CONSTRAINT [PK_StudyTypeVersion] PRIMARY KEY CLUSTERED
    (
        [StudyTypeVersion_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[StudyType_ProjectType] WITH NOCHECK ADD
    CONSTRAINT [PK_StudyType_ProjectType] PRIMARY KEY CLUSTERED
    (
        [StudyType_ProjectType_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[StudyType_TechnicalLine] WITH NOCHECK ADD
    CONSTRAINT [PK_StudyType_TechnicalLine] PRIMARY KEY CLUSTERED
    (
        [StudyType_TechnicalLine_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[StudyType_TechnicalPackage] WITH NOCHECK ADD
    CONSTRAINT [PK_StudyType_TechnicalPackage] PRIMARY KEY CLUSTERED
    (
        [StudyType_TechnicalPackage_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[Table] WITH NOCHECK ADD
    CONSTRAINT [PK_Tables] PRIMARY KEY CLUSTERED
    (
        [Table_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[Table_StoredProcedure] WITH NOCHECK ADD
    CONSTRAINT [PK_Table_StoredProcedure] PRIMARY KEY CLUSTERED
    (
        [StoredProcedure_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[TechnicalField] WITH NOCHECK ADD
    CONSTRAINT [PK_TechnicalField] PRIMARY KEY CLUSTERED
    (
        [TechnicalField_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[TechnicalFieldVersion] WITH NOCHECK ADD
    CONSTRAINT [PK_TechnicalFieldVersion] PRIMARY KEY CLUSTERED
    (
        [TechnicalFieldVersion_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[TechnicalField_Course] WITH NOCHECK ADD
    CONSTRAINT [PK_TechnicalField_Course] PRIMARY KEY CLUSTERED
    (
        [TechnicalField_Course_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[TechnicalLine] WITH NOCHECK ADD
    CONSTRAINT [PK_TechnicalLine] PRIMARY KEY CLUSTERED
    (
        [TechnicalLine_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[TechnicalLineVersion] WITH NOCHECK ADD
    CONSTRAINT [PK_TechnicalLineVersion] PRIMARY KEY CLUSTERED
    (
        [TechnicalLineVersion_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[TechnicalLine_PrerequisiteCourse] WITH NOCHECK ADD
    CONSTRAINT [PK_TechnicalLine_CoursePrerequisite] PRIMARY KEY CLUSTERED
```

```
    (
        [TechnicalLine_PrerequisiteCourse_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[TechnicalLine_PrerequisiteTechnicalPackage] WITH NOCHECK ADD
    CONSTRAINT [PK_TechnicalLine_TechnicalPackagePrerequisite] PRIMARY KEY CLUSTERED
    (
        [TechnicalLine_PrerequisiteTechnicalPackage_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[TechnicalLine_PrerequisiteTechnicalPackageCourse] WITH NOCHECK ADD
    CONSTRAINT [PK_TechnicalLine_PrerequisiteTechnicalPackageCourse] PRIMARY KEY CLUSTERED
    (
        [TechnicalLine_PrerequisiteTechnicalPackageCourse_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[TechnicalLine_Specialization] WITH NOCHECK ADD
    CONSTRAINT [PK_TechnicalLine_Specialization] PRIMARY KEY CLUSTERED
    (
        [TechnicalLine_Specialization_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[TechnicalLine_TechnicalField] WITH NOCHECK ADD
    CONSTRAINT [PK_TechnicalLine_TechnicalField] PRIMARY KEY CLUSTERED
    (
        [TechnicalLine_TechnicalField_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[TechnicalPackage] WITH NOCHECK ADD
    CONSTRAINT [PK_TechnicalPackage] PRIMARY KEY CLUSTERED
    (
        [TechnicalPackage_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[TechnicalPackageVersion] WITH NOCHECK ADD
    CONSTRAINT [PK_TechnicalPackageVersion] PRIMARY KEY CLUSTERED
    (
        [TechnicalPackageVersion_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[TechnicalPackage_FundamentalCourse] WITH NOCHECK ADD
    CONSTRAINT [PK_TechnicalPackage_FundamentalCourseItem] PRIMARY KEY CLUSTERED
    (
        [TechnicalPackage_FundamentalCourse_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[TechnicalPackage_FundamentalCourseItem] WITH NOCHECK ADD
    CONSTRAINT [PK_TechnicalPackage_FundamentalCourse] PRIMARY KEY CLUSTERED
    (
        [TechnicalPackage_FundamentalCourseItem_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[TechnicalPackage_Period] WITH NOCHECK ADD
    CONSTRAINT [PK_TechnicalPackage_Period] PRIMARY KEY CLUSTERED
    (
        [TechnicalPackage_Period_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[TechnicalPackage_PeriodCourse] WITH NOCHECK ADD
    CONSTRAINT [PK_TechnicalPackage_PeriodCourse] PRIMARY KEY CLUSTERED
    (
        [TechnicalPackage_PeriodCourse_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[TechnicalPackage_PeriodCourseItem] WITH NOCHECK ADD
    CONSTRAINT [PK_TechnicalPackage_PeriodCourseItem] PRIMARY KEY CLUSTERED
    (
        [TechnicalPackage_PeriodCourseItem_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[TechnicalPackage_PeriodOptionalCourse] WITH NOCHECK ADD
    CONSTRAINT [PK_TechnicalPackage_PeriodOptionalCourse] PRIMARY KEY CLUSTERED
    (
        [TechnicalPackage_PeriodOptionalCourse_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[TechnicalPackage_Project] WITH NOCHECK ADD
    CONSTRAINT [PK_TechnicalPackage_Project] PRIMARY KEY CLUSTERED
    (
        [TechnicalPackage_Project_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[Text] WITH NOCHECK ADD
    CONSTRAINT [PK_Text] PRIMARY KEY CLUSTERED
    (
```

```
        [Text_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[TextGroup] WITH NOCHECK ADD
    CONSTRAINT [PK_TextGroup] PRIMARY KEY CLUSTERED
    (
        [TextGroup_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[User] WITH NOCHECK ADD
    CONSTRAINT [PK_User] PRIMARY KEY CLUSTERED
    (
        [User_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[User_Login] WITH NOCHECK ADD
    CONSTRAINT [PK_User_Login] PRIMARY KEY CLUSTERED
    (
        [User_Login_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[User_LoginType] WITH NOCHECK ADD
    CONSTRAINT [PK_User_LoginType] PRIMARY KEY CLUSTERED
    (
        [User_LoginType_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[User_PasswordHistory] WITH NOCHECK ADD
    CONSTRAINT [PK_User_PasswordHistory] PRIMARY KEY CLUSTERED
    (
        [User_PasswordHistory_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[User_SecurityRole] WITH NOCHECK ADD
    CONSTRAINT [PK_User_SecurityRole] PRIMARY KEY CLUSTERED
    (
        [User_SecurityRole_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[Weekday] WITH NOCHECK ADD
    CONSTRAINT [PK_Weekday] PRIMARY KEY CLUSTERED
    (
        [Weekday_ID]
    ) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[dtproperties] WITH NOCHECK ADD
    CONSTRAINT [pk_dtproperties] PRIMARY KEY CLUSTERED
    (
        [id],
        [property]
    ) ON [PRIMARY]
GO

ALTER TABLE [dbo].[AssessmentType] ADD
    CONSTRAINT [DF_Assessment_CreatedBy] DEFAULT (newid()) FOR [CreatedBy],
    CONSTRAINT [DF_Assessment_UpdatedBy] DEFAULT (newid()) FOR [UpdatedBy]
GO

 CREATE UNIQUE INDEX [X1_CourseVersion] ON [dbo].[CourseVersion]([Course_ID], [Version] DESC ) WITH FILLFACTOR
   = 90 ON [PRIMARY]
GO

 CREATE UNIQUE INDEX [X2_CourseVersion] ON [dbo].[CourseVersion]([Number], [CourseVersion_ID]) WITH FILLFACTOR
   = 90 ON [PRIMARY]
GO

 CREATE UNIQUE INDEX [X1_Course_Keyword] ON [dbo].[Course_Keyword]([Keyword_ID], [CourseVersion_ID]) WITH
   FILLFACTOR = 90 ON [PRIMARY]
GO

 CREATE UNIQUE INDEX [X1_Course_Period] ON [dbo].[Course_Period]([CourseVersion_ID], [Part], [Period_ID]) WITH
   FILLFACTOR = 90 ON [PRIMARY]
GO

 CREATE INDEX [X1_Course_Period_Module] ON [dbo].[Course_Period_Module]([Course_Period_ID]) WITH FILLFACTOR = 90
   ON [PRIMARY]
GO

 CREATE INDEX [X1_Course_Period_ModuleItem] ON [dbo].[Course_Period_ModuleItem]([Course_Period_Module_ID]) WITH
   FILLFACTOR = 90 ON [PRIMARY]
GO

 CREATE UNIQUE INDEX [X1_Course_Point] ON [dbo].[Course_Point]([CourseVersion_ID], [Part]) WITH FILLFACTOR = 90
   ON [PRIMARY]
GO

ALTER TABLE [dbo].[Keyword] ADD
    CONSTRAINT [DF_Keyword_Keyword_ID] DEFAULT (newid()) FOR [Keyword_ID]
GO
```

```
 CREATE UNIQUE INDEX [X1_Keyword] ON [dbo].[Keyword]([Name], [Culture_ID]) WITH FILLFACTOR = 90 ON [
   PRIMARY]
GO

ALTER TABLE [dbo].[Lecturer] ADD
    CONSTRAINT [DF_Lecturer_Lecturer_ID] DEFAULT (newid()) FOR [Lecturer_ID]
GO

 CREATE UNIQUE INDEX [X1_Period] ON [dbo].[Period]([StartDate], [Period_ID]) WITH FILLFACTOR = 90 ON [PRIMARY
   ]
GO

 CREATE UNIQUE INDEX [X1_PeriodType_Module] ON [dbo].[PeriodType_Module]([PeriodType_ID], [Module_ID]) WITH
   FILLFACTOR = 90 ON [PRIMARY]
GO

 CREATE INDEX [X1_Specialization_Course] ON [dbo].[Specialization_Course]([SpecializationVersion_ID], [Supplementary], [
   Course_ID]) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

 CREATE INDEX [X1_Student_Course] ON [dbo].[Student_Course]([StudentVersion_ID], [CourseVersion_ID]) WITH FILLFACTOR
    = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[StudyPlan] ADD
    CONSTRAINT [DF_StudyPlan_StudyPlan_ID] DEFAULT (newid()) FOR [StudyPlan_ID]
GO

ALTER TABLE [dbo].[StudyPlanCriterion_ContentConcept] ADD
    CONSTRAINT [DF_StudyPlanCriterion_ContentConcept_StudyPlanCriterion_ContentConcept_ID] DEFAULT (newid()) FOR [
      StudyPlanCriterion_ContentConcept_ID]
GO

ALTER TABLE [dbo].[StudyPlanCriterion_Course] ADD
    CONSTRAINT [DF_StudyPlanCriterion_Course_StudyPlanCriterion_Course_ID] DEFAULT (newid()) FOR [
      StudyPlanCriterion_Course_ID]
GO

ALTER TABLE [dbo].[StudyPlanCriterion_CoursePeriod] ADD
    CONSTRAINT [DF_StudyPlanCriterion_CoursePeriod_StudyPlanCriterion_CoursePeriod_ID] DEFAULT (newid()) FOR [
      StudyPlanCriterion_CoursePeriod_ID]
GO

ALTER TABLE [dbo].[StudyPlanCriterion_EvaluationForm] ADD
    CONSTRAINT [DF_StudyPlanCriterion_EvaluationForm_StudyPlanCriterion_EvaluationForm_ID] DEFAULT (newid()) FOR [
      StudyPlanCriterion_EvaluationForm_ID]
GO

/****** The index created by the following statement is for internal use only. ******/
/****** It is not a real index but exists as statistics only. ******/
if (@@microsoftversion > 0x07000000 )
EXEC ('CREATE STATISTICS [Statistic_StudyPlanCriterion_ID] ON [dbo].[StudyPlanCriterion_Keyword] ([StudyPlanCriterion_ID]) ')
GO

 CREATE UNIQUE INDEX [X1_StudyPlanCriterion_Keyword] ON [dbo].[StudyPlanCriterion_Keyword]([StudyPlanCriterion_ID], [
   Keyword_ID]) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

/****** The index created by the following statement is for internal use only. ******/
/****** It is not a real index but exists as statistics only. ******/
if (@@microsoftversion > 0x07000000 )
EXEC ('CREATE STATISTICS [Statistic_Keyword_ID] ON [dbo].[StudyPlanCriterion_Keyword] ([Keyword_ID]) ')
GO

ALTER TABLE [dbo].[StudyPlanCriterion_Language] ADD
    CONSTRAINT [DF_StudyPlanCriterion_Language_StudyPlanCriterion_Language_ID] DEFAULT (newid()) FOR [
      StudyPlanCriterion_Language_ID]
GO

 CREATE UNIQUE INDEX [X1_StudyPlanCriterion_Language] ON [dbo].[StudyPlanCriterion_Language]([StudyPlanCriterion_ID
   ], [Language_ID]) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

/****** The index created by the following statement is for internal use only. ******/
/****** It is not a real index but exists as statistics only. ******/
if (@@microsoftversion > 0x07000000 )
EXEC ('CREATE STATISTICS [Statistic_StudyPlanCriterion_ID] ON [dbo].[StudyPlanCriterion_Lecturer] ([StudyPlanCriterion_ID]) ')
GO

 CREATE UNIQUE INDEX [X1_StudyPlanCriterion_Lecturer] ON [dbo].[StudyPlanCriterion_Lecturer]([StudyPlanCriterion_ID], [
   Lecturer_ID]) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

 CREATE UNIQUE INDEX [X1_StudyPlanCriterion_StudyType] ON [dbo].[StudyPlanCriterion_StudyType]([
   StudyPlanCriterion_ID], [StudyType_ID]) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

 CREATE UNIQUE INDEX [X1_StudyPlanCriterion_WorkloadPeriod] ON [dbo].[StudyPlanCriterion_WorkloadPeriod]([
   StudyPlanCriterion_ID], [Period_ID]) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

 CREATE UNIQUE INDEX [X1_StudyPlanCriterion_WorkloadPeriodType] ON [dbo].[StudyPlanCriterion_WorkloadPeriodType]([
   StudyPlanCriterion_ID], [PeriodType_ID]) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

 CREATE UNIQUE INDEX [X1_StudyPlanCriterion_WorkloadPeriodType_Module] ON [dbo].[
   StudyPlanCriterion_WorkloadPeriodType_Module]([StudyPlanCriterion_WorkloadPeriodType_ID], [Module_ID]) WITH
   FILLFACTOR = 90 ON [PRIMARY]
GO
```

```
/****** The index created by the following statement is for internal use only. ******/
/****** It is not a real index but exists as statistics only. ******/
if (@@microsoftversion > 0x07000000 )
EXEC ('CREATE STATISTICS [Statistic_StudyPlanCriterion_WorkloadPeriod_ID] ON [dbo].[StudyPlanCriterion_WorkloadPeriod_Module] ([
  StudyPlanCriterion_WorkloadPeriod_ID]) ')
GO

 CREATE UNIQUE INDEX [X1_StudyPlanCriterion_WorkloadPeriod_Module] ON [dbo].[
   StudyPlanCriterion_WorkloadPeriod_Module]([StudyPlanCriterion_WorkloadPeriod_ID], [Module_ID]) WITH FILLFACTOR = 90
   ON [PRIMARY]
GO

ALTER TABLE [dbo].[StudyPlan_Period] ADD
    CONSTRAINT [DF_StudyPlan_Period_StudyPlan_Period_ID] DEFAULT (newid()) FOR [StudyPlan_Period_ID]
GO

ALTER TABLE [dbo].[StudyPlan_PeriodCourse] ADD
    CONSTRAINT [DF_StudyPlan_PeriodCourse_StudyPlan_PeriodCourse_ID] DEFAULT (newid()) FOR [
      StudyPlan_PeriodCourse_ID]
GO

 CREATE INDEX [X1_TechnicalLine_PrerequisiteCourse] ON [dbo].[TechnicalLine_PrerequisiteCourse]([TechnicalLineVersion_ID], [
   Course_ID]) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

 CREATE UNIQUE INDEX [X1_TechnicalLine_PrerequisiteTechnicalPackage] ON [dbo].[
   TechnicalLine_PrerequisiteTechnicalPackage]([TechnicalLineVersion_ID], [TechnicalPackage_ID]) WITH FILLFACTOR = 90 ON [
   PRIMARY]
GO

 CREATE INDEX [X1_TechnicalLine_PrerequisiteTechnicalPackageCourse] ON [dbo].[
   TechnicalLine_PrerequisiteTechnicalPackageCourse]([TechnicalLine_PrerequisiteTechnicalPackage_ID], [Course_ID]) WITH
   FILLFACTOR = 90 ON [PRIMARY]
GO

/****** The index created by the following statement is for internal use only. ******/
/****** It is not a real index but exists as statistics only. ******/
if (@@microsoftversion > 0x07000000 )
EXEC ('CREATE STATISTICS [Statistic_TechnicalPackageVersion_ID] ON [dbo].[TechnicalPackage_FundamentalCourse] ([TechnicalPackageVersion_ID]) ')
GO

 CREATE INDEX [X1_TechnicalPackage_FundamentalCourse] ON [dbo].[TechnicalPackage_FundamentalCourse]([
   TechnicalPackageVersion_ID], [TechnicalPackage_FundamentalCourse_ID]) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

 CREATE INDEX [X1_TechnicalPackage_FundamentalCourseItem] ON [dbo].[TechnicalPackage_FundamentalCourseItem]([
   TechnicalPackage_FundamentalCourse_ID], [Course_ID]) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

/****** The index created by the following statement is for internal use only. ******/
/****** It is not a real index but exists as statistics only. ******/
if (@@microsoftversion > 0x07000000 )
EXEC ('CREATE STATISTICS [Statistic_TechnicalPackageVersion_ID] ON [dbo].[TechnicalPackage_Period] ([TechnicalPackageVersion_ID]) ')
GO

/****** The index created by the following statement is for internal use only. ******/
/****** It is not a real index but exists as statistics only. ******/
if (@@microsoftversion > 0x07000000 )
EXEC ('CREATE STATISTICS [Statistic_Period_ID] ON [dbo].[TechnicalPackage_Period] ([Period_ID]) ')
GO

 CREATE INDEX [X1_TechnicalPackage_Period] ON [dbo].[TechnicalPackage_Period]([TechnicalPackageVersion_ID], [Period_ID], [
   TechnicalPackage_Period_ID]) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

/****** The index created by the following statement is for internal use only. ******/
/****** It is not a real index but exists as statistics only. ******/
if (@@microsoftversion > 0x07000000 )
EXEC ('CREATE STATISTICS [Statistic_TechnicalPackage_Period_ID] ON [dbo].[TechnicalPackage_PeriodCourse] ([TechnicalPackage_Period_ID]) ')
GO

 CREATE UNIQUE INDEX [X1_TechnicalPackage_PeriodCourse] ON [dbo].[TechnicalPackage_PeriodCourse]([
   TechnicalPackage_Period_ID], [TechnicalPackage_PeriodCourse_ID]) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[TechnicalPackage_PeriodCourseItem] ADD
    CONSTRAINT [DF_TechnicalPackage_PeriodCourseItem_Part] DEFAULT (1) FOR [Part]
GO

 CREATE INDEX [X1_TechnicalPackage_PeriodCourseItem] ON [dbo].[TechnicalPackage_PeriodCourseItem]([
   TechnicalPackage_PeriodCourse_ID], [Course_ID]) WITH FILLFACTOR = 90 ON [PRIMARY]
GO

ALTER TABLE [dbo].[TechnicalPackage_PeriodOptionalCourse] ADD
    CONSTRAINT [DF_TechnicalPackage_PeriodOptionalCourse_Part] DEFAULT (1) FOR [Part]
GO

ALTER TABLE [dbo].[User] ADD
    CONSTRAINT [DF_User_User_ID] DEFAULT (newid()) FOR [User_ID],
    CONSTRAINT [DF_User_Created] DEFAULT (10 - 21 - 3) FOR [Created]
GO

ALTER TABLE [dbo].[dtproperties] ADD
    CONSTRAINT [DF__dtpropert__versi__2B0A656D] DEFAULT (0) FOR [version]
GO

ALTER TABLE [dbo].[Assessment] ADD
    CONSTRAINT [FK_Assessment_Grade] FOREIGN KEY
```

```
    (
        [Grade_ID]
    ) REFERENCES [dbo].[Grade] (
        [Grade_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[Assessment] nocheck constraint [FK_Assessment_Grade]
GO

ALTER TABLE [dbo].[ContactData] ADD
    CONSTRAINT [FK_ContactData_ContactDataType] FOREIGN KEY
    (
        [ContactDataType_ID]
    ) REFERENCES [dbo].[ContactDataType] (
        [ContactDataType_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[ContactData] nocheck constraint [FK_ContactData_ContactDataType]
GO

ALTER TABLE [dbo].[CourseGrab] ADD
    CONSTRAINT [FK_CourseGrab_Course] FOREIGN KEY
    (
        [Course_ID]
    ) REFERENCES [dbo].[Course] (
        [Course_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_CourseGrab_CourseVersion] FOREIGN KEY
    (
        [CourseVersion_ID]
    ) REFERENCES [dbo].[CourseVersion] (
        [CourseVersion_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[CourseGrab] nocheck constraint [FK_CourseGrab_Course]
GO

alter table [dbo].[CourseGrab] nocheck constraint [FK_CourseGrab_CourseVersion]
GO

ALTER TABLE [dbo].[CourseVersion] ADD
    CONSTRAINT [FK_CourseVersion_AssessmentType] FOREIGN KEY
    (
        [AssessmentType_ID]
    ) REFERENCES [dbo].[AssessmentType] (
        [AssessmentType_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_CourseVersion_Course] FOREIGN KEY
    (
        [Course_ID]
    ) REFERENCES [dbo].[Course] (
        [Course_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_CourseVersion_Language] FOREIGN KEY
    (
        [Language_ID]
    ) REFERENCES [dbo].[Language] (
        [Language_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[CourseVersion] nocheck constraint [FK_CourseVersion_AssessmentType]
GO

alter table [dbo].[CourseVersion] nocheck constraint [FK_CourseVersion_Course]
GO

alter table [dbo].[CourseVersion] nocheck constraint [FK_CourseVersion_Language]
GO

ALTER TABLE [dbo].[Course_Department] ADD
    CONSTRAINT [FK_Course_Department_Course] FOREIGN KEY
    (
        [Course_ID]
    ) REFERENCES [dbo].[Course] (
        [Course_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_Course_Department_Department] FOREIGN KEY
    (
        [Department_ID]
    ) REFERENCES [dbo].[Department] (
        [Department_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[Course_Department] nocheck constraint [FK_Course_Department_Course]
GO

alter table [dbo].[Course_Department] nocheck constraint [FK_Course_Department_Department]
GO

ALTER TABLE [dbo].[Course_EvaluationForm] ADD
    CONSTRAINT [FK_Course_EvaluationForm_CourseVersion] FOREIGN KEY
    (
        [CourseVersion_ID]
    ) REFERENCES [dbo].[CourseVersion] (
```

```
        [CourseVersion_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_Course_EvaluationForm_EvaluationForm] FOREIGN KEY
    (
        [EvaluationForm_ID]
    ) REFERENCES [dbo].[EvaluationForm] (
        [EvaluationForm_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[Course_EvaluationForm] nocheck constraint [FK_Course_EvaluationForm_CourseVersion]
GO

alter table [dbo].[Course_EvaluationForm] nocheck constraint [FK_Course_EvaluationForm_EvaluationForm]
GO

ALTER TABLE [dbo].[Course_Keyword] ADD
    CONSTRAINT [FK_Course_Keyword_CourseVersion] FOREIGN KEY
    (
        [CourseVersion_ID]
    ) REFERENCES [dbo].[CourseVersion] (
        [CourseVersion_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_Course_Keyword_Keyword] FOREIGN KEY
    (
        [Keyword_ID]
    ) REFERENCES [dbo].[Keyword] (
        [Keyword_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[Course_Keyword] nocheck constraint [FK_Course_Keyword_CourseVersion]
GO

alter table [dbo].[Course_Keyword] nocheck constraint [FK_Course_Keyword_Keyword]
GO

ALTER TABLE [dbo].[Course_Lecturer] ADD
    CONSTRAINT [FK_Course_Lecturer_CourseVersion] FOREIGN KEY
    (
        [CourseVersion_ID]
    ) REFERENCES [dbo].[CourseVersion] (
        [CourseVersion_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_Course_Lecturer_Lecturer] FOREIGN KEY
    (
        [Lecturer_ID]
    ) REFERENCES [dbo].[Lecturer] (
        [Lecturer_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[Course_Lecturer] nocheck constraint [FK_Course_Lecturer_CourseVersion]
GO

alter table [dbo].[Course_Lecturer] nocheck constraint [FK_Course_Lecturer_Lecturer]
GO

ALTER TABLE [dbo].[Course_Period] ADD
    CONSTRAINT [FK_Course_Period_CourseVersion] FOREIGN KEY
    (
        [CourseVersion_ID]
    ) REFERENCES [dbo].[CourseVersion] (
        [CourseVersion_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_Course_Period_Period] FOREIGN KEY
    (
        [Period_ID]
    ) REFERENCES [dbo].[Period] (
        [Period_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[Course_Period] nocheck constraint [FK_Course_Period_CourseVersion]
GO

alter table [dbo].[Course_Period] nocheck constraint [FK_Course_Period_Period]
GO

ALTER TABLE [dbo].[Course_Period_Module] ADD
    CONSTRAINT [FK_Course_Period_Module_Course_Period] FOREIGN KEY
    (
        [Course_Period_ID]
    ) REFERENCES [dbo].[Course_Period] (
        [Course_Period_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[Course_Period_Module] nocheck constraint [FK_Course_Period_Module_Course_Period]
GO

ALTER TABLE [dbo].[Course_Period_ModuleItem] ADD
    CONSTRAINT [FK_Course_Period_ModuleItem_Course_Period_Module] FOREIGN KEY
    (
        [Course_Period_Module_ID]
    ) REFERENCES [dbo].[Course_Period_Module] (
        [Course_Period_Module_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_Course_Period_ModuleItem_Module] FOREIGN KEY
```

```
    (
        [Module_ID]
    ) REFERENCES [dbo].[Module] (
        [Module_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[Course_Period_ModuleItem] nocheck constraint [FK_Course_Period_ModuleItem_Course_Period_Module]
GO

alter table [dbo].[Course_Period_ModuleItem] nocheck constraint [FK_Course_Period_ModuleItem_Module]
GO

ALTER TABLE [dbo].[Course_Point] ADD
    CONSTRAINT [FK_Course_Point_CourseVersion] FOREIGN KEY
    (
        [CourseVersion_ID]
    ) REFERENCES [dbo].[CourseVersion] (
        [CourseVersion_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_Course_Point_Point] FOREIGN KEY
    (
        [Point_ID]
    ) REFERENCES [dbo].[Point] (
        [Point_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[Course_Point] nocheck constraint [FK_Course_Point_CourseVersion]
GO

alter table [dbo].[Course_Point] nocheck constraint [FK_Course_Point_Point]
GO

ALTER TABLE [dbo].[Course_RecommendedPlacement] ADD
    CONSTRAINT [FK_Course_RecommendedPlacementSpeci↓c_CourseVersion] FOREIGN KEY
    (
        [CourseVersion_ID]
    ) REFERENCES [dbo].[CourseVersion] (
        [CourseVersion_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_Course_RecommendedPlacementSpeci↓c_Point] FOREIGN KEY
    (
        [Point_ID]
    ) REFERENCES [dbo].[Point] (
        [Point_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_Course_RecommendedPlacementSpeci↓c_RecommendedPlacementConcept] FOREIGN KEY
    (
        [RecommendedPlacementConcept_ID]
    ) REFERENCES [dbo].[RecommendedPlacementConcept] (
        [RecommendedPlacementConcept_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_Course_RecommendedPlacementSpeci↓c_StudyType] FOREIGN KEY
    (
        [StudyType_ID]
    ) REFERENCES [dbo].[StudyType] (
        [StudyType_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[Course_RecommendedPlacement] nocheck constraint [FK_Course_RecommendedPlacementSpeci↓c_CourseVersion]
GO

alter table [dbo].[Course_RecommendedPlacement] nocheck constraint [FK_Course_RecommendedPlacementSpeci↓c_Point]
GO

alter table [dbo].[Course_RecommendedPlacement] nocheck constraint [
  FK_Course_RecommendedPlacementSpeci↓c_RecommendedPlacementConcept]
GO

alter table [dbo].[Course_RecommendedPlacement] nocheck constraint [FK_Course_RecommendedPlacementSpeci↓c_StudyType]
GO

ALTER TABLE [dbo].[Course_RelationCourse] ADD
    CONSTRAINT [FK_Course_RelationCourse_Course_RelationCourseType] FOREIGN KEY
    (
        [Course_RelationCourseType_ID]
    ) REFERENCES [dbo].[Course_RelationCourseType] (
        [Course_RelationCourseType_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_Course_RelationCourse_CourseVersion] FOREIGN KEY
    (
        [CourseVersion_ID]
    ) REFERENCES [dbo].[CourseVersion] (
        [CourseVersion_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[Course_RelationCourse] nocheck constraint [FK_Course_RelationCourse_Course_RelationCourseType]
GO

alter table [dbo].[Course_RelationCourse] nocheck constraint [FK_Course_RelationCourse_CourseVersion]
GO

ALTER TABLE [dbo].[Course_RelationCourseItem] ADD
    CONSTRAINT [FK_Course_RelationCourseItem_Course] FOREIGN KEY
    (
        [Course_ID]
```

```
    ) REFERENCES [dbo].[Course] (
        [Course_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_Course_RelationCourseItem_Course_RelationCourse] FOREIGN KEY
    (
        [Course_RelationCourse_ID]
    ) REFERENCES [dbo].[Course_RelationCourse] (
        [Course_RelationCourse_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[Course_RelationCourseItem] nocheck constraint [FK_Course_RelationCourseItem_Course]
GO

alter table [dbo].[Course_RelationCourseItem] nocheck constraint [FK_Course_RelationCourseItem_Course_RelationCourse]
GO

ALTER TABLE [dbo].[Course_StudyType] ADD
    CONSTRAINT [FK_Course_StudyType_Course_StudyTypeCategory] FOREIGN KEY
    (
        [Course_StudyTypeCategory_ID]
    ) REFERENCES [dbo].[Course_StudyTypeCategory] (
        [Course_StudyTypeCategory_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_Course_StudyType_CourseVersion] FOREIGN KEY
    (
        [CourseVersion_ID]
    ) REFERENCES [dbo].[CourseVersion] (
        [CourseVersion_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_Course_StudyType_StudyType] FOREIGN KEY
    (
        [StudyType_ID]
    ) REFERENCES [dbo].[StudyType] (
        [StudyType_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[Course_StudyType] nocheck constraint [FK_Course_StudyType_Course_StudyTypeCategory]
GO

alter table [dbo].[Course_StudyType] nocheck constraint [FK_Course_StudyType_CourseVersion]
GO

alter table [dbo].[Course_StudyType] nocheck constraint [FK_Course_StudyType_StudyType]
GO

ALTER TABLE [dbo].[Course_StudyTypeCategory] ADD
    CONSTRAINT [FK_Course_StudyTypeCategory_StudyType] FOREIGN KEY
    (
        [StudyType_ID]
    ) REFERENCES [dbo].[StudyType] (
        [StudyType_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[Course_StudyTypeCategory] nocheck constraint [FK_Course_StudyTypeCategory_StudyType]
GO

ALTER TABLE [dbo].[Module] ADD
    CONSTRAINT [FK_Module_Weekday] FOREIGN KEY
    (
        [Weekday_ID]
    ) REFERENCES [dbo].[Weekday] (
        [Weekday_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[Module] nocheck constraint [FK_Module_Weekday]
GO

ALTER TABLE [dbo].[Period] ADD
    CONSTRAINT [FK_Period_PeriodType] FOREIGN KEY
    (
        [PeriodType_ID]
    ) REFERENCES [dbo].[PeriodType] (
        [PeriodType_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[Period] nocheck constraint [FK_Period_PeriodType]
GO

ALTER TABLE [dbo].[PeriodType_Module] ADD
    CONSTRAINT [FK_PeriodType_Module_Module] FOREIGN KEY
    (
        [Module_ID]
    ) REFERENCES [dbo].[Module] (
        [Module_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_PeriodType_Module_PeriodType] FOREIGN KEY
    (
        [PeriodType_ID]
    ) REFERENCES [dbo].[PeriodType] (
        [PeriodType_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[PeriodType_Module] nocheck constraint [FK_PeriodType_Module_Module]
```

```
GO

alter table [dbo].[PeriodType_Module] nocheck constraint [FK_PeriodType_Module_PeriodType]
GO

ALTER TABLE [dbo].[PersonVersion] ADD
    CONSTRAINT [FK_PersonVersion_Gender] FOREIGN KEY
    (
        [Gender_ID]
    ) REFERENCES [dbo].[Gender] (
        [Gender_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_PersonVersion_Person] FOREIGN KEY
    (
        [Person_ID]
    ) REFERENCES [dbo].[Person] (
        [Person_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_PersonVersion_User] FOREIGN KEY
    (
        [User_ID]
    ) REFERENCES [dbo].[User] (
        [User_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[PersonVersion] nocheck constraint [FK_PersonVersion_Gender]
GO

alter table [dbo].[PersonVersion] nocheck constraint [FK_PersonVersion_Person]
GO

alter table [dbo].[PersonVersion] nocheck constraint [FK_PersonVersion_User]
GO

ALTER TABLE [dbo].[Person_ContactData] ADD
    CONSTRAINT [FK_Person_ContactData_ContactData] FOREIGN KEY
    (
        [ContactData_ID]
    ) REFERENCES [dbo].[ContactData] (
        [ContactData_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_Person_ContactData_PersonVersion] FOREIGN KEY
    (
        [PersonVersion_ID]
    ) REFERENCES [dbo].[PersonVersion] (
        [PersonVersion_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[Person_ContactData] nocheck constraint [FK_Person_ContactData_ContactData]
GO

alter table [dbo].[Person_ContactData] nocheck constraint [FK_Person_ContactData_PersonVersion]
GO

ALTER TABLE [dbo].[Project] ADD
    CONSTRAINT [FK_Project_Assessment] FOREIGN KEY
    (
        [AssessmentType_ID]
    ) REFERENCES [dbo].[AssessmentType] (
        [AssessmentType_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_Project_Period] FOREIGN KEY
    (
        [Period_ID]
    ) REFERENCES [dbo].[Period] (
        [Period_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_Project_Point] FOREIGN KEY
    (
        [Point_ID]
    ) REFERENCES [dbo].[Point] (
        [Point_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_Project_ProjectType] FOREIGN KEY
    (
        [ProjectType_ID]
    ) REFERENCES [dbo].[ProjectType] (
        [ProjectType_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[Project] nocheck constraint [FK_Project_Assessment]
GO

alter table [dbo].[Project] nocheck constraint [FK_Project_Period]
GO

alter table [dbo].[Project] nocheck constraint [FK_Project_Point]
GO

alter table [dbo].[Project] nocheck constraint [FK_Project_ProjectType]
GO

ALTER TABLE [dbo].[RecommendedPlacementConcept_StudyType] ADD
    CONSTRAINT [FK_RecommendedPlacementConcept_StudyType_Point] FOREIGN KEY
    (
        [Point_ID]
```

```
    ) REFERENCES [dbo].[Point] (
        [Point_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_RecommendedPlacementConcept_StudyType_RecommendedPlacementConcept] FOREIGN KEY
    (
        [RecommendedPlacementConcept_ID]
    ) REFERENCES [dbo].[RecommendedPlacementConcept] (
        [RecommendedPlacementConcept_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_RecommendedPlacementConcept_StudyType_StudyType] FOREIGN KEY
    (
        [StudyType_ID]
    ) REFERENCES [dbo].[StudyType] (
        [StudyType_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[RecommendedPlacementConcept_StudyType] nocheck constraint [
  FK_RecommendedPlacementConcept_StudyType_Point]
GO

alter table [dbo].[RecommendedPlacementConcept_StudyType] nocheck constraint [
  FK_RecommendedPlacementConcept_StudyType_RecommendedPlacementConcept]
GO

alter table [dbo].[RecommendedPlacementConcept_StudyType] nocheck constraint [
  FK_RecommendedPlacementConcept_StudyType_StudyType]
GO

ALTER TABLE [dbo].[SecurityPermission] ADD
    CONSTRAINT [FK_SecurityPermission_SecurityPermissionCategory] FOREIGN KEY
    (
        [SecurityPermissionCategory_ID]
    ) REFERENCES [dbo].[SecurityPermissionCategory] (
        [SecurityPermissionCategory_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[SecurityPermission] nocheck constraint [FK_SecurityPermission_SecurityPermissionCategory]
GO

ALTER TABLE [dbo].[SecurityRole_SecurityPermission] ADD
    CONSTRAINT [FK_SecurityRole_SecurityPermission_SecurityPermission] FOREIGN KEY
    (
        [SecurityPermission_ID]
    ) REFERENCES [dbo].[SecurityPermission] (
        [SecurityPermission_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_SecurityRole_SecurityPermission_SecurityRole] FOREIGN KEY
    (
        [SecurityRole_ID]
    ) REFERENCES [dbo].[SecurityRole] (
        [SecurityRole_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[SecurityRole_SecurityPermission] nocheck constraint [FK_SecurityRole_SecurityPermission_SecurityPermission]
GO

alter table [dbo].[SecurityRole_SecurityPermission] nocheck constraint [FK_SecurityRole_SecurityPermission_SecurityRole]
GO

ALTER TABLE [dbo].[SpecializationVersion] ADD
    CONSTRAINT [FK_SpecializationVersion_Point] FOREIGN KEY
    (
        [Point_ID]
    ) REFERENCES [dbo].[Point] (
        [Point_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_SpecializationVersion_Specialization] FOREIGN KEY
    (
        [Specialization_ID]
    ) REFERENCES [dbo].[Specialization] (
        [Specialization_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[SpecializationVersion] nocheck constraint [FK_SpecializationVersion_Point]
GO

alter table [dbo].[SpecializationVersion] nocheck constraint [FK_SpecializationVersion_Specialization]
GO

ALTER TABLE [dbo].[Specialization_Course] ADD
    CONSTRAINT [FK_Specialization_Course_Course] FOREIGN KEY
    (
        [Course_ID]
    ) REFERENCES [dbo].[Course] (
        [Course_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_Specialization_Course_SpecializationVersion] FOREIGN KEY
    (
        [SpecializationVersion_ID]
    ) REFERENCES [dbo].[SpecializationVersion] (
        [SpecializationVersion_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[Specialization_Course] nocheck constraint [FK_Specialization_Course_Course]
```

```
GO

alter table [dbo].[Specialization_Course] nocheck constraint [FK_Specialization_Course_SpecializationVersion]
GO

ALTER TABLE [dbo].[StudentVersion] ADD
    CONSTRAINT [FK_StudentVersion_Person] FOREIGN KEY
    (
        [Person_ID]
    ) REFERENCES [dbo].[Person] (
        [Person_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_StudentVersion_Student] FOREIGN KEY
    (
        [Student_ID]
    ) REFERENCES [dbo].[Student] (
        [Student_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_StudentVersion_StudyTypeVersion] FOREIGN KEY
    (
        [StudyTypeVersion_ID]
    ) REFERENCES [dbo].[StudyTypeVersion] (
        [StudyTypeVersion_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_StudentVersion_TechnicalLineVersion] FOREIGN KEY
    (
        [TechnicalLineVersion_ID]
    ) REFERENCES [dbo].[TechnicalLineVersion] (
        [TechnicalLineVersion_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_StudentVersion_TechnicalPackageVersion] FOREIGN KEY
    (
        [TechnicalPackageVersion_ID]
    ) REFERENCES [dbo].[TechnicalPackageVersion] (
        [TechnicalPackageVersion_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[StudentVersion] nocheck constraint [FK_StudentVersion_Person]
GO

alter table [dbo].[StudentVersion] nocheck constraint [FK_StudentVersion_Student]
GO

alter table [dbo].[StudentVersion] nocheck constraint [FK_StudentVersion_StudyTypeVersion]
GO

alter table [dbo].[StudentVersion] nocheck constraint [FK_StudentVersion_TechnicalLineVersion]
GO

alter table [dbo].[StudentVersion] nocheck constraint [FK_StudentVersion_TechnicalPackageVersion]
GO

ALTER TABLE [dbo].[Student_Course] ADD
    CONSTRAINT [FK_Student_Course_Assessment] FOREIGN KEY
    (
        [Assessment_ID]
    ) REFERENCES [dbo].[Assessment] (
        [Assessment_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_Student_Course_CourseVersion] FOREIGN KEY
    (
        [CourseVersion_ID]
    ) REFERENCES [dbo].[CourseVersion] (
        [CourseVersion_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_Student_Course_StudentVersion] FOREIGN KEY
    (
        [StudentVersion_ID]
    ) REFERENCES [dbo].[StudentVersion] (
        [StudentVersion_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[Student_Course] nocheck constraint [FK_Student_Course_Assessment]
GO

alter table [dbo].[Student_Course] nocheck constraint [FK_Student_Course_CourseVersion]
GO

alter table [dbo].[Student_Course] nocheck constraint [FK_Student_Course_StudentVersion]
GO

ALTER TABLE [dbo].[Student_Department] ADD
    CONSTRAINT [FK_Student_Department_Department] FOREIGN KEY
    (
        [Department_ID]
    ) REFERENCES [dbo].[Department] (
        [Department_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_Student_Department_StudentVersion] FOREIGN KEY
    (
        [StudentVersion_ID]
    ) REFERENCES [dbo].[StudentVersion] (
        [StudentVersion_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[Student_Department] nocheck constraint [FK_Student_Department_Department]
```

```
GO

alter table [dbo].[Student_Department] nocheck constraint [FK_Student_Department_StudentVersion]
GO

ALTER TABLE [dbo].[Student_Project] ADD
    CONSTRAINT [FK_Student_Project_Assessment] FOREIGN KEY
    (
        [Assessment_ID]
    ) REFERENCES [dbo].[Assessment] (
        [Assessment_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_Student_Project_Project] FOREIGN KEY
    (
        [Project_ID]
    ) REFERENCES [dbo].[Project] (
        [Project_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_Student_Project_StudentVersion] FOREIGN KEY
    (
        [StudentVersion_ID]
    ) REFERENCES [dbo].[StudentVersion] (
        [StudentVersion_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[Student_Project] nocheck constraint [FK_Student_Project_Assessment]
GO

alter table [dbo].[Student_Project] nocheck constraint [FK_Student_Project_Project]
GO

alter table [dbo].[Student_Project] nocheck constraint [FK_Student_Project_StudentVersion]
GO

ALTER TABLE [dbo].[Student_StudyPlan] ADD
    CONSTRAINT [FK_Student_StudyPlan_Student] FOREIGN KEY
    (
        [Student_ID]
    ) REFERENCES [dbo].[Student] (
        [Student_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_Student_StudyPlan_StudyPlan] FOREIGN KEY
    (
        [StudyPlan_ID]
    ) REFERENCES [dbo].[StudyPlan] (
        [StudyPlan_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[Student_StudyPlan] nocheck constraint [FK_Student_StudyPlan_Student]
GO

alter table [dbo].[Student_StudyPlan] nocheck constraint [FK_Student_StudyPlan_StudyPlan]
GO

ALTER TABLE [dbo].[Student_StudyPlanCriterion] ADD
    CONSTRAINT [FK_Student_StudyPlanCriterion_Student] FOREIGN KEY
    (
        [Student_ID]
    ) REFERENCES [dbo].[Student] (
        [Student_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_Student_StudyPlanCriterion_StudyPlanCriterion] FOREIGN KEY
    (
        [StudyPlanCriterion_ID]
    ) REFERENCES [dbo].[StudyPlanCriterion] (
        [StudyPlanCriterion_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[Student_StudyPlanCriterion] nocheck constraint [FK_Student_StudyPlanCriterion_Student]
GO

alter table [dbo].[Student_StudyPlanCriterion] nocheck constraint [FK_Student_StudyPlanCriterion_StudyPlanCriterion]
GO

ALTER TABLE [dbo].[StudyPlanCriterion] ADD
    CONSTRAINT [FK_StudyPlanCriterion_SpecializationVersion] FOREIGN KEY
    (
        [SpecializationVersion_ID]
    ) REFERENCES [dbo].[SpecializationVersion] (
        [SpecializationVersion_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_StudyPlanCriterion_TechnicalFieldVersion] FOREIGN KEY
    (
        [TechnicalFieldVersion_ID]
    ) REFERENCES [dbo].[TechnicalFieldVersion] (
        [TechnicalFieldVersion_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_StudyPlanCriterion_TechnicalLineVersion] FOREIGN KEY
    (
        [TechnicalLineVersion_ID]
    ) REFERENCES [dbo].[TechnicalLineVersion] (
        [TechnicalLineVersion_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_StudyPlanCriterion_TechnicalPackageVersion] FOREIGN KEY
    (
        [TechnicalPackageVersion_ID]
```

```
    ) REFERENCES [dbo].[TechnicalPackageVersion] (
        [TechnicalPackageVersion_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[StudyPlanCriterion] nocheck constraint [FK_StudyPlanCriterion_SpecializationVersion]
GO

alter table [dbo].[StudyPlanCriterion] nocheck constraint [FK_StudyPlanCriterion_TechnicalFieldVersion]
GO

alter table [dbo].[StudyPlanCriterion] nocheck constraint [FK_StudyPlanCriterion_TechnicalLineVersion]
GO

alter table [dbo].[StudyPlanCriterion] nocheck constraint [FK_StudyPlanCriterion_TechnicalPackageVersion]
GO

ALTER TABLE [dbo].[StudyPlanCriterion_ContentConcept] ADD
    CONSTRAINT [FK_StudyPlanCriterion_ContentConcept_ContentConcept] FOREIGN KEY
    (
        [ContentConcept_ID]
    ) REFERENCES [dbo].[ContentConcept] (
        [ContentConcept_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_StudyPlanCriterion_ContentConcept_StudyPlanCriterion] FOREIGN KEY
    (
        [StudyPlanCriterion_ID]
    ) REFERENCES [dbo].[StudyPlanCriterion] (
        [StudyPlanCriterion_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[StudyPlanCriterion_ContentConcept] nocheck constraint [
  FK_StudyPlanCriterion_ContentConcept_ContentConcept]
GO

alter table [dbo].[StudyPlanCriterion_ContentConcept] nocheck constraint [
  FK_StudyPlanCriterion_ContentConcept_StudyPlanCriterion]
GO

ALTER TABLE [dbo].[StudyPlanCriterion_Course] ADD
    CONSTRAINT [FK_StudyPlanCriterion_Course_Course] FOREIGN KEY
    (
        [Course_ID]
    ) REFERENCES [dbo].[Course] (
        [Course_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_StudyPlanCriterion_Course_StudyPlanCriterion] FOREIGN KEY
    (
        [StudyPlanCriterion_ID]
    ) REFERENCES [dbo].[StudyPlanCriterion] (
        [StudyPlanCriterion_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[StudyPlanCriterion_Course] nocheck constraint [FK_StudyPlanCriterion_Course_Course]
GO

alter table [dbo].[StudyPlanCriterion_Course] nocheck constraint [FK_StudyPlanCriterion_Course_StudyPlanCriterion]
GO

ALTER TABLE [dbo].[StudyPlanCriterion_CoursePeriod] ADD
    CONSTRAINT [FK_StudyPlanCriterion_CoursePeriod_Course] FOREIGN KEY
    (
        [Course_ID]
    ) REFERENCES [dbo].[Course] (
        [Course_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_StudyPlanCriterion_CoursePeriod_Period] FOREIGN KEY
    (
        [Period_ID]
    ) REFERENCES [dbo].[Period] (
        [Period_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_StudyPlanCriterion_CoursePeriod_StudyPlanCriterion] FOREIGN KEY
    (
        [StudyPlanCriterion_ID]
    ) REFERENCES [dbo].[StudyPlanCriterion] (
        [StudyPlanCriterion_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[StudyPlanCriterion_CoursePeriod] nocheck constraint [FK_StudyPlanCriterion_CoursePeriod_Course]
GO

alter table [dbo].[StudyPlanCriterion_CoursePeriod] nocheck constraint [FK_StudyPlanCriterion_CoursePeriod_Period]
GO

alter table [dbo].[StudyPlanCriterion_CoursePeriod] nocheck constraint [FK_StudyPlanCriterion_CoursePeriod_StudyPlanCriterion]
GO

ALTER TABLE [dbo].[StudyPlanCriterion_EvaluationForm] ADD
    CONSTRAINT [FK_StudyPlanCriterion_EvaluationForm_EvaluationForm] FOREIGN KEY
    (
        [EvaluationForm_ID]
    ) REFERENCES [dbo].[EvaluationForm] (
        [EvaluationForm_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_StudyPlanCriterion_EvaluationForm_StudyPlanCriterion] FOREIGN KEY
```

```
    (
        [StudyPlanCriterion_ID]
    ) REFERENCES [dbo].[StudyPlanCriterion] (
        [StudyPlanCriterion_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[StudyPlanCriterion_EvaluationForm] nocheck constraint [FK_StudyPlanCriterion_EvaluationForm_EvaluationForm
  ]
GO

alter table [dbo].[StudyPlanCriterion_EvaluationForm] nocheck constraint [
  FK_StudyPlanCriterion_EvaluationForm_StudyPlanCriterion]
GO

ALTER TABLE [dbo].[StudyPlanCriterion_Keyword] ADD
    CONSTRAINT [FK_StudyPlanCriterion_Keyword_Keyword] FOREIGN KEY
    (
        [Keyword_ID]
    ) REFERENCES [dbo].[Keyword] (
        [Keyword_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_StudyPlanCriterion_Keyword_StudyPlanCriterion] FOREIGN KEY
    (
        [StudyPlanCriterion_ID]
    ) REFERENCES [dbo].[StudyPlanCriterion] (
        [StudyPlanCriterion_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[StudyPlanCriterion_Keyword] nocheck constraint [FK_StudyPlanCriterion_Keyword_Keyword]
GO

alter table [dbo].[StudyPlanCriterion_Keyword] nocheck constraint [FK_StudyPlanCriterion_Keyword_StudyPlanCriterion]
GO

ALTER TABLE [dbo].[StudyPlanCriterion_Language] ADD
    CONSTRAINT [FK_StudyPlanCriterion_Language_Language] FOREIGN KEY
    (
        [Language_ID]
    ) REFERENCES [dbo].[Language] (
        [Language_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_StudyPlanCriterion_Language_StudyPlanCriterion] FOREIGN KEY
    (
        [StudyPlanCriterion_ID]
    ) REFERENCES [dbo].[StudyPlanCriterion] (
        [StudyPlanCriterion_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[StudyPlanCriterion_Language] nocheck constraint [FK_StudyPlanCriterion_Language_Language]
GO

alter table [dbo].[StudyPlanCriterion_Language] nocheck constraint [FK_StudyPlanCriterion_Language_StudyPlanCriterion]
GO

ALTER TABLE [dbo].[StudyPlanCriterion_Lecturer] ADD
    CONSTRAINT [FK_StudyPlanCriterion_Lecturer_Lecturer] FOREIGN KEY
    (
        [Lecturer_ID]
    ) REFERENCES [dbo].[Lecturer] (
        [Lecturer_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_StudyPlanCriterion_Lecturer_StudyPlanCriterion] FOREIGN KEY
    (
        [StudyPlanCriterion_ID]
    ) REFERENCES [dbo].[StudyPlanCriterion] (
        [StudyPlanCriterion_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[StudyPlanCriterion_Lecturer] nocheck constraint [FK_StudyPlanCriterion_Lecturer_Lecturer]
GO

alter table [dbo].[StudyPlanCriterion_Lecturer] nocheck constraint [FK_StudyPlanCriterion_Lecturer_StudyPlanCriterion]
GO

ALTER TABLE [dbo].[StudyPlanCriterion_ProjectPeriodPoint] ADD
    CONSTRAINT [FK_StudyPlanCriterion_ProjectPeriodPoint_Period] FOREIGN KEY
    (
        [Period_ID]
    ) REFERENCES [dbo].[Period] (
        [Period_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_StudyPlanCriterion_ProjectPeriodPoint_Point] FOREIGN KEY
    (
        [Point_ID]
    ) REFERENCES [dbo].[Point] (
        [Point_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_StudyPlanCriterion_ProjectPeriodPoint_Project] FOREIGN KEY
    (
        [Project_ID]
    ) REFERENCES [dbo].[Project] (
        [Project_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_StudyPlanCriterion_ProjectPeriodPoint_StudyPlanCriterion] FOREIGN KEY
    (
```

```
        [StudyPlanCriterion_ID]
    ) REFERENCES [dbo].[StudyPlanCriterion] (
        [StudyPlanCriterion_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[StudyPlanCriterion_ProjectPeriodPoint] nocheck constraint [FK_StudyPlanCriterion_ProjectPeriodPoint_Period]
GO

alter table [dbo].[StudyPlanCriterion_ProjectPeriodPoint] nocheck constraint [FK_StudyPlanCriterion_ProjectPeriodPoint_Point]
GO

alter table [dbo].[StudyPlanCriterion_ProjectPeriodPoint] nocheck constraint [FK_StudyPlanCriterion_ProjectPeriodPoint_Project]
GO

alter table [dbo].[StudyPlanCriterion_ProjectPeriodPoint] nocheck constraint [
  FK_StudyPlanCriterion_ProjectPeriodPoint_StudyPlanCriterion]
GO

ALTER TABLE [dbo].[StudyPlanCriterion_StudyType] ADD
    CONSTRAINT [FK_StudyPlanCriterion_StudyType_StudyPlanCriterion] FOREIGN KEY
    (
        [StudyPlanCriterion_ID]
    ) REFERENCES [dbo].[StudyPlanCriterion] (
        [StudyPlanCriterion_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_StudyPlanCriterion_StudyType_StudyType] FOREIGN KEY
    (
        [StudyType_ID]
    ) REFERENCES [dbo].[StudyType] (
        [StudyType_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[StudyPlanCriterion_StudyType] nocheck constraint [FK_StudyPlanCriterion_StudyType_StudyPlanCriterion]
GO

alter table [dbo].[StudyPlanCriterion_StudyType] nocheck constraint [FK_StudyPlanCriterion_StudyType_StudyType]
GO

ALTER TABLE [dbo].[StudyPlanCriterion_TechnicalPrerequisiteCourse] ADD
    CONSTRAINT [FK_StudyPlanCriterion_TechnicalPrerequisiteCourse_Course] FOREIGN KEY
    (
        [Course_ID]
    ) REFERENCES [dbo].[Course] (
        [Course_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_StudyPlanCriterion_TechnicalPrerequisiteCourse_StudyPlanCriterion] FOREIGN KEY
    (
        [StudyPlanCriterion_ID]
    ) REFERENCES [dbo].[StudyPlanCriterion] (
        [StudyPlanCriterion_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[StudyPlanCriterion_TechnicalPrerequisiteCourse] nocheck constraint [
  FK_StudyPlanCriterion_TechnicalPrerequisiteCourse_Course]
GO

alter table [dbo].[StudyPlanCriterion_TechnicalPrerequisiteCourse] nocheck constraint [
  FK_StudyPlanCriterion_TechnicalPrerequisiteCourse_StudyPlanCriterion]
GO

ALTER TABLE [dbo].[StudyPlanCriterion_WorkloadPeriod] ADD
    CONSTRAINT [FK_StudyPlanCriterion_WorkloadPeriod_Period] FOREIGN KEY
    (
        [Period_ID]
    ) REFERENCES [dbo].[Period] (
        [Period_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_StudyPlanCriterion_WorkloadPeriod_Point] FOREIGN KEY
    (
        [Point_ID]
    ) REFERENCES [dbo].[Point] (
        [Point_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_StudyPlanCriterion_WorkloadPeriod_StudyPlanCriterion] FOREIGN KEY
    (
        [StudyPlanCriterion_ID]
    ) REFERENCES [dbo].[StudyPlanCriterion] (
        [StudyPlanCriterion_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[StudyPlanCriterion_WorkloadPeriod] nocheck constraint [FK_StudyPlanCriterion_WorkloadPeriod_Period]
GO

alter table [dbo].[StudyPlanCriterion_WorkloadPeriod] nocheck constraint [FK_StudyPlanCriterion_WorkloadPeriod_Point]
GO

alter table [dbo].[StudyPlanCriterion_WorkloadPeriod] nocheck constraint [
  FK_StudyPlanCriterion_WorkloadPeriod_StudyPlanCriterion]
GO

ALTER TABLE [dbo].[StudyPlanCriterion_WorkloadPeriodType] ADD
    CONSTRAINT [FK_StudyPlanCriterion_WorkloadPeriodType_PeriodType] FOREIGN KEY
    (
        [PeriodType_ID]
    ) REFERENCES [dbo].[PeriodType] (
```

```
        [PeriodType_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_StudyPlanCriterion_WorkloadPeriodType_Point] FOREIGN KEY
    (
        [Point_ID]
    ) REFERENCES [dbo].[Point] (
        [Point_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_StudyPlanCriterion_WorkloadPeriodType_StudyPlanCriterion] FOREIGN KEY
    (
        [StudyPlanCriterion_ID]
    ) REFERENCES [dbo].[StudyPlanCriterion] (
        [StudyPlanCriterion_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[StudyPlanCriterion_WorkloadPeriodType] nocheck constraint [
  FK_StudyPlanCriterion_WorkloadPeriodType_PeriodType]
GO

alter table [dbo].[StudyPlanCriterion_WorkloadPeriodType] nocheck constraint [FK_StudyPlanCriterion_WorkloadPeriodType_Point]
GO

alter table [dbo].[StudyPlanCriterion_WorkloadPeriodType] nocheck constraint [
  FK_StudyPlanCriterion_WorkloadPeriodType_StudyPlanCriterion]
GO

ALTER TABLE [dbo].[StudyPlanCriterion_WorkloadPeriodType_Module] ADD
    CONSTRAINT [FK_StudyPlanCriterion_WorkloadPeriodType_Module_Module] FOREIGN KEY
    (
        [Module_ID]
    ) REFERENCES [dbo].[Module] (
        [Module_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_StudyPlanCriterion_WorkloadPeriodType_Module_StudyPlanCriterion_WorkloadPeriodType] FOREIGN
      KEY
    (
        [StudyPlanCriterion_WorkloadPeriodType_ID]
    ) REFERENCES [dbo].[StudyPlanCriterion_WorkloadPeriodType] (
        [StudyPlanCriterion_WorkloadPeriodType_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[StudyPlanCriterion_WorkloadPeriodType_Module] nocheck constraint [
  FK_StudyPlanCriterion_WorkloadPeriodType_Module_Module]
GO

alter table [dbo].[StudyPlanCriterion_WorkloadPeriodType_Module] nocheck constraint [
  FK_StudyPlanCriterion_WorkloadPeriodType_Module_StudyPlanCriterion_WorkloadPeriodType]
GO

ALTER TABLE [dbo].[StudyPlanCriterion_WorkloadPeriod_Module] ADD
    CONSTRAINT [FK_StudyPlanCriterion_WorkloadPeriod_Module_Module] FOREIGN KEY
    (
        [Module_ID]
    ) REFERENCES [dbo].[Module] (
        [Module_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_StudyPlanCriterion_WorkloadPeriod_Module_StudyPlanCriterion_WorkloadPeriod] FOREIGN KEY
    (
        [StudyPlanCriterion_WorkloadPeriod_ID]
    ) REFERENCES [dbo].[StudyPlanCriterion_WorkloadPeriod] (
        [StudyPlanCriterion_WorkloadPeriod_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[StudyPlanCriterion_WorkloadPeriod_Module] nocheck constraint [
  FK_StudyPlanCriterion_WorkloadPeriod_Module_Module]
GO

alter table [dbo].[StudyPlanCriterion_WorkloadPeriod_Module] nocheck constraint [
  FK_StudyPlanCriterion_WorkloadPeriod_Module_StudyPlanCriterion_WorkloadPeriod]
GO

ALTER TABLE [dbo].[StudyPlan_Period] ADD
    CONSTRAINT [FK_StudyPlan_Period_Period] FOREIGN KEY
    (
        [Period_ID]
    ) REFERENCES [dbo].[Period] (
        [Period_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_StudyPlan_Period_Project] FOREIGN KEY
    (
        [Project_ID]
    ) REFERENCES [dbo].[Project] (
        [Project_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_StudyPlan_Period_StudyPlan] FOREIGN KEY
    (
        [StudyPlan_ID]
    ) REFERENCES [dbo].[StudyPlan] (
        [StudyPlan_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[StudyPlan_Period] nocheck constraint [FK_StudyPlan_Period_Period]
GO

alter table [dbo].[StudyPlan_Period] nocheck constraint [FK_StudyPlan_Period_Project]
```

```
GO

alter table [dbo].[StudyPlan_Period] nocheck constraint [FK_StudyPlan_Period_StudyPlan]
GO

ALTER TABLE [dbo].[StudyPlan_PeriodCourse] ADD
    CONSTRAINT [FK_StudyPlan_PeriodCourse_CourseVersion] FOREIGN KEY
    (
        [CourseVersion_ID]
    ) REFERENCES [dbo].[CourseVersion] (
        [CourseVersion_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_StudyPlan_PeriodCourse_StudyPlan_Period] FOREIGN KEY
    (
        [StudyPlan_PeriodCourse_ID]
    ) REFERENCES [dbo].[StudyPlan_Period] (
        [StudyPlan_Period_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_StudyPlan_PeriodCourse_StudyPlan_Period1] FOREIGN KEY
    (
        [StudyPlan_Period_ID]
    ) REFERENCES [dbo].[StudyPlan_Period] (
        [StudyPlan_Period_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[StudyPlan_PeriodCourse] nocheck constraint [FK_StudyPlan_PeriodCourse_CourseVersion]
GO

alter table [dbo].[StudyPlan_PeriodCourse] nocheck constraint [FK_StudyPlan_PeriodCourse_StudyPlan_Period]
GO

alter table [dbo].[StudyPlan_PeriodCourse] nocheck constraint [FK_StudyPlan_PeriodCourse_StudyPlan_Period1]
GO

ALTER TABLE [dbo].[StudyTypeVersion] ADD
    CONSTRAINT [FK_StudyTypeVersion_Point] FOREIGN KEY
    (
        [Point_ID]
    ) REFERENCES [dbo].[Point] (
        [Point_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_StudyTypeVersion_StudyType] FOREIGN KEY
    (
        [StudyType_ID]
    ) REFERENCES [dbo].[StudyType] (
        [StudyType_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[StudyTypeVersion] nocheck constraint [FK_StudyTypeVersion_Point]
GO

alter table [dbo].[StudyTypeVersion] nocheck constraint [FK_StudyTypeVersion_StudyType]
GO

ALTER TABLE [dbo].[StudyType_ProjectType] ADD
    CONSTRAINT [FK_StudyType_ProjectType_Point] FOREIGN KEY
    (
        [Point_ID]
    ) REFERENCES [dbo].[Point] (
        [Point_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_StudyType_ProjectType_ProjectType] FOREIGN KEY
    (
        [ProjectType_ID]
    ) REFERENCES [dbo].[ProjectType] (
        [ProjectType_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_StudyType_ProjectType_StudyTypeVersion] FOREIGN KEY
    (
        [StudyTypeVersion_ID]
    ) REFERENCES [dbo].[StudyTypeVersion] (
        [StudyTypeVersion_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[StudyType_ProjectType] nocheck constraint [FK_StudyType_ProjectType_Point]
GO

alter table [dbo].[StudyType_ProjectType] nocheck constraint [FK_StudyType_ProjectType_ProjectType]
GO

alter table [dbo].[StudyType_ProjectType] nocheck constraint [FK_StudyType_ProjectType_StudyTypeVersion]
GO

ALTER TABLE [dbo].[StudyType_TechnicalLine] ADD
    CONSTRAINT [FK_StudyType_TechnicalLine_StudyTypeVersion] FOREIGN KEY
    (
        [StudyTypeVersion_ID]
    ) REFERENCES [dbo].[StudyTypeVersion] (
        [StudyTypeVersion_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_StudyType_TechnicalLine_TechnicalLine] FOREIGN KEY
    (
        [TechnicalLine_ID]
    ) REFERENCES [dbo].[TechnicalLine] (
        [TechnicalLine_ID]
    ) NOT FOR REPLICATION
```

```
GO

alter table [dbo].[StudyType_TechnicalLine] nocheck constraint [FK_StudyType_TechnicalLine_StudyTypeVersion]
GO

alter table [dbo].[StudyType_TechnicalLine] nocheck constraint [FK_StudyType_TechnicalLine_TechnicalLine]
GO

ALTER TABLE [dbo].[StudyType_TechnicalPackage] ADD
    CONSTRAINT [FK_StudyType_TechnicalPackage_StudyTypeVersion] FOREIGN KEY
    (
        [StudyTypeVersion_ID]
    ) REFERENCES [dbo].[StudyTypeVersion] (
        [StudyTypeVersion_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_StudyType_TechnicalPackage_TechnicalPackage] FOREIGN KEY
    (
        [TechnicalPackage_ID]
    ) REFERENCES [dbo].[TechnicalPackage] (
        [TechnicalPackage_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[StudyType_TechnicalPackage] nocheck constraint [FK_StudyType_TechnicalPackage_StudyTypeVersion]
GO

alter table [dbo].[StudyType_TechnicalPackage] nocheck constraint [FK_StudyType_TechnicalPackage_TechnicalPackage]
GO

ALTER TABLE [dbo].[Table_StoredProcedure] ADD
    CONSTRAINT [FK_Table_StoredProcedure_Table] FOREIGN KEY
    (
        [Table_ID]
    ) REFERENCES [dbo].[Table] (
        [Table_ID]
    ) NOT FOR REPLICATION
GO

ALTER TABLE [dbo].[TechnicalFieldVersion] ADD
    CONSTRAINT [FK_TechnicalFieldVersion_TechnicalField] FOREIGN KEY
    (
        [TechnicalField_ID]
    ) REFERENCES [dbo].[TechnicalField] (
        [TechnicalField_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[TechnicalFieldVersion] nocheck constraint [FK_TechnicalFieldVersion_TechnicalField]
GO

ALTER TABLE [dbo].[TechnicalField_Course] ADD
    CONSTRAINT [FK_TechnicalField_Course_Course] FOREIGN KEY
    (
        [Course_ID]
    ) REFERENCES [dbo].[Course] (
        [Course_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_TechnicalField_Course_TechnicalFieldVersion] FOREIGN KEY
    (
        [TechnicalFieldVersion_ID]
    ) REFERENCES [dbo].[TechnicalFieldVersion] (
        [TechnicalFieldVersion_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[TechnicalField_Course] nocheck constraint [FK_TechnicalField_Course_Course]
GO

alter table [dbo].[TechnicalField_Course] nocheck constraint [FK_TechnicalField_Course_TechnicalFieldVersion]
GO

ALTER TABLE [dbo].[TechnicalLineVersion] ADD
    CONSTRAINT [FK_TechnicalLineVersion_Point] FOREIGN KEY
    (
        [Point_ID]
    ) REFERENCES [dbo].[Point] (
        [Point_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_TechnicalLineVersion_TechnicalLine] FOREIGN KEY
    (
        [TechnicalLine_ID]
    ) REFERENCES [dbo].[TechnicalLine] (
        [TechnicalLine_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[TechnicalLineVersion] nocheck constraint [FK_TechnicalLineVersion_Point]
GO

alter table [dbo].[TechnicalLineVersion] nocheck constraint [FK_TechnicalLineVersion_TechnicalLine]
GO

ALTER TABLE [dbo].[TechnicalLine_PrerequisiteCourse] ADD
    CONSTRAINT [FK_TechnicalLine_PrerequisiteCourse_Course] FOREIGN KEY
    (
        [Course_ID]
    ) REFERENCES [dbo].[Course] (
        [Course_ID]
    ) NOT FOR REPLICATION ,
```

```
    CONSTRAINT [FK_TechnicalLine_PrerequisiteCourse_TechnicalLineVersion] FOREIGN KEY
    (
        [TechnicalLineVersion_ID]
    ) REFERENCES [dbo].[TechnicalLineVersion] (
        [TechnicalLineVersion_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[TechnicalLine_PrerequisiteCourse] nocheck constraint [FK_TechnicalLine_PrerequisiteCourse_Course]
GO

alter table [dbo].[TechnicalLine_PrerequisiteCourse] nocheck constraint [FK_TechnicalLine_PrerequisiteCourse_TechnicalLineVersion]
GO

ALTER TABLE [dbo].[TechnicalLine_PrerequisiteTechnicalPackage] ADD
    CONSTRAINT [FK_TechnicalLine_PrerequisiteTechnicalPackage_TechnicalLineVersion] FOREIGN KEY
    (
        [TechnicalLineVersion_ID]
    ) REFERENCES [dbo].[TechnicalLineVersion] (
        [TechnicalLineVersion_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_TechnicalLine_PrerequisiteTechnicalPackage_TechnicalPackage] FOREIGN KEY
    (
        [TechnicalPackage_ID]
    ) REFERENCES [dbo].[TechnicalPackage] (
        [TechnicalPackage_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[TechnicalLine_PrerequisiteTechnicalPackage] nocheck constraint [
  FK_TechnicalLine_PrerequisiteTechnicalPackage_TechnicalLineVersion]
GO

alter table [dbo].[TechnicalLine_PrerequisiteTechnicalPackage] nocheck constraint [
  FK_TechnicalLine_PrerequisiteTechnicalPackage_TechnicalPackage]
GO

ALTER TABLE [dbo].[TechnicalLine_PrerequisiteTechnicalPackageCourse] ADD
    CONSTRAINT [FK_TechnicalLine_PrerequisiteTechnicalPackageCourse_TechnicalLine_PrerequisiteTechnicalPackage]
      FOREIGN KEY
    (
        [TechnicalLine_PrerequisiteTechnicalPackage_ID]
    ) REFERENCES [dbo].[TechnicalLine_PrerequisiteTechnicalPackage] (
        [TechnicalLine_PrerequisiteTechnicalPackage_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[TechnicalLine_PrerequisiteTechnicalPackageCourse] nocheck constraint [
  FK_TechnicalLine_PrerequisiteTechnicalPackageCourse_TechnicalLine_PrerequisiteTechnicalPackage]
GO

ALTER TABLE [dbo].[TechnicalLine_Specialization] ADD
    CONSTRAINT [FK_TechnicalLine_Specialization_Specialization] FOREIGN KEY
    (
        [Specialization_ID]
    ) REFERENCES [dbo].[Specialization] (
        [Specialization_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_TechnicalLine_Specialization_TechnicalLineVersion] FOREIGN KEY
    (
        [TechnicalLineVersion_ID]
    ) REFERENCES [dbo].[TechnicalLineVersion] (
        [TechnicalLineVersion_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[TechnicalLine_Specialization] nocheck constraint [FK_TechnicalLine_Specialization_Specialization]
GO

alter table [dbo].[TechnicalLine_Specialization] nocheck constraint [FK_TechnicalLine_Specialization_TechnicalLineVersion]
GO

ALTER TABLE [dbo].[TechnicalLine_TechnicalField] ADD
    CONSTRAINT [FK_TechnicalLine_TechnicalField_TechnicalField] FOREIGN KEY
    (
        [TechnicalField_ID]
    ) REFERENCES [dbo].[TechnicalField] (
        [TechnicalField_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_TechnicalLine_TechnicalField_TechnicalLineVersion] FOREIGN KEY
    (
        [TechnicalLineVersion_ID]
    ) REFERENCES [dbo].[TechnicalLineVersion] (
        [TechnicalLineVersion_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[TechnicalLine_TechnicalField] nocheck constraint [FK_TechnicalLine_TechnicalField_TechnicalField]
GO

alter table [dbo].[TechnicalLine_TechnicalField] nocheck constraint [FK_TechnicalLine_TechnicalField_TechnicalLineVersion]
GO

ALTER TABLE [dbo].[TechnicalPackageVersion] ADD
    CONSTRAINT [FK_TechnicalPackageVersion_TechnicalPackage] FOREIGN KEY
    (
        [TechnicalPackage_ID]
    ) REFERENCES [dbo].[TechnicalPackage] (
        [TechnicalPackage_ID]
```

```
    ) NOT FOR REPLICATION
GO

alter table [dbo].[TechnicalPackageVersion] nocheck constraint [FK_TechnicalPackageVersion_TechnicalPackage]
GO

ALTER TABLE [dbo].[TechnicalPackage_FundamentalCourse] ADD
    CONSTRAINT [FK_TechnicalPackage_FundamentalCourse_TechnicalPackageVersion] FOREIGN KEY
    (
        [TechnicalPackageVersion_ID]
    ) REFERENCES [dbo].[TechnicalPackageVersion] (
        [TechnicalPackageVersion_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[TechnicalPackage_FundamentalCourse] nocheck constraint [
  FK_TechnicalPackage_FundamentalCourse_TechnicalPackageVersion]
GO

ALTER TABLE [dbo].[TechnicalPackage_FundamentalCourseItem] ADD
    CONSTRAINT [FK_TechnicalPackage_FundamentalCourse_TechnicalPackage_FundamentalCourseItem] FOREIGN KEY
    (
        [TechnicalPackage_FundamentalCourse_ID]
    ) REFERENCES [dbo].[TechnicalPackage_FundamentalCourse] (
        [TechnicalPackage_FundamentalCourse_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_TechnicalPackage_FundamentalCourseItem_Course] FOREIGN KEY
    (
        [Course_ID]
    ) REFERENCES [dbo].[Course] (
        [Course_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[TechnicalPackage_FundamentalCourseItem] nocheck constraint [
  FK_TechnicalPackage_FundamentalCourse_TechnicalPackage_FundamentalCourseItem]
GO

alter table [dbo].[TechnicalPackage_FundamentalCourseItem] nocheck constraint [
  FK_TechnicalPackage_FundamentalCourseItem_Course]
GO

ALTER TABLE [dbo].[TechnicalPackage_Period] ADD
    CONSTRAINT [FK_TechnicalPackage_Period_Period] FOREIGN KEY
    (
        [Period_ID]
    ) REFERENCES [dbo].[Period] (
        [Period_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_TechnicalPackage_Period_TechnicalPackageVersion] FOREIGN KEY
    (
        [TechnicalPackageVersion_ID]
    ) REFERENCES [dbo].[TechnicalPackageVersion] (
        [TechnicalPackageVersion_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[TechnicalPackage_Period] nocheck constraint [FK_TechnicalPackage_Period_Period]
GO

alter table [dbo].[TechnicalPackage_Period] nocheck constraint [FK_TechnicalPackage_Period_TechnicalPackageVersion]
GO

ALTER TABLE [dbo].[TechnicalPackage_PeriodCourse] ADD
    CONSTRAINT [FK_TechnicalPackage_PeriodCourse_TechnicalPackage_Period] FOREIGN KEY
    (
        [TechnicalPackage_Period_ID]
    ) REFERENCES [dbo].[TechnicalPackage_Period] (
        [TechnicalPackage_Period_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[TechnicalPackage_PeriodCourse] nocheck constraint [FK_TechnicalPackage_PeriodCourse_TechnicalPackage_Period]
GO

ALTER TABLE [dbo].[TechnicalPackage_PeriodCourseItem] ADD
    CONSTRAINT [FK_TechnicalPackage_PeriodCourseItem_Course] FOREIGN KEY
    (
        [Course_ID]
    ) REFERENCES [dbo].[Course] (
        [Course_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_TechnicalPackage_PeriodCourseItem_TechnicalPackage_PeriodCourse] FOREIGN KEY
    (
        [TechnicalPackage_PeriodCourse_ID]
    ) REFERENCES [dbo].[TechnicalPackage_PeriodCourse] (
        [TechnicalPackage_PeriodCourse_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[TechnicalPackage_PeriodCourseItem] nocheck constraint [FK_TechnicalPackage_PeriodCourseItem_Course]
GO

alter table [dbo].[TechnicalPackage_PeriodCourseItem] nocheck constraint [
  FK_TechnicalPackage_PeriodCourseItem_TechnicalPackage_PeriodCourse]
GO

ALTER TABLE [dbo].[TechnicalPackage_PeriodOptionalCourse] ADD
    CONSTRAINT [FK_TechnicalPackage_PeriodOptionalCourse_Course] FOREIGN KEY
```

```
    (
        [Course_ID]
    ) REFERENCES [dbo].[Course] (
        [Course_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_TechnicalPackage_PeriodOptionalCourse_TechnicalPackage_Period] FOREIGN KEY
    (
        [TechnicalPackage_Period_ID]
    ) REFERENCES [dbo].[TechnicalPackage_Period] (
        [TechnicalPackage_Period_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[TechnicalPackage_PeriodOptionalCourse] nocheck constraint [FK_TechnicalPackage_PeriodOptionalCourse_Course]
GO

alter table [dbo].[TechnicalPackage_PeriodOptionalCourse] nocheck constraint [
  FK_TechnicalPackage_PeriodOptionalCourse_TechnicalPackage_Period]
GO

ALTER TABLE [dbo].[TechnicalPackage_Project] ADD
    CONSTRAINT [FK_TechnicalPackage_Project_Period] FOREIGN KEY
    (
        [Period_ID]
    ) REFERENCES [dbo].[Period] (
        [Period_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_TechnicalPackage_Project_Project] FOREIGN KEY
    (
        [Project_ID]
    ) REFERENCES [dbo].[Project] (
        [Project_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_TechnicalPackage_Project_TechnicalPackageVersion] FOREIGN KEY
    (
        [TechnicalPackageVersion_ID]
    ) REFERENCES [dbo].[TechnicalPackageVersion] (
        [TechnicalPackageVersion_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[TechnicalPackage_Project] nocheck constraint [FK_TechnicalPackage_Project_Period]
GO

alter table [dbo].[TechnicalPackage_Project] nocheck constraint [FK_TechnicalPackage_Project_Project]
GO

alter table [dbo].[TechnicalPackage_Project] nocheck constraint [FK_TechnicalPackage_Project_TechnicalPackageVersion]
GO

ALTER TABLE [dbo].[Text] ADD
    CONSTRAINT [FK_Text_TextGroup] FOREIGN KEY
    (
        [TextGroup_ID]
    ) REFERENCES [dbo].[TextGroup] (
        [TextGroup_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[Text] nocheck constraint [FK_Text_TextGroup]
GO

ALTER TABLE [dbo].[TextGroup] ADD
    CONSTRAINT [FK_TextGroup_TextGroup] FOREIGN KEY
    (
        [ParentGroup_ID]
    ) REFERENCES [dbo].[TextGroup] (
        [TextGroup_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[TextGroup] nocheck constraint [FK_TextGroup_TextGroup]
GO

ALTER TABLE [dbo].[User_Login] ADD
    CONSTRAINT [FK_User_Login_User] FOREIGN KEY
    (
        [User_ID]
    ) REFERENCES [dbo].[User] (
        [User_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_User_Login_User_LoginType] FOREIGN KEY
    (
        [User_LoginType_ID]
    ) REFERENCES [dbo].[User_LoginType] (
        [User_LoginType_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[User_Login] nocheck constraint [FK_User_Login_User]
GO

alter table [dbo].[User_Login] nocheck constraint [FK_User_Login_User_LoginType]
GO

ALTER TABLE [dbo].[User_PasswordHistory] ADD
    CONSTRAINT [FK_User_PasswordHistory_User] FOREIGN KEY
    (
        [User_ID]
```

```
    ) REFERENCES [dbo].[User] (
        [User_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[User_PasswordHistory] nocheck constraint [FK_User_PasswordHistory_User]
GO

ALTER TABLE [dbo].[User_SecurityRole] ADD
    CONSTRAINT [FK_User_SecurityRole_SecurityRole] FOREIGN KEY
    (
        [SecurityRole_ID]
    ) REFERENCES [dbo].[SecurityRole] (
        [SecurityRole_ID]
    ) NOT FOR REPLICATION ,
    CONSTRAINT [FK_User_SecurityRole_User] FOREIGN KEY
    (
        [User_ID]
    ) REFERENCES [dbo].[User] (
        [User_ID]
    ) NOT FOR REPLICATION
GO

alter table [dbo].[User_SecurityRole] nocheck constraint [FK_User_SecurityRole_SecurityRole]
GO

alter table [dbo].[User_SecurityRole] nocheck constraint [FK_User_SecurityRole_User]
GO
```

Chapter 2

Stored Procedures

2.1 Assessment

2.1.1 Assessment_Delete

```
CREATE PROCEDURE dbo.Assessment_Delete
(
  @Original_Assessment_ID uniqueidenti↓er,
  @Original_Passed bit,
  @Original_Grade_ID int,
  @Original_PointsCredited bit,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
DELETE FROM Assessment
WHERE  (Assessment_ID  = @Original_Assessment_ID)
AND  (Passed  = @Original_Passed)
AND  (Grade_ID  = @Original_Grade_ID)
AND  (PointsCredited = @Original_PointsCredited)
AND  (Created  = @Original_Created)
AND  (CreatedBy  = @Original_CreatedBy)
AND  (Updated  = @Original_Updated)
AND  (UpdatedBy  = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50201', 14, 1);
```

2.1.2 Assessment_Insert

```
CREATE PROCEDURE dbo.Assessment_Insert
(
  @Assessment_ID uniqueidenti↓er,
  @Passed bit,
  @Grade_ID int,
  @PointsCredited bit,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
INSERT INTO Assessment
        (Assessment_ID,
  Passed,
  Grade_ID,
  PointsCredited,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy)
VALUES (@Assessment_ID,
  @Passed,
  @Grade_ID,
  @PointsCredited,
  @Created,
  @CreatedBy,
  @Updated,
  @UpdatedBy);
```

2.1.3 Assessment_Select

```
CREATE PROCEDURE dbo.Assessment_Select
(
  @Assessment_ID uniqueidenti↓er
)
AS
SET NOCOUNT ON;
SELECT  Assessment_ID,
  Passed,
  Grade_ID,
  PointsCredited,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM  Assessment
WHERE  Assessment_ID = @Assessment_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.1.4 Assessment_Update

```
CREATE PROCEDURE dbo.Assessment_Update
(
  @Assessment_ID uniqueidenti↓er,
  @Passed bit,
  @Grade_ID int,
  @PointsCredited bit,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er,
  @Original_Assessment_ID uniqueidenti↓er,
  @Original_Passed bit,
  @Original_Grade_ID int,
  @Original_PointsCredited bit,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
UPDATE Assessment
SET Assessment_ID = @Assessment_ID,
  Passed = @Passed,
  Grade_ID = @Grade_ID,
  PointsCredited = @PointsCredited,
      Created = @Created,
  CreatedBy = @CreatedBy,
  Updated = @Updated,
  UpdatedBy = @UpdatedBy
WHERE (Assessment_ID = @Original_Assessment_ID)
AND (Passed = @Original_Passed)
AND (Grade_ID = @Original_Grade_ID)
AND (PointsCredited = @Original_PointsCredited)
AND (Created = @Original_Created)
AND (CreatedBy = @Original_CreatedBy)
AND (Updated = @Original_Updated)
AND (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50101', 14, 1);
```

2.2 AssessmentType

2.2.1 AssessmentType_Select

```
CREATE PROCEDURE dbo.AssessmentType_Select
(
  @AssessmentType_ID int
)
AS
SET NOCOUNT ON;
SELECT
  AssessmentType_ID,
  Name,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM AssessmentType
WHERE AssessmentType_ID = @AssessmentType_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.3 ContactData

2.3.1 ContactData_Delete

```
CREATE PROCEDURE dbo.ContactData_Delete
(
  @Original_ContactData_ID uniqueidenti↓er,
  @Original_ContactDataType_ID int,
  @Original_Email varchar(100),
  @Original_Mobile varchar(30),
  @Original_Phone varchar(30),
  @Original_Facsimile varchar(30),
  @Original_Homepage varchar(100),
  @Original_Address1 varchar(100),
  @Original_Address2 varchar(100),
  @Original_PostalCode varchar(10),
  @Original_City varchar(50),
  @Original_Building varchar(10),
  @Original_Room varchar(5),
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
```

```
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
DELETE FROM ContactData
WHERE
  (ContactData_ID = @Original_ContactData_ID) AND
  (ContactDataType_ID = @Original_ContactDataType_ID) AND
  (Email = @Original_Email) AND
  (Mobile = @Original_Mobile) AND
  (Phone = @Original_Phone) AND
  (Facsimile = @Original_Facsimile) AND
  (Homepage = @Original_Homepage) AND
  (Address1 = @Original_Address1) AND
  (Address2 = @Original_Address2) AND
  (PostalCode = @Original_PostalCode) AND
  (City = @Original_City) AND
  (Building = @Original_Building) AND
  (Room = @Original_Room) AND
  (Created = @Original_Created) AND
  (CreatedBy = @Original_CreatedBy) AND
  (Updated = @Original_Updated) AND
  (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50201', 14, 1);
```

2.3.2 ContactData_Insert

```
CREATE PROCEDURE dbo.ContactData_Insert
(
  @ContactData_ID uniqueidenti↓er,
  @ContactDataType_ID int,
  @Email varchar(100),
  @Mobile varchar(30),
  @Phone varchar(30),
  @Facsimile varchar(30),
  @Homepage varchar(100),
  @Address1 varchar(100),
  @Address2 varchar(100),
  @PostalCode varchar(10),
  @City varchar(50),
  @Building varchar(10),
  @Room varchar(5),
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
INSERT INTO ContactData (
ContactData_ID,
ContactDataType_ID,
Email,
Mobile,
Phone,
Facsimile,
Homepage,
Address1,
Address2,
PostalCode,
City,
Building,
Room,
Created,
CreatedBy,
Updated,
UpdatedBy
)
VALUES (
@ContactData_ID,
@ContactDataType_ID,
@Email,
@Mobile,
@Phone,
@Facsimile,
@Homepage,
@Address1,
@Address2,
@PostalCode,
@City,
@Building,
@Room,
@Created,
@CreatedBy,
@Updated,
@UpdatedBy
);
```

2.3.3 ContactData_Select

```
CREATE PROCEDURE dbo.ContactData_Select
```

```
(
  @ContactData_ID uniqueidenti↓er
)
AS
SET NOCOUNT ON;
SELECT
  ContactData_ID,
  ContactDataType_ID,
  Email,
  Mobile,
  Phone,
  Facsimile,
  Homepage,
  Address1,
  Address2,
  PostalCode,
  City,
  Building,
  Room,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM ContactData
WHERE ContactData_ID = @ContactData_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.3.4 ContactData_Update

```
CREATE PROCEDURE dbo.ContactData_Update
(
  @ContactData_ID uniqueidenti↓er,
  @ContactDataType_ID int,
  @Email varchar(100),
  @Mobile varchar(30),
  @Phone varchar(30),
  @Facsimile varchar(30),
  @Homepage varchar(100),
  @Address1 varchar(100),
  @Address2 varchar(100),
  @PostalCode varchar(10),
  @City varchar(50),
  @Building varchar(10),
  @Room varchar(5),
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er,
  @Original_ContactData_ID uniqueidenti↓er,
  @Original_ContactDataType_ID int,
  @Original_Email varchar(100),
  @Original_Mobile varchar(30),
  @Original_Phone varchar(30),
  @Original_Facsimile varchar(30),
  @Original_Homepage varchar(100),
  @Original_Address1 varchar(100),
  @Original_Address2 varchar(100),
  @Original_PostalCode varchar(10),
  @Original_City varchar(50),
  @Original_Building varchar(10),
  @Original_Room varchar(5),
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
UPDATE ContactData
SET
  ContactData_ID = @ContactData_ID,
  ContactDataType_ID = @ContactDataType_ID,
  Email = @Email,
  Mobile = @Mobile,
  Phone = @Phone,
  Facsimile = @Facsimile,
  Homepage = @Homepage,
  Address1 = @Address1,
  Address2 = @Address2,
  PostalCode = @PostalCode,
  City = @City,
  Building = @Building,
  Room = @Room,
  Created = @Created,
  CreatedBy = @CreatedBy,
  Updated = @Updated,
  UpdatedBy = @UpdatedBy
WHERE
  (ContactData_ID = @Original_ContactData_ID) AND
  (ContactDataType_ID = @Original_ContactDataType_ID) AND
  (Email = @Original_Email) AND
  (Mobile = @Original_Mobile) AND
  (Phone = @Original_Phone) AND
  (Facsimile = @Original_Facsimile) AND
```

```
  (Homepage = @Original_Homepage) AND
  (Address1 = @Original_Address1) AND
  (Address2 = @Original_Address2) AND
  (PostalCode = @Original_PostalCode) AND
  (City = @Original_City) AND
  (Building = @Original_Building) AND
  (Room = @Original_Room) AND
  (Created = @Original_Created) AND
  (CreatedBy = @Original_CreatedBy) AND
  (Updated = @Original_Updated) AND
  (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50101', 14, 1);
```

2.4 ContactDataType

2.4.1 ContactDataType_Select

```
CREATE PROCEDURE dbo.ContactDataType_Select
(
  @ContactDataType_ID int
)
AS
SET NOCOUNT ON;
SELECT
  ContactDataType_ID,
  Name,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM ContactDataType
WHERE ContactDataType_ID = @ContactDataType_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.5 Course

2.5.1 Course_Delete

```
CREATE PROCEDURE dbo.Course_Delete
(
  @Original_Course_ID uniqueidenti↓er,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
DELETE FROM Course
WHERE  (Course_ID = @Original_Course_ID)
AND  (Created = @Original_Created)
AND  (CreatedBy = @Original_CreatedBy)
AND  (Updated = @Original_Updated)
AND  (UpdatedBy = @Original_UpdatedBy)
```

2.5.2 Course_Insert

```
CREATE PROCEDURE dbo.Course_Insert
(
  @Course_ID uniqueidenti↓er,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
INSERT INTO Course
  (Course_ID,
   Created,
   CreatedBy,
   Updated,
   UpdatedBy)
VALUES (@Course_ID,
   @Created,
   @CreatedBy,
   @Updated,
   @UpdatedBy);
```

2.5.3 Course_Select

```
CREATE PROCEDURE dbo.Course_Select
(
  @Course_ID uniqueidenti↓er,
  @Created datetime OUTPUT,
  @CreatedBy uniqueidenti↓er OUTPUT,
  @Updated datetime OUTPUT,
  @UpdatedBy uniqueidenti↓er OUTPUT
)
AS
  SET NOCOUNT ON;
SELECT @Created=Created, @CreatedBy=CreatedBy, @Updated=Updated, @UpdatedBy=UpdatedBy
FROM Course
WHERE Course_ID = @Course_ID
```

2.5.4 Course_Update

```
CREATE PROCEDURE dbo.Course_Update
(
  @Course_ID uniqueidenti↓er,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er,
  @Original_Course_ID uniqueidenti↓er,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
UPDATE  Course
SET  Course_ID = @Course_ID,
  Created = @Created,
  CreatedBy = @CreatedBy,
  Updated = @Updated,
  UpdatedBy = @UpdatedBy
WHERE  (Course_ID = @Original_Course_ID)
AND  (Created = @Original_Created)
AND  (CreatedBy = @Original_CreatedBy)
AND  (Updated = @Original_Updated)
AND  (UpdatedBy = @Original_UpdatedBy);
```

2.6 Course_Department

2.6.1 Course_Department_Delete

```
CREATE PROCEDURE dbo.Course_Department_Delete
(
  @Original_Course_Department_ID uniqueidenti↓er,
  @Original_Course_ID uniqueidenti↓er,
  @Original_Department_ID int,
  @Original_Responsible bit,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
DELETE FROM Course_Department
WHERE
  (Course_Department_ID = @Original_Course_Department_ID) AND
  (Course_ID = @Original_Course_ID) AND
  (Department_ID = @Original_Department_ID) AND
  (Responsible = @Original_Responsible) AND
  (Created = @Original_Created) AND
  (CreatedBy = @Original_CreatedBy) AND
  (Updated = @Original_Updated) AND
  (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50201', 14, 1);
```

2.6.2 Course_Department_GetDepartments

```
CREATE PROCEDURE dbo.Course_Department_GetDepartments
(
  @Course_ID uniqueidenti↓er
)
AS
SET NOCOUNT ON;
```

```
SELECT
  Department_ID
FROM Course_Department
WHERE Course_ID = @Course_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.6.3 Course_Department_Insert

```
CREATE PROCEDURE dbo.Course_Department_Insert
(
  @Course_Department_ID uniqueidenti↓er,
  @Course_ID uniqueidenti↓er,
  @Department_ID int,
  @Responsible bit,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
INSERT INTO Course_Department (
Course_Department_ID,
Course_ID,
Department_ID,
Responsible,
Created,
CreatedBy,
Updated,
UpdatedBy
)
VALUES (
@Course_Department_ID,
@Course_ID,
@Department_ID,
@Responsible,
@Created,
@CreatedBy,
@Updated,
@UpdatedBy
);
```

2.6.4 Course_Department_Select

```
CREATE PROCEDURE dbo.Course_Department_Select
(
  @Course_Department_ID uniqueidenti↓er
)
AS
SET NOCOUNT ON;
SELECT
  Course_Department_ID,
  Course_ID,
  Department_ID,
  Responsible,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM Course_Department
WHERE Course_Department_ID = @Course_Department_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.6.5 Course_Department_Update

```
CREATE PROCEDURE dbo.Course_Department_Update
(
  @Course_Department_ID uniqueidenti↓er,
  @Course_ID uniqueidenti↓er,
  @Department_ID int,
  @Responsible bit,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er,
  @Original_Course_Department_ID uniqueidenti↓er,
  @Original_Course_ID uniqueidenti↓er,
  @Original_Department_ID int,
  @Original_Responsible bit,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
```

```
UPDATE Course_Department
SET
  Course_Department_ID = @Course_Department_ID,
  Course_ID = @Course_ID,
  Department_ID = @Department_ID,
  Responsible = @Responsible,
  Created = @Created,
  CreatedBy = @CreatedBy,
  Updated = @Updated,
  UpdatedBy = @UpdatedBy
WHERE
  (Course_Department_ID = @Original_Course_Department_ID) AND
  (Course_ID = @Original_Course_ID) AND
  (Department_ID = @Original_Department_ID) AND
  (Responsible = @Original_Responsible) AND
  (Created = @Original_Created) AND
  (CreatedBy = @Original_CreatedBy) AND
  (Updated = @Original_Updated) AND
  (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50101', 14, 1);
```

2.7 Course_EvaluationForm

2.7.1 Course_EvaluationForm_Delete

```
CREATE PROCEDURE dbo.Course_EvaluationForm_Delete
(
  @Original_Course_EvaluationForm_ID uniqueidenti↓er,
  @Original_CourseVersion_ID uniqueidenti↓er,
  @Original_EvaluationForm_ID int,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
DELETE FROM Course_EvaluationForm
WHERE
  (Course_EvaluationForm_ID = @Original_Course_EvaluationForm_ID) AND
  (CourseVersion_ID = @Original_CourseVersion_ID) AND
  (EvaluationForm_ID = @Original_EvaluationForm_ID) AND
  (Created = @Original_Created) AND
  (CreatedBy = @Original_CreatedBy) AND
  (Updated = @Original_Updated) AND
  (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50201', 14, 1);
```

2.7.2 Course_EvaluationForm_Insert

```
CREATE PROCEDURE dbo.Course_EvaluationForm_Insert
(
  @Course_EvaluationForm_ID uniqueidenti↓er,
  @CourseVersion_ID uniqueidenti↓er,
  @EvaluationForm_ID int,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
INSERT INTO Course_EvaluationForm (
Course_EvaluationForm_ID,
CourseVersion_ID,
EvaluationForm_ID,
Created,
CreatedBy,
Updated,
UpdatedBy
)
VALUES (
@Course_EvaluationForm_ID,
@CourseVersion_ID,
@EvaluationForm_ID,
@Created,
@CreatedBy,
@Updated,
@UpdatedBy
);
```

2.7.3 Course_EvaluationForm_Select

```
CREATE PROCEDURE dbo.Course_EvaluationForm_Select
(
  @Course_EvaluationForm_ID uniqueidenti↓er
)
AS
SET NOCOUNT ON;
SELECT
  Course_EvaluationForm_ID,
  CourseVersion_ID,
  EvaluationForm_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM Course_EvaluationForm
WHERE Course_EvaluationForm_ID = @Course_EvaluationForm_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.7.4 Course_EvaluationForm_Update

```
CREATE PROCEDURE dbo.Course_EvaluationForm_Update
(
  @Course_EvaluationForm_ID uniqueidenti↓er,
  @CourseVersion_ID uniqueidenti↓er,
  @EvaluationForm_ID int,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er,
  @Original_Course_EvaluationForm_ID uniqueidenti↓er,
  @Original_CourseVersion_ID uniqueidenti↓er,
  @Original_EvaluationForm_ID int,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
UPDATE Course_EvaluationForm
SET
  Course_EvaluationForm_ID = @Course_EvaluationForm_ID,
  CourseVersion_ID = @CourseVersion_ID,
  EvaluationForm_ID = @EvaluationForm_ID,
  Created = @Created,
  CreatedBy = @CreatedBy,
  Updated = @Updated,
  UpdatedBy = @UpdatedBy
WHERE
  (Course_EvaluationForm_ID = @Original_Course_EvaluationForm_ID) AND
  (CourseVersion_ID = @Original_CourseVersion_ID) AND
  (EvaluationForm_ID = @Original_EvaluationForm_ID) AND
  (Created = @Original_Created) AND
  (CreatedBy = @Original_CreatedBy) AND
  (Updated = @Original_Updated) AND
  (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50101', 14, 1);
```

2.8 Course_Keyword

2.8.1 Course_Keyword_Delete

```
CREATE PROCEDURE dbo.Course_Keyword_Delete
(
  @Original_Course_Keyword_ID uniqueidenti↓er,
  @Original_CourseVersion_ID uniqueidenti↓er,
  @Original_Keyword_ID uniqueidenti↓er,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
DELETE FROM Course_Keyword
WHERE
  (Course_Keyword_ID = @Original_Course_Keyword_ID) AND
  (CourseVersion_ID = @Original_CourseVersion_ID) AND
  (Keyword_ID = @Original_Keyword_ID) AND
  (Created = @Original_Created) AND
  (CreatedBy = @Original_CreatedBy) AND
  (Updated = @Original_Updated) AND
  (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50201', 14, 1);
```

2.8.2 Course_Keyword_GetCoursesFromKeyword

```
CREATE PROCEDURE dbo.Course_Keyword_GetCoursesFromKeyword
(
  @Keyword_ID uniqueidenti↓er
)
AS
SELECT
  CourseVersion_ID
FROM Course_Keyword
WHERE
  Keyword_ID = @Keyword_ID
```

2.8.3 Course_Keyword_Insert

```
CREATE PROCEDURE dbo.Course_Keyword_Insert
(
  @Course_Keyword_ID uniqueidenti↓er,
  @CourseVersion_ID uniqueidenti↓er,
  @Keyword_ID uniqueidenti↓er,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
INSERT INTO Course_Keyword (
Course_Keyword_ID,
CourseVersion_ID,
Keyword_ID,
Created,
CreatedBy,
Updated,
UpdatedBy
)
VALUES (
@Course_Keyword_ID,
@CourseVersion_ID,
@Keyword_ID,
@Created,
@CreatedBy,
@Updated,
@UpdatedBy
);
```

2.8.4 Course_Keyword_Select

```
CREATE PROCEDURE dbo.Course_Keyword_Select
(
  @Course_Keyword_ID uniqueidenti↓er
)
AS
SET NOCOUNT ON;
SELECT
  Course_Keyword_ID,
  CourseVersion_ID,
  Keyword_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM Course_Keyword
WHERE Course_Keyword_ID = @Course_Keyword_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.8.5 Course_Keyword_Update

```
CREATE PROCEDURE dbo.Course_Keyword_Update
(
  @Course_Keyword_ID uniqueidenti↓er,
  @CourseVersion_ID uniqueidenti↓er,
  @Keyword_ID uniqueidenti↓er,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er,
  @Original_Course_Keyword_ID uniqueidenti↓er,
  @Original_CourseVersion_ID uniqueidenti↓er,
  @Original_Keyword_ID uniqueidenti↓er,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
```

```
SET NOCOUNT OFF;
UPDATE Course_Keyword
SET
  Course_Keyword_ID = @Course_Keyword_ID,
  CourseVersion_ID = @CourseVersion_ID,
  Keyword_ID = @Keyword_ID,
  Created = @Created,
  CreatedBy = @CreatedBy,
  Updated = @Updated,
  UpdatedBy = @UpdatedBy
WHERE
  (Course_Keyword_ID = @Original_Course_Keyword_ID) AND
  (CourseVersion_ID = @Original_CourseVersion_ID) AND
  (Keyword_ID = @Original_Keyword_ID) AND
  (Created = @Original_Created) AND
  (CreatedBy = @Original_CreatedBy) AND
  (Updated = @Original_Updated) AND
  (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50101', 14, 1);
```

2.9 Course_Lecturer

2.9.1 Course_Lecturer_Delete

```
CREATE PROCEDURE dbo.Course_Lecturer_Delete
(
  @Original_Course_Lecturer_ID uniqueidenti↓er,
  @Original_CourseVersion_ID uniqueidenti↓er,
  @Original_Lecturer_ID uniqueidenti↓er,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
DELETE FROM Course_Lecturer
WHERE
  (Course_Lecturer_ID = @Original_Course_Lecturer_ID) AND
  (CourseVersion_ID = @Original_CourseVersion_ID) AND
  (Lecturer_ID = @Original_Lecturer_ID) AND
  (Created = @Original_Created) AND
  (CreatedBy = @Original_CreatedBy) AND
  (Updated = @Original_Updated) AND
  (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50201', 14, 1);
```

2.9.2 Course_Lecturer_GetLecturers

```
CREATE PROCEDURE dbo.Course_Lecturer_GetLecturers
(
  @CourseVersion_ID uniqueidenti↓er
)
AS
SET NOCOUNT ON;
SELECT
  Lecturer_ID
FROM Course_Lecturer
WHERE CourseVersion_ID = @CourseVersion_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.9.3 Course_Lecturer_Insert

```
CREATE PROCEDURE dbo.Course_Lecturer_Insert
(
  @Course_Lecturer_ID uniqueidenti↓er,
  @CourseVersion_ID uniqueidenti↓er,
  @Lecturer_ID uniqueidenti↓er,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
INSERT INTO Course_Lecturer (
Course_Lecturer_ID,
CourseVersion_ID,
Lecturer_ID,
Created,
CreatedBy,
Updated,
```

```
UpdatedBy
)
VALUES (
@Course_Lecturer_ID,
@CourseVersion_ID,
@Lecturer_ID,
@Created,
@CreatedBy,
@Updated,
@UpdatedBy
);
```

2.9.4 Course_Lecturer_Select

```
CREATE PROCEDURE dbo.Course_Lecturer_Select
(
  @Course_Lecturer_ID uniqueidenti↓er
)
AS
SET NOCOUNT ON;
SELECT
  Course_Lecturer_ID,
  CourseVersion_ID,
  Lecturer_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM Course_Lecturer
WHERE Course_Lecturer_ID = @Course_Lecturer_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.9.5 Course_Lecturer_Update

```
CREATE PROCEDURE dbo.Course_Lecturer_Update
(
  @Course_Lecturer_ID uniqueidenti↓er,
  @CourseVersion_ID uniqueidenti↓er,
  @Lecturer_ID uniqueidenti↓er,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er,
  @Original_Course_Lecturer_ID uniqueidenti↓er,
  @Original_CourseVersion_ID uniqueidenti↓er,
  @Original_Lecturer_ID uniqueidenti↓er,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
UPDATE Course_Lecturer
SET
  Course_Lecturer_ID = @Course_Lecturer_ID,
  CourseVersion_ID = @CourseVersion_ID,
  Lecturer_ID = @Lecturer_ID,
  Created = @Created,
  CreatedBy = @CreatedBy,
  Updated = @Updated,
  UpdatedBy = @UpdatedBy
WHERE
  (Course_Lecturer_ID = @Original_Course_Lecturer_ID) AND
  (CourseVersion_ID = @Original_CourseVersion_ID) AND
  (Lecturer_ID = @Original_Lecturer_ID) AND
  (Created = @Original_Created) AND
  (CreatedBy = @Original_CreatedBy) AND
  (Updated = @Original_Updated) AND
  (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50101', 14, 1);
```

2.10 Course_Period

2.10.1 Course_Period_Delete

```
CREATE PROCEDURE dbo.Course_Period_Delete
(
  @Original_Course_Period_ID uniqueidenti↓er,
  @Original_CourseVersion_ID uniqueidenti↓er,
  @Original_Period_ID uniqueidenti↓er,
  @Original_Part int,
  @Original_Created datetime,
```

```
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
DELETE FROM Course_Period
WHERE
  (Course_Period_ID = @Original_Course_Period_ID) AND
  (CourseVersion_ID = @Original_CourseVersion_ID) AND
  (Period_ID = @Original_Period_ID) AND
  (Part = @Original_Part) AND
  (Created = @Original_Created) AND
  (CreatedBy = @Original_CreatedBy) AND
  (Updated = @Original_Updated) AND
  (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50201', 14, 1);
```

2.10.2 Course_Period_GetID

```
CREATE PROCEDURE dbo.Course_Period_GetID
(
  @CourseVersion_ID uniqueidenti↓er,
  @Part int,
  @Period_ID uniqueidenti↓er
)
AS
SET NOCOUNT ON;
SELECT
  Course_Period_ID
FROM  Course_Period
WHERE  CourseVersion_ID = @CourseVersion_ID
AND Part = @Part
AND Period_ID = @Period_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.10.3 Course_Period_GetPeriods

```
CREATE PROCEDURE dbo.Course_Period_GetPeriods
(
  @CourseVersion_ID uniqueidenti↓er,
  @Part int
)
AS
SET NOCOUNT ON;
SELECT
  Period_ID
FROM  Course_Period
WHERE  CourseVersion_ID = @CourseVersion_ID
AND Part = @Part
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.10.4 Course_Period_Insert

```
CREATE PROCEDURE dbo.Course_Period_Insert
(
  @Course_Period_ID uniqueidenti↓er,
  @CourseVersion_ID uniqueidenti↓er,
  @Period_ID uniqueidenti↓er,
  @Part int,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
INSERT INTO Course_Period (
Course_Period_ID,
CourseVersion_ID,
Period_ID,
Part,
Created,
CreatedBy,
Updated,
UpdatedBy
)
VALUES (
@Course_Period_ID,
@CourseVersion_ID,
@Period_ID,
@Part,
@Created,
@CreatedBy,
@Updated,
@UpdatedBy
);
```

2.10.5 Course_Period_Select

```
CREATE PROCEDURE dbo.Course_Period_Select
(
  @Course_Period_ID uniqueidenti↓er
)
AS
SET NOCOUNT ON;
SELECT
  Course_Period_ID,
  CourseVersion_ID,
  Period_ID,
  Part,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM Course_Period
WHERE Course_Period_ID = @Course_Period_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.10.6 Course_Period_Update

```
CREATE PROCEDURE dbo.Course_Period_Update
(
  @Course_Period_ID uniqueidenti↓er,
  @CourseVersion_ID uniqueidenti↓er,
  @Period_ID uniqueidenti↓er,
  @Part int,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er,
  @Original_Course_Period_ID uniqueidenti↓er,
  @Original_CourseVersion_ID uniqueidenti↓er,
  @Original_Period_ID uniqueidenti↓er,
  @Original_Part int,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
UPDATE Course_Period
SET
  Course_Period_ID = @Course_Period_ID,
  CourseVersion_ID = @CourseVersion_ID,
  Period_ID = @Period_ID,
  Part = @Part,
  Created = @Created,
  CreatedBy = @CreatedBy,
  Updated = @Updated,
  UpdatedBy = @UpdatedBy
WHERE
  (Course_Period_ID = @Original_Course_Period_ID) AND
  (CourseVersion_ID = @Original_CourseVersion_ID) AND
  (Period_ID = @Original_Period_ID) AND
  (Part = @Original_Part) AND
  (Created = @Original_Created) AND
  (CreatedBy = @Original_CreatedBy) AND
  (Updated = @Original_Updated) AND
  (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50101', 14, 1);
```

2.11 Course_Period_Module

2.11.1 Course_Period_Module_Delete

```
CREATE PROCEDURE dbo.Course_Period_Module_Delete
(
  @Original_Course_Period_Module_ID uniqueidenti↓er,
  @Original_Course_Period_ID uniqueidenti↓er,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
DELETE FROM Course_Period_Module
WHERE
  (Course_Period_Module_ID = @Original_Course_Period_Module_ID) AND
  (Course_Period_ID = @Original_Course_Period_ID) AND
  (Created = @Original_Created) AND
```

```
  (CreatedBy = @Original_CreatedBy) AND
  (Updated = @Original_Updated) AND
  (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50201', 14, 1);
```

2.11.2 Course_Period_Module_GetID

```
CREATE PROCEDURE dbo.Course_Period_Module_GetID
(
  @Course_Period_ID uniqueidenti↓er
)
AS
SET NOCOUNT ON;
SELECT
  Course_Period_Module_ID
FROM  Course_Period_Module
WHERE  Course_Period_ID = @Course_Period_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.11.3 Course_Period_Module_Insert

```
CREATE PROCEDURE dbo.Course_Period_Module_Insert
(
  @Course_Period_Module_ID uniqueidenti↓er,
  @Course_Period_ID uniqueidenti↓er,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
INSERT INTO Course_Period_Module (
Course_Period_Module_ID,
Course_Period_ID,
Created,
CreatedBy,
Updated,
UpdatedBy
)
VALUES (
@Course_Period_Module_ID,
@Course_Period_ID,
@Created,
@CreatedBy,
@Updated,
@UpdatedBy
);
```

2.11.4 Course_Period_Module_Select

```
CREATE PROCEDURE dbo.Course_Period_Module_Select
(
  @Course_Period_Module_ID uniqueidenti↓er
)
AS
SET NOCOUNT ON;
SELECT
  Course_Period_Module_ID,
  Course_Period_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM Course_Period_Module
WHERE Course_Period_Module_ID = @Course_Period_Module_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.11.5 Course_Period_Module_Update

```
CREATE PROCEDURE dbo.Course_Period_Module_Update
(
  @Course_Period_Module_ID uniqueidenti↓er,
  @Course_Period_ID uniqueidenti↓er,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er,
  @Original_Course_Period_Module_ID uniqueidenti↓er,
  @Original_Course_Period_ID uniqueidenti↓er,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
```

```
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
UPDATE Course_Period_Module
SET
  Course_Period_Module_ID = @Course_Period_Module_ID,
  Course_Period_ID = @Course_Period_ID,
  Created = @Created,
  CreatedBy = @CreatedBy,
  Updated = @Updated,
  UpdatedBy = @UpdatedBy
WHERE
  (Course_Period_Module_ID = @Original_Course_Period_Module_ID) AND
  (Course_Period_ID = @Original_Course_Period_ID) AND
  (Created = @Original_Created) AND
  (CreatedBy = @Original_CreatedBy) AND
  (Updated = @Original_Updated) AND
  (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50101', 14, 1);
```

2.12 Course_Period_ModuleItem

2.12.1 Course_Period_ModuleItem_Delete

```
CREATE PROCEDURE dbo.Course_Period_ModuleItem_Delete
(
  @Original_Course_Period_ModuleItem_ID uniqueidenti↓er,
  @Original_Course_Period_Module_ID uniqueidenti↓er,
  @Original_Module_ID int,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
DELETE FROM Course_Period_ModuleItem
WHERE
  (Course_Period_ModuleItem_ID = @Original_Course_Period_ModuleItem_ID) AND
  (Course_Period_Module_ID = @Original_Course_Period_Module_ID) AND
  (Module_ID = @Original_Module_ID) AND
  (Created = @Original_Created) AND
  (CreatedBy = @Original_CreatedBy) AND
  (Updated = @Original_Updated) AND
  (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50201', 14, 1);
```

2.12.2 Course_Period_ModuleItem_GetID

```
CREATE PROCEDURE dbo.Course_Period_ModuleItem_GetID
(
  @Course_Period_Module_ID uniqueidenti↓er
)
AS
SET NOCOUNT ON;
SELECT
  Course_Period_ModuleItem_ID
FROM  Course_Period_ModuleItem
WHERE  Course_Period_Module_ID = @Course_Period_Module_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.12.3 Course_Period_ModuleItem_Insert

```
CREATE PROCEDURE dbo.Course_Period_ModuleItem_Insert
(
  @Course_Period_ModuleItem_ID uniqueidenti↓er,
  @Course_Period_Module_ID uniqueidenti↓er,
  @Module_ID int,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
INSERT INTO Course_Period_ModuleItem (
Course_Period_ModuleItem_ID,
Course_Period_Module_ID,
Module_ID,
Created,
```

```
CreatedBy,
Updated,
UpdatedBy
)
VALUES (
@Course_Period_ModuleItem_ID,
@Course_Period_Module_ID,
@Module_ID,
@Created,
@CreatedBy,
@Updated,
@UpdatedBy
);
```

2.12.4 Course_Period_ModuleItem_Select

```
CREATE PROCEDURE dbo.Course_Period_ModuleItem_Select
(
  @Course_Period_ModuleItem_ID uniqueidenti↓er
)
AS
SET NOCOUNT ON;
SELECT
  Course_Period_ModuleItem_ID,
  Course_Period_Module_ID,
  Module_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM Course_Period_ModuleItem
WHERE Course_Period_ModuleItem_ID = @Course_Period_ModuleItem_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.12.5 Course_Period_ModuleItem_Update

```
CREATE PROCEDURE dbo.Course_Period_ModuleItem_Update
(
  @Course_Period_ModuleItem_ID uniqueidenti↓er,
  @Course_Period_Module_ID uniqueidenti↓er,
  @Module_ID int,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er,
  @Original_Course_Period_ModuleItem_ID uniqueidenti↓er,
  @Original_Course_Period_Module_ID uniqueidenti↓er,
  @Original_Module_ID int,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
UPDATE Course_Period_ModuleItem
SET
  Course_Period_ModuleItem_ID = @Course_Period_ModuleItem_ID,
  Course_Period_Module_ID = @Course_Period_Module_ID,
  Module_ID = @Module_ID,
  Created = @Created,
  CreatedBy = @CreatedBy,
  Updated = @Updated,
  UpdatedBy = @UpdatedBy
WHERE
  (Course_Period_ModuleItem_ID = @Original_Course_Period_ModuleItem_ID) AND
  (Course_Period_Module_ID = @Original_Course_Period_Module_ID) AND
  (Module_ID = @Original_Module_ID) AND
  (Created = @Original_Created) AND
  (CreatedBy = @Original_CreatedBy) AND
  (Updated = @Original_Updated) AND
  (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50101', 14, 1);
```

2.13 Course_Point

2.13.1 Course_Point_Delete

```
CREATE PROCEDURE dbo.Course_Point_Delete
(
  @Original_Course_Point_ID uniqueidenti↓er,
  @Original_CourseVersion_ID uniqueidenti↓er,
  @Original_Part int,
```

```
  @Original_Point_ID uniqueidenti↓er,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
DELETE FROM Course_Point
WHERE
  (Course_Point_ID = @Original_Course_Point_ID) AND
  (CourseVersion_ID = @Original_CourseVersion_ID) AND
  (Part = @Original_Part) AND
  (Point_ID = @Original_Point_ID) AND
  (Created = @Original_Created) AND
  (CreatedBy = @Original_CreatedBy) AND
  (Updated = @Original_Updated) AND
  (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50201', 14, 1);
```

2.13.2 Course_Point_GetPoint

```
CREATE PROCEDURE dbo.Course_Point_GetPoint
(
  @CourseVersion_ID uniqueidenti↓er,
  @Part int,
  @Point_ID uniqueidenti↓er OUTPUT
)
AS
SET NOCOUNT ON;
SELECT
  @Point_ID = Point_ID
FROM Course_Point
WHERE
  CourseVersion_ID = @CourseVersion_ID AND
  Part = @Part

IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.13.3 Course_Point_Insert

```
CREATE PROCEDURE dbo.Course_Point_Insert
(
  @Course_Point_ID uniqueidenti↓er,
  @CourseVersion_ID uniqueidenti↓er,
  @Part int,
  @Point_ID uniqueidenti↓er,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
INSERT INTO Course_Point (
Course_Point_ID,
CourseVersion_ID,
Part,
Point_ID,
Created,
CreatedBy,
Updated,
UpdatedBy
)
VALUES (
@Course_Point_ID,
@CourseVersion_ID,
@Part,
@Point_ID,
@Created,
@CreatedBy,
@Updated,
@UpdatedBy
);
```

2.13.4 Course_Point_Select

```
CREATE PROCEDURE dbo.Course_Point_Select
(
  @Course_Point_ID uniqueidenti↓er
)
AS
SET NOCOUNT ON;
SELECT
  Course_Point_ID,
  CourseVersion_ID,
  Part,
```

```
  Point_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM Course_Point
WHERE Course_Point_ID = @Course_Point_ID

IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.13.5 Course_Point_Update

```
CREATE PROCEDURE dbo.Course_Point_Update
(
  @Course_Point_ID uniqueidenti↓er,
  @CourseVersion_ID uniqueidenti↓er,
  @Part int,
  @Point_ID uniqueidenti↓er,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er,
  @Original_Course_Point_ID uniqueidenti↓er,
  @Original_CourseVersion_ID uniqueidenti↓er,
  @Original_Part int,
  @Original_Point_ID uniqueidenti↓er,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
UPDATE Course_Point
SET
  Course_Point_ID = @Course_Point_ID,
  CourseVersion_ID = @CourseVersion_ID,
  Part = @Part,
  Point_ID = @Point_ID,
  Created = @Created,
  CreatedBy = @CreatedBy,
  Updated = @Updated,
  UpdatedBy = @UpdatedBy
WHERE
  (Course_Point_ID = @Original_Course_Point_ID) AND
  (CourseVersion_ID = @Original_CourseVersion_ID) AND
  (Part = @Original_Part) AND
  (Point_ID = @Original_Point_ID) AND
  (Created = @Original_Created) AND
  (CreatedBy = @Original_CreatedBy) AND
  (Updated = @Original_Updated) AND
  (UpdatedBy = @Original_UpdatedBy)

IF @@ROWCOUNT = 0
  RAISERROR('50101', 14, 1);
```

2.14 Course_RecommendedPlacement

2.14.1 Course_RecommendedPlacement_Delete

```
CREATE PROCEDURE dbo.Course_RecommendedPlacement_Delete
(
  @Original_Course_RecommendedPlacement_ID uniqueidenti↓er,
  @Original_CourseVersion_ID uniqueidenti↓er,
  @Original_StudyType_ID int,
  @Original_Point_ID uniqueidenti↓er,
  @Original_RecommendedPlacementConcept_ID int,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
DELETE FROM Course_RecommendedPlacement
WHERE
  (Course_RecommendedPlacement_ID = @Original_Course_RecommendedPlacement_ID) AND
  (CourseVersion_ID = @Original_CourseVersion_ID) AND
  (StudyType_ID = @Original_StudyType_ID) AND
  (Point_ID = @Original_Point_ID) AND
  (RecommendedPlacementConcept_ID = @Original_RecommendedPlacementConcept_ID) AND
  (Created = @Original_Created) AND
  (CreatedBy = @Original_CreatedBy) AND
  (Updated = @Original_Updated) AND
  (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50201', 14, 1);
```

2.14.2 Course_RecommendedPlacement_GetID

```
CREATE PROCEDURE dbo.Course_RecommendedPlacement_GetID
(
  @CourseVersion_ID uniqueidenti¦er,
  @StudyType_ID int
)
AS
SET NOCOUNT ON;
SELECT
  Course_RecommendedPlacement_ID
FROM  Course_RecommendedPlacement
WHERE  CourseVersion_ID = @CourseVersion_ID
AND StudyType_ID = @StudyType_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.14.3 Course_RecommendedPlacement_Insert

```
CREATE PROCEDURE dbo.Course_RecommendedPlacement_Insert
(
  @Course_RecommendedPlacement_ID uniqueidenti¦er,
  @CourseVersion_ID uniqueidenti¦er,
  @StudyType_ID int,
  @Point_ID uniqueidenti¦er,
  @RecommendedPlacementConcept_ID int,
  @Created datetime,
  @CreatedBy uniqueidenti¦er,
  @Updated datetime,
  @UpdatedBy uniqueidenti¦er
)
AS
SET NOCOUNT OFF;
INSERT INTO Course_RecommendedPlacement (
Course_RecommendedPlacement_ID,
CourseVersion_ID,
StudyType_ID,
Point_ID,
RecommendedPlacementConcept_ID,
Created,
CreatedBy,
Updated,
UpdatedBy
)
VALUES (
@Course_RecommendedPlacement_ID,
@CourseVersion_ID,
@StudyType_ID,
@Point_ID,
@RecommendedPlacementConcept_ID,
@Created,
@CreatedBy,
@Updated,
@UpdatedBy
);
```

2.14.4 Course_RecommendedPlacement_Select

```
CREATE PROCEDURE dbo.Course_RecommendedPlacement_Select
(
  @Course_RecommendedPlacement_ID uniqueidenti¦er
)
AS
SET NOCOUNT ON;
SELECT
  Course_RecommendedPlacement_ID,
  CourseVersion_ID,
  StudyType_ID,
  Point_ID,
  RecommendedPlacementConcept_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM Course_RecommendedPlacement
WHERE Course_RecommendedPlacement_ID = @Course_RecommendedPlacement_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.14.5 Course_RecommendedPlacement_Update

```
CREATE PROCEDURE dbo.Course_RecommendedPlacement_Update
(
  @Course_RecommendedPlacement_ID uniqueidenti¦er,
  @CourseVersion_ID uniqueidenti¦er,
  @StudyType_ID int,
  @Point_ID uniqueidenti¦er,
  @RecommendedPlacementConcept_ID int,
```

```
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er,
  @Original_Course_RecommendedPlacement_ID uniqueidenti↓er,
  @Original_CourseVersion_ID uniqueidenti↓er,
  @Original_StudyType_ID int,
  @Original_Point_ID uniqueidenti↓er,
  @Original_RecommendedPlacementConcept_ID int,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
UPDATE Course_RecommendedPlacement
SET
  Course_RecommendedPlacement_ID = @Course_RecommendedPlacement_ID,
  CourseVersion_ID = @CourseVersion_ID,
  StudyType_ID = @StudyType_ID,
  Point_ID = @Point_ID,
  RecommendedPlacementConcept_ID = @RecommendedPlacementConcept_ID,
  Created = @Created,
  CreatedBy = @CreatedBy,
  Updated = @Updated,
  UpdatedBy = @UpdatedBy
WHERE
  (Course_RecommendedPlacement_ID = @Original_Course_RecommendedPlacement_ID) AND
  (CourseVersion_ID = @Original_CourseVersion_ID) AND
  (StudyType_ID = @Original_StudyType_ID) AND
  (Point_ID = @Original_Point_ID) AND
  (RecommendedPlacementConcept_ID = @Original_RecommendedPlacementConcept_ID) AND
  (Created = @Original_Created) AND
  (CreatedBy = @Original_CreatedBy) AND
  (Updated = @Original_Updated) AND
  (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50101', 14, 1);
```

2.15 Course_RelationCourse

2.15.1 Course_RelationCourse_GetIDFromVersion

```
CREATE PROCEDURE dbo.Course_RelationCourse_GetIDFromVersion
(
  @CourseVersion_ID uniqueidenti↓er,
  @Course_RelationCourseType_ID int
)
AS
SET NOCOUNT ON;
SELECT
  Course_RelationCourse_ID
FROM  Course_RelationCourse
WHERE  CourseVersion_ID = @CourseVersion_ID
AND Course_RelationCourseType_ID = @Course_RelationCourseType_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.15.2 Course_RelationCourse_Select

```
CREATE PROCEDURE dbo.Course_RelationCourse_Select
(
  @Course_RelationCourse_ID uniqueidenti↓er
)
AS
SET NOCOUNT ON;
SELECT
  Course_RelationCourse_ID,
  CourseVersion_ID,
  Course_RelationCourseType_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM Course_RelationCourse
WHERE Course_RelationCourse_ID = @Course_RelationCourse_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.16 Course_RelationCourseItem

2.16.1 Course_RelationCourseItem_GetCourses

```
CREATE PROCEDURE dbo.Course_RelationCourseItem_GetCourses
(
  @Course_RelationCourse_ID uniqueidenti↓er
)
AS
SET NOCOUNT ON;
SELECT
  Course_ID
FROM  Course_RelationCourseItem
WHERE  Course_RelationCourse_ID = @Course_RelationCourse_ID
AND Course_ID IS NOT NULL
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.16.2 Course_RelationCourseItem_Select

```
CREATE PROCEDURE dbo.Course_RelationCourseItem_Select
(
  @Course_RelationCourseItem_ID uniqueidenti↓er
)
AS
SET NOCOUNT ON;
SELECT
  Course_RelationCourseItem_ID,
  Course_RelationCourse_ID,
  Course_ID,
  CourseNumber,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM Course_RelationCourseItem
WHERE Course_RelationCourseItem_ID = @Course_RelationCourseItem_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.17 Course_StudyType

2.17.1 Course_StudyType_Delete

```
CREATE PROCEDURE dbo.Course_StudyType_Delete
(
  @Original_Course_StudyType_ID uniqueidenti↓er,
  @Original_CourseVersion_ID uniqueidenti↓er,
  @Original_StudyType_ID int,
  @Original_Course_StudyTypeCategory_ID int,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
DELETE FROM Course_StudyType
WHERE
  (Course_StudyType_ID = @Original_Course_StudyType_ID) AND
  (CourseVersion_ID = @Original_CourseVersion_ID) AND
  (StudyType_ID = @Original_StudyType_ID) AND
  (Course_StudyTypeCategory_ID = @Original_Course_StudyTypeCategory_ID) AND
  (Created = @Original_Created) AND
  (CreatedBy = @Original_CreatedBy) AND
  (Updated = @Original_Updated) AND
  (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50201', 14, 1);
```

2.17.2 Course_StudyType_GetStudyTypes

```
CREATE PROCEDURE dbo.Course_StudyType_GetStudyTypes
(
  @CourseVersion_ID uniqueidenti↓er
)
AS
SET NOCOUNT ON;
SELECT
  StudyType_ID
FROM  Course_StudyType
WHERE  CourseVersion_ID = @CourseVersion_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.17.3 Course_StudyType_Insert

```
CREATE PROCEDURE dbo.Course_StudyType_Insert
(
  @Course_StudyType_ID uniqueidenti↓er,
  @CourseVersion_ID uniqueidenti↓er,
  @StudyType_ID int,
  @Course_StudyTypeCategory_ID int,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
INSERT INTO Course_StudyType (
Course_StudyType_ID,
CourseVersion_ID,
StudyType_ID,
Course_StudyTypeCategory_ID,
Created,
CreatedBy,
Updated,
UpdatedBy
)
VALUES (
@Course_StudyType_ID,
@CourseVersion_ID,
@StudyType_ID,
@Course_StudyTypeCategory_ID,
@Created,
@CreatedBy,
@Updated,
@UpdatedBy
);
```

2.17.4 Course_StudyType_Select

```
CREATE PROCEDURE dbo.Course_StudyType_Select
(
  @Course_StudyType_ID uniqueidenti↓er
)
AS
SET NOCOUNT ON;
SELECT
  Course_StudyType_ID,
  CourseVersion_ID,
  StudyType_ID,
  Course_StudyTypeCategory_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM Course_StudyType
WHERE Course_StudyType_ID = @Course_StudyType_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.17.5 Course_StudyType_Update

```
CREATE PROCEDURE dbo.Course_StudyType_Update
(
  @Course_StudyType_ID uniqueidenti↓er,
  @CourseVersion_ID uniqueidenti↓er,
  @StudyType_ID int,
  @Course_StudyTypeCategory_ID int,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er,
  @Original_Course_StudyType_ID uniqueidenti↓er,
  @Original_CourseVersion_ID uniqueidenti↓er,
  @Original_StudyType_ID int,
  @Original_Course_StudyTypeCategory_ID int,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
UPDATE Course_StudyType
SET
  Course_StudyType_ID = @Course_StudyType_ID,
  CourseVersion_ID = @CourseVersion_ID,
  StudyType_ID = @StudyType_ID,
  Course_StudyTypeCategory_ID = @Course_StudyTypeCategory_ID,
  Created = @Created,
  CreatedBy = @CreatedBy,
  Updated = @Updated,
  UpdatedBy = @UpdatedBy
WHERE
  (Course_StudyType_ID = @Original_Course_StudyType_ID) AND
  (CourseVersion_ID = @Original_CourseVersion_ID) AND
```

```
  (StudyType_ID = @Original_StudyType_ID) AND
  (Course_StudyTypeCategory_ID = @Original_Course_StudyTypeCategory_ID) AND
  (Created = @Original_Created) AND
  (CreatedBy = @Original_CreatedBy) AND
  (Updated = @Original_Updated) AND
  (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50101', 14, 1);
```

2.18 Course_StudyTypeCategory

2.18.1 Course_StudyTypeCategory_Select

```
CREATE PROCEDURE dbo.Course_StudyTypeCategory_Select
(
  @Course_StudyTypeCategory_ID int
)
AS
SET NOCOUNT ON;
SELECT
  Course_StudyTypeCategory_ID,
  StudyType_ID,
  Name,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM Course_StudyTypeCategory
WHERE Course_StudyTypeCategory_ID = @Course_StudyTypeCategory_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.19 CourseGrab

2.19.1 CourseGrab_Delete

```
CREATE PROCEDURE dbo.CourseGrab_Delete
(
  @Original_CourseGrab_ID uniqueidenti↓er,
  @Original_CourseVersion_ID uniqueidenti↓er,
  @Original_Course_ID uniqueidenti↓er,
  @Original_Number varchar(20),
  @Original_DtuVersion int,
  @Original_Schedule text,
  @Original_Duration text,
  @Original_ExaminationPlacement text,
  @Original_ExaminationAids text,
  @Original_PointBlockingCourses text,
  @Original_PreviousCourseNumbers text,
  @Original_MandatoryPrerequisites text,
  @Original_TechnicalPrerequisites text,
  @Original_DesirablePrerequisites text,
  @Original_Responsibles text,
  @Original_ExternalInstitutions text,
  @Original_LastUpdated datetime,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
DELETE FROM CourseGrab
WHERE
  (CourseGrab_ID = @Original_CourseGrab_ID) AND
  (CourseVersion_ID = @Original_CourseVersion_ID) AND
  (Course_ID = @Original_Course_ID) AND
  (Number = @Original_Number) AND
  (DtuVersion = @Original_DtuVersion) AND
-- (Schedule = @Original_Schedule) AND
  (Duration LIKE @Original_Duration) AND
-- (ExaminationPlacement = @Original_ExaminationPlacement) AND
-- (ExaminationAids = @Original_ExaminationAids) AND
  (PointBlockingCourses LIKE @Original_PointBlockingCourses) AND
  (PreviousCourseNumbers LIKE @Original_PreviousCourseNumbers) AND
  (MandatoryPrerequisites LIKE @Original_MandatoryPrerequisites) AND
  (TechnicalPrerequisites LIKE @Original_TechnicalPrerequisites) AND
  (DesirablePrerequisites LIKE @Original_DesirablePrerequisites) AND
  (Responsibles LIKE @Original_Responsibles) AND
  (ExternalInstitutions LIKE @Original_ExternalInstitutions) AND
  (LastUpdated = @Original_LastUpdated) AND
  (Created = @Original_Created) AND
  (CreatedBy = @Original_CreatedBy) AND
  (Updated = @Original_Updated) AND
  (UpdatedBy = @Original_UpdatedBy)
```

```
IF @@ROWCOUNT = 0
  RAISERROR('50201', 14, 1);
```

2.19.2 CourseGrab_Insert

```
CREATE PROCEDURE dbo.CourseGrab_Insert
(
  @CourseGrab_ID uniqueidenti↓er,
  @CourseVersion_ID uniqueidenti↓er,
  @Course_ID uniqueidenti↓er,
  @Number varchar(20),
  @DtuVersion int,
  @Schedule text,
  @Duration text,
  @ExaminationPlacement text,
  @ExaminationAids text,
  @PointBlockingCourses text,
  @PreviousCourseNumbers text,
  @MandatoryPrerequisites text,
  @TechnicalPrerequisites text,
  @DesirablePrerequisites text,
  @Responsibles text,
  @ExternalInstitutions text,
  @LastUpdated datetime,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
INSERT INTO CourseGrab (
CourseGrab_ID,
CourseVersion_ID,
Course_ID,
Number,
DtuVersion,
Schedule,
Duration,
ExaminationPlacement,
ExaminationAids,
PointBlockingCourses,
PreviousCourseNumbers,
MandatoryPrerequisites,
TechnicalPrerequisites,
DesirablePrerequisites,
Responsibles,
ExternalInstitutions,
LastUpdated,
Created,
CreatedBy,
Updated,
UpdatedBy
)
VALUES (
@CourseGrab_ID,
@CourseVersion_ID,
@Course_ID,
@Number,
@DtuVersion,
@Schedule,
@Duration,
@ExaminationPlacement,
@ExaminationAids,
@PointBlockingCourses,
@PreviousCourseNumbers,
@MandatoryPrerequisites,
@TechnicalPrerequisites,
@DesirablePrerequisites,
@Responsibles,
@ExternalInstitutions,
@LastUpdated,
@Created,
@CreatedBy,
@Updated,
@UpdatedBy
);
```

2.19.3 CourseGrab_Select

```
CREATE PROCEDURE dbo.CourseGrab_Select
(
  @CourseGrab_ID uniqueidenti↓er
)
AS
SET NOCOUNT ON;
SELECT
  CourseGrab_ID,
  CourseVersion_ID,
  Course_ID,
  Number,
  DtuVersion,
```

```
  Schedule,
  Duration,
  ExaminationPlacement,
  ExaminationAids,
  PointBlockingCourses,
  PreviousCourseNumbers,
  MandatoryPrerequisites,
  TechnicalPrerequisites,
  DesirablePrerequisites,
  Responsibles,
  ExternalInstitutions,
  LastUpdated,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM CourseGrab
WHERE CourseGrab_ID = @CourseGrab_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.19.4 CourseGrab_Update

```
CREATE PROCEDURE dbo.CourseGrab_Update
(
  @CourseGrab_ID uniqueidenti↓er,
  @CourseVersion_ID uniqueidenti↓er,
  @Course_ID uniqueidenti↓er,
  @Number varchar(20),
  @DtuVersion int,
  @Schedule text,
  @Duration text,
  @ExaminationPlacement text,
  @ExaminationAids text,
  @PointBlockingCourses text,
  @PreviousCourseNumbers text,
  @MandatoryPrerequisites text,
  @TechnicalPrerequisites text,
  @DesirablePrerequisites text,
  @Responsibles text,
  @ExternalInstitutions text,
  @LastUpdated datetime,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er,
  @Original_CourseGrab_ID uniqueidenti↓er,
  @Original_CourseVersion_ID uniqueidenti↓er,
  @Original_Course_ID uniqueidenti↓er,
  @Original_Number varchar(20),
  @Original_DtuVersion int,
  @Original_Schedule text,
  @Original_Duration text,
  @Original_ExaminationPlacement text,
  @Original_ExaminationAids text,
  @Original_PointBlockingCourses text,
  @Original_PreviousCourseNumbers text,
  @Original_MandatoryPrerequisites text,
  @Original_TechnicalPrerequisites text,
  @Original_DesirablePrerequisites text,
  @Original_Responsibles text,
  @Original_ExternalInstitutions text,
  @Original_LastUpdated datetime,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
UPDATE CourseGrab
SET
  CourseGrab_ID = @CourseGrab_ID,
  CourseVersion_ID = @CourseVersion_ID,
  Course_ID = @Course_ID,
  Number = @Number,
  DtuVersion = @DtuVersion,
  Schedule = @Schedule,
  Duration = @Duration,
  ExaminationPlacement = @ExaminationPlacement,
  ExaminationAids = @ExaminationAids,
  PointBlockingCourses = @PointBlockingCourses,
  PreviousCourseNumbers = @PreviousCourseNumbers,
  MandatoryPrerequisites = @MandatoryPrerequisites,
  TechnicalPrerequisites = @TechnicalPrerequisites,
  DesirablePrerequisites = @DesirablePrerequisites,
  Responsibles = @Responsibles,
  ExternalInstitutions = @ExternalInstitutions,
  LastUpdated = @LastUpdated,
  Created = @Created,
  CreatedBy = @CreatedBy,
  Updated = @Updated,
  UpdatedBy = @UpdatedBy
WHERE
  (CourseGrab_ID = @Original_CourseGrab_ID) AND
```

```
  (CourseVersion_ID = @Original_CourseVersion_ID) AND
  (Course_ID = @Original_Course_ID) AND
  (Number = @Original_Number) AND
  (DtuVersion = @Original_DtuVersion) AND
-- (Schedule = @Original_Schedule) AND
  (Duration LIKE @Original_Duration) AND
-- (ExaminationPlacement = @Original_ExaminationPlacement) AND
-- (ExaminationAids = @Original_ExaminationAids) AND
  (PointBlockingCourses LIKE @Original_PointBlockingCourses) AND
  (PreviousCourseNumbers LIKE @Original_PreviousCourseNumbers) AND
  (MandatoryPrerequisites LIKE @Original_MandatoryPrerequisites) AND
  (TechnicalPrerequisites LIKE @Original_TechnicalPrerequisites) AND
  (DesirablePrerequisites LIKE @Original_DesirablePrerequisites) AND
  (Responsibles LIKE @Original_Responsibles) AND
  (ExternalInstitutions LIKE @Original_ExternalInstitutions) AND
  (LastUpdated = @Original_LastUpdated) AND
  (Created = @Original_Created) AND
  (CreatedBy = @Original_CreatedBy) AND
  (Updated = @Original_Updated) AND
  (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50101', 14, 1);
```

2.20 CourseVersion

2.20.1 CourseVersion_Delete

```
CREATE PROCEDURE dbo.CourseVersion_Delete
(
  @Original_CourseVersion_ID uniqueidenti↓er,
  @Original_Course_ID uniqueidenti↓er,
  @Original_Version int,
  @Original_Number varchar(20),
  @Original_Name text,
  @Original_Language_ID char(2),
  @Original_TeachingForm text,
  @Original_Parts int,
  @Original_AssessmentType_ID int,
  @Original_EvaluationFormDescription text,
  @Original_ParticipantLimitationMin int,
  @Original_ParticipantLimitationMax int,
  @Original_Objective text,
  @Original_Contents text,
  @Original_Remark text,
  @Original_Url varchar(100),
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
DELETE FROM CourseVersion
WHERE
  (CourseVersion_ID = @Original_CourseVersion_ID) AND
  (Course_ID = @Original_Course_ID) AND
  (Version = @Original_Version) AND
  (Number = @Original_Number) AND
-- (Name LIKE @Original_Name) AND
  (Language_ID = @Original_Language_ID) AND
-- (TeachingForm LIKE @Original_TeachingForm) AND
  (Parts = @Original_Parts) AND
  (AssessmentType_ID = @Original_AssessmentType_ID) AND
-- (EvaluationFormDescription LIKE @Original_EvaluationFormDescription) AND
  (ParticipantLimitationMin = @Original_ParticipantLimitationMin) AND
  (ParticipantLimitationMax = @Original_ParticipantLimitationMax) AND
-- (Objective LIKE @Original_Objective) AND
-- (Contents LIKE @Original_Contents) AND
-- (Remark LIKE @Original_Remark) AND
  (Url = @Original_Url) AND
  (Created = @Original_Created) AND
  (CreatedBy = @Original_CreatedBy) AND
  (Updated = @Original_Updated) AND
  (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50201', 14, 1);
```

2.20.2 CourseVersion_GetNewestVersionFromCourseID

```
CREATE PROCEDURE dbo.CourseVersion_GetNewestVersionFromCourseID
(
  @Course_ID uniqueidenti↓er,
  @CourseVersion_ID uniqueidenti↓er OUTPUT
)
AS
SET NOCOUNT ON;
SELECT
  @CourseVersion_ID = CourseVersion_ID
FROM CourseVersion
```

```
WHERE
  Course_ID = @Course_ID AND
  Version = (SELECT MAX(Version) FROM CourseVersion WHERE Course_ID = @Course_ID)
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.20.3 CourseVersion_GetVersions

```
CREATE PROCEDURE dbo.CourseVersion_GetVersions
(
  @Course_ID uniqueidenti↓er
)
AS
SET NOCOUNT ON;
SELECT
  CourseVersion_ID
FROM CourseVersion
WHERE
  Course_ID = @Course_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.20.4 CourseVersion_Insert

```
CREATE PROCEDURE dbo.CourseVersion_Insert
(
  @CourseVersion_ID uniqueidenti↓er,
  @Course_ID uniqueidenti↓er,
  @Version int,
  @Number varchar(20),
  @Name text,
  @Language_ID char(2),
  @TeachingForm text,
  @Parts int,
  @AssessmentType_ID int,
  @EvaluationFormDescription text,
  @ParticipantLimitationMin int,
  @ParticipantLimitationMax int,
  @Objective text,
  @Contents text,
  @Remark text,
  @Url varchar(100),
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
INSERT INTO CourseVersion (
CourseVersion_ID,
Course_ID,
Version,
Number,
Name,
Language_ID,
TeachingForm,
Parts,
AssessmentType_ID,
EvaluationFormDescription,
ParticipantLimitationMin,
ParticipantLimitationMax,
Objective,
Contents,
Remark,
Url,
Created,
CreatedBy,
Updated,
UpdatedBy
)
VALUES (
@CourseVersion_ID,
@Course_ID,
@Version,
@Number,
@Name,
@Language_ID,
@TeachingForm,
@Parts,
@AssessmentType_ID,
@EvaluationFormDescription,
@ParticipantLimitationMin,
@ParticipantLimitationMax,
@Objective,
@Contents,
@Remark,
@Url,
@Created,
@CreatedBy,
@Updated,
@UpdatedBy
);
```

2.20.5 CourseVersion_Select

```
CREATE PROCEDURE dbo.CourseVersion_Select
(
  @CourseVersion_ID uniqueidenti↓er
)
AS
SET NOCOUNT ON;
SELECT
  CourseVersion_ID,
  Course_ID,
  Version,
  Number,
  Name,
  Language_ID,
  TeachingForm,
  Parts,
  AssessmentType_ID,
  EvaluationFormDescription,
  ParticipantLimitationMin,
  ParticipantLimitationMax,
  Objective,
  Contents,
  Remark,
  Url,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM CourseVersion
WHERE CourseVersion_ID = @CourseVersion_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.20.6 CourseVersion_Update

```
CREATE PROCEDURE dbo.CourseVersion_Update
(
  @CourseVersion_ID uniqueidenti↓er,
  @Course_ID uniqueidenti↓er,
  @Version int,
  @Number varchar(20),
  @Name text,
  @Language_ID char(2),
  @TeachingForm text,
  @Parts int,
  @AssessmentType_ID int,
  @EvaluationFormDescription text,
  @ParticipantLimitationMin int,
  @ParticipantLimitationMax int,
  @Objective text,
  @Contents text,
  @Remark text,
  @Url varchar(100),
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er,
  @Original_CourseVersion_ID uniqueidenti↓er,
  @Original_Course_ID uniqueidenti↓er,
  @Original_Version int,
  @Original_Number varchar(20),
  @Original_Name text,
  @Original_Language_ID char(2),
  @Original_TeachingForm text,
  @Original_Parts int,
  @Original_AssessmentType_ID int,
  @Original_EvaluationFormDescription text,
  @Original_ParticipantLimitationMin int,
  @Original_ParticipantLimitationMax int,
  @Original_Objective text,
  @Original_Contents text,
  @Original_Remark text,
  @Original_Url varchar(100),
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
UPDATE CourseVersion
SET
  CourseVersion_ID = @CourseVersion_ID,
  Course_ID = @Course_ID,
  Version = @Version,
  Number = @Number,
  Name = @Name,
  Language_ID = @Language_ID,
  TeachingForm = @TeachingForm,
  Parts = @Parts,
  AssessmentType_ID = @AssessmentType_ID,
  EvaluationFormDescription = @EvaluationFormDescription,
  ParticipantLimitationMin = @ParticipantLimitationMin,
```

```
  ParticipantLimitationMax = @ParticipantLimitationMax,
  Objective = @Objective,
  Contents = @Contents,
  Remark = @Remark,
  Url = @Url,
  Created = @Created,
  CreatedBy = @CreatedBy,
  Updated = @Updated,
  UpdatedBy = @UpdatedBy
WHERE
  (CourseVersion_ID = @Original_CourseVersion_ID) AND
  (Course_ID = @Original_Course_ID) AND
  (Version = @Original_Version) AND
  (Number = @Original_Number) AND
-- (Name LIKE @Original_Name) AND
  (Language_ID = @Original_Language_ID) AND
-- (TeachingForm LIKE @Original_TeachingForm) AND
  (Parts = @Original_Parts) AND
  (AssessmentType_ID = @Original_AssessmentType_ID) AND
-- (EvaluationFormDescription LIKE @Original_EvaluationFormDescription) AND
  (ParticipantLimitationMin = @Original_ParticipantLimitationMin) AND
  (ParticipantLimitationMax = @Original_ParticipantLimitationMax) AND
-- (Objective LIKE @Original_Objective) AND
-- (Contents LIKE @Original_Contents) AND
-- (Remark LIKE @Original_Remark) AND
  (Url = @Original_Url) AND
  (Created = @Original_Created) AND
  (CreatedBy = @Original_CreatedBy) AND
  (Updated = @Original_Updated) AND
  (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50101', 14, 1);
```

2.21 Department

2.21.1 Department_GetDepartments

```
CREATE PROCEDURE dbo.Department_GetDepartments
AS
SET NOCOUNT ON;
SELECT
  Department_ID
FROM Department
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.21.2 Department_GetIDFromNumber

```
CREATE PROCEDURE dbo.Department_GetIDFromNumber
(
  @Number int,
  @Department_ID int OUTPUT
)
AS
SELECT
  @Department_ID = Department_ID
FROM Department
WHERE Number=@Number
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.21.3 Department_Select

```
CREATE PROCEDURE dbo.Department_Select
(
  @Department_ID int
)
AS
SET NOCOUNT ON;
SELECT
  Department_ID,
  ContactData_ID,
  Name,
  Number,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM Department
WHERE Department_ID = @Department_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.22 Gender

2.22.1 Gender_Select

```
CREATE PROCEDURE dbo.Gender_Select
(
  @Gender_ID int
)
AS
SET NOCOUNT ON;
SELECT  Gender_ID,
  Name,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM  Gender
WHERE  Gender_ID = @Gender_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.23 Grade

2.23.1 Grade_Select

```
CREATE PROCEDURE dbo.Grade_Select
(
  @Grade_ID int
)
AS
SET NOCOUNT ON;
SELECT
  Grade_ID,
  Number,
  Name,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM Grade
WHERE Grade_ID = @Grade_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.24 Keyword

2.24.1 Keyword_Delete

```
CREATE PROCEDURE dbo.Keyword_Delete
(
  @Original_Keyword_ID uniqueidenti↓er,
  @Original_Name varchar(100),
  @Original_Culture_ID varchar(10),
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
DELETE FROM Keyword
WHERE  (Keyword_ID = @Original_Keyword_ID)
AND  (Name = @Original_Name)
AND (Culture_ID = @Original_Culture_ID)
AND  (Created = @Original_Created)
AND  (CreatedBy = @Original_CreatedBy)
AND  (Updated = @Original_Updated)
AND  (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50201', 14, 1);
```

2.24.2 Keyword_Insert

```
CREATE PROCEDURE dbo.Keyword_Insert
(
  @Keyword_ID uniqueidenti↓er,
  @Name varchar(100),
```

```
  @Culture_ID varchar(10),
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
INSERT INTO Keyword
        (Keyword_ID,
  Name,
  Culture_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy)
VALUES (@Keyword_ID,
  @Name,
  @Culture_ID,
  @Created,
  @CreatedBy,
  @Updated,
  @UpdatedBy)
```

2.24.3 Keyword_SearchForKeywords

```
CREATE PROCEDURE dbo.Keyword_SearchForKeywords
(
  @Keyword varchar(100),
  @Culture_ID varchar(10)
)
AS
SELECT
  Keyword_ID
FROM Keyword
WHERE
  Name = @Keyword
  AND Culture_ID = @Culture_ID
```

2.24.4 Keyword_Select

```
CREATE PROCEDURE dbo.Keyword_Select
(
  @Keyword_ID uniqueidenti↓er
)
AS
SET NOCOUNT ON;
SELECT  Keyword_ID,
  Name,
  Culture_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM  Keyword
WHERE  Keyword_ID = @Keyword_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.24.5 Keyword_Update

```
CREATE PROCEDURE dbo.Keyword_Update
(
  @Keyword_ID uniqueidenti↓er,
  @Name varchar(100),
  @Culture_ID varchar(10),
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er,
  @Original_Keyword_ID uniqueidenti↓er,
  @Original_Name varchar(100),
  @Original_Culture_ID varchar(10),
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
UPDATE  Keyword
SET  Keyword_ID = @Keyword_ID,
  Name = @Name,
  Culture_ID = @Culture_ID,
      Created = @Created,
  CreatedBy = @CreatedBy,
  Updated = @Updated,
  UpdatedBy = @UpdatedBy
```

```
WHERE  (Keyword_ID = @Original_Keyword_ID)
AND (Name = @Original_Name)
AND (Culture_ID = @Original_Culture_ID)
AND  (Created = @Original_Created)
AND (CreatedBy = @Original_CreatedBy)
AND (Updated = @Original_Updated)
AND (UpdatedBy = @Original_UpdatedBy);
IF @@ROWCOUNT = 0
  RAISERROR('50101', 14, 1);
```

2.25 Language

2.25.1 Language_Select

```
CREATE PROCEDURE dbo.Language_Select
(
  @Language_ID char(2)
)
AS
SET NOCOUNT ON;
SELECT
  Language_ID,
  Name,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM Language
WHERE Language_ID = @Language_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.26 Lecturer

2.26.1 Lecturer_Delete

```
CREATE PROCEDURE dbo.Lecturer_Delete
(
  @Original_Lecturer_ID uniqueidenti↓er,
  @Original_CwisNumber int,
  @Original_Person_ID uniqueidenti↓er,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
DELETE FROM Lecturer
WHERE
  (Lecturer_ID = @Original_Lecturer_ID) AND
  (CwisNumber = @Original_CwisNumber) AND
  (Person_ID = @Original_Person_ID) AND
  (Created = @Original_Created) AND
  (CreatedBy = @Original_CreatedBy) AND
  (Updated = @Original_Updated) AND
  (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50201', 14, 1);
```

2.26.2 Lecturer_GetIDFromCwisNumber

```
CREATE PROCEDURE dbo.Lecturer_GetIDFromCwisNumber
(
  @CwisNumber int,
  @Lecturer_ID uniqueidenti↓er OUTPUT
)
AS
SELECT
  @Lecturer_ID = Lecturer_ID
FROM Lecturer
WHERE CwisNumber=@CwisNumber
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.26.3 Lecturer_Insert

```
CREATE PROCEDURE dbo.Lecturer_Insert
(
  @Lecturer_ID uniqueidenti↓er,
  @CwisNumber int,
  @Person_ID uniqueidenti↓er,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
INSERT INTO Lecturer (
Lecturer_ID,
CwisNumber,
Person_ID,
Created,
CreatedBy,
Updated,
UpdatedBy
)
VALUES (
@Lecturer_ID,
@CwisNumber,
@Person_ID,
@Created,
@CreatedBy,
@Updated,
@UpdatedBy
);
```

2.26.4 Lecturer_Select

```
CREATE PROCEDURE dbo.Lecturer_Select
(
  @Lecturer_ID uniqueidenti↓er
)
AS
SET NOCOUNT ON;
SELECT
  Lecturer_ID,
  CwisNumber,
  Person_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM Lecturer
WHERE Lecturer_ID = @Lecturer_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.26.5 Lecturer_Update

```
CREATE PROCEDURE dbo.Lecturer_Update
(
  @Lecturer_ID uniqueidenti↓er,
  @CwisNumber int,
  @Person_ID uniqueidenti↓er,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er,
  @Original_Lecturer_ID uniqueidenti↓er,
  @Original_CwisNumber int,
  @Original_Person_ID uniqueidenti↓er,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
UPDATE Lecturer
SET
  Lecturer_ID = @Lecturer_ID,
  CwisNumber = @CwisNumber,
  Person_ID = @Person_ID,
  Created = @Created,
  CreatedBy = @CreatedBy,
  Updated = @Updated,
  UpdatedBy = @UpdatedBy
WHERE
  (Lecturer_ID = @Original_Lecturer_ID) AND
  (CwisNumber = @Original_CwisNumber) AND
  (Person_ID = @Original_Person_ID) AND
  (Created = @Original_Created) AND
  (CreatedBy = @Original_CreatedBy) AND
  (Updated = @Original_Updated) AND
  (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50101', 14, 1);
```

2.27 Module

2.27.1 Module_Select

```
CREATE PROCEDURE dbo.Module_Select
(
  @Module_ID int
)
AS
SET NOCOUNT ON;
SELECT  Module_ID,
  Name,
  StartTime,
  EndTime,
  Weekday_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM  Module
WHERE  Module_ID = @Module_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.28 Period

2.28.1 Period_Delete

```
CREATE PROCEDURE dbo.Period_Delete
(
  @Original_Period_ID uniqueidenti¦er,
  @Original_PeriodType_ID int,
  @Original_Name text,
  @Original_StartDate datetime,
  @Original_EndDate datetime,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti¦er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti¦er
)
AS
SET NOCOUNT OFF;
DELETE FROM Period
WHERE  (Period_ID  = @Original_Period_ID)
AND  (PeriodType_ID = @Original_PeriodType_ID)
--AND  (Name like @Original_Name)
AND  (StartDate  = @Original_StartDate)
AND  (EndDate  = @Original_EndDate)
AND  (Created  = @Original_Created)
AND  (CreatedBy  = @Original_CreatedBy)
AND  (Updated  = @Original_Updated)
AND  (UpdatedBy  = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50201', 14, 1);
```

2.28.2 Period_GetNextPeriod

```
CREATE PROCEDURE dbo.Period_GetNextPeriod
(
  @Period_ID uniqueidenti¦er
)
AS
SET NOCOUNT ON;

DECLARE @Startdate datetime
SET @Startdate = (SELECT Startdate
      FROM Period
      WHERE Period_id = @Period_ID)

IF @Startdate IS NULL
BEGIN
  SET @Startdate = getdate()
END

SELECT  TOP 1 Period_id
FROM Period
WHERE Startdate > @Startdate
AND PeriodType_ID IN (1,2)
ORDER BY Startdate

IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.28.3 Period_Insert

```
CREATE PROCEDURE dbo.Period_Insert
(
  @Period_ID uniqueidenti↓er,
  @PeriodType_ID int,
  @Name text,
  @StartDate datetime,
  @EndDate datetime,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
INSERT INTO Period
        (Period_ID,
  PeriodType_ID,
  Name,
  StartDate,
  EndDate,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy)
VALUES (@Period_ID,
  @PeriodType_ID,
  @Name,
  @StartDate,
  @EndDate,
  @Created,
  @CreatedBy,
  @Updated,
  @UpdatedBy);
```

2.28.4 Period_Select

```
CREATE PROCEDURE dbo.Period_Select
(
  @Period_ID uniqueidenti↓er
)
AS
SET NOCOUNT ON;
SELECT  Period_ID,
  PeriodType_ID,
  Name,
  StartDate,
  EndDate,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM  Period
WHERE  Period_ID = @Period_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.28.5 Period_Update

```
CREATE PROCEDURE dbo.Period_Update
(
  @Period_ID uniqueidenti↓er,
  @PeriodType_ID int,
  @Name text,
  @StartDate datetime,
  @EndDate datetime,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er,
  @Original_Period_ID uniqueidenti↓er,
  @Original_PeriodType_ID int,
  @Original_Name text,
  @Original_StartDate datetime,
  @Original_EndDate datetime,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
UPDATE  Period
SET  Period_ID = @Period_ID,
  PeriodType_ID = @PeriodType_ID,
  Name = @Name,
  StartDate = @StartDate,
  EndDate = @EndDate,
      Created = @Created,
  CreatedBy = @CreatedBy,
```

```
  Updated = @Updated,
  UpdatedBy = @UpdatedBy
WHERE (Period_ID = @Original_Period_ID)
AND (PeriodType_ID = @Original_PeriodType_ID)
--AND (Name like @Name)
AND (StartDate = @StartDate)
AND (EndDate = @EndDate)
AND (Created = @Original_Created)
AND (CreatedBy = @Original_CreatedBy)
AND (Updated = @Original_Updated)
AND (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50101', 14, 1);
```

2.29 PeriodType

2.29.1 PeriodType_Select

```
CREATE PROCEDURE dbo.PeriodType_Select
(
  @PeriodType_ID int
)
AS
SET NOCOUNT ON;
SELECT PeriodType_ID,
  Name,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM PeriodType
WHERE PeriodType_ID = @PeriodType_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.30 PeriodType_Module

2.30.1 PeriodType_Module_GetModules

```
CREATE PROCEDURE dbo.PeriodType_Module_GetModules
(
  @PeriodType_ID int
)
AS
SET NOCOUNT ON;
SELECT Module_ID
FROM PeriodType_Module
WHERE PeriodType_ID = @PeriodType_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.30.2 PeriodType_Module_Select

```
CREATE PROCEDURE dbo.PeriodType_Module_Select
(
  @PeriodType_Module_ID int
)
AS
SET NOCOUNT ON;
SELECT PeriodType_Module_ID,
  PeriodType_ID,
  Module_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM PeriodType_Module
WHERE PeriodType_Module_ID = @PeriodType_Module_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.31 Person

2.31.1 Person_Delete

```
CREATE PROCEDURE dbo.Person_Delete
(
  @Original_Person_ID uniqueidenti↓er,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
DELETE FROM Person
WHERE  (Person_ID = @Original_Person_ID)
AND  (Created = @Original_Created)
AND  (CreatedBy = @Original_CreatedBy)
AND  (Updated = @Original_Updated)
AND  (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50201', 14, 1);
```

2.31.2 Person_Insert

```
CREATE PROCEDURE dbo.Person_Insert
(
  @Person_ID uniqueidenti↓er,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
INSERT INTO Person
  (Person_ID,
   Created,
   CreatedBy,
   Updated,
   UpdatedBy)
VALUES (@Person_ID,
  @Created,
  @CreatedBy,
  @Updated,
  @UpdatedBy);
```

2.31.3 Person_Select

```
CREATE PROCEDURE dbo.Person_Select
(
  @Person_ID uniqueidenti↓er
)
AS
SET NOCOUNT ON;
SELECT
  Person_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM Person
WHERE Person_ID = @Person_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.31.4 Person_Update

```
CREATE PROCEDURE dbo.Person_Update
(
  @Person_ID uniqueidenti↓er,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er,
  @Original_Person_ID uniqueidenti↓er,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
UPDATE  Person
SET Person_ID = @Person_ID,
  Created = @Created,
  CreatedBy = @CreatedBy,
  Updated = @Updated,
  UpdatedBy = @UpdatedBy
WHERE Person_ID = @Original_Person_ID
AND  (Created = @Original_Created)
AND (CreatedBy = @Original_CreatedBy)
AND (Updated = @Original_Updated)
```

```
AND (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50101', 14, 1);
```

2.32 Person_ContactData

2.32.1 Person_ContactData_Delete

```
CREATE PROCEDURE dbo.Person_ContactData_Delete
(
  @Original_Person_ContactData_ID uniqueidenti↓er,
  @Original_PersonVersion_ID uniqueidenti↓er,
  @Original_ContactData_ID uniqueidenti↓er,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
DELETE FROM Person_ContactData
WHERE  (Person_ContactData_ID = @Original_Person_ContactData_ID)
AND (PersonVersion_ID  = @Original_PersonVersion_ID)
AND (ContactData_ID  = @Original_ContactData_ID)
AND (Created   = @Original_Created)
AND (CreatedBy   = @Original_CreatedBy)
AND (Updated   = @Original_Updated)
AND (UpdatedBY   = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50201', 14, 1);
```

2.32.2 Person_ContactData_Insert

```
CREATE PROCEDURE dbo.Person_ContactData_Insert
(
  @Person_ContactData_ID uniqueidenti↓er,
  @PersonVersion_ID uniqueidenti↓er,
  @ContactData_ID uniqueidenti↓er,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
INSERT INTO Person_ContactData
       (Person_ContactData_ID,
  PersonVersion_ID,
  ContactData_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy)
VALUES (@Person_ContactData_ID,
  @PersonVersion_ID,
  @ContactData_ID,
  @Created,
  @CreatedBy,
  @Updated,
  @UpdatedBy);
```

2.32.3 Person_ContactData_Select

```
CREATE PROCEDURE dbo.Person_ContactData_Select
(
  @Person_ContactData_ID uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
SELECT Person_ContactData_ID,
  PersonVersion_ID,
  ContactData_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM Person_ContactData
WHERE Person_ContactData_ID = @Person_ContactData_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.32.4 Person_ContactData_Update

```
CREATE PROCEDURE dbo.Person_ContactData_Update
(
  @Person_ContactData_ID uniqueidenti↓er,
  @PersonVersion_ID uniqueidenti↓er,
  @ContactData_ID uniqueidenti↓er,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er,
  @Original_Person_ContactData_ID uniqueidenti↓er,
  @Original_PersonVersion_ID uniqueidenti↓er,
  @Original_ContactData_ID uniqueidenti↓er,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
UPDATE Person_ContactData
SET  Person_ContactData_ID  = @Person_ContactData_ID,
  PersonVersion_ID  = @PersonVersion_ID,
  ContactData_ID  = @ContactData_ID,
  Created  = @Created,
  CreatedBy  = @CreatedBy,
  Updated  = @Updated,
  UpdatedBy  = @UpdatedBy
WHERE  (Person_ContactData_ID = @Original_Person_ContactData_ID)
AND (PersonVersion_ID  = @Original_PersonVersion_ID)
AND (ContactData_ID  = @Original_ContactData_ID)
AND (Created  = @Original_Created)
AND (CreatedBy  = @Original_CreatedBy)
AND (Updated  = @Original_Updated)
AND (UpdatedBY  = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50101', 14, 1);
```

2.33 PersonVersion

2.33.1 PersonVersion_Delete

```
CREATE PROCEDURE dbo.PersonVersion_Delete
(
  @Original_PersonVersion_ID uniqueidenti↓er,
  @Original_Person_ID uniqueidenti↓er,
  @Original_Version int,
  @Original_GivenName varchar(200),
  @Original_MiddleName varchar(200),
  @Original_FamilyName varchar(200),
  @Original_CivilRegistrationNumber varchar(20),
  @Original_Initials varchar(20),
  @Original_Title varchar(100),
  @Original_Birthday datetime,
  @Original_Gender_ID int,
  @Original_User_ID uniqueidenti↓er,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
DELETE FROM PersonVersion
WHERE PersonVersion_ID  = @Original_PersonVersion_ID
AND Person_ID  = @Original_Person_ID
AND version  = @Original_Version
AND GivenName  = @Original_GivenName
AND MiddleName  = @Original_MiddleName
AND FamilyName  = @Original_FamilyName
AND CivilRegistrationNumber = @Original_CivilRegistrationNumber
AND Initials  = @Original_Initials
AND Title  = @Original_Title
AND Birthday  = @Original_Birthday
AND Gender_ID  = @Original_Gender_ID
AND User_ID  = @Original_User_ID
AND Created  = @Original_Created
AND CreatedBy  = @Original_CreatedBy
AND Updated  = @Original_Updated
AND UpdatedBy  = @Original_UpdatedBy
IF @@ROWCOUNT = 0
  RAISERROR('50201', 14, 1);
```

2.33.2 PersonVersion_Insert

```
CREATE PROCEDURE dbo.PersonVersion_Insert
(
  @PersonVersion_ID uniqueidenti↓er,
  @Person_ID uniqueidenti↓er,
  @Version int,
```

```
  @GivenName varchar(200),
  @MiddleName varchar(200),
  @FamilyName varchar(200),
  @CivilRegistrationNumber varchar(20),
  @Initials varchar(20),
  @Title varchar(100),
  @Birthday datetime,
  @Gender_ID int,
  @User_ID uniqueidenti↓er,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
INSERT INTO PersonVersion
  (PersonVersion_ID
  ,Person_ID
  ,Version
  ,GivenName
  ,MiddleName
  ,FamilyName
  ,CivilRegistrationNumber
  ,Initials
  ,Title
  ,Birthday
  ,Gender_ID
  ,User_ID
  ,Created
  ,CreatedBy
  ,Updated
  ,UpdatedBy)
VALUES (@PersonVersion_ID
  ,@Person_ID
  ,@version
  ,@GivenName
  ,@MiddleName
  ,@FamilyName
  ,@CivilRegistrationNumber
  ,@Initials
  ,@Title
  ,@Birthday
  ,@Gender_ID
  ,@User_ID
  ,@Created
  ,@CreatedBy
  ,@Updated
  ,@UpdatedBy)
```

2.33.3 PersonVersion_Select

```
CREATE PROCEDURE dbo.PersonVersion_Select
(
  @PersonVersion_ID uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
SELECT PersonVersion_ID,
  Person_ID,
  Version,
  GivenName,
  MiddleName,
  FamilyName,
  CivilRegistrationNumber,
  Initials,
  Title,
  Birthday,
  Gender_ID,
  User_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM PersonVersion
WHERE PersonVersion_ID  = PersonVersion_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.33.4 PersonVersion_Update

```
CREATE PROCEDURE dbo.PersonVersion_Update
(
  @PersonVersion_ID uniqueidenti↓er,
  @Person_ID uniqueidenti↓er,
  @Version int,
  @GivenName varchar(200),
  @MiddleName varchar(200),
  @FamilyName varchar(200),
  @CivilRegistrationNumber varchar(20),
  @Initials varchar(20),
  @Title varchar(100),
```

```
  @Birthday datetime,
  @Gender_ID int,
  @User_ID uniqueidenti↓er,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er,
  @Original_PersonVersion_ID uniqueidenti↓er,
  @Original_Person_ID uniqueidenti↓er,
  @Original_Version int,
  @Original_GivenName varchar(200),
  @Original_MiddleName varchar(200),
  @Original_FamilyName varchar(200),
  @Original_CivilRegistrationNumber varchar(20),
  @Original_Initials varchar(20),
  @Original_Title varchar(100),
  @Original_Birthday datetime,
  @Original_Gender_ID int,
  @Original_User_ID uniqueidenti↓er,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
UPDATE PersonVersion
SET PersonVersion_ID = @PersonVersion_ID,
  Person_ID = @Person_ID,
  Version = @Version,
  GivenName = @GivenName,
  MiddleName = @MiddleName,
  FamilyName = @FamilyName,
  CivilRegistrationNumber = @CivilRegistrationNumber,
  Initials = @Initials,
  Title = @Title,
  Birthday = @Birthday,
  Gender_ID = @Gender_ID,
  User_ID = @User_ID,
  Created = @Created,
  CreatedBy = @CreatedBy,
  Updated = @Updated,
  UpdatedBy = @UpdatedBy
WHERE PersonVersion_ID = @Original_PersonVersion_ID
AND Person_ID = @Original_Person_ID
AND version = @Original_Version
AND GivenName = @Original_GivenName
AND MiddleName = @Original_MiddleName
AND FamilyName = @Original_FamilyName
AND CivilRegistrationNumber = @Original_CivilRegistrationNumber
AND Initials = @Original_Initials
AND Title = @Original_Title
AND Birthday = @Original_Birthday
AND Gender_ID = @Original_Gender_ID
AND User_ID = @Original_User_ID
AND Created = @Original_Created
AND CreatedBy = @Original_CreatedBy
AND Updated = @Original_Updated
AND UpdatedBy = @Original_UpdatedBy
IF @@ROWCOUNT = 0
  RAISERROR('50101', 14, 1);
```

2.34 Point

2.34.1 Point_Delete

```
CREATE PROCEDURE dbo.Point_Delete
(
  @Original_Point_ID uniqueidenti↓er,
  @Original_PointMin int,
  @Original_PointMax int,
  @Original_AmsGiving bit,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
DELETE FROM Point
WHERE (Point_ID = @Original_Point_ID)
AND (PointMin = @Original_PointMin)
AND (PointMax = @Original_PointMax)
AND (AmsGiving = @Original_AmsGiving)
AND (Created = @Original_Created)
AND (CreatedBy = @Original_CreatedBy)
AND (Updated = @Original_Updated)
AND (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50201', 14, 1);
```

2.34.2 Point_Insert

```
CREATE PROCEDURE dbo.Point_Insert
(
  @Point_ID uniqueidenti↓er,
  @PointMin float,
  @PointMax float,
  @AmsGiving bit,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
INSERT INTO Point
        (Point_ID,
  PointMin,
  PointMax,
  AmsGiving,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy)
VALUES (@Point_ID,
  @PointMin,
  @PointMax,
  @AmsGiving,
  @Created,
  @CreatedBy,
  @Updated,
  @UpdatedBy);
```

2.34.3 Point_Select

```
CREATE PROCEDURE dbo.Point_Select
(
  @Point_ID uniqueidenti↓er
)
AS
SET NOCOUNT ON;
SELECT Point_ID,
  PointMin,
  PointMax,
  AmsGiving,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM  Point
WHERE  Point_ID = @Point_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.34.4 Point_Update

```
CREATE PROCEDURE dbo.Point_Update
(
  @Point_ID uniqueidenti↓er,
  @PointMin float,
  @PointMax float,
  @AmsGiving bit,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er,
  @Original_Point_ID uniqueidenti↓er,
  @Original_PointMin float,
  @Original_PointMax float,
  @Original_AmsGiving bit,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
UPDATE  Point
SET  Point_ID = @Point_ID,
  PointMin = @PointMin,
  PointMax = @PointMax,
  AmsGiving = @AmsGiving,
      Created = @Created,
  CreatedBy = @CreatedBy,
  Updated = @Updated,
  UpdatedBy = @UpdatedBy
WHERE  (Point_ID = @Original_Point_ID)
AND (PointMin = @Original_PointMin)
AND (PointMax = @Original_PointMax)
AND (AmsGiving = @Original_AmsGiving)
AND  (Created = @Original_Created)
```

```
AND (CreatedBy = @Original_CreatedBy)
AND (Updated = @Original_Updated)
AND (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50101', 14, 1);
```

2.35 Project

2.35.1 Project_Delete

```
CREATE PROCEDURE dbo.Project_Delete
(
  @Original_Project_ID uniqueidenti↓er,
  @Original_Number varchar(20),
  @Original_Name text,
  @Original_ProjectType_ID int,
  @Original_AssessmentType_ID int,
  @Original_Period_ID uniqueidenti↓er,
  @Original_Point_ID uniqueidenti↓er,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
DELETE FROM Project
WHERE
  (Project_ID = @Original_Project_ID)
  AND (Number = @Original_Number)
-- AND (Name = @Original_Name)
  AND (ProjectType_ID = @Original_ProjectType_ID)
  AND (AssessmentType_ID = @Original_AssessmentType_ID)
  AND (Period_ID = @Original_Period_ID)
  AND (Point_ID = @Original_Point_ID)
  AND  (Created = @Original_Created)
  AND (CreatedBy = @Original_CreatedBy)
  AND (Updated = @Original_Updated)
  AND (UpdatedBy = @Original_UpdatedBy)

IF @@ROWCOUNT = 0
  RAISERROR('50101', 14, 1);
```

2.35.2 Project_Insert

```
CREATE PROCEDURE dbo.Project_Insert
(
  @Project_ID uniqueidenti↓er,
  @Number varchar(20),
  @Name text,
  @ProjectType_ID int,
  @AssessmentType_ID int,
  @Period_ID uniqueidenti↓er,
  @Point_ID uniqueidenti↓er,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
INSERT INTO Project (
  Project_ID,
  Number,
  Name,
  ProjectType_ID,
  AssessmentType_ID,
  Period_ID,
  Point_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy)
VALUES (@Project_ID,
  @Number,
  @Name,
  @ProjectType_ID,
  @AssessmentType_ID,
  @Period_ID,
  @Point_ID,
  @Created,
  @CreatedBy,
  @Updated,
  @UpdatedBy);
```

2.35.3 Project_Select

```
CREATE PROCEDURE dbo.Project_Select
(
  @Project_ID uniqueidenti↓er
)
AS
SET NOCOUNT ON;
SELECT  Project_ID,
  Number,
  Name,
  ProjectType_ID,
  AssessmentType_ID,
  Period_ID,
  Point_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM  Project
WHERE  Project_ID = @Project_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.35.4 Project_Update

```
CREATE PROCEDURE dbo.Project_Update
(
  @Project_ID uniqueidenti↓er,
  @Number varchar(20),
  @Name text,
  @ProjectType_ID int,
  @AssessmentType_ID int,
  @Period_ID uniqueidenti↓er,
  @Point_ID uniqueidenti↓er,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er,
  @Original_Project_ID uniqueidenti↓er,
  @Original_Number varchar(20),
  @Original_Name text,
  @Original_ProjectType_ID int,
  @Original_AssessmentType_ID int,
  @Original_Period_ID uniqueidenti↓er,
  @Original_Point_ID uniqueidenti↓er,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
UPDATE Project
SET
  Project_ID = @Project_ID,
  Number = @Number,
  Name = @Name,
  ProjectType_ID = @ProjectType_ID,
  AssessmentType_ID = @AssessmentType_ID,
  Period_ID = @Period_ID,
  Point_ID = @Point_ID,
  Created = @Created,
  CreatedBy = @CreatedBy,
  Updated = @Updated,
  UpdatedBy = @UpdatedBy
WHERE
  (Project_ID = @Original_Project_ID)
  AND (Number = @Original_Number)
-- AND (Name = @Original_Name)
  AND (ProjectType_ID = @Original_ProjectType_ID)
  AND (AssessmentType_ID = @Original_AssessmentType_ID)
  AND (Period_ID = @Original_Period_ID)
  AND (Point_ID = @Original_Point_ID)
  AND  (Created = @Original_Created)
  AND (CreatedBy = @Original_CreatedBy)
  AND (Updated = @Original_Updated)
  AND (UpdatedBy = @Original_UpdatedBy)

IF @@ROWCOUNT = 0
  RAISERROR('50101', 14, 1);
```

2.36 ProjectType

2.36.1 ProjectType_Select

```
CREATE PROCEDURE dbo.ProjectType_Select
(
  @ProjectType_ID int
)
```

```
AS
SET NOCOUNT ON;
SELECT
  ProjectType_ID,
  Name,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM ProjectType
WHERE ProjectType_ID = @ProjectType_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.37 RecommendedPlacementConcept

2.37.1 RecommendedPlacementConcept_Select

```
CREATE PROCEDURE dbo.RecommendedPlacementConcept_Select
(
  @RecommendedPlacementConcept_ID int
)
AS
SET NOCOUNT ON;
SELECT
  RecommendedPlacementConcept_ID,
  Name,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM RecommendedPlacementConcept
WHERE RecommendedPlacementConcept_ID = @RecommendedPlacementConcept_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.38 RecommendedPlacementConcept_StudyType

2.38.1 RecommendedPlacementConcept_StudyType_GetID

```
CREATE PROCEDURE dbo.RecommendedPlacementConcept_StudyType_GetID
(
  @RecommendedPlacementConcept_ID int,
  @StudyType_ID int
)
AS
SET NOCOUNT ON;
SELECT
  RecommendedPlacementConcept_StudyType_ID
FROM  RecommendedPlacementConcept_StudyType
WHERE  RecommendedPlacementConcept_ID = @RecommendedPlacementConcept_ID
AND StudyType_ID = @StudyType_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.38.2 RecommendedPlacementConcept_StudyType_Select

```
CREATE PROCEDURE dbo.RecommendedPlacementConcept_StudyType_Select
(
  @RecommendedPlacementConcept_StudyType_ID int
)
AS
SET NOCOUNT ON;
SELECT
  RecommendedPlacementConcept_StudyType_ID,
  RecommendedPlacementConcept_ID,
  StudyType_ID,
  Point_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM RecommendedPlacementConcept_StudyType
WHERE RecommendedPlacementConcept_StudyType_ID = @RecommendedPlacementConcept_StudyType_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.39 SecurityRole_SecurityPermission

2.39.1 SecurityRole_SecurityPermission_Delete

```
CREATE PROCEDURE dbo.SecurityRole_SecurityPermission_Delete
(
  @Original_SecurityRole_SecurityPermission_ID int,
  @Original_SecurityRole_ID int,
  @Original_SecurityPermission_ID int,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
DELETE FROM SecurityRole_SecurityPermission
WHERE
  (SecurityRole_SecurityPermission_ID = @Original_SecurityRole_SecurityPermission_ID)
  AND (SecurityRole_ID = @Original_SecurityRole_ID)
  AND (SecurityPermission_ID = @Original_SecurityPermission_ID)
  AND (Created = @Original_Created)
  AND (CreatedBy = @Original_CreatedBy)
  AND (Updated = @Original_Updated)
  AND (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50201', 14, 1);
```

2.39.2 SecurityRole_SecurityPermission_GetPermissionsFromRoleID

```
CREATE PROCEDURE dbo.SecurityRole_SecurityPermission_GetPermissionsFromRoleID
(
  @SecurityRole_ID int
)
AS
SET NOCOUNT ON;
SELECT
  SecurityPermission_ID
FROM  SecurityRole_SecurityPermission
WHERE  SecurityRole_ID = @SecurityRole_ID

IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.39.3 SecurityRole_SecurityPermission_Insert

```
CREATE PROCEDURE dbo.SecurityRole_SecurityPermission_Insert
(
  @SecurityRole_SecurityPermission_ID int,
  @SecurityRole_ID int,
  @SecurityPermission_ID int,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
INSERT INTO SecurityRole_SecurityPermission
  (SecurityRole_SecurityPermission_ID,
  SecurityRole_ID,
  SecurityPermission_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy)
VALUES (@SecurityRole_SecurityPermission_ID,
  @SecurityRole_ID,
  @SecurityPermission_ID,
  @Created,
  @CreatedBy,
  @Updated,
  @UpdatedBy);
```

2.39.4 SecurityRole_SecurityPermission_Select

```
CREATE PROCEDURE dbo.SecurityRole_SecurityPermission_Select
(
  @SecurityRole_SecurityPermission_ID int
)
AS
SET NOCOUNT ON;
SELECT
  SecurityRole_SecurityPermission_ID,
```

```
  SecurityRole_ID,
  SecurityPermission_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM SecurityRole_SecurityPermission
WHERE SecurityRole_SecurityPermission_ID = @SecurityRole_SecurityPermission_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.39.5 SecurityRole_SecurityPermission_Update

```
CREATE PROCEDURE dbo.SecurityRole_SecurityPermission_Update
(
  @SecurityRole_SecurityPermission_ID int,
  @SecurityRole_ID int,
  @SecurityPermission_ID int,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er,
  @Original_SecurityRole_SecurityPermission_ID int,
  @Original_SecurityRole_ID int,
  @Original_SecurityPermission_ID int,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
UPDATE SecurityRole_SecurityPermission
SET
  SecurityRole_SecurityPermission_ID = @SecurityRole_SecurityPermission_ID,
  SecurityRole_ID = @SecurityRole_ID,
  SecurityPermission_ID = @SecurityPermission_ID,
  Created = @Created,
  CreatedBy = @CreatedBy,
  Updated = @Updated,
  UpdatedBy = @UpdatedBy
WHERE
  (SecurityRole_SecurityPermission_ID = @Original_SecurityRole_SecurityPermission_ID)
  AND (SecurityRole_ID = @Original_SecurityRole_ID)
  AND (SecurityPermission_ID = @Original_SecurityPermission_ID)
  AND (Created = @Original_Created)
  AND (CreatedBy = @Original_CreatedBy)
  AND (Updated = @Original_Updated)
  AND (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50101', 14, 1);
```

2.40 Specialization

2.40.1 Specialization_Select

```
CREATE PROCEDURE dbo.Specialization_Select
(
  @Specialization_ID int
)
AS
SET NOCOUNT ON;
SELECT Specialization_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM Specialization
WHERE Specialization_ID = @Specialization_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.41 Specialization_Course

2.41.1 Specialization_Course_GetCourses

```
CREATE PROCEDURE dbo.Specialization_Course_GetCourses
(
  @SpecializationVersion_ID int,
  @Supplementary bit
)
```

```
AS
SELECT
  Course_ID
FROM Specialization_Course
WHERE
  SpecializationVersion_ID = @SpecializationVersion_ID
  AND Supplementary = @Supplementary
  AND Course_ID IS NOT NULL
```

2.41.2 Specialization_Course_Select

```
CREATE PROCEDURE dbo.Specialization_Course_Select
(
  @Specialization_Course_ID int
)
AS
SET NOCOUNT ON;
SELECT  Specialization_Course_ID,
  SpecializationVersion_ID,
  Course_ID,
  Number,
  PointGiving,
  Supplementary,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM  Specialization_Course
WHERE  Specialization_Course_ID = @Specialization_Course_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.42 SpecializationVersion

2.42.1 SpecializationVersion_Select

```
CREATE PROCEDURE dbo.SpecializationVersion_Select
(
  @SpecializationVersion_ID int
)
AS
SET NOCOUNT ON;
SELECT  SpecializationVersion_ID,
  Specialization_ID,
  Version,
  Name,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM  SpecializationVersion
WHERE  SpecializationVersion_ID = @SpecializationVersion_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.43 Student

2.43.1 Student_Delete

```
CREATE PROCEDURE dbo.Student_Delete
(
  @Original_Student_ID uniqueidenti↓er,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
DELETE FROM Student
WHERE (Student_ID = @Original_Student_ID)
AND (Created = @Original_Created)
AND (CreatedBy = @Original_CreatedBy)
AND (Updated = @Original_Updated)
AND (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50201', 14, 1);
```

2.43.2 Student_Insert

```
CREATE PROCEDURE dbo.Student_Insert
(
  @Student_ID uniqueidenti↓er,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
INSERT INTO Student(
  Student_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy)
VALUES (
  @Student_ID,
  @Created,
  @CreatedBy,
  @Updated,
  @UpdatedBy);
SELECT
  Student_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM Student
WHERE (Student_ID = @Student_ID)
```

2.43.3 Student_Select

```
CREATE PROCEDURE dbo.Student_Select
(
  @Student_ID uniqueidenti↓er
)
AS
SET NOCOUNT ON;
SELECT
  Student_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM Student
WHERE Student_ID = @Student_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.43.4 Student_Update

```
CREATE PROCEDURE dbo.Student_Update
(
  @Student_ID uniqueidenti↓er,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er,
  @Original_Student_ID uniqueidenti↓er,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
UPDATE Student
SET
  Student_ID = @Student_ID,
  Created = @Created,
  CreatedBy = @CreatedBy,
  Updated = @Updated,
  UpdatedBy = @UpdatedBy
WHERE
  Student_ID = @Original_Student_ID
  AND Created = @Original_Created
  AND CreatedBy = @Original_CreatedBy
  AND Updated = @Original_Updated
  AND UpdatedBy = @Original_UpdatedBy
IF @@ROWCOUNT = 0
  RAISERROR('50101', 14, 1);
```

2.44 Student_Course

2.44.1 Student_Course_GetCourses

```
CREATE PROCEDURE dbo.Student_Course_GetCourses
(
  @StudentVersion_ID uniqueidenti↓er
)
AS
SET NOCOUNT ON;
SELECT
  Student_Course_ID
FROM  Student_Course
WHERE StudentVersion_ID = @StudentVersion_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.44.2 Student_Course_GetIDsFromStudentAndCourse

```
CREATE PROCEDURE dbo.Student_Course_GetIDsFromStudentAndCourse
(
  @StudentVersion_ID uniqueidenti↓er,
  @CourseVersion_ID uniqueidenti↓er
)
AS
SET NOCOUNT ON;
SELECT
  Student_Course_ID
FROM  Student_Course
WHERE
  StudentVersion_ID = @StudentVersion_ID
AND CourseVersion_ID = @CourseVersion_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.44.3 Student_Course_Select

```
CREATE PROCEDURE dbo.Student_Course_Select
(
  @Student_Course_ID uniqueidenti↓er
)
AS
SET NOCOUNT ON;
SELECT
  Student_Course_ID,
  StudentVersion_ID,
  CourseVersion_ID,
  Part,
  DateOfSignUp,
  DateOfCancel,
  DateOfExaminationSignUp,
  DateOfExaminationCancel,
  Assessment_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM Student_Course
WHERE Student_Course_ID = @Student_Course_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.45 Student_Department

2.45.1 Student_Department_Select

```
CREATE PROCEDURE dbo.Student_Department_Select
(
  @Student_Department_ID uniqueidenti↓er
)
AS
SET NOCOUNT ON;
SELECT
  Student_Department_ID,
  StudentVersion_ID,
  Department_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM Student_Department
WHERE Student_Department_ID = @Student_Department_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.46 Student_Project

2.46.1 Student_Project_Select

```
CREATE PROCEDURE dbo.Student_Project_Select
(
  @Student_Project_ID uniqueidenti↓er
)
AS
SET NOCOUNT ON;
SELECT
  Student_Project_ID,
  StudentVersion_ID,
  Project_ID,
  Assessment_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM Student_Project
WHERE Student_Project_ID = @Student_Project_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.47 Student_StudyPlan

2.47.1 Student_StudyPlan_Delete

```
CREATE PROCEDURE dbo.Student_StudyPlan_Delete
(
  @Original_Student_StudyPlan_ID uniqueidenti↓er,
  @Original_Student_ID uniqueidenti↓er,
  @Original_StudyPlan_ID uniqueidenti↓er,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
DELETE FROM Student_StudyPlan
WHERE
  (Student_StudyPlan_ID = @Original_Student_StudyPlan_ID) AND
  (Student_ID = @Original_Student_ID) AND
  (StudyPlan_ID = @Original_StudyPlan_ID) AND
  (Created = @Original_Created) AND
  (CreatedBy = @Original_CreatedBy) AND
  (Updated = @Original_Updated) AND
  (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50201', 14, 1);
```

2.47.2 Student_StudyPlan_GetID

2.47.3 Student_StudyPlan_GetStudyPlans

```
CREATE PROCEDURE dbo.Student_StudyPlan_GetStudyPlans
(
  @Student_ID uniqueidenti↓er
)
AS
SET NOCOUNT ON;
SELECT
  StudyPlan_ID
FROM  Student_StudyPlan
WHERE  Student_ID = @Student_ID
```

2.47.4 Student_StudyPlan_Insert

```
CREATE PROCEDURE dbo.Student_StudyPlan_Insert
(
  @Student_StudyPlan_ID uniqueidenti↓er,
  @Student_ID uniqueidenti↓er,
  @StudyPlan_ID uniqueidenti↓er,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er
```

```
)
AS
SET NOCOUNT OFF;
INSERT INTO Student_StudyPlan (
Student_StudyPlan_ID,
Student_ID,
StudyPlan_ID,
Created,
CreatedBy,
Updated,
UpdatedBy
)
VALUES (
@Student_StudyPlan_ID,
@Student_ID,
@StudyPlan_ID,
@Created,
@CreatedBy,
@Updated,
@UpdatedBy
);
```

2.47.5 Student_StudyPlan_Select

```
CREATE PROCEDURE dbo.Student_StudyPlan_Select
(
  @Student_StudyPlan_ID uniqueidenti¦er
)
AS
SET NOCOUNT ON;
SELECT
  Student_StudyPlan_ID,
  Student_ID,
  StudyPlan_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM Student_StudyPlan
WHERE Student_StudyPlan_ID = @Student_StudyPlan_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.47.6 Student_StudyPlan_Update

```
CREATE PROCEDURE dbo.Student_StudyPlan_Update
(
  @Student_StudyPlan_ID uniqueidenti¦er,
  @Student_ID uniqueidenti¦er,
  @StudyPlan_ID uniqueidenti¦er,
  @Created datetime,
  @CreatedBy uniqueidenti¦er,
  @Updated datetime,
  @UpdatedBy uniqueidenti¦er,
  @Original_Student_StudyPlan_ID uniqueidenti¦er,
  @Original_Student_ID uniqueidenti¦er,
  @Original_StudyPlan_ID uniqueidenti¦er,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti¦er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti¦er
)
AS
SET NOCOUNT OFF;
UPDATE Student_StudyPlan
SET
  Student_StudyPlan_ID = @Student_StudyPlan_ID,
  Student_ID = @Student_ID,
  StudyPlan_ID = @StudyPlan_ID,
  Created = @Created,
  CreatedBy = @CreatedBy,
  Updated = @Updated,
  UpdatedBy = @UpdatedBy
WHERE
  (Student_StudyPlan_ID = @Original_Student_StudyPlan_ID) AND
  (Student_ID = @Original_Student_ID) AND
  (StudyPlan_ID = @Original_StudyPlan_ID) AND
  (Created = @Original_Created) AND
  (CreatedBy = @Original_CreatedBy) AND
  (Updated = @Original_Updated) AND
  (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50101', 14, 1);
```

2.48 Student_StudyPlanCriterion

2.48.1 Student_StudyPlanCriterion_Delete

```
CREATE PROCEDURE dbo.Student_StudyPlanCriterion_Delete
(
  @Original_Student_StudyPlanCriterion_ID uniqueidenti↓er,
  @Original_Student_ID uniqueidenti↓er,
  @Original_StudyPlanCriterion_ID uniqueidenti↓er,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
DELETE FROM Student_StudyPlanCriterion
WHERE
  (Student_StudyPlanCriterion_ID = @Original_Student_StudyPlanCriterion_ID) AND
  (Student_ID = @Original_Student_ID) AND
  (StudyPlanCriterion_ID = @Original_StudyPlanCriterion_ID) AND
  (Created = @Original_Created) AND
  (CreatedBy = @Original_CreatedBy) AND
  (Updated = @Original_Updated) AND
  (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50201', 14, 1);
```

2.48.2 Student_StudyPlanCriterion_GetID

2.48.3 Student_StudyPlanCriterion_GetStudyPlanCriteria

```
CREATE PROCEDURE dbo.Student_StudyPlanCriterion_GetStudyPlanCriteria
(
  @Student_ID uniqueidenti↓er
)
AS
SET NOCOUNT ON;
SELECT
  StudyPlanCriterion_ID
FROM  Student_StudyPlanCriterion
WHERE  Student_ID = @Student_ID
```

2.48.4 Student_StudyPlanCriterion_Insert

```
CREATE PROCEDURE dbo.Student_StudyPlanCriterion_Insert
(
  @Student_StudyPlanCriterion_ID uniqueidenti↓er,
  @Student_ID uniqueidenti↓er,
  @StudyPlanCriterion_ID uniqueidenti↓er,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
INSERT INTO Student_StudyPlanCriterion (
Student_StudyPlanCriterion_ID,
Student_ID,
StudyPlanCriterion_ID,
Created,
CreatedBy,
Updated,
UpdatedBy
)
VALUES (
@Student_StudyPlanCriterion_ID,
@Student_ID,
@StudyPlanCriterion_ID,
@Created,
@CreatedBy,
@Updated,
@UpdatedBy
);
```

2.48.5 Student_StudyPlanCriterion_Select

```
CREATE PROCEDURE dbo.Student_StudyPlanCriterion_Select
(
  @Student_StudyPlanCriterion_ID uniqueidenti↓er
)
AS
SET NOCOUNT ON;
SELECT
  Student_StudyPlanCriterion_ID,
  Student_ID,
  StudyPlanCriterion_ID,
  Created,
  CreatedBy,
```

```
  Updated,
  UpdatedBy
FROM Student_StudyPlanCriterion
WHERE Student_StudyPlanCriterion_ID = @Student_StudyPlanCriterion_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.48.6 Student_StudyPlanCriterion_Update

```
CREATE PROCEDURE dbo.Student_StudyPlanCriterion_Update
(
  @Student_StudyPlanCriterion_ID uniqueidenti↓er,
  @Student_ID uniqueidenti↓er,
  @StudyPlanCriterion_ID uniqueidenti↓er,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er,
  @Original_Student_StudyPlanCriterion_ID uniqueidenti↓er,
  @Original_Student_ID uniqueidenti↓er,
  @Original_StudyPlanCriterion_ID uniqueidenti↓er,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
UPDATE Student_StudyPlanCriterion
SET
  Student_StudyPlanCriterion_ID = @Student_StudyPlanCriterion_ID,
  Student_ID = @Student_ID,
  StudyPlanCriterion_ID = @StudyPlanCriterion_ID,
  Created = @Created,
  CreatedBy = @CreatedBy,
  Updated = @Updated,
  UpdatedBy = @UpdatedBy
WHERE
  (Student_StudyPlanCriterion_ID = @Original_Student_StudyPlanCriterion_ID) AND
  (Student_ID = @Original_Student_ID) AND
  (StudyPlanCriterion_ID = @Original_StudyPlanCriterion_ID) AND
  (Created = @Original_Created) AND
  (CreatedBy = @Original_CreatedBy) AND
  (Updated = @Original_Updated) AND
  (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50101', 14, 1);
```

2.49 StudentVersion

2.49.1 StudentVersion_Select

```
CREATE PROCEDURE dbo.StudentVersion_Select
(
  @StudentVersion_ID uniqueidenti↓er
)
AS
SET NOCOUNT ON;
SELECT
  StudentVersion_ID,
  Student_ID,
  Version,
  StudyNumber,
  DateOfSignUp,
  Person_ID,
  StudyTypeVersion_ID,
  TechnicalPackageVersion_ID,
  TechnicalLineVersion_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM StudentVersion
WHERE StudentVersion_ID = @StudentVersion_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.50 StudyPlan

2.50.1 StudyPlan_Delete

```
CREATE PROCEDURE dbo.StudyPlan_Delete
(
  @Original_StudyPlan_ID uniqueidenti↓er,
  @Original_Name varchar(200),
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
DELETE FROM StudyPlan
WHERE
  (StudyPlan_ID = @Original_StudyPlan_ID) AND
  (Name = @Original_Name) AND
  (Created = @Original_Created) AND
  (CreatedBy = @Original_CreatedBy) AND
  (Updated = @Original_Updated) AND
  (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50201', 14, 1);
```

2.50.2 StudyPlan_Insert

```
CREATE PROCEDURE dbo.StudyPlan_Insert
(
  @StudyPlan_ID uniqueidenti↓er,
  @Name varchar(200),
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
INSERT INTO StudyPlan (
StudyPlan_ID,
Name,
Created,
CreatedBy,
Updated,
UpdatedBy
)
VALUES (
@StudyPlan_ID,
@Name,
@Created,
@CreatedBy,
@Updated,
@UpdatedBy
);
```

2.50.3 StudyPlan_Select

```
CREATE PROCEDURE dbo.StudyPlan_Select
(
  @StudyPlan_ID uniqueidenti↓er
)
AS
SET NOCOUNT ON;
SELECT
  StudyPlan_ID,
  Name,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM StudyPlan
WHERE StudyPlan_ID = @StudyPlan_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.50.4 StudyPlan_Update

```
CREATE PROCEDURE dbo.StudyPlan_Update
(
  @StudyPlan_ID uniqueidenti↓er,
  @Name varchar(200),
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er,
  @Original_StudyPlan_ID uniqueidenti↓er,
  @Original_Name varchar(200),
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
```

```
)
AS
SET NOCOUNT OFF;
UPDATE StudyPlan
SET
  StudyPlan_ID = @StudyPlan_ID,
  Name = @Name,
  Created = @Created,
  CreatedBy = @CreatedBy,
  Updated = @Updated,
  UpdatedBy = @UpdatedBy
WHERE
  (StudyPlan_ID = @Original_StudyPlan_ID) AND
  (Name = @Original_Name) AND
  (Created = @Original_Created) AND
  (CreatedBy = @Original_CreatedBy) AND
  (Updated = @Original_Updated) AND
  (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50101', 14, 1);
```

2.51 StudyPlan_Period

2.51.1 StudyPlan_Period_Delete

```
CREATE PROCEDURE dbo.StudyPlan_Period_Delete
(
  @Original_StudyPlan_Period_ID uniqueidenti↓er,
  @Original_StudyPlan_ID uniqueidenti↓er,
  @Original_Period_ID uniqueidenti↓er,
  @Original_Project_ID uniqueidenti↓er,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
DELETE FROM StudyPlan_Period
WHERE
  (StudyPlan_Period_ID = @Original_StudyPlan_Period_ID) AND
  (StudyPlan_ID = @Original_StudyPlan_ID) AND
  (Period_ID = @Original_Period_ID) AND
  (Project_ID = @Original_Project_ID) AND
  (Created = @Original_Created) AND
  (CreatedBy = @Original_CreatedBy) AND
  (Updated = @Original_Updated) AND
  (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50201', 14, 1);
```

2.51.2 StudyPlan_Period_GetPeriods

2.51.3 StudyPlan_Period_Insert

```
CREATE PROCEDURE dbo.StudyPlan_Period_Insert
(
  @StudyPlan_Period_ID uniqueidenti↓er,
  @StudyPlan_ID uniqueidenti↓er,
  @Period_ID uniqueidenti↓er,
  @Project_ID uniqueidenti↓er,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
INSERT INTO StudyPlan_Period (
StudyPlan_Period_ID,
StudyPlan_ID,
Period_ID,
Project_ID,
Created,
CreatedBy,
Updated,
UpdatedBy
)
VALUES (
@StudyPlan_Period_ID,
@StudyPlan_ID,
@Period_ID,
@Project_ID,
@Created,
@CreatedBy,
@Updated,
```

```
@UpdatedBy
);
```

2.51.4 StudyPlan_Period_Select

```
CREATE PROCEDURE dbo.StudyPlan_Period_Select
(
  @StudyPlan_Period_ID uniqueidenti↓er
)
AS
SET NOCOUNT ON;
SELECT
  StudyPlan_Period_ID,
  StudyPlan_ID,
  Period_ID,
  Project_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM StudyPlan_Period
WHERE StudyPlan_Period_ID = @StudyPlan_Period_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.51.5 StudyPlan_Period_Update

```
CREATE PROCEDURE dbo.StudyPlan_Period_Update
(
  @StudyPlan_Period_ID uniqueidenti↓er,
  @StudyPlan_ID uniqueidenti↓er,
  @Period_ID uniqueidenti↓er,
  @Project_ID uniqueidenti↓er,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er,
  @Original_StudyPlan_Period_ID uniqueidenti↓er,
  @Original_StudyPlan_ID uniqueidenti↓er,
  @Original_Period_ID uniqueidenti↓er,
  @Original_Project_ID uniqueidenti↓er,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
UPDATE StudyPlan_Period
SET
  StudyPlan_Period_ID = @StudyPlan_Period_ID,
  StudyPlan_ID = @StudyPlan_ID,
  Period_ID = @Period_ID,
  Project_ID = @Project_ID,
  Created = @Created,
  CreatedBy = @CreatedBy,
  Updated = @Updated,
  UpdatedBy = @UpdatedBy
WHERE
  (StudyPlan_Period_ID = @Original_StudyPlan_Period_ID) AND
  (StudyPlan_ID = @Original_StudyPlan_ID) AND
  (Period_ID = @Original_Period_ID) AND
  (Project_ID = @Original_Project_ID) AND
  (Created = @Original_Created) AND
  (CreatedBy = @Original_CreatedBy) AND
  (Updated = @Original_Updated) AND
  (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50101', 14, 1);
```

2.52 StudyPlan_PeriodCourse

2.52.1 StudyPlan_PeriodCourse_Delete

```
CREATE PROCEDURE dbo.StudyPlan_PeriodCourse_Delete
(
  @Original_StudyPlan_PeriodCourse_ID uniqueidenti↓er,
  @Original_StudyPlan_Period_ID uniqueidenti↓er,
  @Original_CourseVersion_ID uniqueidenti↓er,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
```

```
AS
SET NOCOUNT OFF;
DELETE FROM StudyPlan_PeriodCourse
WHERE
  (StudyPlan_PeriodCourse_ID = @Original_StudyPlan_PeriodCourse_ID) AND
  (StudyPlan_Period_ID = @Original_StudyPlan_Period_ID) AND
  (CourseVersion_ID = @Original_CourseVersion_ID) AND
  (Created = @Original_Created) AND
  (CreatedBy = @Original_CreatedBy) AND
  (Updated = @Original_Updated) AND
  (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50201', 14, 1);
```

2.52.2 StudyPlan_PeriodCourse_GetCourses

2.52.3 StudyPlan_PeriodCourse_Insert

```
CREATE PROCEDURE dbo.StudyPlan_PeriodCourse_Insert
(
  @StudyPlan_PeriodCourse_ID uniqueidenti↓er,
  @StudyPlan_Period_ID uniqueidenti↓er,
  @CourseVersion_ID uniqueidenti↓er,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
INSERT INTO StudyPlan_PeriodCourse (
StudyPlan_PeriodCourse_ID,
StudyPlan_Period_ID,
CourseVersion_ID,
Created,
CreatedBy,
Updated,
UpdatedBy
)
VALUES (
@StudyPlan_PeriodCourse_ID,
@StudyPlan_Period_ID,
@CourseVersion_ID,
@Created,
@CreatedBy,
@Updated,
@UpdatedBy
);
```

2.52.4 StudyPlan_PeriodCourse_Select

```
CREATE PROCEDURE dbo.StudyPlan_PeriodCourse_Select
(
  @StudyPlan_PeriodCourse_ID uniqueidenti↓er
)
AS
SET NOCOUNT ON;
SELECT
  StudyPlan_PeriodCourse_ID,
  StudyPlan_Period_ID,
  CourseVersion_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM StudyPlan_PeriodCourse
WHERE StudyPlan_PeriodCourse_ID = @StudyPlan_PeriodCourse_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.52.5 StudyPlan_PeriodCourse_Update

```
CREATE PROCEDURE dbo.StudyPlan_PeriodCourse_Update
(
  @StudyPlan_PeriodCourse_ID uniqueidenti↓er,
  @StudyPlan_Period_ID uniqueidenti↓er,
  @CourseVersion_ID uniqueidenti↓er,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er,
  @Original_StudyPlan_PeriodCourse_ID uniqueidenti↓er,
  @Original_StudyPlan_Period_ID uniqueidenti↓er,
  @Original_CourseVersion_ID uniqueidenti↓er,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
```

```
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
UPDATE StudyPlan_PeriodCourse
SET
  StudyPlan_PeriodCourse_ID = @StudyPlan_PeriodCourse_ID,
  StudyPlan_Period_ID = @StudyPlan_Period_ID,
  CourseVersion_ID = @CourseVersion_ID,
  Created = @Created,
  CreatedBy = @CreatedBy,
  Updated = @Updated,
  UpdatedBy = @UpdatedBy
WHERE
  (StudyPlan_PeriodCourse_ID = @Original_StudyPlan_PeriodCourse_ID) AND
  (StudyPlan_Period_ID = @Original_StudyPlan_Period_ID) AND
  (CourseVersion_ID = @Original_CourseVersion_ID) AND
  (Created = @Original_Created) AND
  (CreatedBy = @Original_CreatedBy) AND
  (Updated = @Original_Updated) AND
  (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50101', 14, 1);
```

2.53 StudyPlanCriterion

2.53.1 StudyPlanCriterion_Delete

```
CREATE PROCEDURE dbo.StudyPlanCriterion_Delete
(
  @Original_StudyPlanCriterion_ID uniqueidenti↓er,
  @Original_Name varchar(200),
  @Original_AllowPointBlockCourses bit,
  @Original_RecommendedPlacementCogent bit,
  @Original_AllowForTechnicalPrerequisites bit,
  @Original_TechnicalPackageVersion_ID int,
  @Original_TechnicalLineVersion_ID int,
  @Original_TechnicalFieldVersion_ID int,
  @Original_SpecializationVersion_ID int,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
DELETE FROM StudyPlanCriterion
WHERE (StudyPlanCriterion_ID = @Original_StudyPlanCriterion_ID)
AND (Name = @Original_Name)
AND (AllowPointBlockCourses = @Original_AllowPointBlockCourses)
AND (RecommendedPlacementCogent = @Original_RecommendedPlacementCogent)
AND (AllowForTechnicalPrerequisites = @Original_AllowForTechnicalPrerequisites)
AND (TechnicalPackageVersion_ID = @Original_TechnicalPackageVersion_ID)
AND (TechnicalLineVersion_ID = @Original_TechnicalLineVersion_ID)
AND (TechnicalFieldVersion_ID = @Original_TechnicalFieldVersion_ID)
AND (SpecializationVersion_ID = @Original_SpecializationVersion_ID)
AND (Created = @Original_Created)
AND (CreatedBy = @Original_CreatedBy)
AND (Updated = @Original_Updated)
AND (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50201', 14, 1);
```

2.53.2 StudyPlanCriterion_Insert

```
CREATE PROCEDURE dbo.StudyPlanCriterion_Insert
(
  @StudyPlanCriterion_ID uniqueidenti↓er,
  @Name varchar(200),
  @AllowPointBlockCourses bit,
  @RecommendedPlacementCogent bit,
  @AllowForTechnicalPrerequisites bit,
  @TechnicalPackageVersion_ID int,
  @TechnicalLineVersion_ID int,
  @TechnicalFieldVersion_ID int,
  @SpecializationVersion_ID int,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
INSERT INTO StudyPlanCriterion
      (StudyPlanCriterion_ID,
  Name,
  AllowPointBlockCourses,
```

```
  RecommendedPlacementCogent,
  AllowForTechnicalPrerequisites,
  TechnicalPackageVersion_ID,
  TechnicalLineVersion_ID,
  TechnicalFieldVersion_ID,
  SpecializationVersion_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy)
VALUES (@StudyPlanCriterion_ID,
  @Name,
  @AllowPointBlockCourses,
  @RecommendedPlacementCogent,
  @AllowForTechnicalPrerequisites,
  @TechnicalPackageVersion_ID,
  @TechnicalLineVersion_ID,
  @TechnicalFieldVersion_ID,
  @SpecializationVersion_ID,
  @Created,
  @CreatedBy,
  @Updated,
  @UpdatedBy);
```

2.53.3 StudyPlanCriterion_Select

```
CREATE PROCEDURE dbo.StudyPlanCriterion_Select
(
  @StudyPlanCriterion_ID uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
SELECT StudyPlanCriterion_ID,
  Name,
  AllowPointBlockCourses,
  RecommendedPlacementCogent,
  AllowForTechnicalPrerequisites,
  TechnicalPackageVersion_ID,
  TechnicalLineVersion_ID,
  TechnicalFieldVersion_ID,
  SpecializationVersion_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM  StudyPlanCriterion
WHERE StudyPlanCriterion_ID = @StudyPlanCriterion_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.53.4 StudyPlanCriterion_Update

```
CREATE PROCEDURE dbo.StudyPlanCriterion_Update
(
  @StudyPlanCriterion_ID uniqueidenti↓er,
  @Name varchar(200),
  @AllowPointBlockCourses bit,
  @RecommendedPlacementCogent bit,
  @AllowForTechnicalPrerequisites bit,
  @TechnicalPackageVersion_ID int,
  @TechnicalLineVersion_ID int,
  @TechnicalFieldVersion_ID int,
  @SpecializationVersion_ID int,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er,
  @Original_StudyPlanCriterion_ID uniqueidenti↓er,
  @Original_Name varchar(200),
  @Original_AllowPointBlockCourses bit,
  @Original_RecommendedPlacementCogent bit,
  @Original_AllowForTechnicalPrerequisites bit,
  @Original_TechnicalPackageVersion_ID int,
  @Original_TechnicalLineVersion_ID int,
  @Original_TechnicalFieldVersion_ID int,
  @Original_SpecializationVersion_ID int,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
UPDATE  StudyPlanCriterion
SET  StudyPlanCriterion_ID  = @StudyPlanCriterion_ID,
  Name     = @Name,
  AllowPointBlockCourses   = @AllowPointBlockCourses,
  RecommendedPlacementCogent  = @RecommendedPlacementCogent,
  AllowForTechnicalPrerequisites = @AllowForTechnicalPrerequisites,
  TechnicalPackageVersion_ID  = @TechnicalPackageVersion_ID,
  TechnicalLineVersion_ID  = @TechnicalLineVersion_ID,
  TechnicalFieldVersion_ID = @TechnicalFieldVersion_ID,
```

```
  SpecializationVersion_ID  = @SpecializationVersion_ID,
  Created    = @Created,
  CreatedBy    = @CreatedBy,
  Updated    = @Updated,
  UpdatedBy    = @UpdatedBy
WHERE (StudyPlanCriterion_ID  = @Original_StudyPlanCriterion_ID)
AND (Name     = @Original_Name)
AND (AllowPointBlockCourses  = @Original_AllowPointBlockCourses)
AND (RecommendedPlacementCogent  = @Original_RecommendedPlacementCogent)
AND (AllowForTechnicalPrerequisites = @Original_AllowForTechnicalPrerequisites)
AND (TechnicalPackageVersion_ID  = @Original_TechnicalPackageVersion_ID)
AND (TechnicalLineVersion_ID = @Original_TechnicalLineVersion_ID)
AND (TechnicalFieldVersion_ID = @Original_TechnicalFieldVersion_ID)
AND (SpecializationVersion_ID  = @Original_SpecializationVersion_ID)
AND (Created    = @Original_Created)
AND (CreatedBy    = @Original_CreatedBy)
AND (Updated    = @Original_Updated)
AND (UpdatedBy    = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50101', 14, 1);
```

2.54 StudyPlanCriterion_Course

2.54.1 StudyPlanCriterion_Course_Delete

```
CREATE PROCEDURE dbo.StudyPlanCriterion_Course_Delete
(
  @Original_StudyPlanCriterion_Course_ID uniqueidenti↓er,
  @Original_StudyPlanCriterion_ID uniqueidenti↓er,
  @Original_Course_ID uniqueidenti↓er,
  @Original_AdditionalChoice bit,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
DELETE FROM StudyPlanCriterion_Course
WHERE (StudyPlanCriterion_Course_ID  = @Original_StudyPlanCriterion_Course_ID)
AND (StudyPlanCriterion_ID  = @Original_StudyPlanCriterion_ID)
AND (Course_ID    = @Original_Course_ID)
AND (AdditionalChoice   = @Original_AdditionalChoice)
AND (Created    = @Original_Created)
AND (CreatedBy    = @Original_CreatedBy)
AND (Updated    = @Original_Updated)
AND (UpdatedBY    = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50201', 14, 1);
```

2.54.2 StudyPlanCriterion_Course_GetCourses

```
CREATE PROCEDURE dbo.StudyPlanCriterion_Course_GetCourses
(
  @StudyPlanCriterion_ID uniqueidenti↓er,
  @AdditionalChoice bit
)
AS
SET NOCOUNT OFF;
SELECT Course_ID
FROM StudyPlanCriterion_Course
WHERE StudyPlanCriterion_ID = @StudyPlanCriterion_ID
AND AdditionalChoice = @AdditionalChoice
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.54.3 StudyPlanCriterion_Course_Insert

```
CREATE PROCEDURE dbo.StudyPlanCriterion_Course_Insert
(
  @StudyPlanCriterion_Course_ID uniqueidenti↓er,
  @StudyPlanCriterion_ID uniqueidenti↓er,
  @Course_ID uniqueidenti↓er,
  @AdditionalChoice bit,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
INSERT INTO StudyPlanCriterion_Course
       (StudyPlanCriterion_Course_ID,
  StudyPlanCriterion_ID,
```

```
  Course_ID,
  AdditionalChoice,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy)
VALUES (@StudyPlanCriterion_Course_ID,
  @StudyPlanCriterion_ID,
  @Course_ID,
  @AdditionalChoice,
  @Created,
  @CreatedBy,
  @Updated,
  @UpdatedBy);
```

2.54.4 StudyPlanCriterion_Course_Select

```
CREATE PROCEDURE dbo.StudyPlanCriterion_Course_Select
(
  @StudyPlanCriterion_Course_ID uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
SELECT StudyPlanCriterion_Course_ID,
  StudyPlanCriterion_ID,
  Course_ID,
  AdditionalChoice,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM StudyPlanCriterion_Course
WHERE StudyPlanCriterion_Course_ID = @StudyPlanCriterion_Course_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.54.5 StudyPlanCriterion_Course_Update

```
CREATE PROCEDURE dbo.StudyPlanCriterion_Course_Update
(
  @StudyPlanCriterion_Course_ID uniqueidenti↓er,
  @StudyPlanCriterion_ID uniqueidenti↓er,
  @Course_ID uniqueidenti↓er,
  @AdditionalChoice bit,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er,
  @Original_StudyPlanCriterion_Course_ID uniqueidenti↓er,
  @Original_StudyPlanCriterion_ID uniqueidenti↓er,
  @Original_Course_ID uniqueidenti↓er,
  @Original_AdditionalChoice bit,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
UPDATE StudyPlanCriterion_Course
SET  StudyPlanCriterion_Course_ID  = @StudyPlanCriterion_Course_ID,
  StudyPlanCriterion_ID   = @StudyPlanCriterion_ID,
  Course_ID    = @Course_ID,
  AdditionalChoice   = @AdditionalChoice,
  Created    = @Created,
  CreatedBy    = @CreatedBy,
  Updated    = @Updated,
  UpdatedBy    = @UpdatedBy
WHERE  (StudyPlanCriterion_Course_ID  = @Original_StudyPlanCriterion_Course_ID)
AND (StudyPlanCriterion_ID  = @Original_StudyPlanCriterion_ID)
AND (Course_ID    = @Original_Course_ID)
AND (AdditionalChoice   = @Original_AdditionalChoice)
AND (Created    = @Original_Created)
AND (CreatedBy    = @Original_CreatedBy)
AND (Updated    = @Original_Updated)
AND (UpdatedBY    = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50101', 14, 1);
```

2.55 StudyPlanCriterion_CoursePeriod

2.55.1 StudyPlanCriterion_CoursePeriod_Delete

```
CREATE PROCEDURE dbo.StudyPlanCriterion_CoursePeriod_Delete
(
  @Original_StudyPlanCriterion_CoursePeriod_ID uniqueidenti↓er,
  @Original_StudyPlanCriterion_ID uniqueidenti↓er,
  @Original_Period_ID uniqueidenti↓er,
  @Original_Course_ID uniqueidenti↓er,
  @Original_AdditionalChoice bit,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
DELETE FROM StudyPlanCriterion_CoursePeriod
WHERE (StudyPlanCriterion_CoursePeriod_ID = @Original_StudyPlanCriterion_CoursePeriod_ID)
AND (StudyPlanCriterion_ID = @Original_StudyPlanCriterion_ID)
AND (Period_ID = @Original_Period_ID)
AND (Course_ID = @Original_Course_ID)
AND (AdditionalChoice = @Original_AdditionalChoice)
AND (Created = @Original_Created)
AND (CreatedBy = @Original_CreatedBy)
AND (Updated = @Original_Updated)
AND (UpdatedBY = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50201', 14, 1);
```

2.55.2 StudyPlanCriterion_CoursePeriod_GetCourses

```
CREATE PROCEDURE dbo.StudyPlanCriterion_CoursePeriod_GetCourses
(
  @StudyPlanCriterion_ID uniqueidenti↓er,
  @Period_ID uniqueidenti↓er,
  @AdditionalChoice bit
)
AS
SET NOCOUNT OFF;
SELECT Course_ID
FROM StudyPlanCriterion_CoursePeriod
WHERE StudyPlanCriterion_ID = @StudyPlanCriterion_ID
AND Period_ID = @Period_ID
AND AdditionalChoice = @AdditionalChoice
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.55.3 StudyPlanCriterion_CoursePeriod_Insert

```
CREATE PROCEDURE dbo.StudyPlanCriterion_CoursePeriod_Insert
(
  @StudyPlanCriterion_CoursePeriod_ID uniqueidenti↓er,
  @StudyPlanCriterion_ID uniqueidenti↓er,
  @Period_ID uniqueidenti↓er,
  @Course_ID uniqueidenti↓er,
  @AdditionalChoice bit,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
INSERT INTO StudyPlanCriterion_CoursePeriod
      (StudyPlanCriterion_CoursePeriod_ID,
  StudyPlanCriterion_ID,
  Period_ID,
  Course_ID,
  AdditionalChoice,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy)
VALUES (@StudyPlanCriterion_CoursePeriod_ID,
  @StudyPlanCriterion_ID,
  @Period_ID,
  @Course_ID,
  @AdditionalChoice,
  @Created,
  @CreatedBy,
  @Updated,
  @UpdatedBy);
```

2.55.4 StudyPlanCriterion_CoursePeriod_Select

```
CREATE PROCEDURE dbo.StudyPlanCriterion_CoursePeriod_Select
(
  @StudyPlanCriterion_CoursePeriod_ID uniqueidenti↓er
)
AS
```

```
SET NOCOUNT OFF;
SELECT StudyPlanCriterion_CoursePeriod_ID,
  StudyPlanCriterion_ID,
  Period_ID,
  Course_ID,
  AdditionalChoice,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM StudyPlanCriterion_CoursePeriod
WHERE StudyPlanCriterion_CoursePeriod_ID = @StudyPlanCriterion_CoursePeriod_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.55.5 StudyPlanCriterion_CoursePeriod_Update

```
CREATE PROCEDURE dbo.StudyPlanCriterion_CoursePeriod_Update
(
  @StudyPlanCriterion_CoursePeriod_ID uniqueidenti↓er,
  @StudyPlanCriterion_ID uniqueidenti↓er,
  @Period_ID uniqueidenti↓er,
  @Course_ID uniqueidenti↓er,
  @AdditionalChoice bit,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er,
  @Original_StudyPlanCriterion_CoursePeriod_ID uniqueidenti↓er,
  @Original_StudyPlanCriterion_ID uniqueidenti↓er,
  @Original_Period_ID uniqueidenti↓er,
  @Original_Course_ID uniqueidenti↓er,
  @Original_AdditionalChoice bit,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
UPDATE StudyPlanCriterion_CoursePeriod
SET StudyPlanCriterion_CoursePeriod_ID = @StudyPlanCriterion_CoursePeriod_ID,
  StudyPlanCriterion_ID = @StudyPlanCriterion_ID,
  Period_ID = @Period_ID,
  Course_ID = @Course_ID,
  AdditionalChoice = @AdditionalChoice,
  Created = @Created,
  CreatedBy = @CreatedBy,
  Updated = @Updated,
  UpdatedBy = @UpdatedBy
WHERE (StudyPlanCriterion_CoursePeriod_ID = @Original_StudyPlanCriterion_CoursePeriod_ID)
AND (StudyPlanCriterion_ID = @Original_StudyPlanCriterion_ID)
AND (Period_ID = @Original_Period_ID)
AND (Course_ID = @Original_Course_ID)
AND (AdditionalChoice = @Original_AdditionalChoice)
AND (Created = @Original_Created)
AND (CreatedBy = @Original_CreatedBy)
AND (Updated = @Original_Updated)
AND (UpdatedBY = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50101', 14, 1);
```

2.56 StudyPlanCriterion_EvaluationForm

2.56.1 StudyPlanCriterion_EvaluationForm_Delete

```
CREATE PROCEDURE dbo.StudyPlanCriterion_EvaluationForm_Delete
(
  @Original_StudyPlanCriterion_EvaluationForm_ID uniqueidenti↓er,
  @Original_StudyPlanCriterion_ID uniqueidenti↓er,
  @Original_EvaluationForm_ID int,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
DELETE FROM StudyPlanCriterion_EvaluationForm
WHERE (StudyPlanCriterion_EvaluationForm_ID = @Original_StudyPlanCriterion_EvaluationForm_ID)
AND (StudyPlanCriterion_ID = @Original_StudyPlanCriterion_ID)
AND (EvaluationForm_ID = @Original_EvaluationForm_ID)
AND (Created = @Original_Created)
AND (CreatedBy = @Original_CreatedBy)
AND (Updated = @Original_Updated)
AND (UpdatedBY = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50201', 14, 1);
```

2.56.2 StudyPlanCriterion_EvaluationForm_Insert

```
CREATE PROCEDURE dbo.StudyPlanCriterion_EvaluationForm_Insert
(
  @StudyPlanCriterion_EvaluationForm_ID uniqueidenti↓er,
  @StudyPlanCriterion_ID uniqueidenti↓er,
  @EvaluationForm_ID int,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
INSERT INTO StudyPlanCriterion_EvaluationForm
       (StudyPlanCriterion_EvaluationForm_ID,
  StudyPlanCriterion_ID,
  EvaluationForm_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy)
VALUES (@StudyPlanCriterion_EvaluationForm_ID,
  @StudyPlanCriterion_ID,
  @EvaluationForm_ID,
  @Created,
  @CreatedBy,
  @Updated,
  @UpdatedBy);
```

2.56.3 StudyPlanCriterion_EvaluationForm_Select

```
CREATE PROCEDURE dbo.StudyPlanCriterion_EvaluationForm_Select
(
  @StudyPlanCriterion_EvaluationForm_ID uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
SELECT StudyPlanCriterion_EvaluationForm_ID,
  StudyPlanCriterion_ID,
  EvaluationForm_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM StudyPlanCriterion_EvaluationForm
WHERE StudyPlanCriterion_EvaluationForm_ID = @StudyPlanCriterion_EvaluationForm_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.56.4 StudyPlanCriterion_EvaluationForm_Update

```
CREATE PROCEDURE dbo.StudyPlanCriterion_EvaluationForm_Update
(
  @StudyPlanCriterion_EvaluationForm_ID uniqueidenti↓er,
  @StudyPlanCriterion_ID uniqueidenti↓er,
  @EvaluationForm_ID int,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er,
  @Original_StudyPlanCriterion_EvaluationForm_ID uniqueidenti↓er,
  @Original_StudyPlanCriterion_ID uniqueidenti↓er,
  @Original_EvaluationForm_ID int,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
UPDATE StudyPlanCriterion_EvaluationForm
SET  StudyPlanCriterion_EvaluationForm_ID  = @StudyPlanCriterion_EvaluationForm_ID,
  StudyPlanCriterion_ID    = @StudyPlanCriterion_ID,
  EvaluationForm_ID   = @EvaluationForm_ID,
  Created     = @Created,
  CreatedBy     = @CreatedBy,
  Updated     = @Updated,
  UpdatedBy     = @UpdatedBy
WHERE  (StudyPlanCriterion_EvaluationForm_ID = @Original_StudyPlanCriterion_EvaluationForm_ID)
AND (StudyPlanCriterion_ID    = @Original_StudyPlanCriterion_ID)
AND (EvaluationForm_ID   = @Original_EvaluationForm_ID)
AND (Created     = @Original_Created)
AND (CreatedBy     = @Original_CreatedBy)
AND (Updated     = @Original_Updated)
AND (UpdatedBY     = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50101', 14, 1);
```

2.57 StudyPlanCriterion_Keyword

2.57.1 StudyPlanCriterion_Keyword_Delete

```
CREATE PROCEDURE dbo.StudyPlanCriterion_Keyword_Delete
(
  @Original_StudyPlanCriterion_Keyword_ID uniqueidenti↓er,
  @Original_StudyPlanCriterion_ID uniqueidenti↓er,
  @Original_Keyword_ID uniqueidenti↓er,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
DELETE FROM StudyPlanCriterion_Keyword
WHERE  (StudyPlanCriterion_Keyword_ID  = @Original_StudyPlanCriterion_Keyword_ID)
AND (StudyPlanCriterion_ID   = @Original_StudyPlanCriterion_ID)
AND (Keyword_ID   = @Original_Keyword_ID)
AND (Created    = @Original_Created)
AND (CreatedBy    = @Original_CreatedBy)
AND (Updated    = @Original_Updated)
AND (UpdatedBY    = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50201', 14, 1);
```

2.57.2 StudyPlanCriterion_Keyword_GetKeywords

```
CREATE PROCEDURE dbo.StudyPlanCriterion_Keyword_GetKeywords
(
  @StudyPlanCriterion_ID uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
SELECT Keyword_ID
FROM StudyPlanCriterion_Keyword
WHERE StudyPlanCriterion_ID = @StudyPlanCriterion_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.57.3 StudyPlanCriterion_Keyword_Insert

```
CREATE PROCEDURE dbo.StudyPlanCriterion_Keyword_Insert
(
  @StudyPlanCriterion_Keyword_ID uniqueidenti↓er,
  @StudyPlanCriterion_ID uniqueidenti↓er,
  @Keyword_ID uniqueidenti↓er,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
INSERT INTO StudyPlanCriterion_Keyword
       (StudyPlanCriterion_Keyword_ID,
  StudyPlanCriterion_ID,
  Keyword_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy)
VALUES (@StudyPlanCriterion_Keyword_ID,
  @StudyPlanCriterion_ID,
  @Keyword_ID,
  @Created,
  @CreatedBy,
  @Updated,
  @UpdatedBy);
```

2.57.4 StudyPlanCriterion_Keyword_Select

```
CREATE PROCEDURE dbo.StudyPlanCriterion_Keyword_Select
(
  @StudyPlanCriterion_Keyword_ID uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
SELECT StudyPlanCriterion_Keyword_ID,
  StudyPlanCriterion_ID,
  Keyword_ID,
  Created,
  CreatedBy,
```

```
  Updated,
  UpdatedBy
FROM StudyPlanCriterion_Keyword
WHERE StudyPlanCriterion_Keyword_ID = @StudyPlanCriterion_Keyword_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.57.5 StudyPlanCriterion_Keyword_Update

```
CREATE PROCEDURE dbo.StudyPlanCriterion_Keyword_Update
(
  @StudyPlanCriterion_Keyword_ID uniqueidenti↓er,
  @StudyPlanCriterion_ID uniqueidenti↓er,
  @Keyword_ID uniqueidenti↓er,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er,
  @Original_StudyPlanCriterion_Keyword_ID uniqueidenti↓er,
  @Original_StudyPlanCriterion_ID uniqueidenti↓er,
  @Original_Keyword_ID uniqueidenti↓er,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
UPDATE StudyPlanCriterion_Keyword
SET  StudyPlanCriterion_Keyword_ID  = @StudyPlanCriterion_Keyword_ID,
  StudyPlanCriterion_ID  = @StudyPlanCriterion_ID,
  Keyword_ID  = @Keyword_ID,
  Created  = @Created,
  CreatedBy  = @CreatedBy,
  Updated  = @Updated,
  UpdatedBy  = @UpdatedBy
WHERE  (StudyPlanCriterion_Keyword_ID = @Original_StudyPlanCriterion_Keyword_ID)
AND (StudyPlanCriterion_ID  = @Original_StudyPlanCriterion_ID)
AND (Keyword_ID  = @Original_Keyword_ID)
AND (Created  = @Original_Created)
AND (CreatedBy  = @Original_CreatedBy)
AND (Updated  = @Original_Updated)
AND (UpdatedBY  = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50101', 14, 1);
```

2.58 StudyPlanCriterion_Language

2.58.1 StudyPlanCriterion_Language_Delete

```
CREATE PROCEDURE dbo.StudyPlanCriterion_Language_Delete
(
  @Original_StudyPlanCriterion_Language_ID uniqueidenti↓er,
  @Original_StudyPlanCriterion_ID uniqueidenti↓er,
  @Original_Language_ID char(2),
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
DELETE FROM StudyPlanCriterion_Language
WHERE  (StudyPlanCriterion_Language_ID = @Original_StudyPlanCriterion_Language_ID)
AND (StudyPlanCriterion_ID  = @Original_StudyPlanCriterion_ID)
AND (Language_ID  = @Original_Language_ID)
AND (Created  = @Original_Created)
AND (CreatedBy  = @Original_CreatedBy)
AND (Updated  = @Original_Updated)
AND (UpdatedBY  = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50201', 14, 1);
```

2.58.2 StudyPlanCriterion_Language_GetLanguages

```
CREATE PROCEDURE dbo.StudyPlanCriterion_Language_GetLanguages
(
  @StudyPlanCriterion_ID uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
SELECT Language_ID
FROM StudyPlanCriterion_Language
WHERE StudyPlanCriterion_ID = @StudyPlanCriterion_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.58.3 StudyPlanCriterion_Language_Insert

```
CREATE PROCEDURE dbo.StudyPlanCriterion_Language_Insert
(
  @StudyPlanCriterion_Language_ID uniqueidenti↓er,
  @StudyPlanCriterion_ID uniqueidenti↓er,
  @Language_ID char(2),
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
INSERT INTO StudyPlanCriterion_Language
       (StudyPlanCriterion_Language_ID,
  StudyPlanCriterion_ID,
  Language_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy)
VALUES (@StudyPlanCriterion_Language_ID,
  @StudyPlanCriterion_ID,
  @Language_ID,
  @Created,
  @CreatedBy,
  @Updated,
  @UpdatedBy);
```

2.58.4 StudyPlanCriterion_Language_Select

```
CREATE PROCEDURE dbo.StudyPlanCriterion_Language_Select
(
  @StudyPlanCriterion_Language_ID uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
SELECT StudyPlanCriterion_Language_ID,
  StudyPlanCriterion_ID,
  Language_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM StudyPlanCriterion_Language
WHERE StudyPlanCriterion_Language_ID = @StudyPlanCriterion_Language_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.58.5 StudyPlanCriterion_Language_Update

```
CREATE PROCEDURE dbo.StudyPlanCriterion_Language_Update
(
  @StudyPlanCriterion_Language_ID uniqueidenti↓er,
  @StudyPlanCriterion_ID uniqueidenti↓er,
  @Language_ID char(2),
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er,
  @Original_StudyPlanCriterion_Language_ID uniqueidenti↓er,
  @Original_StudyPlanCriterion_ID uniqueidenti↓er,
  @Original_Language_ID char(2),
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
UPDATE StudyPlanCriterion_Language
SET  StudyPlanCriterion_Language_ID  = @StudyPlanCriterion_Language_ID,
  StudyPlanCriterion_ID   = @StudyPlanCriterion_ID,
  Language_ID   = @Language_ID,
  Created    = @Created,
  CreatedBy    = @CreatedBy,
  Updated    = @Updated,
  UpdatedBy    = @UpdatedBy
WHERE  (StudyPlanCriterion_Language_ID = @Original_StudyPlanCriterion_Language_ID)
AND (StudyPlanCriterion_ID   = @Original_StudyPlanCriterion_ID)
AND (Language_ID   = @Original_Language_ID)
AND (Created    = @Original_Created)
AND (CreatedBy    = @Original_CreatedBy)
AND (Updated    = @Original_Updated)
AND (UpdatedBY    = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50101', 14, 1);
```

2.59 StudyPlanCriterion_Lecturer

2.59.1 StudyPlanCriterion_Lecturer_Delete

```
CREATE PROCEDURE dbo.StudyPlanCriterion_Lecturer_Delete
(
  @Original_StudyPlanCriterion_Lecturer_ID uniqueidenti↓er,
  @Original_StudyPlanCriterion_ID uniqueidenti↓er,
  @Original_Lecturer_ID uniqueidenti↓er,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
DELETE FROM StudyPlanCriterion_Lecturer
WHERE  (StudyPlanCriterion_Lecturer_ID = @Original_StudyPlanCriterion_Lecturer_ID)
AND (StudyPlanCriterion_ID   = @Original_StudyPlanCriterion_ID)
AND (Lecturer_ID   = @Original_Lecturer_ID)
AND (Created    = @Original_Created)
AND (CreatedBy    = @Original_CreatedBy)
AND (Updated    = @Original_Updated)
AND (UpdatedBY    = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50201', 14, 1);
```

2.59.2 StudyPlanCriterion_Lecturer_GetLecturers

```
CREATE PROCEDURE dbo.StudyPlanCriterion_Lecturer_GetLecturers
(
  @StudyPlanCriterion_ID uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
SELECT Lecturer_ID
FROM StudyPlanCriterion_Lecturer
WHERE StudyPlanCriterion_ID = @StudyPlanCriterion_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.59.3 StudyPlanCriterion_Lecturer_Insert

```
CREATE PROCEDURE dbo.StudyPlanCriterion_Lecturer_Insert
(
  @StudyPlanCriterion_Lecturer_ID uniqueidenti↓er,
  @StudyPlanCriterion_ID uniqueidenti↓er,
  @Lecturer_ID uniqueidenti↓er,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
INSERT INTO StudyPlanCriterion_Lecturer
       (StudyPlanCriterion_Lecturer_ID,
  StudyPlanCriterion_ID,
  Lecturer_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy)
VALUES (@StudyPlanCriterion_Lecturer_ID,
  @StudyPlanCriterion_ID,
  @Lecturer_ID,
  @Created,
  @CreatedBy,
  @Updated,
  @UpdatedBy);
```

2.59.4 StudyPlanCriterion_Lecturer_Select

```
CREATE PROCEDURE dbo.StudyPlanCriterion_Lecturer_Select
(
  @StudyPlanCriterion_Lecturer_ID uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
SELECT StudyPlanCriterion_Lecturer_ID,
  StudyPlanCriterion_ID,
  Lecturer_ID,
  Created,
  CreatedBy,
```

```
  Updated,
  UpdatedBy
FROM StudyPlanCriterion_Lecturer
WHERE StudyPlanCriterion_Lecturer_ID = @StudyPlanCriterion_Lecturer_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.59.5 StudyPlanCriterion_Lecturer_Update

```
CREATE PROCEDURE dbo.StudyPlanCriterion_Lecturer_Update
(
  @StudyPlanCriterion_Lecturer_ID uniqueidenti↓er,
  @StudyPlanCriterion_ID uniqueidenti↓er,
  @Lecturer_ID uniqueidenti↓er,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er,
  @Original_StudyPlanCriterion_Lecturer_ID uniqueidenti↓er,
  @Original_StudyPlanCriterion_ID uniqueidenti↓er,
  @Original_Lecturer_ID uniqueidenti↓er,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
UPDATE StudyPlanCriterion_Lecturer
SET  StudyPlanCriterion_Lecturer_ID  = @StudyPlanCriterion_Lecturer_ID,
  StudyPlanCriterion_ID   = @StudyPlanCriterion_ID,
  Lecturer_ID   = @Lecturer_ID,
  Created    = @Created,
  CreatedBy    = @CreatedBy,
  Updated    = @Updated,
  UpdatedBy    = @UpdatedBy
WHERE  (StudyPlanCriterion_Lecturer_ID = @Original_StudyPlanCriterion_Lecturer_ID)
AND (StudyPlanCriterion_ID   = @Original_StudyPlanCriterion_ID)
AND (Lecturer_ID   = @Original_Lecturer_ID)
AND (Created    = @Original_Created)
AND (CreatedBy    = @Original_CreatedBy)
AND (Updated    = @Original_Updated)
AND (UpdatedBY    = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50101', 14, 1);
```

2.60 StudyPlanCriterion_ProjectWorkload

2.60.1 StudyPlanCriterion_ProjectWorkload_Delete

2.60.2 StudyPlanCriterion_ProjectWorkload_GetID

2.60.3 StudyPlanCriterion_ProjectWorkload_Insert

2.60.4 StudyPlanCriterion_ProjectWorkload_Select

2.60.5 StudyPlanCriterion_ProjectWorkload_Update

2.61 StudyPlanCriterion_StudyType

2.61.1 StudyPlanCriterion_StudyType_Delete

```
CREATE PROCEDURE dbo.StudyPlanCriterion_StudyType_Delete
(
  @Original_StudyPlanCriterion_StudyType_ID uniqueidenti↓er,
  @Original_StudyPlanCriterion_ID uniqueidenti↓er,
  @Original_StudyType_ID int,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
```

```
SET NOCOUNT OFF;
DELETE FROM StudyPlanCriterion_StudyType
WHERE  (StudyPlanCriterion_StudyType_ID= @Original_StudyPlanCriterion_StudyType_ID)
AND (StudyPlanCriterion_ID   = @Original_StudyPlanCriterion_ID)
AND (StudyType_ID   = @Original_StudyType_ID)
AND (Created    = @Original_Created)
AND (CreatedBy    = @Original_CreatedBy)
AND (Updated    = @Original_Updated)
AND (UpdatedBY    = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50201', 14, 1);
```

2.61.2 StudyPlanCriterion_StudyType_GetStudyTypes

```
CREATE PROCEDURE dbo.StudyPlanCriterion_StudyType_GetStudyTypes
(
  @StudyPlanCriterion_ID uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
SELECT StudyType_ID
FROM StudyPlanCriterion_StudyType
WHERE StudyPlanCriterion_ID = @StudyPlanCriterion_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.61.3 StudyPlanCriterion_StudyType_Insert

```
CREATE PROCEDURE dbo.StudyPlanCriterion_StudyType_Insert
(
  @StudyPlanCriterion_StudyType_ID uniqueidenti↓er,
  @StudyPlanCriterion_ID uniqueidenti↓er,
  @StudyType_ID int,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
INSERT INTO StudyPlanCriterion_StudyType
       (StudyPlanCriterion_StudyType_ID,
  StudyPlanCriterion_ID,
  StudyType_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy)
VALUES (@StudyPlanCriterion_StudyType_ID,
  @StudyPlanCriterion_ID,
  @StudyType_ID,
  @Created,
  @CreatedBy,
  @Updated,
  @UpdatedBy);
```

2.61.4 StudyPlanCriterion_StudyType_Select

```
CREATE PROCEDURE dbo.StudyPlanCriterion_StudyType_Select
(
  @StudyPlanCriterion_StudyType_ID uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
SELECT StudyPlanCriterion_StudyType_ID,
  StudyPlanCriterion_ID,
  StudyType_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM StudyPlanCriterion_StudyType
WHERE StudyPlanCriterion_StudyType_ID = @StudyPlanCriterion_StudyType_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.61.5 StudyPlanCriterion_StudyType_Update

```
CREATE PROCEDURE dbo.StudyPlanCriterion_StudyType_Update
(
  @StudyPlanCriterion_StudyType_ID uniqueidenti↓er,
  @StudyPlanCriterion_ID uniqueidenti↓er,
  @StudyType_ID int,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
```

```
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er,
  @Original_StudyPlanCriterion_StudyType_ID uniqueidenti↓er,
  @Original_StudyPlanCriterion_ID uniqueidenti↓er,
  @Original_StudyType_ID int,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
UPDATE StudyPlanCriterion_StudyType
SET  StudyPlanCriterion_StudyType_ID = @StudyPlanCriterion_StudyType_ID,
  StudyPlanCriterion_ID   = @StudyPlanCriterion_ID,
  StudyType_ID   = @StudyType_ID,
  Created    = @Created,
  CreatedBy    = @CreatedBy,
  Updated    = @Updated,
  UpdatedBy    = @UpdatedBy
WHERE  (StudyPlanCriterion_StudyType_ID= @Original_StudyPlanCriterion_StudyType_ID)
AND (StudyPlanCriterion_ID   = @Original_StudyPlanCriterion_ID)
AND (StudyType_ID   = @Original_StudyType_ID)
AND (Created    = @Original_Created)
AND (CreatedBy    = @Original_CreatedBy)
AND (Updated    = @Original_Updated)
AND (UpdatedBY    = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50101', 14, 1);
```

2.62 StudyPlanCriterion_TechnicalPrerequisiteCourse

2.62.1 StudyPlanCriterion_TechnicalPrerequisiteCourse_Delete

```
CREATE PROCEDURE dbo.StudyPlanCriterion_TechnicalPrerequisiteCourse_Delete
(
  @Original_StudyPlanCriterion_TechnicalPrerequisiteCourse_ID uniqueidenti↓er,
  @Original_StudyPlanCriterion_ID uniqueidenti↓er,
  @Original_Course_ID uniqueidenti↓er,
  @Original_AdditionalChoice bit,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
DELETE FROM StudyPlanCriterion_TechnicalPrerequisiteCourse
WHERE  (StudyPlanCriterion_TechnicalPrerequisiteCourse_ID = @Original_StudyPlanCriterion_TechnicalPrerequisiteCourse_ID)
AND (StudyPlanCriterion_ID   = @Original_StudyPlanCriterion_ID)
AND (Course_ID    = @Original_Course_ID)
AND (AdditionalChoice   = @Original_AdditionalChoice)
AND (Created    = @Original_Created)
AND (CreatedBy    = @Original_CreatedBy)
AND (Updated    = @Original_Updated)
AND (UpdatedBY    = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50201', 14, 1);
```

2.62.2 StudyPlanCriterion_TechnicalPrerequisiteCourse_Insert

```
CREATE PROCEDURE dbo.StudyPlanCriterion_TechnicalPrerequisiteCourse_Insert
(
  @StudyPlanCriterion_TechnicalPrerequisiteCourse_ID uniqueidenti↓er,
  @StudyPlanCriterion_ID uniqueidenti↓er,
  @Course_ID uniqueidenti↓er,
  @AdditionalChoice bit,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
INSERT INTO StudyPlanCriterion_TechnicalPrerequisiteCourse
        (StudyPlanCriterion_TechnicalPrerequisiteCourse_ID,
  StudyPlanCriterion_ID,
  Course_ID,
  AdditionalChoice,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy)
VALUES (@StudyPlanCriterion_TechnicalPrerequisiteCourse_ID,
  @StudyPlanCriterion_ID,
  @Course_ID,
  @AdditionalChoice,
  @Created,
```

```
  @CreatedBy,
  @Updated,
  @UpdatedBy);
```

2.62.3 StudyPlanCriterion_TechnicalPrerequisiteCourse_RespectTP

```
CREATE PROCEDURE dbo.StudyPlanCriterion_TechnicalPrerequisiteCourse_RespectTP
(
  @StudyPlanCriterion_ID uniqueidenti↓er,
  @Course_ID uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
SELECT AdditionalChoice
FROM StudyPlanCriterion_TechnicalPrerequisiteCourse
WHERE StudyPlanCriterion_ID = @StudyPlanCriterion_ID
AND Course_ID = @Course_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.62.4 StudyPlanCriterion_TechnicalPrerequisiteCourse_Select

```
CREATE PROCEDURE dbo.StudyPlanCriterion_TechnicalPrerequisiteCourse_Select
(
  @StudyPlanCriterion_TechnicalPrerequisiteCourse_ID uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
SELECT StudyPlanCriterion_TechnicalPrerequisiteCourse_ID,
  StudyPlanCriterion_ID,
  Course_ID,
  AdditionalChoice,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM StudyPlanCriterion_TechnicalPrerequisiteCourse
WHERE StudyPlanCriterion_TechnicalPrerequisiteCourse_ID = @StudyPlanCriterion_TechnicalPrerequisiteCourse_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.62.5 StudyPlanCriterion_TechnicalPrerequisiteCourse_Update

```
CREATE PROCEDURE dbo.StudyPlanCriterion_TechnicalPrerequisiteCourse_Update
(
  @StudyPlanCriterion_TechnicalPrerequisiteCourse_ID uniqueidenti↓er,
  @StudyPlanCriterion_ID uniqueidenti↓er,
  @Course_ID uniqueidenti↓er,
  @AdditionalChoice bit,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er,
  @Original_StudyPlanCriterion_TechnicalPrerequisiteCourse_ID uniqueidenti↓er,
  @Original_StudyPlanCriterion_ID uniqueidenti↓er,
  @Original_Course_ID uniqueidenti↓er,
  @Original_AdditionalChoice bit,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
UPDATE StudyPlanCriterion_TechnicalPrerequisiteCourse
SET  StudyPlanCriterion_TechnicalPrerequisiteCourse_ID = @StudyPlanCriterion_TechnicalPrerequisiteCourse_ID,
  StudyPlanCriterion_ID   = @StudyPlanCriterion_ID,
  Course_ID    = @Course_ID,
  AdditionalChoice   = @AdditionalChoice,
  Created    = @Created,
  CreatedBy    = @CreatedBy,
  Updated    = @Updated,
  UpdatedBy    = @UpdatedBy
WHERE  (StudyPlanCriterion_TechnicalPrerequisiteCourse_ID = @Original_StudyPlanCriterion_TechnicalPrerequisiteCourse_ID)
AND (StudyPlanCriterion_ID  = @Original_StudyPlanCriterion_ID)
AND (Course_ID    = @Original_Course_ID)
AND (AdditionalChoice   = @Original_AdditionalChoice)
AND (Created    = @Original_Created)
AND (CreatedBy    = @Original_CreatedBy)
AND (Updated    = @Original_Updated)
AND (UpdatedBY    = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50101', 14, 1);
```

2.63 StudyPlanCriterion_WorkloadPeriod

2.63.1 StudyPlanCriterion_WorkloadPeriod_Delete

```
CREATE PROCEDURE dbo.StudyPlanCriterion_WorkloadPeriod_Delete
(
  @Original_StudyPlanCriterion_WorkloadPeriod_ID uniqueidenti↓er,
  @Original_StudyPlanCriterion_ID uniqueidenti↓er,
  @Original_Period_ID uniqueidenti↓er,
  @Original_Point_ID uniqueidenti↓er,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
DELETE FROM StudyPlanCriterion_WorkloadPeriod
WHERE  (StudyPlanCriterion_WorkloadPeriod_ID  = @Original_StudyPlanCriterion_WorkloadPeriod_ID)
AND (StudyPlanCriterion_ID    = @Original_StudyPlanCriterion_ID)
AND (Period_ID    = @Original_Period_ID)
AND (Point_ID    = @Original_Point_ID)
AND (Created    = @Original_Created)
AND (CreatedBy    = @Original_CreatedBy)
AND (Updated    = @Original_Updated)
AND (UpdatedBY    = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50201', 14, 1);
```

2.63.2 StudyPlanCriterion_WorkloadPeriod_GetID

```
CREATE PROCEDURE dbo.StudyPlanCriterion_WorkloadPeriod_GetID
(
  @StudyPlanCriterion_ID uniqueidenti↓er,
  @Period_ID uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
SELECT StudyPlanCriterion_WorkloadPeriod_ID
FROM StudyPlanCriterion_WorkloadPeriod
WHERE  StudyPlanCriterion_ID = @StudyPlanCriterion_ID
AND Period_ID = @Period_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.63.3 StudyPlanCriterion_WorkloadPeriod_GetPoint

```
CREATE PROCEDURE dbo.StudyPlanCriterion_WorkloadPeriod_GetPoint
(
  @StudyPlanCriterion_ID uniqueidenti↓er,
  @Period_ID uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
SELECT Point_ID
FROM StudyPlanCriterion_WorkloadPeriod
WHERE  StudyPlanCriterion_ID = @StudyPlanCriterion_ID
AND Period_ID = @Period_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.63.4 StudyPlanCriterion_WorkloadPeriod_Insert

```
CREATE PROCEDURE dbo.StudyPlanCriterion_WorkloadPeriod_Insert
(
  @StudyPlanCriterion_WorkloadPeriod_ID uniqueidenti↓er,
  @StudyPlanCriterion_ID uniqueidenti↓er,
  @Period_ID uniqueidenti↓er,
  @Point_ID uniqueidenti↓er,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
INSERT INTO StudyPlanCriterion_WorkloadPeriod
      (StudyPlanCriterion_WorkloadPeriod_ID,
  StudyPlanCriterion_ID,
  Period_ID,
  Point_ID,
  Created,
  CreatedBy,
  Updated,
```

```
  UpdatedBy)
VALUES (@StudyPlanCriterion_WorkloadPeriod_ID,
  @StudyPlanCriterion_ID,
  @Period_ID,
  @Point_ID,
  @Created,
  @CreatedBy,
  @Updated,
  @UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50101', 14, 1);
```

2.63.5 StudyPlanCriterion_WorkloadPeriod_Select

```
CREATE PROCEDURE dbo.StudyPlanCriterion_WorkloadPeriod_Select
(
  @StudyPlanCriterion_WorkloadPeriod_ID uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
SELECT StudyPlanCriterion_WorkloadPeriod_ID,
  StudyPlanCriterion_ID,
  Period_ID,
  Point_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM StudyPlanCriterion_WorkloadPeriod
WHERE StudyPlanCriterion_WorkloadPeriod_ID = @StudyPlanCriterion_WorkloadPeriod_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.63.6 StudyPlanCriterion_WorkloadPeriod_Update

```
CREATE PROCEDURE dbo.StudyPlanCriterion_WorkloadPeriod_Update
(
  @StudyPlanCriterion_WorkloadPeriod_ID uniqueidenti↓er,
  @StudyPlanCriterion_ID uniqueidenti↓er,
  @Period_ID uniqueidenti↓er,
  @Point_ID uniqueidenti↓er,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er,
  @Original_StudyPlanCriterion_WorkloadPeriod_ID uniqueidenti↓er,
  @Original_StudyPlanCriterion_ID uniqueidenti↓er,
  @Original_Period_ID uniqueidenti↓er,
  @Original_Point_ID uniqueidenti↓er,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
UPDATE StudyPlanCriterion_WorkloadPeriod
SET StudyPlanCriterion_WorkloadPeriod_ID = @StudyPlanCriterion_WorkloadPeriod_ID,
  StudyPlanCriterion_ID = @StudyPlanCriterion_ID,
  Period_ID = @Period_ID,
  Point_ID = @Point_ID,
  Created = @Created,
  CreatedBy = @CreatedBy,
  Updated = @Updated,
  UpdatedBy = @UpdatedBy
WHERE (StudyPlanCriterion_WorkloadPeriod_ID = @Original_StudyPlanCriterion_WorkloadPeriod_ID)
AND (StudyPlanCriterion_ID = @Original_StudyPlanCriterion_ID)
AND (Period_ID = @Original_Period_ID)
AND (Point_ID = @Original_Point_ID)
AND (Created = @Original_Created)
AND (CreatedBy = @Original_CreatedBy)
AND (Updated = @Original_Updated)
AND (UpdatedBY = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50101', 14, 1);
```

2.64 StudyPlanCriterion_WorkloadPeriod_Module

2.64.1 StudyPlanCriterion_WorkloadPeriod_Module_Delete

```
CREATE PROCEDURE dbo.StudyPlanCriterion_WorkloadPeriod_Module_Delete
(
  @Original_StudyPlanCriterion_WorkloadPeriod_Module_ID uniqueidenti↓er,
  @Original_StudyPlanCriterion_WorkloadPeriod_ID uniqueidenti↓er,
```

```
  @Original_Module_ID int,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
DELETE FROM StudyPlanCriterion_WorkloadPeriod_Module
WHERE  (StudyPlanCriterion_WorkloadPeriod_Module_ID = @Original_StudyPlanCriterion_WorkloadPeriod_Module_ID)
AND (StudyPlanCriterion_WorkloadPeriod_ID   = @Original_StudyPlanCriterion_WorkloadPeriod_ID)
AND (Module_ID     = @Original_Module_ID)
AND (Created     = @Original_Created)
AND (CreatedBy     = @Original_CreatedBy)
AND (Updated     = @Original_Updated)
AND (UpdatedBY     = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50201', 14, 1);
```

2.64.2 StudyPlanCriterion_WorkloadPeriod_Module_GetModules

```
CREATE PROCEDURE dbo.StudyPlanCriterion_WorkloadPeriod_Module_GetModules
(
  @StudyPlanCriterion_WorkloadPeriod_ID uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
SELECT Module_ID
FROM StudyPlanCriterion_WorkloadPeriod_Module
WHERE StudyPlanCriterion_WorkloadPeriod_ID = @StudyPlanCriterion_WorkloadPeriod_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.64.3 StudyPlanCriterion_WorkloadPeriod_Module_Insert

```
CREATE PROCEDURE dbo.StudyPlanCriterion_WorkloadPeriod_Module_Insert
(
  @StudyPlanCriterion_WorkloadPeriod_Module_ID uniqueidenti↓er,
  @StudyPlanCriterion_WorkloadPeriod_ID uniqueidenti↓er,
  @Module_ID int,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
INSERT INTO StudyPlanCriterion_WorkloadPeriod_Module
        (StudyPlanCriterion_WorkloadPeriod_Module_ID,
  StudyPlanCriterion_WorkloadPeriod_ID,
  Module_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy)
VALUES (@StudyPlanCriterion_WorkloadPeriod_Module_ID,
  @StudyPlanCriterion_WorkloadPeriod_ID,
  @Module_ID,
  @Created,
  @CreatedBy,
  @Updated,
  @UpdatedBy);
```

2.64.4 StudyPlanCriterion_WorkloadPeriod_Module_Select

```
CREATE PROCEDURE dbo.StudyPlanCriterion_WorkloadPeriod_Module_Select
(
  @StudyPlanCriterion_WorkloadPeriod_Module_ID uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
SELECT StudyPlanCriterion_WorkloadPeriod_Module_ID,
  StudyPlanCriterion_WorkloadPeriod_ID,
  Module_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM StudyPlanCriterion_WorkloadPeriod_Module
WHERE StudyPlanCriterion_WorkloadPeriod_Module_ID = @StudyPlanCriterion_WorkloadPeriod_Module_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.64.5 StudyPlanCriterion_WorkloadPeriod_Module_Update

```
CREATE PROCEDURE dbo.StudyPlanCriterion_WorkloadPeriod_Module_Update
(
  @StudyPlanCriterion_WorkloadPeriod_Module_ID uniqueidenti↓er,
  @StudyPlanCriterion_WorkloadPeriod_ID uniqueidenti↓er,
  @Module_ID int,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er,
  @Original_StudyPlanCriterion_WorkloadPeriod_Module_ID uniqueidenti↓er,
  @Original_StudyPlanCriterion_WorkloadPeriod_ID uniqueidenti↓er,
  @Original_Module_ID int,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
UPDATE StudyPlanCriterion_WorkloadPeriod_Module
SET  StudyPlanCriterion_WorkloadPeriod_Module_ID = @StudyPlanCriterion_WorkloadPeriod_Module_ID,
  StudyPlanCriterion_WorkloadPeriod_ID  = @StudyPlanCriterion_WorkloadPeriod_ID,
  Module_ID    = @Module_ID,
  Created     = @Created,
  CreatedBy     = @CreatedBy,
  Updated     = @Updated,
  UpdatedBy    = @UpdatedBy
WHERE  (StudyPlanCriterion_WorkloadPeriod_Module_ID= @Original_StudyPlanCriterion_WorkloadPeriod_Module_ID)
AND (StudyPlanCriterion_WorkloadPeriod_ID  = @Original_StudyPlanCriterion_WorkloadPeriod_ID)
AND (Module_ID    = @Original_Module_ID)
AND (Created     = @Original_Created)
AND (CreatedBy     = @Original_CreatedBy)
AND (Updated     = @Original_Updated)
AND (UpdatedBY     = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50101', 14, 1);
```

2.65 StudyPlanCriterion_WorkloadPeriodType

2.65.1 StudyPlanCriterion_WorkloadPeriodType_Delete

```
CREATE PROCEDURE dbo.StudyPlanCriterion_WorkloadPeriodType_Delete
(
  @Original_StudyPlanCriterion_WorkloadPeriodType_ID uniqueidenti↓er,
  @Original_StudyPlanCriterion_ID uniqueidenti↓er,
  @Original_PeriodType_ID int,
  @Original_Point_ID uniqueidenti↓er,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
DELETE FROM StudyPlanCriterion_WorkloadPeriodType
WHERE  (StudyPlanCriterion_WorkloadPeriodType_ID = @Original_StudyPlanCriterion_WorkloadPeriodType_ID)
AND (StudyPlanCriterion_ID    = @Original_StudyPlanCriterion_ID)
AND (PeriodType_ID    = @Original_PeriodType_ID)
AND (Point_ID    = @Original_Point_ID)
AND (Created     = @Original_Created)
AND (CreatedBy     = @Original_CreatedBy)
AND (Updated     = @Original_Updated)
AND (UpdatedBY     = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50201', 14, 1);
```

2.65.2 StudyPlanCriterion_WorkloadPeriodType_GetID

```
CREATE PROCEDURE dbo.StudyPlanCriterion_WorkloadPeriodType_GetID
(
  @StudyPlanCriterion_ID uniqueidenti↓er,
  @PeriodType_ID int
)
AS
SET NOCOUNT OFF;
SELECT StudyPlanCriterion_WorkloadPeriodType_ID
FROM StudyPlanCriterion_WorkloadPeriodType
WHERE  StudyPlanCriterion_ID = @StudyPlanCriterion_ID
AND PeriodType_ID = @PeriodType_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.65.3 StudyPlanCriterion_WorkloadPeriodType_GetPoint

```
CREATE PROCEDURE dbo.StudyPlanCriterion_WorkloadPeriodType_GetPoint
(
  @StudyPlanCriterion_ID uniqueidenti↓er,
  @PeriodType_ID int
)
AS
SET NOCOUNT OFF;
SELECT Point_ID
FROM StudyPlanCriterion_WorkloadPeriodType
WHERE  StudyPlanCriterion_ID = @StudyPlanCriterion_ID
AND PeriodType_ID = @PeriodType_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.65.4 StudyPlanCriterion_WorkloadPeriodType_Insert

```
CREATE PROCEDURE dbo.StudyPlanCriterion_WorkloadPeriodType_Insert
(
  @StudyPlanCriterion_WorkloadPeriodType_ID uniqueidenti↓er,
  @StudyPlanCriterion_ID uniqueidenti↓er,
  @PeriodType_ID int,
  @Point_ID uniqueidenti↓er,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
INSERT INTO StudyPlanCriterion_WorkloadPeriodType
        (StudyPlanCriterion_WorkloadPeriodType_ID,
  StudyPlanCriterion_ID,
  PeriodType_ID,
  Point_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy)
VALUES (@StudyPlanCriterion_WorkloadPeriodType_ID,
  @StudyPlanCriterion_ID,
  @PeriodType_ID,
  @Point_ID,
  @Created,
  @CreatedBy,
  @Updated,
  @UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50101', 14, 1);
```

2.65.5 StudyPlanCriterion_WorkloadPeriodType_Select

```
CREATE PROCEDURE dbo.StudyPlanCriterion_WorkloadPeriodType_Select
(
  @StudyPlanCriterion_WorkloadPeriodType_ID uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
SELECT StudyPlanCriterion_WorkloadPeriodType_ID,
  StudyPlanCriterion_ID,
  PeriodType_ID,
  Point_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM StudyPlanCriterion_WorkloadPeriodType
WHERE  StudyPlanCriterion_WorkloadPeriodType_ID = @StudyPlanCriterion_WorkloadPeriodType_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.65.6 StudyPlanCriterion_WorkloadPeriodType_Update

```
CREATE PROCEDURE dbo.StudyPlanCriterion_WorkloadPeriodType_Update
(
  @StudyPlanCriterion_WorkloadPeriodType_ID uniqueidenti↓er,
  @StudyPlanCriterion_ID uniqueidenti↓er,
  @PeriodType_ID int,
  @Point_ID uniqueidenti↓er,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er,
  @Original_StudyPlanCriterion_WorkloadPeriodType_ID uniqueidenti↓er,
  @Original_StudyPlanCriterion_ID uniqueidenti↓er,
  @Original_PeriodType_ID int,
  @Original_Point_ID uniqueidenti↓er,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
```

```
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
UPDATE StudyPlanCriterion_WorkloadPeriodType
SET  StudyPlanCriterion_WorkloadPeriodType_ID= @StudyPlanCriterion_WorkloadPeriodType_ID,
  StudyPlanCriterion_ID    = @StudyPlanCriterion_ID,
  PeriodType_ID    = @PeriodType_ID,
  Point_ID    = @Point_ID,
  Created    = @Created,
  CreatedBy    = @CreatedBy,
  Updated    = @Updated,
  UpdatedBy    = @UpdatedBy
WHERE  (StudyPlanCriterion_WorkloadPeriodType_ID= @Original_StudyPlanCriterion_WorkloadPeriodType_ID)
AND (StudyPlanCriterion_ID    = @Original_StudyPlanCriterion_ID)
AND (PeriodType_ID    = @Original_PeriodType_ID)
AND (Point_ID    = @Original_Point_ID)
AND (Created    = @Original_Created)
AND (CreatedBy    = @Original_CreatedBy)
AND (Updated    = @Original_Updated)
AND (UpdatedBY    = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50101', 14, 1);
```

2.66 StudyPlanCriterion_WorkloadPeriodType_Module

2.66.1 StudyPlanCriterion_WorkloadPeriodType_Module_Delete

```
CREATE PROCEDURE dbo.StudyPlanCriterion_WorkloadPeriodType_Module_Delete
(
  @Original_StudyPlanCriterion_WorkloadPeriodType_Module_ID uniqueidenti↓er,
  @Original_StudyPlanCriterion_WorkloadPeriodType_ID uniqueidenti↓er,
  @Original_Module_ID int,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
DELETE FROM StudyPlanCriterion_WorkloadPeriodType_Module
WHERE  (StudyPlanCriterion_WorkloadPeriodType_Module_ID= @Original_StudyPlanCriterion_WorkloadPeriodType_Module_ID)
AND (StudyPlanCriterion_WorkloadPeriodType_ID  = @Original_StudyPlanCriterion_WorkloadPeriodType_ID)
AND (Module_ID    = @Original_Module_ID)
AND (Created    = @Original_Created)
AND (CreatedBy    = @Original_CreatedBy)
AND (Updated    = @Original_Updated)
AND (UpdatedBY    = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50201', 14, 1);
```

2.66.2 StudyPlanCriterion_WorkloadPeriodType_Module_GetModules

```
CREATE PROCEDURE dbo.StudyPlanCriterion_WorkloadPeriodType_Module_GetModules
(
  @StudyPlanCriterion_WorkloadPeriodType_ID uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
SELECT Module_ID
FROM StudyPlanCriterion_WorkloadPeriodType_Module
WHERE StudyPlanCriterion_WorkloadPeriodType_ID = @StudyPlanCriterion_WorkloadPeriodType_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.66.3 StudyPlanCriterion_WorkloadPeriodType_Module_Insert

```
CREATE PROCEDURE dbo.StudyPlanCriterion_WorkloadPeriodType_Module_Insert
(
  @StudyPlanCriterion_WorkloadPeriodType_Module_ID uniqueidenti↓er,
  @StudyPlanCriterion_WorkloadPeriodType_ID uniqueidenti↓er,
  @Module_ID int,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
INSERT INTO StudyPlanCriterion_WorkloadPeriodType_Module
      (StudyPlanCriterion_WorkloadPeriodType_Module_ID,
```

```
  StudyPlanCriterion_WorkloadPeriodType_ID,
  Module_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy)
VALUES (@StudyPlanCriterion_WorkloadPeriodType_Module_ID,
  @StudyPlanCriterion_WorkloadPeriodType_ID,
  @Module_ID,
  @Created,
  @CreatedBy,
  @Updated,
  @UpdatedBy);
```

2.66.4 StudyPlanCriterion_WorkloadPeriodType_Module_Select

```
CREATE PROCEDURE dbo.StudyPlanCriterion_WorkloadPeriodType_Module_Select
(
  @StudyPlanCriterion_WorkloadPeriodType_Module_ID uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
SELECT StudyPlanCriterion_WorkloadPeriodType_Module_ID,
  StudyPlanCriterion_WorkloadPeriodType_ID,
  Module_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM StudyPlanCriterion_WorkloadPeriodType_Module
WHERE StudyPlanCriterion_WorkloadPeriodType_Module_ID = @StudyPlanCriterion_WorkloadPeriodType_Module_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.66.5 StudyPlanCriterion_WorkloadPeriodType_Module_Update

```
CREATE PROCEDURE dbo.StudyPlanCriterion_WorkloadPeriodType_Module_Update
(
  @StudyPlanCriterion_WorkloadPeriodType_Module_ID uniqueidenti↓er,
  @StudyPlanCriterion_WorkloadPeriodType_ID uniqueidenti↓er,
  @Module_ID int,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er,
  @Original_StudyPlanCriterion_WorkloadPeriodType_Module_ID uniqueidenti↓er,
  @Original_StudyPlanCriterion_WorkloadPeriodType_ID uniqueidenti↓er,
  @Original_Module_ID int,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
UPDATE StudyPlanCriterion_WorkloadPeriodType_Module
SET  StudyPlanCriterion_WorkloadPeriodType_Module_ID = @StudyPlanCriterion_WorkloadPeriodType_Module_ID,
  StudyPlanCriterion_WorkloadPeriodType_ID  = @StudyPlanCriterion_WorkloadPeriodType_ID,
  Module_ID     = @Module_ID,
  Created      = @Created,
  CreatedBy      = @CreatedBy,
  Updated      = @Updated,
  UpdatedBy     = @UpdatedBy
WHERE  (StudyPlanCriterion_WorkloadPeriodType_Module_ID= @Original_StudyPlanCriterion_WorkloadPeriodType_Module_ID)
AND (StudyPlanCriterion_WorkloadPeriodType_ID  = @Original_StudyPlanCriterion_WorkloadPeriodType_ID)
AND (Module_ID     = @Original_Module_ID)
AND (Created      = @Original_Created)
AND (CreatedBy      = @Original_CreatedBy)
AND (Updated      = @Original_Updated)
AND (UpdatedBY      = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50101', 14, 1);
```

2.67 StudyType

2.67.1 StudyType_Select

```
CREATE PROCEDURE dbo.StudyType_Select
(
  @StudyType_ID int
)
AS
SET NOCOUNT ON;
SELECT
```

```
  StudyType_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM StudyType
WHERE StudyType_ID = @StudyType_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.68 StudyType_ProjectType

2.68.1 StudyType_ProjectType_GetProjectTypes

```
CREATE PROCEDURE dbo.StudyType_ProjectType_GetProjectTypes
(
  @StudyTypeVersion_ID int
)
AS
SET NOCOUNT ON;
SELECT
  StudyType_ProjectType_ID
FROM StudyType_ProjectType
WHERE
  StudyTypeVersion_ID = @StudyTypeVersion_ID
ORDER BY SequenceNumber

IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.69 StudyType_TechnicalLine

2.69.1 StudyType_TechnicalLine_Select

```
CREATE PROCEDURE dbo.StudyType_TechnicalLine_Select
(
  @StudyType_TechnicalLine_ID int
)
AS
SET NOCOUNT ON;
SELECT
  StudyType_TechnicalLine_ID,
  StudyTypeVersion_ID,
  TechnicalLine_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM StudyType_TechnicalLine
WHERE StudyType_TechnicalLine_ID = @StudyType_TechnicalLine_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.70 StudyTypeVersion

2.70.1 StudyTypeVersion_Select

```
CREATE PROCEDURE dbo.StudyTypeVersion_Select
(
  @StudyTypeVersion_ID int
)
AS
SET NOCOUNT ON;
SELECT
  StudyTypeVersion_ID,
  StudyType_ID,
  Version,
  Name,
  Point_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM StudyTypeVersion
WHERE StudyTypeVersion_ID = @StudyTypeVersion_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.71 TechnicalField

2.71.1 TechnicalField_Select

```
CREATE PROCEDURE dbo.TechnicalField_Select
(
  @TechnicalField_ID int
)
AS
SET NOCOUNT ON;
SELECT  TechnicalField_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM  TechnicalField
WHERE  TechnicalField_ID = @TechnicalField_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.72 TechnicalField_Course

2.72.1 TechnicalField_Course_GetIDFromVersion

```
CREATE PROCEDURE dbo.TechnicalField_Course_GetIDFromVersion
(
  @TechnicalFieldVersion_ID int
)
AS
SET NOCOUNT ON;
SELECT  TechnicalField_Course_ID
FROM  TechnicalField_Course
WHERE  TechnicalFieldVersion_ID = @TechnicalFieldVersion_ID
AND  Course_ID IS NOT NULL
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.72.2 TechnicalField_Course_Select

```
CREATE PROCEDURE dbo.TechnicalField_Course_Select
(
  @TechnicalField_Course_ID int
)
AS
SET NOCOUNT ON;
SELECT  TechnicalField_Course_ID,
  TechnicalFieldVersion_ID,
  Course_ID,
  Number,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM  TechnicalField_Course
WHERE  TechnicalField_Course_ID = @TechnicalField_Course_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.73 TechnicalFieldVersion

2.73.1 TechnicalFieldVersion_Select

```
CREATE PROCEDURE dbo.TechnicalFieldVersion_Select
(
  @TechnicalFieldVersion_ID int
)
AS
SET NOCOUNT ON;
SELECT  TechnicalFieldVersion_ID,
  TechnicalField_ID,
  Version,
  Name,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM  TechnicalFieldVersion
WHERE  TechnicalFieldVersion_ID = @TechnicalFieldVersion_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.74 TechnicalLine

2.74.1 TechnicalLine_Select

```
CREATE PROCEDURE dbo.TechnicalLine_Select
(
  @TechnicalLine_ID int
)
AS
SET NOCOUNT ON;
SELECT  TechnicalLine_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM  TechnicalLine
WHERE  TechnicalLine_ID = @TechnicalLine_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.75 TechnicalLine_PrerequisiteCourse

2.75.1 TechnicalLine_PrerequisiteCourse_GetCourses

```
CREATE PROCEDURE dbo.TechnicalLine_PrerequisiteCourse_GetCourses
(
  @TechnicalLineVersion_ID int
)
AS
SELECT
  Course_ID
FROM TechnicalLine_PrerequisiteCourse
WHERE
  TechnicalLineVersion_ID = @TechnicalLineVersion_ID
  AND Course_ID IS NOT NULL
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.75.2 TechnicalLine_PrerequisiteCourse_Select

```
CREATE PROCEDURE dbo.TechnicalLine_PrerequisiteCourse_Select
(
  @TechnicalLine_PrerequisiteCourse_ID int
)
AS
SET NOCOUNT ON;
SELECT  TechnicalLine_PrerequisiteCourse_ID,
  TechnicalLineVersion_ID,
  Course_ID,
  Number,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM  TechnicalLine_PrerequisiteCourse
WHERE  TechnicalLine_PrerequisiteCourse_ID = @TechnicalLine_PrerequisiteCourse_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.76 TechnicalLine_PrerequisiteTechnicalPackage

2.76.1 TechnicalLine_PrerequisiteTechnicalPackage_GetIDFromTechnicalPackage

```
CREATE PROCEDURE dbo.TechnicalLine_PrerequisiteTechnicalPackage_GetIDFromTechnicalPackage
(
  @TechnicalLineVersion_ID int,
  @TechnicalPackage_ID int,
  @Technical_PrerequisiteTechnicalPackage int OUTPUT
)
AS
SELECT
  @Technical_PrerequisiteTechnicalPackage = TechnicalLine_PrerequisiteTechnicalPackage_ID
FROM TechnicalLine_PrerequisiteTechnicalPackage
WHERE
  TechnicalLineVersion_ID = @TechnicalLineVersion_ID
```

```
  AND TechnicalPackage_ID = @TechnicalPackage_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.76.2 TechnicalLine_PrerequisiteTechnicalPackage_GetTechnicalPackages

```
CREATE PROCEDURE dbo.TechnicalLine_PrerequisiteTechnicalPackage_GetTechnicalPackages
(
  @TechnicalLineVersion_ID int
)
AS
SELECT
  TechnicalPackage_ID
FROM TechnicalLine_PrerequisiteTechnicalPackage
WHERE
  TechnicalLineVersion_ID = @TechnicalLineVersion_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.76.3 TechnicalLine_PrerequisiteTechnicalPackage_Select

```
CREATE PROCEDURE dbo.TechnicalLine_PrerequisiteTechnicalPackage_Select
(
  @TechnicalLine_PrerequisiteTechnicalPackage_ID int
)
AS
SET NOCOUNT ON;
SELECT  TechnicalLine_PrerequisiteTechnicalPackage_ID,
  TechnicalLineVersion_ID,
  TechnicalPackage_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM  TechnicalLine_PrerequisiteTechnicalPackage
WHERE  TechnicalLine_PrerequisiteTechnicalPackage_ID = @TechnicalLine_PrerequisiteTechnicalPackage_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.77 TechnicalLine_PrerequisiteTechnicalPackageCourse

2.77.1 TechnicalLine_PrerequisiteTechnicalPackageCourse_GetCourses

```
CREATE PROCEDURE dbo.TechnicalLine_PrerequisiteTechnicalPackageCourse_GetCourses
(
  @TechnicalLine_PrerequisiteTechnicalPackage_ID int
)
AS
SELECT
  Course_ID
FROM TechnicalLine_PrerequisiteTechnicalPackageCourse
WHERE
  TechnicalLine_PrerequisiteTechnicalPackage_ID = @TechnicalLine_PrerequisiteTechnicalPackage_ID
  AND Course_ID IS NOT NULL
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.78 TechnicalLine_Specialization

2.78.1 TechnicalLine_Specialization_Select

```
CREATE PROCEDURE dbo.TechnicalLine_Specialization_Select
(
  @TechnicalLine_Specialization_ID int
)
AS
SET NOCOUNT ON;
SELECT  TechnicalLine_Specialization_ID,
  TechnicalLineVersion_ID,
  Specialization_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM  TechnicalLine_Specialization
WHERE  TechnicalLine_Specialization_ID = @TechnicalLine_Specialization_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.79 TechnicalLine_TechnicalField

2.79.1 TechnicalLine_TechnicalField_Select

```
CREATE PROCEDURE dbo.TechnicalLine_TechnicalField_Select
(
  @TechnicalLine_TechnicalField_ID int
)
AS
SET NOCOUNT ON;
SELECT  TechnicalLine_TechnicalField_ID,
  TechnicalLineVersion_ID,
  TechnicalField_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM  TechnicalLine_TechnicalField
WHERE  TechnicalLine_TechnicalField_ID = @TechnicalLine_TechnicalField_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.80 TechnicalLineVersion

2.80.1 TechnicalLineVersion_Select

```
CREATE PROCEDURE dbo.TechnicalLineVersion_Select
(
  @TechnicalLineVersion_ID int
)
AS
SET NOCOUNT ON;
SELECT  TechnicalLineVersion_ID,
  TechnicalLine_ID,
  Version,
  Name,
  Point_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM  TechnicalLineVersion
WHERE  TechnicalLineVersion_ID = @TechnicalLineVersion_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.81 TechnicalPackage

2.81.1 TechnicalPackage_Select

```
CREATE PROCEDURE dbo.TechnicalPackage_Select
(
  @TechnicalPackage_ID int
)
AS
SET NOCOUNT ON;
SELECT  TechnicalPackage_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM  TechnicalPackage
WHERE  TechnicalPackage_ID = @TechnicalPackage_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.82 TechnicalPackage_FundamentalCourse

2.82.1 TechnicalPackage_FundamentalCourse_GetIDFromVersion

```
CREATE PROCEDURE dbo.TechnicalPackage_FundamentalCourse_GetIDFromVersion
(
  @TechnicalPackageVersion_ID int
)
AS
SET NOCOUNT ON;
SELECT  TechnicalPackage_FundamentalCourse_ID
FROM  TechnicalPackage_FundamentalCourse
WHERE  TechnicalPackageVersion_ID = @TechnicalPackageVersion_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.82.2 TechnicalPackage_FundamentalCourse_Select

```
CREATE PROCEDURE dbo.TechnicalPackage_FundamentalCourse_Select
(
  @TechnicalPackage_FundamentalCourse_ID int
)
AS
SET NOCOUNT ON;
SELECT  TechnicalPackage_FundamentalCourse_ID,
  TechnicalPackageVersion_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM  TechnicalPackage_FundamentalCourse
WHERE  TechnicalPackage_FundamentalCourse_ID = @TechnicalPackage_FundamentalCourse_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.83 TechnicalPackage_FundamentalCourseItem

2.83.1 TechnicalPackage_FundamentalCourseItem_GetCourses

```
CREATE PROCEDURE dbo.TechnicalPackage_FundamentalCourseItem_GetCourses
(
  @TechnicalPackage_FundamentalCourse_ID int
)
AS
SET NOCOUNT ON;
SELECT  Course_ID
FROM  TechnicalPackage_FundamentalCourseItem
WHERE  TechnicalPackage_FundamentalCourse_ID = @TechnicalPackage_FundamentalCourse_ID
AND  Course_ID IS NOT NULL
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.83.2 TechnicalPackage_FundamentalCourseItem_Select

```
CREATE PROCEDURE dbo.TechnicalPackage_FundamentalCourseItem_Select
(
  @TechnicalPackage_FundamentalCourseItem_ID int
)
AS
SET NOCOUNT ON;
SELECT  TechnicalPackage_FundamentalCourseItem_ID,
  TechnicalPackage_FundamentalCourse_ID,
  Course_ID,
  Number,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM  TechnicalPackage_FundamentalCourseItem
WHERE  TechnicalPackage_FundamentalCourseItem_ID = @TechnicalPackage_FundamentalCourseItem_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.84 TechnicalPackage_Period

2.84.1 sp_createtpp_bachelor

```
CREATE PROCEDURE dbo.sp_createtpp_bachelor
@startdate datetime,
@tpvid int,
@count int,
```

```
@tppid int output
as
declare @userid uniqueidenti↓er
declare @curdate datetime
declare @pid uniqueidenti↓er
set @userid = 'D36DD218-AE59-4046-977C-659E4EB92C92'
set @curdate = getdate()
declare period_cursor cursor for
select top 14 period_id
from period
where startdate >= @startdate
and periodtype_id in (1,2)
order by startdate
for read only
open period_cursor
fetch next from period_cursor into @pid
while @@fetch_status = 0
begin
  set @count = @count + 1
  insert into technicalpackage_period
  values (@count, @tpvid, @pid, @curdate, @userid, @curdate, @userid)
  fetch next from period_cursor into @pid
end
close period_cursor
deallocate period_cursor
set @tppid = @count
return
```

2.84.2 sp_createtpp_master

```
CREATE PROCEDURE dbo.sp_createtpp_master
@startdate datetime,
@tpvid int,
@count int,
@tppid int output
as
declare @userid uniqueidenti↓er
declare @curdate datetime
declare @pid uniqueidenti↓er
set @userid = 'D36DD218-AE59-4046-977C-659E4EB92C92'
set @curdate = getdate()
declare period_cursor cursor for
select top 8 period_id
from period
where startdate >= @startdate
and periodtype_id in (1,2)
order by startdate
for read only
open period_cursor
fetch next from period_cursor into @pid
while @@fetch_status = 0
begin
  set @count = @count + 1
  insert into technicalpackage_period
  values (@count, @tpvid, @pid, @curdate, @userid, @curdate, @userid)
  fetch next from period_cursor into @pid
end
close period_cursor
deallocate period_cursor
set @tppid = @count
return
```

2.84.3 TechnicalPackage_Period_GetIDFromVersion

```
CREATE PROCEDURE dbo.TechnicalPackage_Period_GetIDFromVersion
(
  @TechnicalPackageVersion_ID int,
  @Period_ID uniqueidenti↓er
)
AS
SET NOCOUNT ON;
SELECT  TechnicalPackage_Period_ID
FROM  TechnicalPackage_Period
WHERE  TechnicalPackageVersion_ID = @TechnicalPackageVersion_ID
AND Period_ID = @Period_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.84.4 TechnicalPackage_Period_GetPeriods

```
CREATE PROCEDURE dbo.TechnicalPackage_Period_GetPeriods
(
  @TechnicalPackageVersion_ID int
)
AS
SET NOCOUNT ON;
SELECT  Period_ID
FROM  TechnicalPackage_Period
```

```
WHERE  TechnicalPackageVersion_ID = @TechnicalPackageVersion_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.84.5 TechnicalPackage_Period_Select

```
CREATE PROCEDURE dbo.TechnicalPackage_Period_Select
(
  @TechnicalPackage_Period_ID int
)
AS
SET NOCOUNT ON;
SELECT  TechnicalPackage_Period_ID,
  TechnicalPackageVersion_ID,
  Period_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM  TechnicalPackage_Period
WHERE  TechnicalPackage_Period_ID = @TechnicalPackage_Period_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.85 TechnicalPackage_PeriodCourse

2.85.1 TechnicalPackage_PeriodCourse_GetIDFromPeriod

```
CREATE PROCEDURE dbo.TechnicalPackage_PeriodCourse_GetIDFromPeriod
(
  @TechnicalPackage_Period_ID int
)
AS
SET NOCOUNT ON;
SELECT  TechnicalPackage_PeriodCourse_ID
FROM  TechnicalPackage_PeriodCourse
WHERE  TechnicalPackage_Period_ID = @TechnicalPackage_Period_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.85.2 TechnicalPackage_PeriodCourse_Select

```
CREATE PROCEDURE dbo.TechnicalPackage_PeriodCourse_Select
(
  @TechnicalPackage_PeriodCourse_ID int
)
AS
SET NOCOUNT ON;
SELECT  TechnicalPackage_PeriodCourse_ID,
  TechnicalPackage_Period_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM  TechnicalPackage_PeriodCourse
WHERE  TechnicalPackage_PeriodCourse_ID = @TechnicalPackage_PeriodCourse_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.86 TechnicalPackage_PeriodCourseItem

2.86.1 TechnicalPackage_PeriodCourseItem_GetCourses

```
CREATE PROCEDURE dbo.TechnicalPackage_PeriodCourseItem_GetCourses
(
  @TechnicalPackage_PeriodCourse_ID int
)
AS
SET NOCOUNT ON;
SELECT  Course_ID,
  Part
FROM  TechnicalPackage_PeriodCourseItem
WHERE  TechnicalPackage_PeriodCourse_ID = @TechnicalPackage_PeriodCourse_ID
AND Course_ID IS NOT NULL
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.86.2 TechnicalPackage_PeriodCourseItem_Select

```
CREATE PROCEDURE dbo.TechnicalPackage_PeriodCourseItem_Select
(
  @TechnicalPackage_PeriodCourseItem_ID int
)
AS
SET NOCOUNT ON;
SELECT  TechnicalPackage_PeriodCourseItem_ID,
  TechnicalPackage_PeriodCourse_ID,
  Course_ID,
  Part,
  Number,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM  TechnicalPackage_PeriodCourseItem
WHERE  TechnicalPackage_PeriodCourseItem_ID = @TechnicalPackage_PeriodCourseItem_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.87 TechnicalPackage_PeriodOptionalCourse

2.87.1 TechnicalPackage_PeriodOptionalCourse_GetCourses

```
CREATE PROCEDURE dbo.TechnicalPackage_PeriodOptionalCourse_GetCourses
(
  @TechnicalPackage_Period_ID int
)
AS
SET NOCOUNT ON;
SELECT  Course_ID,
  Part
FROM  TechnicalPackage_PeriodOptionalCourse
WHERE  TechnicalPackage_Period_ID = @TechnicalPackage_Period_ID
AND Course_ID IS NOT NULL
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.87.2 TechnicalPackage_PeriodOptionalCourse_Select

```
CREATE PROCEDURE dbo.TechnicalPackage_PeriodOptionalCourse_Select
(
  @TechnicalPackage_PeriodOptionalCourse_ID int
)
AS
SET NOCOUNT ON;
SELECT  TechnicalPackage_PeriodOptionalCourse_ID,
  TechnicalPackage_Period_ID,
  Course_ID,
  Part,
  Number,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM  TechnicalPackage_PeriodOptionalCourse
WHERE  TechnicalPackage_PeriodOptionalCourse_ID = @TechnicalPackage_PeriodOptionalCourse_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.88 TechnicalPackage_Project

2.88.1 TechnicalPackage_Project_Select

```
CREATE PROCEDURE dbo.TechnicalPackage_Project_Select
(
  @TechnicalPackage_Project_ID int
)
AS
SET NOCOUNT ON;
SELECT  TechnicalPackage_Project_ID,
  TechnicalPackageVersion_ID,
  Project_ID,
  Period_ID,
  Point_ID,
  Months,
  Created,
  CreatedBy,
```

```
  Updated,
  UpdatedBy
FROM  TechnicalPackage_Project
WHERE  TechnicalPackage_Project_ID = @TechnicalPackage_Project_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.89 TechnicalPackageVersion

2.89.1 TechnicalPackageVersion_Select

```
CREATE PROCEDURE dbo.TechnicalPackageVersion_Select
(
  @TechnicalPackageVersion_ID int
)
AS
SET NOCOUNT ON;
SELECT  TechnicalPackageVersion_ID,
  TechnicalPackage_ID,
  Version,
  Number,
  Name,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM  TechnicalPackageVersion
WHERE  TechnicalPackageVersion_ID = @TechnicalPackageVersion_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.90 Text

2.90.1 Text_GetTextsInGroup

```
CREATE PROCEDURE dbo.Text_GetTextsInGroup
(
  @TextGroup_ID int,
  @Culture_ID varchar(10)
)
AS
SET NOCOUNT ON;
SELECT
  NumberInGroup,
  Text
FROM  Text
WHERE
  TextGroup_ID = @TextGroup_ID
  AND Culture_ID = @Culture_ID
```

2.90.2 Text_Select

```
CREATE PROCEDURE dbo.Text_Select
(
  @TextGroup_ID int,
  @NumberInGroup varchar(25),
  @Culture_ID varchar(10)
)
AS
SET NOCOUNT ON;
SELECT Text_ID, TextGroup_ID, NumberInGroup, Culture_ID, Text, Description, Created, CreatedBy, Updated, UpdatedBy
FROM Text AS T
WHERE
  TextGroup_ID = @TextGroup_ID AND
  NumberInGroup = @NumberInGroup AND
  Culture_ID = @Culture_ID
```

2.91 User

2.91.1 User_Delete

```
CREATE PROCEDURE dbo.User_Delete
(
  @Original_User_ID uniqueidenti↓er,
  @Original_Username varchar(20),
  @Original_Password varchar(50),
  @Original_ChangePasswordAtNextLogon bit,
  @Original_UserCannotChangePassword bit,
  @Original_PasswordNeverExpires bit,
  @Original_PasswordLastChanged datetime,
  @Original_Locked bit,
  @Original_LockedAt datetime,
  @Original_Disabled bit,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
DELETE FROM [User]
WHERE
  (User_ID = @Original_User_ID)
  AND (Username = @Original_Username)
  AND (Password = @Original_Password)
  AND (ChangePasswordAtNextLogon = @Original_ChangePasswordAtNextLogon)
  AND (UserCannotChangePassword = @Original_UserCannotChangePassword)
  AND (PasswordNeverExpires = @Original_PasswordNeverExpires)
  AND (PasswordLastChanged = @Original_PasswordLastChanged)
  AND (Locked = @Original_Locked)
  AND (LockedAt = @Original_LockedAt)
  AND (Disabled = @Original_Disabled)
  AND (Created = @Original_Created)
  AND (CreatedBy = @Original_CreatedBy)
  AND (Updated = @Original_Updated)
  AND (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50201', 14, 1);
```

2.91.2 User_GetIDFromUsername

```
CREATE PROCEDURE dbo.User_GetIDFromUsername
(
  @Username varchar(20),
  @User_ID uniqueidenti↓er OUTPUT
)
AS
SELECT
  @User_ID = User_ID
FROM [User]
WHERE Username=@Username
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.91.3 User_Insert

```
CREATE PROCEDURE dbo.User_Insert
(
  @User_ID uniqueidenti↓er,
  @Username varchar(20),
  @Password varchar(50),
  @ChangePasswordAtNextLogon bit,
  @UserCannotChangePassword bit,
  @PasswordNeverExpires bit,
  @PasswordLastChanged datetime,
  @Locked bit,
  @LockedAt datetime,
  @Disabled bit,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
INSERT INTO [User]
  (User_ID,
  Username,
  Password,
  ChangePasswordAtNextLogon,
  UserCannotChangePassword,
  PasswordNeverExpires,
  PasswordLastChanged,
  Locked,
  LockedAt,
  Disabled,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy)
VALUES (@User_ID,
  @Username,
  @Password,
```

```
  @ChangePasswordAtNextLogon,
  @UserCannotChangePassword,
  @PasswordNeverExpires,
  @PasswordLastChanged,
  @Locked,
  @LockedAt,
  @Disabled,
  @Created,
  @CreatedBy,
  @Updated,
  @UpdatedBy);
```

2.91.4 User_Select

```
CREATE PROCEDURE dbo.User_Select
(
  @User_ID uniqueidenti¦er
)
AS
SET NOCOUNT ON;
SELECT
  User_ID,
  Username,
  Password,
  ChangePasswordAtNextLogon,
  UserCannotChangePassword,
  PasswordNeverExpires,
  PasswordLastChanged,
  Locked,
  LockedAt,
  Disabled,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM  [User]
WHERE  User_ID = @User_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.91.5 User_Update

```
CREATE PROCEDURE dbo.User_Update
(
  @User_ID uniqueidenti¦er,
  @Username varchar(20),
  @Password varchar(50),
  @ChangePasswordAtNextLogon bit,
  @UserCannotChangePassword bit,
  @PasswordNeverExpires bit,
  @PasswordLastChanged datetime,
  @Locked bit,
  @LockedAt datetime,
  @Disabled bit,
  @Created datetime,
  @CreatedBy uniqueidenti¦er,
  @Updated datetime,
  @UpdatedBy uniqueidenti¦er,
  @Original_User_ID uniqueidenti¦er,
  @Original_Username varchar(20),
  @Original_Password varchar(50),
  @Original_ChangePasswordAtNextLogon bit,
  @Original_UserCannotChangePassword bit,
  @Original_PasswordNeverExpires bit,
  @Original_PasswordLastChanged datetime,
  @Original_Locked bit,
  @Original_LockedAt datetime,
  @Original_Disabled bit,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti¦er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti¦er
)
AS
SET NOCOUNT OFF;
UPDATE [User]
SET
  User_ID = @User_ID,
  Username = @Username,
  Password = @Password,
  ChangePasswordAtNextLogon = @ChangePasswordAtNextLogon,
  UserCannotChangePassword = @UserCannotChangePassword,
  PasswordNeverExpires = @PasswordNeverExpires,
  PasswordLastChanged = @PasswordLastChanged,
  Locked = @Locked,
  LockedAt = @LockedAt,
  Disabled = @Disabled,
  Created = @Created,
  CreatedBy = @CreatedBy,
  Updated = @Updated,
  UpdatedBy = @UpdatedBy
WHERE
```

```
  (User_ID = @Original_User_ID)
  AND (Username = @Original_Username)
  AND (Password = @Original_Password)
  AND (ChangePasswordAtNextLogon = @Original_ChangePasswordAtNextLogon)
  AND (UserCannotChangePassword = @Original_UserCannotChangePassword)
  AND (PasswordNeverExpires = @Original_PasswordNeverExpires)
  AND (PasswordLastChanged = @Original_PasswordLastChanged)
  AND (Locked = @Original_Locked)
  AND (LockedAt = @Original_LockedAt)
  AND (Disabled = @Original_Disabled)
  AND (Created = @Original_Created)
  AND (CreatedBy = @Original_CreatedBy)
  AND (Updated = @Original_Updated)
  AND (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50101', 14, 1);
```

2.92 User_Login

2.92.1 User_Login_Delete

```
CREATE PROCEDURE dbo.User_Login_Delete
(
  @Original_User_Login_ID uniqueidenti↓er,
  @Original_User_ID uniqueidenti↓er,
  @Original_User_LoginType_ID int,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
DELETE FROM User_Login
WHERE
  (User_Login_ID = @Original_User_Login_ID)
  AND (User_ID = @Original_User_ID)
  AND (User_LoginType_ID = @Original_User_LoginType_ID)
  AND (Created = @Original_Created)
  AND (CreatedBy = @Original_CreatedBy)
  AND (Updated = @Original_Updated)
  AND (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50201', 14, 1);
```

2.92.2 User_Login_GetNumberOfLogins

```
CREATE PROCEDURE dbo.User_Login_GetNumberOfLogins
(
  @User_ID uniqueidenti↓er,
  @LoginType_ID int,
  @CurrentDateAndTime datetime,
  @WithinMinutes int,
  @NumberOfLogins int OUTPUT
)
AS

SELECT @NumberOfLogins=COUNT(*) FROM User_Login
WHERE
User_ID = @User_ID
AND User_LoginType_ID=@LoginType_ID
AND Created > (SELECT @CurrentDateAndTime - @WithinMinutes/1440)
```

2.92.3 User_Login_Insert

```
CREATE PROCEDURE dbo.User_Login_Insert
(
  @User_Login_ID uniqueidenti↓er,
  @User_ID uniqueidenti↓er,
  @User_LoginType_ID int,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
INSERT INTO User_Login
  (User_Login_ID,
  User_ID,
  User_LoginType_ID,
  Created,
  CreatedBy,
  Updated,
```

```
  UpdatedBy)
VALUES (@User_Login_ID,
  @User_ID,
  @User_LoginType_ID,
  @Created,
  @CreatedBy,
  @Updated,
  @UpdatedBy);
```

2.92.4 User_Login_Select

```
CREATE PROCEDURE dbo.User_Login_Select
(
  @User_Login_ID uniqueidenti↓er
)
AS
SET NOCOUNT ON;
SELECT
  User_Login_ID,
  User_ID,
  User_LoginType_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM  User_Login
WHERE  User_Login_ID = @User_Login_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.92.5 User_Login_Update

```
CREATE PROCEDURE dbo.User_Login_Update
(
  @User_Login_ID uniqueidenti↓er,
  @User_ID uniqueidenti↓er,
  @User_LoginType_ID int,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er,
  @Original_User_Login_ID uniqueidenti↓er,
  @Original_User_ID uniqueidenti↓er,
  @Original_User_LoginType_ID int,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
UPDATE User_Login
SET
  User_Login_ID = @User_Login_ID,
  User_ID = @User_ID,
  User_LoginType_ID = @User_LoginType_ID,
  Created = @Created,
  CreatedBy = @CreatedBy,
  Updated = @Updated,
  UpdatedBy = @UpdatedBy
WHERE
  (User_Login_ID = @Original_User_Login_ID)
  AND (User_ID = @Original_User_ID)
  AND (User_LoginType_ID = @Original_User_LoginType_ID)
  AND (Created = @Original_Created)
  AND (CreatedBy = @Original_CreatedBy)
  AND (Updated = @Original_Updated)
  AND (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50101', 14, 1);
```

2.93 User_PasswordHistory

2.93.1 User_PasswordHistory_Delete

```
CREATE PROCEDURE dbo.User_PasswordHistory_Delete
(
  @Original_User_PasswordHistory_ID uniqueidenti↓er,
  @Original_User_ID uniqueidenti↓er,
  @Original_Password varchar(50),
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
```

```
)
AS
SET NOCOUNT OFF;
DELETE FROM User_PasswordHistory
WHERE
  (User_PasswordHistory_ID = @Original_User_PasswordHistory_ID)
  AND (User_ID = @Original_User_ID)
  AND (Password = @Original_Password)
  AND (Created = @Original_Created)
  AND (CreatedBy = @Original_CreatedBy)
  AND (Updated = @Original_Updated)
  AND (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50201', 14, 1);
```

2.93.2 User_PasswordHistory_GetPreviousPasswords

```
CREATE PROCEDURE dbo.User_PasswordHistory_GetPreviousPasswords
(
  @User_ID uniqueidenti↓er
)
AS
SET NOCOUNT ON;

SELECT User_PasswordHistory_ID
FROM User_PasswordHistory
WHERE
  User_ID = @User_ID
ORDER BY Created DESC
```

2.93.3 User_PasswordHistory_Insert

```
CREATE PROCEDURE dbo.User_PasswordHistory_Insert
(
  @User_PasswordHistory_ID uniqueidenti↓er,
  @User_ID uniqueidenti↓er,
  @Password varchar(50),
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
INSERT INTO User_PasswordHistory
  (User_PasswordHistory_ID,
  User_ID,
  Password,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy)
VALUES (@User_PasswordHistory_ID,
  @User_ID,
  @Password,
  @Created,
  @CreatedBy,
  @Updated,
  @UpdatedBy);
```

2.93.4 User_PasswordHistory_Select

```
CREATE PROCEDURE dbo.User_PasswordHistory_Select
(
  @User_PasswordHistory_ID uniqueidenti↓er
)
AS
SET NOCOUNT ON;
SELECT
  User_PasswordHistory_ID,
  User_ID,
  Password,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM  User_PasswordHistory
WHERE  User_PasswordHistory_ID = @User_PasswordHistory_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.93.5 User_PasswordHistory_Update

```
CREATE PROCEDURE dbo.User_PasswordHistory_Update
(
  @User_PasswordHistory_ID uniqueidenti↓er,
  @User_ID uniqueidenti↓er,
  @Password varchar(50),
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er,
  @Original_User_PasswordHistory_ID uniqueidenti↓er,
  @Original_User_ID uniqueidenti↓er,
  @Original_Password varchar(50),
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
UPDATE User_PasswordHistory
SET
  User_PasswordHistory_ID = @User_PasswordHistory_ID,
  User_ID = @User_ID,
  Password = @Password,
  Created = @Created,
  CreatedBy = @CreatedBy,
  Updated = @Updated,
  UpdatedBy = @UpdatedBy
WHERE
  (User_PasswordHistory_ID = @Original_User_PasswordHistory_ID)
  AND (User_ID = @Original_User_ID)
  AND (Password = @Original_Password)
  AND (Created = @Original_Created)
  AND (CreatedBy = @Original_CreatedBy)
  AND (Updated = @Original_Updated)
  AND (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50101', 14, 1);
```

2.94 User_SecurityRole

2.94.1 User_SecurityRole_Delete

```
CREATE PROCEDURE dbo.User_SecurityRole_Delete
(
  @Original_User_SecurityRole_ID uniqueidenti↓er,
  @Original_User_ID uniqueidenti↓er,
  @Original_SecurityRole_ID int,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
DELETE FROM User_SecurityRole
WHERE
  (User_SecurityRole_ID = @Original_User_SecurityRole_ID)
  AND (User_ID = @Original_User_ID)
  AND (SecurityRole_ID = @Original_SecurityRole_ID)
  AND (Created = @Original_Created)
  AND (CreatedBy = @Original_CreatedBy)
  AND (Updated = @Original_Updated)
  AND (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50201', 14, 1);
```

2.94.2 User_SecurityRole_GetRolesFromUserID

```
CREATE PROCEDURE dbo.User_SecurityRole_GetRolesFromUserID
(
  @User_ID uniqueidenti↓er
)
AS
SET NOCOUNT ON;
SELECT
  SecurityRole_ID
FROM  User_SecurityRole
WHERE  User_ID = @User_ID

IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.94.3 User_SecurityRole_Insert

```
CREATE PROCEDURE dbo.User_SecurityRole_Insert
(
  @User_SecurityRole_ID uniqueidenti↓er,
  @User_ID uniqueidenti↓er,
  @SecurityRole_ID int,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
INSERT INTO User_SecurityRole
  (User_SecurityRole_ID,
  User_ID,
  SecurityRole_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy)
VALUES (@User_SecurityRole_ID,
  @User_ID,
  @SecurityRole_ID,
  @Created,
  @CreatedBy,
  @Updated,
  @UpdatedBy);
```

2.94.4 User_SecurityRole_Select

```
CREATE PROCEDURE dbo.User_SecurityRole_Select
(
  @User_SecurityRole_ID uniqueidenti↓er
)
AS
SET NOCOUNT ON;
SELECT
  User_SecurityRole_ID,
  User_ID,
  SecurityRole_ID,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM  User_SecurityRole
WHERE  User_SecurityRole_ID = @User_SecurityRole_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

2.94.5 User_SecurityRole_Update

```
CREATE PROCEDURE dbo.User_SecurityRole_Update
(
  @User_SecurityRole_ID uniqueidenti↓er,
  @User_ID uniqueidenti↓er,
  @SecurityRole_ID int,
  @Created datetime,
  @CreatedBy uniqueidenti↓er,
  @Updated datetime,
  @UpdatedBy uniqueidenti↓er,
  @Original_User_SecurityRole_ID uniqueidenti↓er,
  @Original_User_ID uniqueidenti↓er,
  @Original_SecurityRole_ID int,
  @Original_Created datetime,
  @Original_CreatedBy uniqueidenti↓er,
  @Original_Updated datetime,
  @Original_UpdatedBy uniqueidenti↓er
)
AS
SET NOCOUNT OFF;
UPDATE User_SecurityRole
SET
  User_SecurityRole_ID = @User_SecurityRole_ID,
  User_ID = @User_ID,
  SecurityRole_ID = @SecurityRole_ID,
  Created = @Created,
  CreatedBy = @CreatedBy,
  Updated = @Updated,
  UpdatedBy = @UpdatedBy
WHERE
  (User_SecurityRole_ID = @Original_User_SecurityRole_ID)
  AND (User_ID = @Original_User_ID)
  AND (SecurityRole_ID = @Original_SecurityRole_ID)
  AND (Created = @Original_Created)
  AND (CreatedBy = @Original_CreatedBy)
  AND (Updated = @Original_Updated)
  AND (UpdatedBy = @Original_UpdatedBy)
IF @@ROWCOUNT = 0
  RAISERROR('50101', 14, 1);
```

2.95 Weekday

2.95.1 Weekday_Select

```
CREATE PROCEDURE dbo.Weekday_Select
(
  @Weekday_ID int
)
AS
SET NOCOUNT ON;
SELECT  Weekday_ID,
  Name,
  Created,
  CreatedBy,
  Updated,
  UpdatedBy
FROM  Weekday
WHERE  Weekday_ID = @Weekday_ID
IF @@ROWCOUNT = 0
  RAISERROR('50001', 14, 1);
```

Chapter 3

Data Files

3.1 AssessmentType

```
declare @userid uniqueidenti↓er
declare @curdate datetime

set @userid = 'D36DD218-AE59-4046-977C-659E4EB92C92'
set @curdate = getdate()

insert into assessmentType
values
(1,
'<cultures>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>13-skala, intern censur</value>
  </culture>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>Scale of marks (0-13), internal examiner</value>
  </culture>
</cultures>',
@curdate, @userid, @curdate, @userid)

insert into assessmentType
values
(2,
'<cultures>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>13-skala, ekstern censur</value>
  </culture>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>Scale of marks (0-13), external examiner</value>
  </culture>
</cultures>',
@curdate, @userid, @curdate, @userid)

insert into assessmentType
values
(3,
'<cultures>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>bestået/ikke bestået, intern censur</value>
  </culture>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>pass/not pass, internal examiner</value>
  </culture>
</cultures>',
@curdate, @userid, @curdate, @userid)

insert into assessmentType
values
(4,
'<cultures>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>bestået/ikke bestået, ekstern censur</value>
  </culture>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>pass/not pass, external examiner</value>
  </culture>
</cultures>',
@curdate, @userid, @curdate, @userid)
```

3.2 Project

3.2.1 Bachelor of Science Trainee Projects

```
declare @userid uniqueidenti↓er
declare @curdate datetime
declare @projecttype int

set @userid = '{BC61A572-A19A-41FB-97B8-9B0B463F2DBB}'
set @curdate = getdate()
set @projecttype = 5 -- Trainee Project

-------------------------------------------
-- Arctic Engineering Training Project
-------------------------------------------

insert into Project
values (
'{AE87834F-535B-4dbf-B093-88F1F3D2B9B4}',
'11805',
'<cultures>
  <culture>
```

```
    <cultureID>da-DK</cultureID>
    <value>Arktisk ingeniørpraktik</value>
  </culture>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>Arctic Engineering Training</value>
  </culture>
</cultures>',
@projecttype,
3,
null,
'{B347A190-1A9B-4c23-9576-9FE7CF2CA62B}',
@curdate, @userid, @curdate, @userid)

-------------------------------------------
-- Engineering Training Project
-------------------------------------------

insert into Project
values (
'{C1AB722E-01F3-473a-AFB8-DFCEC7DB493B}',
'11751',
'<cultures>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>Bygningspraktikprojekt</value>
  </culture>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>Engineering Training</value>
  </culture>
</cultures>',
@projecttype,
3,
null,
'{E8FA9A2F-A0D8-4b7d-ADE8-A091818DC8DC}',
@curdate, @userid, @curdate, @userid)

-------------------------------------------
-- Civil Engineering Design Project
-------------------------------------------

insert into Project
values (
'{06F28C4C-E7C1-4b95-987B-DE09838A7AE6}',
'11951',
'<cultures>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>Projekteringspraktik</value>
  </culture>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>Civil Engineering Design</value>
  </culture>
</cultures>',
@projecttype,
3,
null,
'{FE3F0B68-E455-49c5-856E-96D61F44F732}',
@curdate, @userid, @curdate, @userid)

-----------------------------------------------------------------
-- Industrial Training Project (Electrical Engineering)
-----------------------------------------------------------------

insert into Project
values (
'{32B5D8AA-16FF-4c88-93B8-02B29372F3E9}',
'31031',
'<cultures>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>Ingeniørpraktik</value>
  </culture>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>Industrial Training</value>
  </culture>
</cultures>',
@projecttype,
3,
null,
'{2F3A67B5-4E5E-4168-A1BA-D0A5371DAA0A}',
@curdate, @userid, @curdate, @userid)

-------------------------------------------------------
-- Industrial Placement (Mechanical Engineering)
-------------------------------------------------------

insert into Project
values (
'{22E27B35-4609-4124-933D-563FCD350E24}',
'41845',
'<cultures>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>Ingeniørpraktik for konstruktions- og skibslinien</value>
  </culture>
```

```
  <culture>
    <cultureID>en-GB</cultureID>
    <value>Industrial Placement</value>
  </culture>
</cultures>',
@projecttype,
3,
null,
'{8C72F4C9-C27C-4530-8778-5F16E322F744}',
@curdate, @userid, @curdate, @userid)

-------------------------------------------------------------
-- Industrial Placement (Production and Plastic)
-------------------------------------------------------------

insert into Project
values (
'{40BC7783-154F-4c07-8BF7-20E2BFC6A7D0}',
'42845',
'<cultures>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>Ingeniørpraktik for produktions- og plastlinien</value>
  </culture>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>Industrial Placement (production and plastic)</value>
  </culture>
</cultures>',
@projecttype,
3,
null,
'{56E03A48-788B-4cd0-911D-BCBAC02DD3B6}',
@curdate, @userid, @curdate, @userid)

-------------------------------------------------------------
-- Engineering Trainee (Chemical Engineering)
-------------------------------------------------------------

insert into Project
values (
'{AF585023-3773-44fe-B3E8-22BFC9718E8B}',
'28181',
'<cultures>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>Ingeniørpraktik (kemi)</value>
  </culture>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>Engineering Trainee (Chemical Engineering)</value>
  </culture>
</cultures>',
@projecttype,
3,
null,
'{FEA8D4C1-8CE5-49cf-B1B5-F9E18E9F515D}',
@curdate, @userid, @curdate, @userid)

-------------------------------------------------------------
-- Industrial Training (Information Technology)
-------------------------------------------------------------

insert into Project
values (
'{870899CF-B9E0-49ae-9800-094EBAFEC157}',
'02365',
'<cultures>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>Ingeniørpraktik (IT)</value>
  </culture>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>Industrial Training (Information Technology)</value>
  </culture>
</cultures>',
@projecttype,
3,
null,
'{102A0A1F-2956-48c3-AF1F-DAFEEDF16D26}',
@curdate, @userid, @curdate, @userid)
```

3.3 ContactDataType

```
declare @userid uniqueidenti↓er
declare @curdate datetime

set @userid = 'D36DD218-AE59-4046-977C-659E4EB92C92'
set @curdate = getdate()

insert into contactdatatype
values
(1,
```

```
'<cultures>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>Person privat</value>
  </culture>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>Person private</value>
  </culture>
</cultures>',
@curdate, @userid, @curdate, @userid)

insert into contactdatatype
values
(2,
'<cultures>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>Person arbejde</value>
  </culture>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>Person work</value>
  </culture>
</cultures>',
@curdate, @userid, @curdate, @userid)

insert into contactdatatype
values
(3,
'<cultures>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>Institut</value>
  </culture>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>Department</value>
  </culture>
</cultures>',
@curdate, @userid, @curdate, @userid)
```

3.4 Course_RelationCourseType

```
declare @userid uniqueidenti↓er
set @userid = '{BC61A572-A19A-41FB-97B8-9B0B463F2DBB}'

declare @curdate datetime
set @curdate = getdate()

insert into Course_RelationCourseType values (
0,
'Point Blocking Course',
@curdate, @userid, @curdate, @userid)

insert into Course_RelationCourseType values (
1,
'Mandatory Prerequisite Course',
@curdate, @userid, @curdate, @userid)

insert into Course_RelationCourseType values (
2,
'Technical Prerequisite Course',
@curdate, @userid, @curdate, @userid)

insert into Course_RelationCourseType values (
3,
'Desirable Prerequisite Course',
@curdate, @userid, @curdate, @userid)
```

3.5 Course_StudyTypeCategory

```
------------------------------------------------------------------------
--Common declarations
------------------------------------------------------------------------
declare @userid uniqueidenti↓er
set @userid = '{BC61A572-A19A-41FB-97B8-9B0B463F2DBB}'

declare @curdate datetime
set @curdate = getdate()

declare @studytypecategory_id int
declare @studytype_id int

declare @da varchar(255), @en varchar(255)

------------------------------------------------------------------------
-- MASTER OF SCIENCE CATEGORIES
-- MASTER OF SCIENCE CATEGORIES
```

```
-- MASTER OF SCIENCE CATEGORIES
-------------------------------------------------------------------------

set @studytypecategory_id = 1
set @studytype_id = 1

set @da = 'Initiativkursus'
set @en = 'Initiative Course'

insert into Course_StudyTypeCategory
values (@studytypecategory_id,
@studytype_id,
'<cultures>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>' + @en + '</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>' + @da + '</value>
  </culture>
</cultures>',
@curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------

set @studytypecategory_id = 2
set @studytype_id = 1

set @da = 'Basiskursus'
set @en = 'Basis Course'

insert into Course_StudyTypeCategory
values (@studytypecategory_id,
@studytype_id,
'<cultures>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>' + @en + '</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>' + @da + '</value>
  </culture>
</cultures>',
@curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------
-- BACHELOR OF SCIENCE CATEGORIES
-- BACHELOR OF SCIENCE CATEGORIES
-- BACHELOR OF SCIENCE CATEGORIES
-------------------------------------------------------------------------

set @studytypecategory_id = 20
set @studytype_id = 2

set @da = 'Kemi'
set @en = 'Chemical Engineering'

insert into Course_StudyTypeCategory
values (@studytypecategory_id,
@studytype_id,
'<cultures>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>' + @en + '</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>' + @da + '</value>
  </culture>
</cultures>',
@curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------

set @studytypecategory_id = 21
set @studytype_id = 2

set @da = 'Bygning'
set @en = 'Construction'

insert into Course_StudyTypeCategory
values (@studytypecategory_id,
@studytype_id,
'<cultures>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>' + @en + '</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>' + @da + '</value>
  </culture>
</cultures>',
@curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------
```

```
set @studytypecategory_id = 22
set @studytype_id = 2

set @da = 'Arktisk teknologi'
set @en = 'Arctic technology'

insert into Course_StudyTypeCategory
values (@studytypecategory_id,
@studytype_id,
'<cultures>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>' + @en + '</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>' + @da + '</value>
  </culture>
</cultures>',
@curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------

set @studytypecategory_id = 23
set @studytype_id = 2

set @da = 'By- og Bygningsingeniør'
set @en = 'City and Construction Engineering'

insert into Course_StudyTypeCategory
values (@studytypecategory_id,
@studytype_id,
'<cultures>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>' + @en + '</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>' + @da + '</value>
  </culture>
</cultures>',
@curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------

set @studytypecategory_id = 24
set @studytype_id = 2

set @da = 'IT'
set @en = 'IT'

insert into Course_StudyTypeCategory
values (@studytypecategory_id,
@studytype_id,
'<cultures>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>' + @en + '</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>' + @da + '</value>
  </culture>
</cultures>',
@curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------

set @studytypecategory_id = 25
set @studytype_id = 2

set @da = 'Elektro'
set @en = 'Electrical Engineering'

insert into Course_StudyTypeCategory
values (@studytypecategory_id,
@studytype_id,
'<cultures>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>' + @en + '</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>' + @da + '</value>
  </culture>
</cultures>',
@curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------

set @studytypecategory_id = 26
set @studytype_id = 2

set @da = 'Maskin'
set @en = 'Machine Engineering'

insert into Course_StudyTypeCategory
```

```
values (@studytypecategory_id,
@studytype_id,
'<cultures>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>' + @en + '</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>' + @da + '</value>
  </culture>
</cultures>',
@curdate, @userid, @curdate, @userid)

-----------------------------------------------------------------------

set @studytypecategory_id = 27
set @studytype_id = 2

set @da = 'Matematik'
set @en = 'Mathematics'

insert into Course_StudyTypeCategory
values (@studytypecategory_id,
@studytype_id,
'<cultures>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>' + @en + '</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>' + @da + '</value>
  </culture>
</cultures>',
@curdate, @userid, @curdate, @userid)

-----------------------------------------------------------------------

set @studytypecategory_id = 28
set @studytype_id = 2

set @da = 'Fysik og informatik'
set @en = 'Physics and Informatics'

insert into Course_StudyTypeCategory
values (@studytypecategory_id,
@studytype_id,
'<cultures>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>' + @en + '</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>' + @da + '</value>
  </culture>
</cultures>',
@curdate, @userid, @curdate, @userid)

-----------------------------------------------------------------------

set @studytypecategory_id = 29
set @studytype_id = 2

set @da = 'Kemi og bioteknologi'
set @en = 'Chemistry and Biotechnology'

insert into Course_StudyTypeCategory
values (@studytypecategory_id,
@studytype_id,
'<cultures>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>' + @en + '</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>' + @da + '</value>
  </culture>
</cultures>',
@curdate, @userid, @curdate, @userid)

-----------------------------------------------------------------------

set @studytypecategory_id = 30
set @studytype_id = 2

set @da = 'Elektronik og Elteknik'
set @en = 'Electronics and Electrical Engineering'

insert into Course_StudyTypeCategory
values (@studytypecategory_id,
@studytype_id,
'<cultures>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>' + @en + '</value>
  </culture>
```

```
  <culture>
    <cultureID>da-DK</cultureID>
    <value>' + @da + '</value>
  </culture>
</cultures>',
@curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------
-- Ph.D. CATEGORIES
-- Ph.D. CATEGORIES
-- Ph.D. CATEGORIES
-------------------------------------------------------------------------

set @studytypecategory_id = 50
set @studytype_id = 3

set @da = 'Kemi og bioteknologi'
set @en = 'Chemical Engineering and Biotechnology'

insert into Course_StudyTypeCategory
values (@studytypecategory_id,
@studytype_id,
'<cultures>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>' + @en + '</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>' + @da + '</value>
  </culture>
</cultures>',
@curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------

set @studytypecategory_id = 51
set @studytype_id = 3

set @da = 'Konstruktion'
set @en = 'Construction Engineering'

insert into Course_StudyTypeCategory
values (@studytypecategory_id,
@studytype_id,
'<cultures>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>' + @en + '</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>' + @da + '</value>
  </culture>
</cultures>',
@curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------

set @studytypecategory_id = 52
set @studytype_id = 3

set @da = 'Produktion'
set @en = 'Manufacturing Engineering'

insert into Course_StudyTypeCategory
values (@studytypecategory_id,
@studytype_id,
'<cultures>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>' + @en + '</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>' + @da + '</value>
  </culture>
</cultures>',
@curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------

set @studytypecategory_id = 53
set @studytype_id = 3

set @da = 'Byggeri'
set @en = 'Building Engineering'

insert into Course_StudyTypeCategory
values (@studytypecategory_id,
@studytype_id,
'<cultures>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>' + @en + '</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>' + @da + '</value>
```

```
  </culture>
</cultures>',
@curdate, @userid, @curdate, @userid)

-----------------------------------------------------------------------

set @studytypecategory_id = 54
set @studytype_id = 3

set @da = 'Miljø og transport'
set @en = 'Environmental and Transport Engineering'

insert into Course_StudyTypeCategory
values (@studytypecategory_id,
@studytype_id,
'<cultures>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>' + @en + '</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>' + @da + '</value>
  </culture>
</cultures>',
@curdate, @userid, @curdate, @userid)

-----------------------------------------------------------------------

set @studytypecategory_id = 55
set @studytype_id = 3

set @da = 'Matematik'
set @en = 'Mathematics'

insert into Course_StudyTypeCategory
values (@studytypecategory_id,
@studytype_id,
'<cultures>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>' + @en + '</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>' + @da + '</value>
  </culture>
</cultures>',
@curdate, @userid, @curdate, @userid)

-----------------------------------------------------------------------

set @studytypecategory_id = 56
set @studytype_id = 3

set @da = 'Fysik og Informatik'
set @en = 'Physics and Informatics'

insert into Course_StudyTypeCategory
values (@studytypecategory_id,
@studytype_id,
'<cultures>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>' + @en + '</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>' + @da + '</value>
  </culture>
</cultures>',
@curdate, @userid, @curdate, @userid)

-----------------------------------------------------------------------

set @studytypecategory_id = 57
set @studytype_id = 3

set @da = 'Elektronik og elteknik'
set @en = 'Electronics and Electrical Engineering'

insert into Course_StudyTypeCategory
values (@studytypecategory_id,
@studytype_id,
'<cultures>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>' + @en + '</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>' + @da + '</value>
  </culture>
</cultures>',
@curdate, @userid, @curdate, @userid)

-----------------------------------------------------------------------

set @studytypecategory_id = 58
```

```
set @studytype_id = 3

set @da = 'Basiskursus'
set @en = 'Basis Course'

insert into Course_StudyTypeCategory
values (@studytypecategory_id,
@studytype_id,
'<cultures>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>' + @en + '</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>' + @da + '</value>
  </culture>
</cultures>',
@curdate, @userid, @curdate, @userid)
```

3.6 Department

```
declare @userid uniqueidenti↓er
set @userid = '{BC61A572-A19A-41FB-97B8-9B0B463F2DBB}'

declare @curdate datetime
set @curdate = getdate()

insert into Department
values (1, NULL,
'<cultures>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>BioCentrum-DTU</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>BioCentrum-DTU</value>
  </culture>
</cultures>',
27,
@curdate, @userid, @curdate, @userid)

insert into Department
values (2, NULL,
'<cultures>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>Department of Civil Engineering</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>BYG.DTU</value>
  </culture>
</cultures>',
11,
@curdate, @userid, @curdate, @userid)

insert into Department
values (3, NULL,
'<cultures>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>Centre for Traffic and Transport</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>Center for Trafik og Transport</value>
  </culture>
</cultures>',
13,
@curdate, @userid, @curdate, @userid)

insert into Department
values (4, NULL,
'<cultures>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>Research Center COM</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>Forskningscenter COM</value>
  </culture>
</cultures>',
34,
@curdate, @userid, @curdate, @userid)

insert into Department
values (5, NULL,
'<cultures>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>Informatics and Mathematical Modelling</value>
```

```
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>Informatik og Matematisk Modellering</value>
  </culture>
</cultures>',
2,
@curdate, @userid, @curdate, @userid)

insert into Department
values (7, NULL,
'<cultures>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>Department of Physics</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>Institut for Fysik</value>
  </culture>
</cultures>',
10,
@curdate, @userid, @curdate, @userid)

insert into Department
values (8, NULL,
'<cultures>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>Department of Chemical Engineering</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>Institut for Kemiteknik</value>
  </culture>
</cultures>',
28,
@curdate, @userid, @curdate, @userid)

insert into Department
values (9, NULL,
'<cultures>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>Department of Mathematics</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>Institut for Matematik</value>
  </culture>
</cultures>',
1,
@curdate, @userid, @curdate, @userid)

insert into Department
values (10, NULL,
'<cultures>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>Department of Mechanical Engineering</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>Institut for Mekanik, Energi og Konstruktion</value>
  </culture>
</cultures>',
41,
@curdate, @userid, @curdate, @userid)

insert into Department
values (11, NULL,
'<cultures>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>Department of Manufacturing Engineering and Management</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>Institut for Produktion og Ledelse</value>
  </culture>
</cultures>',
42,
@curdate, @userid, @curdate, @userid)

insert into Department
values (12, NULL,
'<cultures>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>Department of Chemistry</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>Kemisk Institut</value>
  </culture>
</cultures>',
26,
@curdate, @userid, @curdate, @userid)
```

```
insert into Department
values (13, NULL,
'<cultures>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>Mikroelektronik Centret</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>Mikroelektronikcentret</value>
  </culture>
</cultures>',
33,
@curdate, @userid, @curdate, @userid)

insert into Department
values (14, NULL,
'<cultures>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>Environment & Resources DTU</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>Miljø & Ressourcer DTU</value>
  </culture>
</cultures>',
12,
@curdate, @userid, @curdate, @userid)

insert into Department
values (15, NULL,
'<cultures>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>Ørsted DTU</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>Ørsted DTU</value>
  </culture>
</cultures>',
31,
@curdate, @userid, @curdate, @userid)

insert into Department
values (16, NULL,
'<cultures>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>Study Division</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>Studieforvaltningen</value>
  </culture>
</cultures>',
83,
@curdate, @userid, @curdate, @userid)
```

3.7 Gender

```
declare @userid uniqueidenti⌊er
set @userid = '{BC61A572-A19A-41FB-97B8-9B0B463F2DBB}'

declare @curdate datetime
set @curdate = getdate()

insert into Gender
values (0,
'<cultures>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>Not known</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>Ukendt</value>
  </culture>
</cultures>',
@curdate, @userid, @curdate, @userid)

insert into Gender
values (1,
'<cultures>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>Male</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>Hankøn</value>
  </culture>
</cultures>',
```

```
@curdate, @userid, @curdate, @userid)

insert into Gender
values (2,
'<cultures>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>Female</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>Hunkøn</value>
  </culture>
</cultures>',
@curdate, @userid, @curdate, @userid)

insert into Gender
values (3,
'<cultures>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>Not specified</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>Uspecificeret</value>
  </culture>
</cultures>',
@curdate, @userid, @curdate, @userid)
```

3.8 Grade

```
declare @userid uniqueidenti↓er
set @userid = '{BC61A572-A19A-41FB-97B8-9B0B463F2DBB}'

declare @curdate datetime
set @curdate = getdate()

insert into Grade
values (1, 0, '00', @curdate, @userid, @curdate, @userid)

insert into Grade
values (2, 3, '03', @curdate, @userid, @curdate, @userid)

insert into Grade
values (3, 5, '5', @curdate, @userid, @curdate, @userid)

insert into Grade
values (4, 6, '6', @curdate, @userid, @curdate, @userid)

insert into Grade
values (5, 7, '7', @curdate, @userid, @curdate, @userid)

insert into Grade
values (6, 8, '8', @curdate, @userid, @curdate, @userid)

insert into Grade
values (7, 9, '9', @curdate, @userid, @curdate, @userid)

insert into Grade
values (8, 10, '10', @curdate, @userid, @curdate, @userid)

insert into Grade
values (9, 11, '11', @curdate, @userid, @curdate, @userid)

insert into Grade
values (10, 13, '13', @curdate, @userid, @curdate, @userid)
```

3.9 Language

```
declare @userid uniqueidenti↓er
set @userid = '{BC61A572-A19A-41FB-97B8-9B0B463F2DBB}'

declare @curdate datetime
set @curdate = getdate()

insert into Language
values ('da',
'<cultures>
  <culture>
    <cultureID>en</cultureID>
    <value>Danish</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>Dansk</value>
  </culture>
</cultures>',
@curdate, @userid, @curdate, @userid)
```

```
insert into Language
values ('en',
'<cultures>
  <culture>
    <cultureID>en</cultureID>
    <value>English</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>Engelsk</value>
  </culture>
</cultures>',
@curdate, @userid, @curdate, @userid)
```

3.10 Module

```
declare @userid uniqueidenti↓er
declare @curdate datetime

set @userid = 'D36DD218-AE59-4046-977C-659E4EB92C92'
set @curdate = getdate()

insert into module
values
(1, '1A', '01-01-1900 08:00:00', '01-01-1900 12:00:00', 1,
@curdate, @userid, @curdate, @userid)

insert into module
values
(2, '1B', '01-01-1900 13:00:00', '01-01-1900 17:00:00', 4,
@curdate, @userid, @curdate, @userid)

insert into module
values
(3, '2A', '01-01-1900 13:00:00', '01-01-1900 17:00:00', 1,
@curdate, @userid, @curdate, @userid)

insert into module
values
(4, '2B', '01-01-1900 08:00:00', '01-01-1900 12:00:00', 4,
@curdate, @userid, @curdate, @userid)

insert into module
values
(5, '3A', '01-01-1900 08:00:00', '01-01-1900 12:00:00', 2,
@curdate, @userid, @curdate, @userid)

insert into module
values
(6, '3B', '01-01-1900 13:00:00', '01-01-1900 17:00:00', 5,
@curdate, @userid, @curdate, @userid)

insert into module
values
(7, '4A', '01-01-1900 13:00:00', '01-01-1900 17:00:00', 2,
@curdate, @userid, @curdate, @userid)

insert into module
values
(8, '4B', '01-01-1900 08:00:00', '01-01-1900 12:00:00', 5,
@curdate, @userid, @curdate, @userid)

insert into module
values
(9, '5A', '01-01-1900 08:00:00', '01-01-1900 12:00:00', 3,
@curdate, @userid, @curdate, @userid)

insert into module
values
(10, '5B', '01-01-1900 13:00:00', '01-01-1900 17:00:00', 3,
@curdate, @userid, @curdate, @userid)

insert into module
values
(11, '6', '01-01-1900 17:00:00', '01-01-1900 21:00:00', 1,
@curdate, @userid, @curdate, @userid)

insert into module
values
(12, '7', '01-01-1900 17:00:00', '01-01-1900 21:00:00', 2,
@curdate, @userid, @curdate, @userid)

insert into module
values
(13, '8', '01-01-1900 17:00:00', '01-01-1900 21:00:00', 3,
@curdate, @userid, @curdate, @userid)

insert into module
values
(14, '9', '01-01-1900 17:00:00', '01-01-1900 21:00:00', 4,
@curdate, @userid, @curdate, @userid)

insert into module
```

```
values
(15, '10', '01-01-1900 17:00:00', '01-01-1900 21:00:00', 5,
@curdate, @userid, @curdate, @userid)
```

3.11 Period

```
-------------------------------------------------------------------------
-- Common declarations
-------------------------------------------------------------------------
declare @userid uniqueidenti↓er
declare @curdate datetime
declare @da varchar(255), @en varchar(255)
declare @id uniqueidenti↓er
declare @13week int, @3week int
declare @start datetime, @end datetime
set @userid = 'D36DD218-AE59-4046-977C-659E4EB92C92'
set @curdate = getdate()
set @13week = 1
set @3week = 2

set @id = 'f8adcced-7dfa-4482-91e4-5b30d6eaddac'
set @da = '13-ugers perioden efterår 1999'
set @en = '13-week period fall 1999'
set @start = '1999-08-30'
set @end = '1999-12-10'

insert into period
  values (@id, @13week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = '9d2470b2-5b03-4634-8020-5e32ba840800'
set @da = '3-ugers perioden efterår 1999'
set @en = '3-week period fall 1999'
set @start = '2000-01-10'
set @end = '2000-01-28'

insert into period
  values (@id, @3week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = 'fd2c7042-37f8-442c-9682-6fdf7894064d'
set @da = '13-ugers perioden forår 2000'
set @en = '13-week period spring 2000'
set @start = '2000-01-31'
set @end = '2000-05-16'

insert into period
  values (@id, @13week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = '2bbf418a-3858-457a-b1d3-13a8cfa01595'
set @da = '3-ugers perioden forår 2000'
set @en = '3-week period spring 2000'
set @start = '2000-06-13'
set @end = '2000-06-30'

insert into period
  values (@id, @3week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
```

```
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = 'b622ada2-a0f6-4513-ab30-20aeb9aaccdf'
set @da = '13-ugers perioden efterår 2000'
set @en = '13-week period fall 2000'
set @start = '2000-09-04'
set @end = '2000-12-15'

insert into period
  values (@id, @13week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = 'a3aa81dd-3f6e-41a6-b300-5b0e8063302a'
set @da = '3-ugers perioden efterår 2000'
set @en = '3-week period fall 2000'
set @start = '2001-01-15'
set @end = '2001-02-02'

insert into period
  values (@id, @3week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = '6576004e-9d2d-49fc-8604-7bd85383e914'
set @da = '13-ugers perioden forår 2001'
set @en = '13-week period forår 2001'
set @start = '2001-02-15'
set @end = '2001-05-23'

insert into period
  values (@id, @13week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = 'ac952191-71b2-4e08-8689-858c15e18ecf'
set @da = '3-ugers perioden forår 2001'
set @en = '3-week period forår 2001'
set @start = '2001-06-18'
set @end = '2001-07-06'

insert into period
  values (@id, @3week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = 'fa6fbc9c-a515-4b6c-aa87-633f90f26627'
set @da = '13-ugers perioden efterår 2001'
set @en = '13-week period fall 2001'
set @start = '2001-09-03'
set @end = '2001-12-17'

insert into period
  values (@id, @13week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
```

```
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = '1dad1ee7-4167-4d84-8546-d808561d1e60'
set @da = '3-ugers perioden efterår 2001'
set @en = '3-week period fall 2001'
set @start = '2002-01-07'
set @end = '2002-01-25'

insert into period
  values (@id, @3week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = 'c90bd3d8-9ea2-4df8-a15f-2cd117b4b6e5'
set @da = '13-ugers perioden forår 2002'
set @en = '13-week period spring 2002'
set @start = '2002-02-04'
set @end = '2002-05-17'

insert into period
  values (@id, @13week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = '574a854f-e07b-4918-b6ae-8ac4767db158'
set @da = '3-ugers perioden forår 2002'
set @en = '3-week period spring 2002'
set @start = '2002-06-10'
set @end = '2002-06-28'

insert into period
  values (@id, @3week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = '754215d2-0aec-4519-8e68-cdcdedee00c7'
set @da = '13-ugers perioden efterår 2002'
set @en = '13-week period fall 2002'
set @start = '2002-09-02'
set @end = '2002-12-06'

insert into period
  values (@id, @13week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = '689a9a32-ac3c-4677-ad85-58d3a47c476b'
set @da = '3-ugers perioden efterår 2002'
set @en = '3-week period fall 2002'
set @start = '2003-01-06'
set @end = '2003-01-24'

insert into period
  values (@id, @3week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
```

```
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = 'f1fa563b-6baa-4643-8d02-4bb5d0fd880e'
set @da = '13-ugers perioden forår 2003'
set @en = '13-week period spring 2003'
set @start = '2003-02-03'
set @end = '2003-05-13'

insert into period
  values (@id, @13week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = '26a06200-cc04-4d6f-b585-6c55cf408be9'
set @da = '3-ugers perioden forår 2003'
set @en = '3-week period spring 2003'
set @start = '2003-06-16'
set @end = '2003-07-04'

insert into period
  values (@id, @3week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = '8c264cc5-caec-4e13-addb-a380a1cb7c24'
set @da = '13-ugers perioden efterår 2003'
set @en = '13-week period fall 2003'
set @start = '2003-09-01'
set @end = '2003-12-05'

insert into period
  values (@id, @13week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = 'b39d8aef-92e7-4c58-8893-30aef6651a95'
set @da = '3-ugers perioden efterår 2003'
set @en = '3-week period fall 2003'
set @start = '2004-01-05'
set @end = '2004-01-23'

insert into period
  values (@id, @3week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = '58cb73fd-facc-4f9a-bc26-7778edd477f1'
set @da = '13-ugers perioden forår 2004'
set @en = '13-week period spring 2004'
set @start = '2004-02-02'
set @end = '2004-05-11'

insert into period
  values (@id, @13week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
```

```
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = '09fc3db8-beaf-4efd-9fc5-3fba302ccb37'
set @da = '3-ugers perioden forår 2004'
set @en = '3-week period spring 2004'
set @start = '2004-06-07'
set @end = '2004-06-25'

insert into period
  values (@id, @3week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = '9f7c935a-5a79-4362-b353-3a0d16ae2c45'
set @da = '13-ugers perioden efterår 2004'
set @en = '13-week period fall 2004'
set @start = '2004-08-30'
set @end = '2004-12-03'

insert into period
  values (@id, @13week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = '6ce3daf5-8b1d-4732-9565-126b885c2844'
set @da = '3-ugers perioden efterår 2004'
set @en = '3-week period fall 2004'
set @start = '2005-01-10'
set @end = '2005-01-28'

insert into period
  values (@id, @3week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = '6ba71b64-e217-4c9e-a54e-f527b2e6db6a'
set @da = '13-ugers perioden forår 2005'
set @en = '13-week period spring 2005'
set @start = '2005-02-07'
set @end = '2005-05-13'

insert into period
  values (@id, @13week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = '4ba753f2-c343-45a0-9b4c-da430a3866a6'
set @da = '3-ugers perioden forår 2005'
set @en = '3-week period spring 2005'
set @start = '2005-06-13'
set @end = '2005-07-01'

insert into period
  values (@id, @3week,
  '<cultures>
    <culture>
```

```
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = '3fe2e855-1e2e-4154-b509-84c5e0cefe7a'
set @da = '13-ugers perioden efterår 2005'
set @en = '13-week period fall 2005'
set @start = '2005-09-05'
set @end = '2005-12-16'

insert into period
  values (@id, @13week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = 'c43d4565-0b27-478c-82e5-acc96f2cfc61'
set @da = '3-ugers perioden efterår 2005'
set @en = '3-week period fall 2005'
set @start = '2006-01-09'
set @end = '2006-01-27'

insert into period
  values (@id, @3week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = '1140a8e2-646c-482b-9c76-e1bf805ccef9'
set @da = '13-ugers perioden forår 2006'
set @en = '13-week period spring 2006'
set @start = '2006-02-06'
set @end = '2006-05-19'

insert into period
  values (@id, @13week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = '1f3f20d8-c3d2-4544-8e18-e2c76e80bd10'
set @da = '3-ugers perioden forår 2006'
set @en = '3-week period spring 2006'
set @start = '2006-06-05'
set @end = '2006-06-23'

insert into period
  values (@id, @3week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = '9b0068ed-a35a-45eb-830c-d9c8ac8a83f8'
set @da = '13-ugers perioden efterår 2006'
set @en = '13-week period fall 2006'
set @start = '2006-09-04'
set @end = '2006-12-15'

insert into period
  values (@id, @13week,
  '<cultures>
```

```
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = '566a30c2-fa73-46a3-96a6-5cbd666ecdec'
set @da = '3-ugers perioden efterår 2006'
set @en = '3-week period fall 2006'
set @start = '2007-01-08'
set @end = '2007-01-26'

insert into period
  values (@id, @3week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = '05867a9e-520f-4f74-b2d9-79fcd1b0764a'
set @da = '13-ugers perioden forår 2007'
set @en = '13-week period spring 2007'
set @start = '2007-02-05'
set @end = '2007-05-18'

insert into period
  values (@id, @13week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = '48d572b1-50a9-496f-bd47-3d50ac3af34e'
set @da = '3-ugers perioden forår 2007'
set @en = '3-week period spring 2007'
set @start = '2007-06-11'
set @end = '2007-06-29'

insert into period
  values (@id, @3week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = '1ba70c44-84a0-40b4-8740-b40b14a13928'
set @da = '13-ugers perioden efterår 2007'
set @en = '13-week period fall 2007'
set @start = '2007-09-03'
set @end = '2007-12-14'

insert into period
  values (@id, @13week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = '438ca3f7-5ece-48c9-8807-fd857bd25723'
set @da = '3-ugers perioden efterår 2007'
set @en = '3-week period fall 2007'
set @start = '2008-01-07'
set @end = '2008-01-25'

insert into period
  values (@id, @3week,
```

```
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = 'aaf0b7b8-201f-4aad-bd7b-06c43c3a5923'
set @da = '13-ugers perioden forår 2008'
set @en = '13-week period spring 2008'
set @start = '2008-02-04'
set @end = '2008-05-16'

insert into period
  values (@id, @13week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = '78c4114f-1234-4217-a7f7-0de32de5b460'
set @da = '3-ugers perioden forår 2008'
set @en = '3-week period spring 2008'
set @start = '2008-06-09'
set @end = '2008-06-27'

insert into period
  values (@id, @3week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = 'a75a3b73-143c-4b8b-ac1b-b5e753e491ce'
set @da = '13-ugers perioden efterår 2008'
set @en = '13-week period fall 2008'
set @start = '2008-09-01'
set @end = '2008-12-12'

insert into period
  values (@id, @13week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = 'e979c701-c244-46fa-b037-e28a9cd6f8ad'
set @da = '3-ugers perioden efterår 2008'
set @en = '3-week period fall 2008'
set @start = '2009-01-05'
set @end = '2009-01-23'

insert into period
  values (@id, @3week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = '19c863f2-1f68-4eae-8b25-54820823dfdf'
set @da = '13-ugers perioden forår 2009'
set @en = '13-week period spring 2009'
set @start = '2009-02-02'
set @end = '2009-05-15'

insert into period
```

```
  values (@id, @13week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = '57bbe48c-071a-4aed-ba0d-636c460ad791'
set @da = '3-ugers perioden forår 2009'
set @en = '3-week period spring 2009'
set @start = '2009-06-08'
set @end = '2009-06-26'

insert into period
  values (@id, @3week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = '2f2c6edd-9c32-4fe6-ad86-a3b26730bc7d'
set @da = '13-ugers perioden efterår 2009'
set @en = '13-week period efterår 2009'
set @start = '2009-08-31'
set @end = '2009-12-11'

insert into period
  values (@id, @13week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = 'f678f377-1496-4110-9223-2bd7277ea4c9'
set @da = '3-ugers perioden efterår 2009'
set @en = '3-week period efterår 2009'
set @start = '2010-01-11'
set @end = '2010-01-29'

insert into period
  values (@id, @3week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = '277e2de4-edeb-4e7b-b6a4-75b35f59c5f8'
set @da = '13-ugers perioden forår 2010'
set @en = '13-week period spring 2010'
set @start = '2010-02-01'
set @end = '2010-05-14'

insert into period
  values (@id, @13week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = '8e502cc1-a408-407d-8616-6cd6d40aeabc'
set @da = '3-ugers perioden forår 2010'
set @en = '3-week period spring 2010'
set @start = '2010-06-07'
set @end = '2010-06-25'
```

```
insert into period
  values (@id, @3week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = 'a562cd66-27ac-495b-aa58-7d6e49174027'
set @da = '13-ugers perioden efterår 2010'
set @en = '13-week period fall 2010'
set @start = '2010-08-30'
set @end = '2010-12-10'

insert into period
  values (@id, @13week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = '12021911-cc73-43b4-82df-a747938c2903'
set @da = '3-ugers perioden efterår 2010'
set @en = '3-week period fall 2010'
set @start = '2011-01-10'
set @end = '2011-01-28'

insert into period
  values (@id, @3week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = '7339211c-95cb-4b32-b3af-7fce42c4c8af'
set @da = '13-ugers perioden forår 2011'
set @en = '13-week period spring 2011'
set @start = '2011-02-07'
set @end = '2011-05-13'

insert into period
  values (@id, @13week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = '99a22b16-e867-4d08-bf13-ae6a92730f7a'
set @da = '3-ugers perioden forår 2011'
set @en = '3-week period spring 2011'
set @start = '2011-06-06'
set @end = '2011-06-24'

insert into period
  values (@id, @3week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = '754276e3-75da-46e9-a525-4fe7142a66a2'
set @da = '13-ugers perioden efterår 2011'
set @en = '13-week period fall 2011'
set @start = '2011-08-29'
set @end = '2011-12-02'
```

```
insert into period
  values (@id, @13week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = '723b7728-2d6d-4574-ad89-ba27bd58e9c0'
set @da = '3-ugers perioden efterår 2011'
set @en = '3-week period fall 2011'
set @start = '2012-01-09'
set @end = '2012-01-27'

insert into period
  values (@id, @3week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = '66c8e7d0-e4d9-49d7-ad2b-edcae7b8c467'
set @da = '13-ugers perioden forår 2012'
set @en = '13-week period spring 2012'
set @start = '2012-02-06'
set @end = '2012-05-11'

insert into period
  values (@id, @13week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = 'ef9f1b68-21f7-4790-9afe-072c8bc754c3'
set @da = '3-ugers perioden forår 2012'
set @en = '3-week period spring 2012'
set @start = '2012-06-11'
set @end = '2012-06-29'

insert into period
  values (@id, @3week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = '60d59062-91fb-40fc-acd7-6312ba5bd7e4'
set @da = '13-ugers perioden efterår 2012'
set @en = '13-week period fall 2012'
set @start = '2012-09-03'
set @end = '2012-12-07'

insert into period
  values (@id, @13week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = '0128b683-4583-4672-b4fc-6bec711b94de'
set @da = '3-ugers perioden efterår 2012'
set @en = '3-week period fall 2012'
set @start = '2013-01-14'
```

```
set @end = '2013-02-01'

insert into period
  values (@id, @3week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = 'e56ee5e6-b09a-4b32-92d0-9a3200f017b6'
set @da = '13-ugers perioden forår 2013'
set @en = '13-week period spring 2013'
set @start = '2013-02-11'
set @end = '2013-05-10'

insert into period
  values (@id, @13week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = '7bf411e6-a712-45e1-acaf-eb127f979aa5'
set @da = '3-ugers perioden forår 2013'
set @en = '3-week period spring 2013'
set @start = '2013-06-10'
set @end = '2013-06-28'

insert into period
  values (@id, @3week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = 'ab9224c1-b47c-4dd2-b07c-0cd083f446dc'
set @da = '13-ugers perioden efterår 2013'
set @en = '13-week period fall 2013'
set @start = '2013-09-02'
set @end = '2013-11-29'

insert into period
  values (@id, @13week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)

set @id = 'c559fd87-bf66-4381-86ee-b5ea5416173e'
set @da = '3-ugers perioden efterår 2013'
set @en = '3-week period fall 2013'
set @start = '2014-01-13'
set @end = '2014-01-31'

insert into period
  values (@id, @3week,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @start, @end, @curdate, @userid, @curdate, @userid)
```

3.12 PeriodType

```
------------------------------------------------------------------------
--Common declarations
------------------------------------------------------------------------
declare @userid uniqueidenti↓er
declare @curdate datetime
declare @da varchar(255), @en varchar(255)
declare @id int
set @userid = 'D36DD218-AE59-4046-977C-659E4EB92C92'
set @curdate = getdate()
set @id = 0

set @id = @id + 1
set @da = '13-ugers periode'
set @en = '13-week period'

insert into periodtype
  values (@id,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @curdate, @userid, @curdate, @userid)

set @id = @id + 1
set @da = '3-ugers periode'
set @en = '3-week period'

insert into periodtype
  values (@id,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @curdate, @userid, @curdate, @userid)

set @id = @id + 1
set @da = 'Individuel periode'
set @en = 'Individual period'

insert into periodtype
  values (@id,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
  @curdate, @userid, @curdate, @userid)
```

3.12.1 PeriodType_Module

```
------------------------------------------------------------------------
--Common declarations
------------------------------------------------------------------------
declare @userid uniqueidenti↓er
declare @curdate datetime
declare @id int
declare @module int
set @userid = 'D36DD218-AE59-4046-977C-659E4EB92C92'
set @curdate = getdate()
set @id = 0
set @module = 1

while @module < 16
begin
  set @id = @id + 1
  insert into periodtype_module
  values(@id, 1, @module, @curdate, @userid, @curdate, @userid)
  set @module = @module + 1
end
```

3.13 Point

3.13.1 Bachelor of Science Trainee Projects

```
declare @userid uniqueidenti↓er
declare @curdate datetime

set @userid = '{BC61A572-A19A-41FB-97B8-9B0B463F2DBB}'
set @curdate = getdate()

declare @pid uniqueidenti↓er

------------------------------------------------------------------------

------------------------------------------
-- Arctic Engineering Training Project
------------------------------------------
set @pid = '{B347A190-1A9B-4c23-9576-9FE7CF2CA62B}'
insert into Point values(@pid, 30, 30, 0, @curdate, @userid, @curdate, @userid)

------------------------------------------
-- Engineering Training Project
------------------------------------------
set @pid = '{E8FA9A2F-A0D8-4b7d-ADE8-A091818DC8DC}'
insert into Point values(@pid, 30, 30, 0, @curdate, @userid, @curdate, @userid)

------------------------------------------
-- Civil Engineering Design Project
------------------------------------------
set @pid = '{FE3F0B68-E455-49c5-856E-96D61F44F732}'
insert into Point values(@pid, 30, 30, 0, @curdate, @userid, @curdate, @userid)

------------------------------------------------------------
-- Industrial Training Project (Electrical Engineering)
------------------------------------------------------------
set @pid = '{2F3A67B5-4E5E-4168-A1BA-D0A5371DAA0A}'
insert into Point values(@pid, 30, 30, 0, @curdate, @userid, @curdate, @userid)

------------------------------------------------------
-- Industrial Placement (Mechanical Engineering)
------------------------------------------------------
set @pid = '{8C72F4C9-C27C-4530-8778-5F16E322F744}'
insert into Point values(@pid, 30, 30, 0, @curdate, @userid, @curdate, @userid)

------------------------------------------------------
-- Industrial Placement (Production and Plastic)
------------------------------------------------------
set @pid = '{56E03A48-788B-4cd0-911D-BCBAC02DD3B6}'
insert into Point values(@pid, 30, 30, 0, @curdate, @userid, @curdate, @userid)

------------------------------------------------------
-- Engineering Trainee (Chemical Engineering)
------------------------------------------------------
set @pid = '{FEA8D4C1-8CE5-49cf-B1B5-F9E18E9F515D}'
insert into Point values(@pid, 30, 30, 0, @curdate, @userid, @curdate, @userid)

------------------------------------------------------
-- Industrial Training (Information Technology)
------------------------------------------------------
set @pid = '{102A0A1F-2956-48c3-AF1F-DAFEEDF16D26}'
insert into Point values(@pid, 30, 30, 0, @curdate, @userid, @curdate, @userid)
```

3.13.2 StudyType

```
------------------------------------------------------------------------------------------------

--Common declarations
------------------------------------------------------------------------------------------------

declare @userid uniqueidenti↓er
declare @curdate datetime
declare @pid uniqueidenti↓er
set @userid = 'D36DD218-AE59-4046-977C-659E4EB92C92'
set @curdate = getdate()

set @pid = 'c345c73c-1043-4167-b4a4-f7ebb3be2f23' --Master
insert into point values(@pid, 300, 300, 0, @curdate, @userid, @curdate, @userid)

set @pid = 'f57cec33-2b5c-4c0c-a6c0-288abd01f198' --Bachelor
insert into point values(@pid, 210, 210, 0, @curdate, @userid, @curdate, @userid)

set @pid = '502161fe-7dfe-4df8-9e2c-e2d925458e54' --Ph.D.
insert into point values(@pid, 90, 90, 0, @curdate, @userid, @curdate, @userid)

set @pid = 'a03eee09-810d-450b-b530-08c6853700ef' --Open university
insert into point values(@pid, 90, 90, 0, @curdate, @userid, @curdate, @userid)

set @pid = '55dd6271-e386-4a88-8ffe-c2d4729b9fe6' --Food science
insert into point values(@pid, 90, 90, 0, @curdate, @userid, @curdate, @userid)
```

3.13.3 StudyType_ProjectType

```
declare @userid uniqueidenti↓er
declare @curdate datetime

set @userid = '{BC61A572-A19A-41FB-97B8-9B0B463F2DBB}'
set @curdate = getdate()

declare @pid uniqueidenti↓er

-------------------------------------------------------------------------------

-----------------------------
-- Master of Science
-----------------------------

-- Midway Project - placement
-- Confer page 73 in the "Studiehåndbog - Studieplaner, Regelsamling"
set @pid = '{037B9864-77CD-4299-B5E2-6D5A70BFB02F}'
insert into Point values(@pid, 80, 190, 0, @curdate, @userid, @curdate, @userid)

-- Midway Project - workload
-- Confer page 73 in the "Studiehåndbog - Studieplaner, Regelsamling"
set @pid = '{c6418073-365e-4f6b-a410-543db11df74a}'
insert into Point values(@pid, 12.5, 12.5, 0, @curdate, @userid, @curdate, @userid)

-- Thesis - placement
-- Confer page 76 in the "Studiehåndbog - Studieplaner, Regelsamling"
set @pid = '{19032106-A382-4b6e-BB4E-D8BE7AC6E572}'
insert into Point values(@pid, 235, null, 0, @curdate, @userid, @curdate, @userid)

-- Thesis - workload
-- Confer page 76 in the "Studiehåndbog - Studieplaner, Regelsamling"
set @pid = '{a7a7eda2-0251-49ec-8fb6-27cf40462e27}'
insert into Point values(@pid, 30, 30, 0, @curdate, @userid, @curdate, @userid)

-----------------------------
-- Bachelor of Science
-----------------------------

-- Thesis - placement
-- Confer chapter 5 in the "Studiehåndbog - Studieplaner, Regelsamling". The
-- bachelor thesis project is placed at 7th term i.e. at the last term of
-- the bachelor of science study.
set @pid = '{B2C990CA-F827-4945-B683-5F711EFBC2DE}'
insert into point values(@pid, 180, null, 0, @curdate, @userid, @curdate, @userid)

-- Thesis - workload
-- Confer chapter 5 in the "Studiehåndbog - Studieplaner, Regelsamling". The
-- bachelor thesis project is placed at 7th term i.e. at the last term of
-- the bachelor of science study.
set @pid = '{46dff3f0-ddad-473d-ae93-a39b3bd05b78}'
insert into point values(@pid, 15, 15, 0, @curdate, @userid, @curdate, @userid)
```

3.13.4 TechnicalLine

```
-------------------------------------------------------------------------------------------------------

--Common declarations
-------------------------------------------------------------------------------------------------------

declare @userid uniqueidenti↓er
declare @curdate datetime
declare @pid uniqueidenti↓er
set @userid = 'D36DD218-AE59-4046-977C-659E4EB92C92'
set @curdate = getdate()

set @pid = '4054b9a7-ae3a-40b6-b96e-14dc212afb81'
insert into point values(@pid, 90, 90, 0, @curdate, @userid, @curdate, @userid)

set @pid = '035e46a9-b98f-49cb-9ed0-30521bdb6dc0'
insert into point values(@pid, 90, 90, 0, @curdate, @userid, @curdate, @userid)

set @pid = '95877f88-e661-4657-a874-ff2065b7e653'
insert into point values(@pid, 90, 90, 0, @curdate, @userid, @curdate, @userid)

set @pid = 'bce9eddf-0ed6-4111-adef-88ac9f256895'
insert into point values(@pid, 90, 90, 0, @curdate, @userid, @curdate, @userid)

set @pid = '720c1413-5fc1-48ac-bb67-ffa9f3f4cc3b'
insert into point values(@pid, 90, 90, 0, @curdate, @userid, @curdate, @userid)

set @pid = 'f3ea98c4-1811-4c65-911a-e1217db87271'
insert into point values(@pid, 90, 90, 0, @curdate, @userid, @curdate, @userid)

set @pid = '575f214c-5ced-4d8e-a2d0-53ec109fc770'
insert into point values(@pid, 90, 90, 0, @curdate, @userid, @curdate, @userid)

set @pid = 'df49fff5-e6f2-4610-9f63-b6e5e8b0e522'
insert into point values(@pid, 90, 90, 0, @curdate, @userid, @curdate, @userid)

set @pid = '13afba8b-93a9-4e4f-b07b-4349a4651528'
insert into point values(@pid, 90, 90, 0, @curdate, @userid, @curdate, @userid)
```

```
set @pid = '896e1da4-eb3f-471e-8938-ffa550731c9d'
insert into point values(@pid, 90, 90, 0, @curdate, @userid, @curdate, @userid)

set @pid = '305a078e-6c54-40cc-a8b6-8202f2422cbb'
insert into point values(@pid, 90, 90, 0, @curdate, @userid, @curdate, @userid)

set @pid = 'a7a395ef-56b5-4753-9b70-9976cd9aafd0'
insert into point values(@pid, 90, 90, 0, @curdate, @userid, @curdate, @userid)

set @pid = '2932da0b-621e-4f8c-b7da-b318f57d2ff2'
insert into point values(@pid, 90, 90, 0, @curdate, @userid, @curdate, @userid)

set @pid = '5a3a437c-834e-41fd-8861-e867b74bf62c'
insert into point values(@pid, 90, 90, 0, @curdate, @userid, @curdate, @userid)
```

3.14 ProjectType

```
declare @userid uniqueidenti⌋er
declare @curdate datetime

set @userid = '{BC61A572-A19A-41FB-97B8-9B0B463F2DBB}'
set @curdate = getdate()

-- Notice, a technical package project is considered as a
-- course and does therefore not exists as a project type.

insert into ProjectType
values
(1,
'<cultures>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>Midtvejsprojekt</value>
  </culture>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>Midway Project</value>
  </culture>
</cultures>',
@curdate, @userid, @curdate, @userid)

insert into ProjectType
values
(2,
'<cultures>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>Eksamensprojekt</value>
  </culture>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>Thesis</value>
  </culture>
</cultures>',
@curdate, @userid, @curdate, @userid)

insert into ProjectType
values
(3,
'<cultures>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>Forprojekt til eksamensprojekt</value>
  </culture>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>Thesis Pre-project</value>
  </culture>
</cultures>',
@curdate, @userid, @curdate, @userid)

insert into ProjectType
values
(4,
'<cultures>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>Specialprojekt</value>
  </culture>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>Special Project</value>
  </culture>
</cultures>',
@curdate, @userid, @curdate, @userid)

insert into ProjectType
values
(5,
'<cultures>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>Praktikprojekt</value>
  </culture>
```

```
  <culture>
    <cultureID>en-GB</cultureID>
    <value>Trainee Project</value>
  </culture>
</cultures>',
@curdate, @userid, @curdate, @userid)
```

3.15 RecommendedPlacementConcept

```
declare @userid uniqueidenti↓er
set @userid = '{BC61A572-A19A-41FB-97B8-9B0B463F2DBB}'

declare @curdate datetime
set @curdate = getdate()
```

```
insert into RecommendedPlacementConcept
values (1,
'<cultures>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>Early in the study</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>Først i studiet</value>
  </culture>
</cultures>',
@curdate, @userid, @curdate, @userid)

    insert into RecommendedPlacementConcept_StudyType
    values(1,
    1,
    1,
    '{84F7B4B9-67C3-455c-837E-19E9FC0E20B8}',
    @curdate, @userid, @curdate, @userid)

    insert into Point
    values('{84F7B4B9-67C3-455c-837E-19E9FC0E20B8}',
    0,
    100,
    null,
    @curdate, @userid, @curdate, @userid)

    ---

    insert into RecommendedPlacementConcept_StudyType
    values(2,
    1,
    2,
    '{90F55A05-8B20-4c9b-9292-9DDC2F15D8F9}',
    @curdate, @userid, @curdate, @userid)

    insert into Point
    values('{90F55A05-8B20-4c9b-9292-9DDC2F15D8F9}',
    0,
    70,
    null,
    @curdate, @userid, @curdate, @userid)
```

```
insert into RecommendedPlacementConcept
values (2,
'<cultures>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>In the middle of the study.</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>Midt i studiet.</value>
  </culture>
</cultures>',
@curdate, @userid, @curdate, @userid)

    insert into RecommendedPlacementConcept_StudyType
    values(10,
    2,
    1,
    '{BF837A7D-3EC0-43ed-9C73-FBB433AD5D64}',
    @curdate, @userid, @curdate, @userid)

    insert into Point
    values('{BF837A7D-3EC0-43ed-9C73-FBB433AD5D64}',
    100,
    200,
    null,
    @curdate, @userid, @curdate, @userid)

    ---
```

```
    insert into RecommendedPlacementConcept_StudyType
    values(11,
    2,
    2,
    '{68648134-ED60-4907-8AB5-7024A00CF26F}',
    @curdate, @userid, @curdate, @userid)

    insert into Point
    values('{68648134-ED60-4907-8AB5-7024A00CF26F}',
    70,
    140,
    null,
    @curdate, @userid, @curdate, @userid)
```

```
insert into RecommendedPlacementConcept
values (3,
'<cultures>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>At the end of the study.</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>Sidst i studiet.</value>
  </culture>
</cultures>',
@curdate, @userid, @curdate, @userid)

    insert into RecommendedPlacementConcept_StudyType
    values(20,
    3,
    1,
    '{4E5C63F9-3726-4330-96B6-6293BC23337E}',
    @curdate, @userid, @curdate, @userid)

    insert into Point
    values('{4E5C63F9-3726-4330-96B6-6293BC23337E}',
    200,
    300,
    null,
    @curdate, @userid, @curdate, @userid)

    ---

    insert into RecommendedPlacementConcept_StudyType
    values(21,
    3,
    2,
    '{D4749A01-B044-41c9-A167-19831899205C}',
    @curdate, @userid, @curdate, @userid)

    insert into Point
    values('{D4749A01-B044-41c9-A167-19831899205C}',
    140,
    210,
    null,
    @curdate, @userid, @curdate, @userid)
```

3.16 SecurityRole

```
declare @userid uniqueidenti⌊er
set @userid = '{BC61A572-A19A-41FB-97B8-9B0B463F2DBB}'

declare @curdate datetime
set @curdate = getdate()

insert into SecurityRole
values (1, 'Student', @curdate, @userid, @curdate, @userid)

insert into SecurityRole
values (2, 'Lecturer', @curdate, @userid, @curdate, @userid)
```

3.17 Specialization

```
-------------------------------------------------------------------
--Common declarations
-------------------------------------------------------------------
declare @userid uniqueidenti⌊er
declare @curdate datetime
declare @da varchar(255), @en varchar(255)
declare @sid int, @svid int, @version int, @p_min int, @p_max int, @scid int
set @userid = 'D36DD218-AE59-4046-977C-659E4EB92C92'
```

```
set @curdate = getdate()
set @sid = 0
set @svid = 0
set @scid = 0
set @version = 0
set @p_min = 0
set @p_max = 0
------------------------------------------------------------------------------
--Data Security and Communication
------------------------------------------------------------------------------
set @da = 'Datasikkerhed og kommunikation'
set @en = 'Data Security and Communication'
set @pid = '{51FCE253-99A1-4baf-BD37-2309DFF40090}'
set @p_min = 1
set @p_max = 1

set @sid = @sid + 1
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)

set @svid = @svid + 1
set @version = 1

insert into specializationversion
  values (@svid, @sid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '01227', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '01259', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '01425', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02230', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02240', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34230', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34240', 1, 0, @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------
--Operations Research
------------------------------------------------------------------------------
set @da = 'Operationsanalyse'
set @en = 'Operations Research'
set @pid = '{BEEA1AA8-1BB6-4252-B232-0B2324F99C1D}'
set @p_min = 1
set @p_max = 1

set @sid = @sid + 1
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)

set @svid = @svid + 1
set @version = 1

insert into specializationversion
  values (@svid, @sid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02443', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02709', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02711', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
```

```
insert into specialization_course
values(@scid, @svid, null, '02713', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02715', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02721', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02723', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '13430', 1, 0, @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------
--Dynamic Systems and Nonlinear Dynamics
------------------------------------------------------------------------------
set @da = 'Dynamiske systemer og ikke-lineær dynamik'
set @en = 'Dynamic Systems and Nonlinear Dynamics'
set @pid = '{9482B2C0-6309-4e41-8C2D-2209C6B19A79}'
set @p_min = 1
set @p_max = 1

set @sid = @sid + 1
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)

set @svid = @svid + 1
set @version = 1

insert into specializationversion
  values (@svid, @sid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '01246', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '01449', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02417', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02421', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02443', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02613', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02623', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02645', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02655', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02685', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02687', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10342', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10344', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '28260', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31310', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31320', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31360', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41223', 1, 0, @curdate, @userid, @curdate, @userid)
```

```
------------------------------------------------------------------------------
--Scientific Computing
------------------------------------------------------------------------------
set @da = 'Scientific computing'
set @en = 'Scientific Computing'
set @pid = '{B6E61897-448D-42de-BD1F-D70B621F5852}'
set @p_min = 1
set @p_max = 1

set @sid = @sid + 1
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)

set @svid = @svid + 1
set @version = 1

insert into specializationversion
  values (@svid, @sid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '01243', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '01246', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '01250', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '01456', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02581', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02611', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02613', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02623', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02645', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02685', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02715', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10112', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11105', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11513', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31430', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41124', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41223', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41320', 1, 0, @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------
--Data Analysis and Applied Statistics
------------------------------------------------------------------------------
set @da = 'Dataanalyse og anvendt statistik'
set @en = 'Data Analysis and Applied Statistics'
set @pid = '{E97ECD5D-5952-43b3-83CC-64261A7822F0}'
set @p_min = 1
set @p_max = 1

set @sid = @sid + 1
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)
```

```
set @svid = @svid + 1
set @version = 1

insert into specializationversion
  values (@svid, @sid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '01250', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '01227', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02407', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02409', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02411', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02413', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02417', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02421', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02423', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02427', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02443', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02451', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02453', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02457', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02503', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02551', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02561', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02581', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27511', 1, 0, @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------
--Signal and Image Processing
------------------------------------------------------------------------------
set @da = 'Signal- og billedbehandling'
set @en = 'Signal and Image Processing'
set @pid = '{DF32C4D6-69BD-4c99-BCF2-6E1837565F35}'
set @p_min = 1
set @p_max = 1

set @sid = @sid + 1
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)

set @svid = @svid + 1
set @version = 1

insert into specializationversion
  values (@svid, @sid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
```

```
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '01243', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '01456', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02409', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02417', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02451', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02453', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02457', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02503', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02505', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02541', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02551', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02561', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02563', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02565', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27011', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31610', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31655', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34230', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34241', 1, 0, @curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------
--Molecular Microbiology
-------------------------------------------------------------------------------
set @da = 'Molekylær mikrobiologi'
set @en = 'Molecular Microbiology'
set @pid = '{188be628-a776-42c0-ac93-9410428e27f6}'
set @p_min = 1
set @p_max = 1

set @sid = @sid + 1
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)

set @svid = @svid + 1
set @version = 1

insert into specializationversion
  values (@svid, @sid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27214', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
```

```
values(@scid, @svid, null, '27217', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27221', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27231', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27443', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27251', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27252', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27405', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27722', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27723', 1, 0, @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------
--Environmental Biology and Bio-Diversity
------------------------------------------------------------------------------
set @da = 'Miljømikrobiologi og biodiversitet'
set @en = 'Environmental Biology and Bio-Diversity'
set @pid = '{12daaad3-3921-4c58-a026-6e65065f0f53}'
set @p_min = 1
set @p_max = 1

set @sid = @sid + 1
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)

set @svid = @svid + 1
set @version = 1

insert into specializationversion
  values (@svid, @sid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27262', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27443', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27251', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27252', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27511', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27550', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27552', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27582', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27583', 1, 0, @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------
--Bio-Chemistry and Nutrition
------------------------------------------------------------------------------
set @da = 'Biokemi og ernæring'
set @en = 'Bio-Chemistry and Nutrition'
set @pid = '{3fcb9ed7-d28d-46dc-bd68-7e5f613714e3}'
set @p_min = 1
set @p_max = 1

set @sid = @sid + 1
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)

set @svid = @svid + 1
set @version = 1
```

```
insert into specializationversion
  values (@svid, @sid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27301', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27302', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27311', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27321', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27333', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27443', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27732', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27500', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27733', 1, 0, @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------
--Bio-Informatics and System Biology
------------------------------------------------------------------------------
set @da = 'Bioinformatik og system biologi'
set @en = 'Bio-Informatics and System Biology'
set @pid = '{79ebfd9e-35a4-4bda-ba8e-0de1dae04509}'
set @p_min = 1
set @p_max = 1

set @sid = @sid + 1
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)

set @svid = @svid + 1
set @version = 1

insert into specializationversion
  values (@svid, @sid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27443', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27251', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27252', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27311', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27321', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27403', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27405', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27442', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
```

```
values(@scid, @svid, null, '27583', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '33232', 1, 0, @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------
--Food-Biotechnology
------------------------------------------------------------------------
set @da = 'Levnedsmiddelbioteknologi'
set @en = 'Food-Biotechnology'
set @pid = '{0115eadd-48fe-4b7c-990b-6dc8a14ae509}'
set @p_min = 1
set @p_max = 1

set @sid = @sid + 1
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)

set @svid = @svid + 1
set @version = 1

insert into specialization version
  values (@svid, @sid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27333', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27500', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27505', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27520', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27580', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27581', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27582', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27583', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27717', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27752', 1, 0, @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------
--Process-Biotechnology
------------------------------------------------------------------------
set @da = 'Procesbioteknologi'
set @en = 'Process-Biotechnology'
set @pid = '{ba9d5abc-74f8-4641-8862-f508045bf2cf}'
set @p_min = 1
set @p_max = 1

set @sid = @sid + 1
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)

set @svid = @svid + 1
set @version = 1

insert into specialization version
  values (@svid, @sid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27404', 1, 0, @curdate, @userid, @curdate, @userid)
```

```
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27406', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27443', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27442', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27505', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27733', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27967', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '28231', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '28354', 1, 0, @curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------
--Medical Microbiology
-------------------------------------------------------------------------------
set @da = 'Medicinsk mikrobiologi'
set @en = 'Medical Microbiology'
set @pid = '{70d57f78-2253-40c3-8d1e-2ec4c6428c71}'
set @p_min = 1
set @p_max = 1

set @sid = @sid + 1
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)

set @svid = @svid + 1
set @version = 1

insert into specializationversion
  values (@svid, @sid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

-- No courses specified!

-------------------------------------------------------------------------------
--Plan Constructions and Geotechnology
-------------------------------------------------------------------------------
set @da = 'Anlægskonstruktioner og Geoteknik'
set @en = 'Plan Constructions and Geotechnology'
set @pid = '{8996fc80-8191-4ac8-a38f-81fcdf989a7c}'
set @p_min = 1
set @p_max = 1

set @sid = @sid + 1
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)

set @svid = @svid + 1
set @version = 1

insert into specializationversion
  values (@svid, @sid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02541', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02543', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11251', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11412', 1, 0, @curdate, @userid, @curdate, @userid)
```

```
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11413', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11414', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11415', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11512', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11513', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11514', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11521', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11531', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12321', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '13310', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '13311', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '13320', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31200', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41110', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41121', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41122', 1, 0, @curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02532', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02534', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11301', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11411', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11501', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12400', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12401', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12420', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '13000', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '13003', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '13110', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '13210', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31250', 0, 1, @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------
--Building Physics and Installations
------------------------------------------------------------------------
set @da = 'Bygningsfysik og -installationer'
set @en = 'Building Physics and Installations'
set @pid = '{3369c515-da87-4e60-ab70-8e618ece8ba2}'
set @p_min = 1
set @p_max = 1

set @sid = @sid + 1
```

```
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)

set @svid = @svid + 1
set @version = 1

insert into specializationversion
  values (@svid, @sid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11005', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11021', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11102', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11103', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11104', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11105', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11106', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11302', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11310', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11505', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11506', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11546', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27552', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31200', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41110', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41112', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41722', 1, 0, @curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11306', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11511', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '32490', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '32810', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41320', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41323', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41403', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41721', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41815', 0, 1, @curdate, @userid, @curdate, @userid)
```

```
------------------------------------------------------------------------------
--Building Constructions and Materials
------------------------------------------------------------------------------
set @da = 'Bygningskonstruktioner og Materialer'
set @en = 'Building Constructions and Materials'
set @pid = '{53d7950f-d9df-48bf-acf9-b87e296ef419}'
set @p_min = 1
set @p_max = 1

set @sid = @sid + 1
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)

set @svid = @svid + 1
set @version = 1

insert into specializationversion
  values (@svid, @sid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11003', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11021', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11022', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11102', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11251', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11252', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11254', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11502', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11503', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11506', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11512', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11513', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11521', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11522', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11531', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11542', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11544', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11601', 1, 0, @curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02401', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '01142', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '28852', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '32810', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
```

```
insert into specialization_course
values(@scid, @svid, null, '42412', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42415', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42983', 0, 1, @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------
--Town Planning
------------------------------------------------------------------------
set @da = 'Byplanlægning'
set @en = 'Town Planning'
set @pid = '{baff7c47-2ea1-4960-b857-c6ababf55516}'
set @p_min = 1
set @p_max = 1

set @sid = @sid + 1
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)

set @svid = @svid + 1
set @version = 1

insert into specializationversion
  values (@svid, @sid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11005', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11253', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11302', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11303', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11304', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11306', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11310', 1, 0, @curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02401', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02701', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02725', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10305', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11251', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11252', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '13000', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '13003', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '13110', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '13120', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '13210', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42412', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
```

```
values(@scid, @svid, null, '42541', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42642', 0, 1, @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------
--Sound and Acoustic Technology
------------------------------------------------------------------------
set @da = 'Lyd og akustisk teknologi'
set @en = 'Sound and Acoustic Technology'
set @pid = '{f745f15b-ec71-4374-b89f-1f3e40578523}'
set @p_min = 1
set @p_max = 1

set @sid = @sid + 1
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)

set @svid = @svid + 1
set @version = 1

insert into specializationversion
  values (@svid, @sid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31200', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31220', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31230', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31240', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31250', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31260', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31270', 1, 0, @curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '01032', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '01246', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '01257', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02411', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02623', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11003', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11201', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11551', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31610', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31620', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31650', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31652', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41311', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41512', 0, 1, @curdate, @userid, @curdate, @userid)
```

```
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41621', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41722', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02593', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31029', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31030', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31671', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31657', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31658', 0, 1, @curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------
--Physical Electronics
-------------------------------------------------------------------------
set @da = 'Fysisk elektronik'
set @en = 'Physical Electronics'
set @pid = '{8ec5f879-349f-4fd0-b110-892d2c878efc}'
set @p_min = 1
set @p_max = 1

set @sid = @sid + 1
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)

set @svid = @svid + 1
set @version = 1

insert into specializationversion
  values (@svid, @sid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31415', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31620', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31625', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31630', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31635', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31640', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31641', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31820', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34349', 1, 0, @curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31000', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31405', 0, 1, @curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------
--Medico Technology
-------------------------------------------------------------------------
set @da = 'Medikoteknik'
set @en = 'Medico Technology'
set @pid = '{a9380f7a-e4af-4756-88d8-246e7dde9ab9}'
set @p_min = 1
set @p_max = 1
```

```
set @sid = @sid + 1
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)

set @svid = @svid + 1
set @version = 1

insert into specializationversion
  values (@svid, @sid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31650', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31651', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31655', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31610', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31656', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31661', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31662', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31230', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31260', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31620', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31630', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31820', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31825', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31830', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31840', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31860', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31800', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02453', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02455', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02457', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02503', 1, 0, @curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41621', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31657', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31652', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31030', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
```

```
values(@scid, @svid, null, '31671', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31200', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31235', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31300', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '01141', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '01246', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '01248', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '01257', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '01449', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02401', 0, 1, @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------
--Robot Technology and Intelligent Automation Systems
------------------------------------------------------------------------------
set @da = 'Robotteknologi og intelligente automationssystemer'
set @en = 'Robot Technology and Intelligent Automation Systems'
set @pid = '{1b084b69-5f83-4f15-93dc-c78d8a350ced}'
set @p_min = 1
set @p_max = 1

set @sid = @sid + 1
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)

set @svid = @svid + 1
set @version = 1

insert into specializationversion
  values (@svid, @sid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31305', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31310', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31320', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31340', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31360', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31365', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31370', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31380', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31385', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31390', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31825', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31860', 1, 0, @curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02220', 0, 1, @curdate, @userid, @curdate, @userid)
```

```
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02224', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02284', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02286', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02330', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02333', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02393', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02421', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02451', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02453', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02501', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31050', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31610', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31820', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31840', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41624', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41625', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41637', 0, 1, @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------
-- Wireless Communication
------------------------------------------------------------------------------
set @da = 'Trådløs kommunikation'
set @en = 'Wireless Communication'
set @pid = '{4594da25-7d9c-46dc-a968-d34062b342a7}'
set @p_min = 1
set @p_max = 1

set @sid = @sid + 1
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)

set @svid = @svid + 1
set @version = 1

insert into specializationversion
  values (@svid, @sid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31415', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31420', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31425', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31430', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31435', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
```

```
values(@scid, @svid, null, '31440', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31230', 1, 0, @curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31400', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31405', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31410', 0, 1, @curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------
--Manufacturing Processes
-------------------------------------------------------------------------
set @da = 'Fremstillingsprocesser'
set @en = 'Manufacturing Processes'
set @pid = '{ebbf84b6-77cf-44be-816c-a0647c9a46c8}'
set @p_min = 1
set @p_max = 1

set @sid = @sid + 1
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)

set @svid = @svid + 1
set @version = 1

insert into specializationversion
  values (@svid, @sid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42130', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42155', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42213', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42221', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42222', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42224', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42230', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42250', 1, 0, @curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41502', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41816', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42110', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42120', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42201', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42226', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42232', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42234', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42341', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
```

```
values(@scid, @svid, null, '42301', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42302', 0, 1, @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------
--Product and Production Modelling
------------------------------------------------------------------------
set @da = 'Produkt- og produktionsmodellering'
set @en = 'Product and Production Modelling'
set @pid = '{b8051033-7833-440b-aac9-1a1d89bf49eb}'
set @p_min = 1
set @p_max = 1

set @sid = @sid + 1
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)

set @svid = @svid + 1
set @version = 1

insert into specializationversion
  values (@svid, @sid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41812', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42213', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42221', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42224', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42250', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42371', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42450', 1, 0, @curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02623', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02661', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41320', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41502', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41816', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42110', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42301', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42302', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42110', 0, 1, @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------
--Production and Company Management
------------------------------------------------------------------------
set @da = 'Produktions- og virksomhedsledelse'
set @en = 'Production and Company Management'
set @pid = '{5c91aa78-c983-41fe-be72-03546ecc592c}'
set @p_min = 1
set @p_max = 1

set @sid = @sid + 1
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)

set @svid = @svid + 1
```

```
set @version = 1

insert into specializationversion
  values (@svid, @sid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42342', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42371', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42430', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42435', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42450', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42455', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42460', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42465', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42470', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42532', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02725', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42540', 1, 0, @curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42301', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42341', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42405', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42412', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42415', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42410', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42422', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42425', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42440', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42521', 0, 1, @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------
--Innovation
------------------------------------------------------------------------
set @da = 'Innovation'
set @en = 'Innovation'
set @pid = '{db74e1b5-380f-4cfd-a483-bade977ea0e9}'
set @p_min = 1
set @p_max = 1

set @sid = @sid + 1
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)

set @svid = @svid + 1
set @version = 1
```

```
insert into specializationversion
  values (@svid, @sid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41612', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41629', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42374', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42430', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42435', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42465', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42532', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42541', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42640', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '88312', 1, 0, @curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02259', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41816', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42301', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42302', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42342', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42373', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42440', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42521', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42531', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42642', 0, 1, @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------
--Viable Production
------------------------------------------------------------------------------
set @da = 'Bæredygtig produktion'
set @en = 'Viable Production'
set @pid = '{ec6e6265-020e-4535-9f1c-445c32569480}'
set @p_min = 1
set @p_max = 1

set @sid = @sid + 1
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)

set @svid = @svid + 1
set @version = 1

insert into specializationversion
  values (@svid, @sid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
```

```
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42342', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42455', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42470', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42540', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42531', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42532', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42541', 1, 0, @curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42372', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42405', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42415', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42420', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42521', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42630', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42642', 0, 1, @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------
--Secure and Reliable IT--Systems
------------------------------------------------------------------------
set @da = 'Sikre og pålidelige IT-systemer'
set @en = 'Secure and Reliable IT-Systems'
set @pid = '{2c5cbf09-7217-48f0-bee3-46e07f4548e5}'
set @p_min = 1
set @p_max = 1

set @sid = @sid + 1
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)

set @svid = @svid + 1
set @version = 1

insert into specializationversion
  values (@svid, @sid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02220', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02222', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02224', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02226', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02230', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
```

```
values(@scid, @svid, null, '02240', 1, 0, @curdate, @userid, @curdate, @userid)
--xxxxx (missing course number)
--yyyyy (missing course number)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '01425', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34320', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34240', 0, 1, @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------
--System-on-Chip
------------------------------------------------------------------------
set @da = 'System-on-Chip'
set @en = 'System-on-Chip'
set @pid = '{71fda069-f2f2-42be-ba1e-8df52da7e891}'
set @p_min = 1
set @p_max = 1

set @sid = @sid + 1
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)

set @svid = @svid + 1
set @version = 1

insert into specializationversion
  values (@svid, @sid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02200', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02202', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02204', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02206', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02208', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02220', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02222', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02224', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02260', 1, 0, @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------
--Requirements and Knowledge Engineering
------------------------------------------------------------------------
set @da = 'Krav og Viden'
set @en = 'Requirements and Knowledge Engineering'
set @pid = '{654b9938-57ac-4329-ad39-d88e76589d5e}'
set @p_min = 1
set @p_max = 1

set @sid = @sid + 1
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)

set @svid = @svid + 1
set @version = 1

insert into specializationversion
  values (@svid, @sid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
```

```
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02260', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02261', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02262', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02264', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02266', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02268', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02280', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02282', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02284', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02286', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02288', 1, 0, @curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------
--Chemical and Biochemical Process Technology
-------------------------------------------------------------------------
set @da = 'Kemisk og biokemisk procesteknik'
set @en = 'Chemical and Biochemical Process Technology'
set @pid = '{c50f1ad7-3262-4d84-af53-da36161074f4}'
set @p_min = 1
set @p_max = 1

set @sid = @sid + 1
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)

set @svid = @svid + 1
set @version = 1

insert into specializationversion
  values (@svid, @sid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27404', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27406', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '28230', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '28240', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '28250', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '28320', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '28350', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '26128', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '28231', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '28244', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
```

```
values(@scid, @svid, null, '28416', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '28434', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '28443', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '28851', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '28852', 1, 0, @curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12131', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41814', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27405', 0, 1, @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------
--Kemisk produktudvikling
------------------------------------------------------------------------------
set @da = 'Kemisk produktudvikling'
set @en = 'Chemical Product Development'
set @pid = '{4a486ad1-5454-4d42-ba41-7e3a98aaec50}'
set @p_min = 1
set @p_max = 1

set @sid = @sid + 1
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)

set @svid = @svid + 1
set @version = 1

insert into specializationversion
  values (@svid, @sid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '26124', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '26128', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '26140', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '26320', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '26430', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27403', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '28212', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '28213', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '28310', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '28315', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '26142', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '26322', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '26334', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '26510', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '28316', 1, 0, @curdate, @userid, @curdate, @userid)
```

```
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '28414', 1, 0, @curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10312', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10472', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '26260', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27505', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '33442', 0, 1, @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------
--Applied Chemistry
------------------------------------------------------------------------------
set @da = 'Anvendt kemi'
set @en = 'Applied Chemistry'
set @pid = '{b1c478e8-0513-4ca7-b157-56d8d3da7c47}'
set @p_min = 1
set @p_max = 1

set @sid = @sid + 1
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)

set @svid = @svid + 1
set @version = 1

insert into specializationversion
  values (@svid, @sid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '26124', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '26126', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '26128', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '26230', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '26235', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '26240', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '26250', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '26310', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '26312', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '26320', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '26322', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '26330', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '26334', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '26430', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27511', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '28241', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
```

```
insert into specialization_course
values(@scid, @svid, null, '28423', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '33232', 1, 0, @curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02411', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10100', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10312', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10322', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10340', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12231', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '26260', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27583', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27733', 0, 1, @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------
--Chemistry of Nano-materials: Catalysis
------------------------------------------------------------------------------
set @da = 'Nanomaterialers kemi: Katalyse'
set @en = 'Chemistry of Nano-materials: Catalysis'
set @pid = '{2402a8c2-0f93-4889-8bb9-1cb03bea1c42}'
set @p_min = 1
set @p_max = 1

set @sid = @sid + 1
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)

set @svid = @svid + 1
set @version = 1

insert into specializationversion
  values (@svid, @sid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10111', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10112', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10120', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10300', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10302', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10304', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10308', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10322', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '26140', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '26235', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '26320', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
```

```
values(@scid, @svid, null, '26330', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '26510', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '28250', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '28240', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '28241', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '28443', 1, 0, @curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '28845', 0, 1, @curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------
--Construction and Product Development
-------------------------------------------------------------------------
set @da = 'Konstruktion og produktudvikling'
set @en = 'Construction and Product Development'
set @pid = '{113f61fd-fed5-4ed7-9bbb-f4782e3c649b}'
set @p_min = 1
set @p_max = 1

set @sid = @sid + 1
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)

set @svid = @svid + 1
set @version = 1

insert into specializationversion
  values (@svid, @sid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41612', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41627', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41628', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41629', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41623', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42374', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42450', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42460', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42640', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41624', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41625', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41626', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41511', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41811', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41614', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41622', 1, 0, @curdate, @userid, @curdate, @userid)
```

```
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41611', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41502', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41512', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41312', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31300', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41503', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42130', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42211', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42232', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42230', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42220', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41816', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41515', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41812', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42435', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42405', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42415', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42532', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42410', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42373', 0, 1, @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------
--Applied Mechanics and Dynamics
------------------------------------------------------------------------
set @da = 'Anvendt mekanik og dynamik'
set @en = 'Applied Mechanics and Dynamics'
set @pid = '{1b285942-d3ee-484f-8f26-e28134174d16}'
set @p_min = 1
set @p_max = 1

set @sid = @sid + 1
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)

set @svid = @svid + 1
set @version = 1

insert into specializationversion
  values (@svid, @sid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41511', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41811', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
```

```
insert into specialization_course
values(@scid, @svid, null, '41521', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41522', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41523', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41525', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41222', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41614', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41313', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41320', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41323', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41822', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41612', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41623', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41625', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41626', 1, 0, @curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41501', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41502', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41503', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41512', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41812', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41515', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41816', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41312', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41602', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41611', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31300', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42130', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42120', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '01030', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '01141', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '01243', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '01246', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '01248', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '01250', 0, 1, @curdate, @userid, @curdate, @userid)
```

```
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02417', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02451', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02701', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02601', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02623', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02685', 0, 1, @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------
--Industrial Flow Mechanics and Windpower
------------------------------------------------------------------------------
set @da = 'Industriel strømningsmekanik og vindkraft'
set @en = 'Industrial Flow Mechanics and Windpower'
set @pid = '{720b5cd5-93d7-497b-b771-eeb466b7fd66}'
set @p_min = 1
set @p_max = 1

set @sid = @sid + 1
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)

set @svid = @svid + 1
set @version = 1

insert into specializationversion
  values (@svid, @sid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41320', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41323', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41822', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41313', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41322', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41120', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41811', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41614', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41622', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41624', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41221', 1, 0, @curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41312', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41401', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41416', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41814', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41501', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
```

```
insert into specialization_course
values(@scid, @svid, null, '41502', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41611', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31300', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41201', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41861', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41515', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '01030', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02601', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '01246', 0, 1, @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------
--Maritime Technology
------------------------------------------------------------------------------
set @da = 'Maritim teknik'
set @en = 'Maritime Technology'
set @pid = '{afe40078-ab33-4033-bc20-8a0f30c3df20}'
set @p_min = 1
set @p_max = 1

set @sid = @sid + 1
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)

set @svid = @svid + 1
set @version = 1

insert into specializationversion
  values (@svid, @sid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41211', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41221', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41222', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41223', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41271', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41272', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41612', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41627', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41629', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41511', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41521', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41522', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41525', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41320', 1, 0, @curdate, @userid, @curdate, @userid)
```

```
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41323', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41811', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41822', 1, 0, @curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '01141', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02685', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12411', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '28415', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41121', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41124', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41201', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41212', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41260', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41312', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41401', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41414', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41501', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41502', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41512', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41515', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41812', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41813', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41816', 0, 1, @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------
--Environmental Technology
------------------------------------------------------------------------------
set @da = 'Miljøteknologi'
set @en = 'Environmental Technology'
set @pid = '{473c16a1-381c-4d2b-bfdc-6f77ae8d0ba6}'
set @p_min = 1
set @p_max = 1

set @sid = @sid + 1
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)

set @svid = @svid + 1
set @version = 1

insert into specializationversion
  values (@svid, @sid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
```

```
insert into specialization_course
values(@scid, @svid, null, '12225', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12230', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12130', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12120', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12131', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12320', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12330', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12121', 1, 0, @curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11302', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12231', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12233', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12140', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12240', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12242', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12243', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12321', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12322', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12323', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12332', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12331', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12333', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12340', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12341', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12420', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27262', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '28244', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '28375', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '28376', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '28852', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41120', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41124', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41413', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42521', 1, 1, @curdate, @userid, @curdate, @userid)
```

```
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '43372', 1, 1, @curdate, @userid, @curdate, @userid)set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '01246', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02100', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02401', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02407', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02411', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02413', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02417', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02541', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02593', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02601', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02643', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02685', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10474', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '26470', 0, 1, @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------
-- Water Resources
------------------------------------------------------------------------------
set @da = 'Vandressourcer'
set @en = 'Water Resources'
set @pid = '{313e40d7-9c78-4ef5-ac25-fc5ac0d28439}'
set @p_min = 1
set @p_max = 1

set @sid = @sid + 1
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)

set @svid = @svid + 1
set @version = 1

insert into specializationversion
  values (@svid, @sid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12320', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12321', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12322', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12323', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12332', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12341', 1, 0, @curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02541', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
```

```
values(@scid, @svid, null, '11423', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12120', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12121', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12130', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12131', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12140', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12210', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12252', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12230', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12231', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12233', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12240', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12242', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12243', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12330', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12331', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12333', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12420', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12421', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12422', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41122', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41124', 1, 1, @curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '01246', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '01257', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02100', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02401', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02407', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02411', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02413', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02417', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02532', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02601', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02643', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
```

```
insert into specialization_course
values(@scid, @svid, null, '02645', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02661', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02685', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10474', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42412', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41415', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42630', 0, 1, @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------
--Oil, Gas and Other Raw Materials
------------------------------------------------------------------------
set @da = 'Olie, gas og andre råstoffer'
set @en = 'Oil, Gas and Other Raw Materials'
set @pid = '{971b6e2c-cc91-49d3-ab20-ff0b7e79ce1b}'
set @p_min = 1
set @p_max = 1

set @sid = @sid + 1
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)

set @svid = @svid + 1
set @version = 1

insert into specializationversion
  values (@svid, @sid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12411', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12420', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12422', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12440', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12444', 1, 0, @curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11423', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12320', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12340', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12421', 1, 1, @curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10312', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11415', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11501', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11502', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '28415', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '28515', 0, 1, @curdate, @userid, @curdate, @userid)
```

```
------------------------------------------------------------------------------
--Industrial Environmental Work
------------------------------------------------------------------------------
set @da = 'Industrielt miljøarbejde'
set @en = 'Industrial Environmental Work'
set @pid = '{0beb9f8f-4789-42cd-864c-c75a8367fe13}'
set @p_min = 1
set @p_max = 1

set @sid = @sid + 1
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)

set @svid = @svid + 1
set @version = 1

insert into specializationversion
  values (@svid, @sid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42342', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42372', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42540', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42640', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12230', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11310', 1, 0, @curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42470', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12130', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12131', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '28375', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '28376', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42521', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31250', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27262', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42531', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41722', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41414', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41413', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '28244', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '28852', 1, 1, @curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41634', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '41627', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
```

```
insert into specialization_course
values(@scid, @svid, null, '31200', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '28320', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '28434', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42412', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42415', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02401', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02413', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '26470', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '28021', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '28120', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '28121', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '28153', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27000', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27403', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '27711', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42630', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10474', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12133', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '28875', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42420', 0, 1, @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------
--Environmental Work in Developing Countries
------------------------------------------------------------------------
set @da = 'Miljøarbejde i udviklingslande'
set @en = 'Environmental Work in Developing Countries'
set @pid = '{d302a4aa-6ba2-4990-8e01-2256fbb7dbd0}'
set @p_min = 1
set @p_max = 1

set @sid = @sid + 1
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)

set @svid = @svid + 1
set @version = 1

insert into specializationversion
  values (@svid, @sid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12210', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12242', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42644', 1, 0, @curdate, @userid, @curdate, @userid)
```

```
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42645', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12243', 1, 0, @curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11302', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11303', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '11310', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12121', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12130', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12131', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12120', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12225', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12231', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12233', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12240', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12320', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12333', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '12341', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42342', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42372', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42470', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42521', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42531', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42540', 1, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42640', 1, 1, @curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02725', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42541', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42642', 0, 1, @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------
--Nanoscale Material Physics
------------------------------------------------------------------------
set @da = 'Nanoskala materialefysik'
set @en = 'Nanoscale Material Physics'
set @pid = '{e3fe3e6e-3d73-4503-8e58-05d9c8968e02}'
set @p_min = 1
set @p_max = 1

set @sid = @sid + 1
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)

set @svid = @svid + 1
set @version = 1

insert into specializationversion
  values (@svid, @sid, @version,
```

```
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10111', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10120', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10112', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10300', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10302', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10304', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10306', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10308', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10310', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10322', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10346', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10467', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10469', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '33205', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '33250', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '33251', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '33254', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '33321', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '33355', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '33441', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '33442', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '33471', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02613', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '26510', 1, 0, @curdate, @userid, @curdate, @userid)--
  ----------------------------------------------------------------------
--Biophysics and Complex Systems
------------------------------------------------------------------------
set @da = 'Biofysik og komplekse systemer'
set @en = 'Biophysics and Complex Systems'
set @pid = '{fe056555-d220-46c6-8d05-2de497369c62}'
set @p_min = 1
set @p_max = 1

set @sid = @sid + 1
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)

set @svid = @svid + 1
set @version = 1
```

```
insert into specializationversion
  values (@svid, @sid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10111', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10120', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10112', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10302', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10343', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10344', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10345', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10346', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10347', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10350', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10381', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10467', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10469', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10477', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '33232', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '33430', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '01449', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02655', 1, 0, @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------
--Photonics
------------------------------------------------------------------------
set @da = 'Fotonik'
set @en = 'Photonics'
set @pid = '{33fb4337-409b-448b-9284-242c0779a9cc}'
set @p_min = 1
set @p_max = 1

set @sid = @sid + 1
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)

set @svid = @svid + 1
set @version = 1

insert into specializationversion
  values (@svid, @sid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)
```

```
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10111', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10120', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10112', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10300', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10370', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10372', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10373', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10374', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10376', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10378', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10381', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10392', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10467', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10469', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34030', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34040', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34049', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34051', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34055', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34056', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34059', 1, 0, @curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------
--Heterogeneous Networks and Service Development
-------------------------------------------------------------------------
set @da = 'Heterogene net og tjenesteudvikling'
set @en = 'Heterogeneous Networks and Service Development'
set @pid = '{33106a63-1e52-40e8-beb8-cc862b76cced}'
set @p_min = 1
set @p_max = 1

set @sid = @sid + 1
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)

set @svid = @svid + 1
set @version = 1

insert into specializationversion
  values (@svid, @sid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02501', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
```

```
insert into specialization_course
values(@scid, @svid, null, '11201', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34230', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34240', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34241', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34320', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34321', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34341', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34351', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34531', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34631', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34641', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34643', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34841', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34842', 1, 0, @curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42412', 0, 1, @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------
--Interactive Media
------------------------------------------------------------------------
set @da = 'Interaktive medier'
set @en = 'Interactive Media'
set @pid = '{5bf79356-042c-4dc8-ace2-b4e1afced2bc}'
set @p_min = 1
set @p_max = 1

set @sid = @sid + 1
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)

set @svid = @svid + 1
set @version = 1

insert into specializationversion
  values (@svid, @sid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02260', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02261', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02264', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02266', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02725', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34320', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34531', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
```

```
values(@scid, @svid, null, '34631', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34641', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34642', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34643', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34742', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34743', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34841', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34842', 1, 0, @curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42430', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '42541', 0, 1, @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------
--Optical Components
------------------------------------------------------------------------------
set @da = 'Optiske komponenter'
set @en = 'Optical Components'
set @pid = '{5c95e256-27d2-4059-b04c-199cbcc0be17}'
set @p_min = 1
set @p_max = 1

set @sid = @sid + 1
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)

set @svid = @svid + 1
set @version = 1

insert into specializationversion
  values (@svid, @sid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10374', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10376', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34030', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34040', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34049', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34050', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34055', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34140', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34150', 1, 0, @curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '01030', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10100', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10110', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
```

```
values(@scid, @svid, null, '10300', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10370', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '10392', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '33253', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '33470', 0, 1, @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------
--Signals and Tranmission Technology
------------------------------------------------------------------------
set @da = 'Signaler og tranmissionsteknik'
set @en = 'Signals and Tranmission Technology'
set @pid = '{538a828a-c041-445f-bf1d-9cbc632e1c08}'
set @p_min = 1
set @p_max = 1

set @sid = @sid + 1
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)

set @svid = @svid + 1
set @version = 1

insert into specializationversion
  values (@svid, @sid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '01259', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '01425', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02501', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02455', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31405', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31420', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31430', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34140', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34150', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34151', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34230', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34240', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34241', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34259', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34320', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34350', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34351', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34631', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
```

```
insert into specialization_course
values(@scid, @svid, null, '02202', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02220', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02222', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02264', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02266', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02445', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31415', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31641', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34030', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34040', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34349', 1, 0, @curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '01016', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '01142', 0, 1, @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------
-- Telecommunication Networks
------------------------------------------------------------------------
set @da = 'Telenet'
set @en = 'Telecommunication Networks'
set @pid = '{90ea4cb2-51a8-474e-9e4c-dd4c5a4f5c9d}'
set @p_min = 1
set @p_max = 1

set @sid = @sid + 1
insert into specialization
values (@sid, @curdate, @userid, @curdate, @userid)

set @svid = @svid + 1
set @version = 1

insert into specializationversion
  values (@svid, @sid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34140', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34150', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34151', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34230', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34240', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34320', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34321', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34340', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34341', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
```

```
values(@scid, @svid, null, '34350', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34351', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34531', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34631', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34643', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34841', 1, 0, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34842', 1, 0, @curdate, @userid, @curdate, @userid)

set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02220', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '02222', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '01142', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '31405', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34352', 0, 1, @curdate, @userid, @curdate, @userid)
set @scid = @scid + 1
insert into specialization_course
values(@scid, @svid, null, '34390', 0, 1, @curdate, @userid, @curdate, @userid)
```

3.18 StudyType

```
-----------------------------------------------------------------------
--Common declarations
-----------------------------------------------------------------------
declare @userid uniqueidenti↓er
declare @curdate datetime
declare @studytype int, @version int, @svid int
declare @da varchar(255), @en varchar(255)
declare @pid uniqueidenti↓er
set @userid = 'D36DD218-AE59-4046-977C-659E4EB92C92'
set @curdate = getdate()
set @version = 1
set @svid = 0
-----------------------------------------------------------------------
set @studytype = 1
insert into studytype values(@studytype, @curdate, @userid, @curdate, @userid)

set @da = 'Civil'
set @en = 'Master of Science'
set @pid = 'c345c73c-1043-4167-b4a4-f7ebb3be2f23'

set @svid = @svid + 1
insert into studytypeversion
values(@svid, @studytype, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>', @pid,
@curdate, @userid, @curdate, @userid)
-----------------------------------------------------------------------
set @studytype = 2
insert into studytype values(@studytype, @curdate, @userid, @curdate, @userid)

set @da = 'Diplom'
set @en = 'Bachelor of Science'
set @pid = 'f57cec33-2b5c-4c0c-a6c0-288abd01f198'

set @svid = @svid + 1
insert into studytypeversion
values(@svid, @studytype, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
```

```
    </culture>
  </cultures>', @pid,
@curdate, @userid, @curdate, @userid)
-------------------------------------------------------------------------------
set @studytype = 3
insert into studytype values(@studytype, @curdate, @userid, @curdate, @userid)

set @da = 'Ph.D.'
set @en = 'Ph.D.'
set @pid = '502161fe-7dfe-4df8-9e2c-e2d925458e54'

set @svid = @svid + 1
insert into studytypeversion
values(@svid, @studytype, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>', @pid,
@curdate, @userid, @curdate, @userid)
-------------------------------------------------------------------------------
set @studytype = 4
insert into studytype values(@studytype, @curdate, @userid, @curdate, @userid)

set @da = 'Åben uddannelse'
set @en = 'Open University'
set @pid = 'a03eee09-810d-450b-b530-08c6853700ef'

set @svid = @svid + 1
insert into studytypeversion
values(@svid, @studytype, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>', @pid,
@curdate, @userid, @curdate, @userid)
-------------------------------------------------------------------------------
set @studytype = 5
insert into studytype values(@studytype, @curdate, @userid, @curdate, @userid)

set @da = 'Levnedsmiddel'
set @en = 'Food Science'
set @pid = '55dd6271-e386-4a88-8ffe-c2d4729b9fe6'

set @svid = @svid + 1
insert into studytypeversion
values(@svid, @studytype, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>', @pid,
@curdate, @userid, @curdate, @userid)
```

3.18.1 StudyType_ProjectType

```
declare @userid uniqueidenti⌊er
set @userid = '{BC61A572-A19A-41FB-97B8-9B0B463F2DBB}'

declare @curdate datetime
set @curdate = getdate()

-------------------------------------------------------------------------
-- Master of Science
-------------------------------------------------------------------------

-- Notice, a technical package project is considered as part
-- of a technical package and is therefore not related
-- directly to a study type.

-- Midway Project
insert into StudyType_ProjectType
values (1, 1, 1, 1, '{037B9864-77CD-4299-B5E2-6D5A70BFB02F}',
'{c6418073-365e-4f6b-a410-543db11df74a}',
@curdate, @userid, @curdate, @userid)

-- Thesis
insert into StudyType_ProjectType
values (2, 1, 2, 2, '{19032106-A382-4b6e-BB4E-D8BE7AC6E572}',
```

```
'{a7a7eda2-0251-49ec-8fb6-27cf40462e27}',
@curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------
-- Bachelor of Science
-------------------------------------------------------------------

-- Notice, a trainee project is considered as part
-- of a technical package and is therefore not related
-- directly to a study type.

-- Thesis
insert into StudyType_ProjectType
values (11, 2, 2, 1, '{B2C990CA-F827-4945-B683-5F711EFBC2DE}',
'{46dff3f0-ddad-473d-ae93-a39b3bd05b78}',
@curdate, @userid, @curdate, @userid)
```

3.18.2 StudyType_TechnicalLine

```
declare @userid uniqueidenti↓er
declare @curdate datetime

set @userid = '{BC61A572-A19A-41FB-97B8-9B0B463F2DBB}'
set @curdate = getdate()

-------------------------------------------------------------------
-- Master of Science, Applied Mathematics (Anvendt matematik)
-------------------------------------------------------------------
insert into StudyType_TechnicalLine
values (
1,
1,
1,
@curdate, @userid, @curdate, @userid)

-------------------------------------------------------
-- Master of Science, Biotechnology (Bioteknologi)
-------------------------------------------------------
insert into StudyType_TechnicalLine
values (
2,
1,
2,
@curdate, @userid, @curdate, @userid)

-------------------------------------------------------
-- Master of Science, Civil engineering (Bygning)
-------------------------------------------------------
insert into StudyType_TechnicalLine
values (
3,
1,
3,
@curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------------
-- Master of Science, Electrical and Electronic Engineering (Elektro)
-------------------------------------------------------------------------------------
insert into StudyType_TechnicalLine
values (
4,
1,
4,
@curdate, @userid, @curdate, @userid)

-------------------------------------------
-- Master of Science, Energy (Energi)
-------------------------------------------
insert into StudyType_TechnicalLine
values (
5,
1,
5,
@curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------------------

-- Master of Science, Manufacturing and Management (Industriel produktion)
-------------------------------------------------------------------------------------------

insert into StudyType_TechnicalLine
values (
6,
1,
6,
@curdate, @userid, @curdate, @userid)

-------------------------------------------------------
-- Master of Science, Informatics (Informatikretningen)
-------------------------------------------------------
insert into StudyType_TechnicalLine
values (
7,
1,
7,
```

```
@curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------------------------------

-- Master of Science, Applied Chemistry and Chemical Engineering (Kemi og kemiteknik)
-------------------------------------------------------------------------------------------------------

insert into StudyType_TechnicalLine
values (
8,
1,
8,
@curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------------------------------

-- Master of Science, Engineering Design and Applied Mechanics (Konstruktion og mekanik)
-------------------------------------------------------------------------------------------------------

insert into StudyType_TechnicalLine
values (
9,
1,
9,
@curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------
-- Master of Science, Material Technology (Materialeteknologi)
-------------------------------------------------------------------
insert into StudyType_TechnicalLine
values (
10,
1,
10,
@curdate, @userid, @curdate, @userid)

-------------------------------------------------------------
-- Master of Science, Environmental Engineering (Miljø)
-------------------------------------------------------------
insert into StudyType_TechnicalLine
values (
11,
1,
11,
@curdate, @userid, @curdate, @userid)

-----------------------------------------------------------------------------
-- Master of Science, Planning and Management (Planlægning og ledelse)
-----------------------------------------------------------------------------
insert into StudyType_TechnicalLine
values (
12,
1,
12,
@curdate, @userid, @curdate, @userid)

-------------------------------------------------------------
-- Master of Science, Engineering Physics (Teknisk fysik)
-------------------------------------------------------------
insert into StudyType_TechnicalLine
values (
13,
1,
13,
@curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------
-- Master of Science, Telecommunications (Telekommunikation)
-------------------------------------------------------------------
insert into StudyType_TechnicalLine
values (
14,
1,
14,
@curdate, @userid, @curdate, @userid)
```

3.18.3 StudyType_TechnicalPackage

```
declare @userid uniqueidenti↓er
declare @curdate datetime
declare @studytypeversionid int

set @userid = '{BC61A572-A19A-41FB-97B8-9B0B463F2DBB}'
set @curdate = getdate()

-------------------------------------------------------------------------------------------------------

-- MASTER OF SCIENCE MASTER OF SCIENCE MASTER OF SCIENCE
-- MASTER OF SCIENCE MASTER OF SCIENCE MASTER OF SCIENCE
-- MASTER OF SCIENCE MASTER OF SCIENCE MASTER OF SCIENCE
-------------------------------------------------------------------------------------------------------

set @studytypeversionid = 1
```

```
-------------------------------------------------------------------------------------------

-- The Biotechnology Technical Package (Bioteknologifagpakken)
-------------------------------------------------------------------------------------------

insert into StudyType_TechnicalPackage
values (
1,
@studytypeversionid,
1,
@curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------------------

-- The Construction Engineering Technical Package (Bygningsfagpakken)
-------------------------------------------------------------------------------------------

insert into StudyType_TechnicalPackage
values (
2,
@studytypeversionid,
2,
@curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------------------

-- The Design & Innovation Technical Package (Design & innovationsfagpakken)
-------------------------------------------------------------------------------------------

insert into StudyType_TechnicalPackage
values (
3,
@studytypeversionid,
3,
@curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------------------

-- The Electrical Engineering Technical Package (Elektrofagpakken)
-------------------------------------------------------------------------------------------

insert into StudyType_TechnicalPackage
values (
4,
@studytypeversionid,
4,
@curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------
-- The Energy Technical Package (Energifagpakken)
-------------------------------------------------------------------------
insert into StudyType_TechnicalPackage
values (
5,
@studytypeversionid,
5,
@curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------------------

-- The Information Technology Technical Package (Informatikfagpakken)
-------------------------------------------------------------------------------------------

insert into StudyType_TechnicalPackage
values (
6,
@studytypeversionid,
6,
@curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------------------

-- The Chemical Engineering Technical Package (Kemifagpakken)
-------------------------------------------------------------------------------------------

insert into StudyType_TechnicalPackage
values (
7,
@studytypeversionid,
7,
@curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------------------

-- The Mechanical Engineering Technical Package (Maskinfagpakken)
-------------------------------------------------------------------------------------------

insert into StudyType_TechnicalPackage
values (
8,
@studytypeversionid,
8,
@curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------------------

-- The Enviromental Engineering Technical Package (Miljøfagpakken)
-------------------------------------------------------------------------------------------
```

```
	insert into StudyType_TechnicalPackage
	values (
	9,
	@studytypeversionid,
	9,
	@curdate, @userid, @curdate, @userid)

	-- The Technical Physics Technical Package (Teknisk fysikfagpakken)

	insert into StudyType_TechnicalPackage
	values (
	10,
	@studytypeversionid,
	10,
	@curdate, @userid, @curdate, @userid)

	-- The Medication and Technology Technical Package (Medicin og teknologifagpakken)

	insert into StudyType_TechnicalPackage
	values (
	11,
	@studytypeversionid,
	11,
	@curdate, @userid, @curdate, @userid)

-- BACHELOR OF SCIENCE BACHELOR OF SCIENCE BACHELOR OF SCIENCE
-- BACHELOR OF SCIENCE BACHELOR OF SCIENCE BACHELOR OF SCIENCE
-- BACHELOR OF SCIENCE BACHELOR OF SCIENCE BACHELOR OF SCIENCE

set @studytypeversionid = 2

	-- Construction Engineering February (Bygning februar)
	insert into StudyType_TechnicalPackage
	values (
	12,
	@studytypeversionid,
	12,
	@curdate, @userid, @curdate, @userid)

	-- Construction Engineering September (Bygning september)
	insert into StudyType_TechnicalPackage
	values (
	13,
	@studytypeversionid,
	13,
	@curdate, @userid, @curdate, @userid)

	-- Urban Planning and Construction (By og bygning)
	insert into StudyType_TechnicalPackage
	values (
	14,
	@studytypeversionid,
	14,
	@curdate, @userid, @curdate, @userid)

	-- Electrical Engineering February (Elektro februar)
	insert into StudyType_TechnicalPackage
	values (
	15,
	@studytypeversionid,
	15,
	@curdate, @userid, @curdate, @userid)

	-- Electrical Engineering September (Elektro september)
	insert into StudyType_TechnicalPackage
	values (
	16,
	@studytypeversionid,
	16,
	@curdate, @userid, @curdate, @userid)

	-- Chemical Engineering February (Kemi februar)
	insert into StudyType_TechnicalPackage
	values (
	17,
	@studytypeversionid,
	17,
```

```
@curdate, @userid, @curdate, @userid)

-------------------------------------------------------------
-- Chemical Engineering September (Kemi september)
-------------------------------------------------------------
insert into StudyType_TechnicalPackage
values (
18,
@studytypeversionid,
18,
@curdate, @userid, @curdate, @userid)

-------------------------------------------------------------
-- IT February (IT februar)
-------------------------------------------------------------
insert into StudyType_TechnicalPackage
values (
19,
@studytypeversionid,
19,
@curdate, @userid, @curdate, @userid)

-------------------------------------------------------------
-- IT September (IT september)
-------------------------------------------------------------
insert into StudyType_TechnicalPackage
values (
20,
@studytypeversionid,
20,
@curdate, @userid, @curdate, @userid)

-------------------------------------------------------------
-- Mechanical Engineering [construction] February
-- Maskin [konstruktion] februar
-------------------------------------------------------------
insert into StudyType_TechnicalPackage
values (
21,
@studytypeversionid,
21,
@curdate, @userid, @curdate, @userid)

-------------------------------------------------------------
-- Mechanical Engineering [construction] September
-- Maskin [konstruktion] september
-------------------------------------------------------------
insert into StudyType_TechnicalPackage
values (
22,
@studytypeversionid,
22,
@curdate, @userid, @curdate, @userid)

-------------------------------------------------------------
-- Mechanical Engineering [construction - ship] February
-- Maskin [konstruktion - skib] februar
-------------------------------------------------------------
insert into StudyType_TechnicalPackage
values (
23,
@studytypeversionid,
23,
@curdate, @userid, @curdate, @userid)

-------------------------------------------------------------
-- Mechanical Engineering [construction - ship] September
-- Maskin [konstruktion - skib] - september
-------------------------------------------------------------
insert into StudyType_TechnicalPackage
values (
24,
@studytypeversionid,
24,
@curdate, @userid, @curdate, @userid)

-------------------------------------------------------------
-- Mechanical Engineering [construction - energy] February
-- Maskin [konstruktion - energi] februar
-------------------------------------------------------------
insert into StudyType_TechnicalPackage
values (
25,
@studytypeversionid,
25,
@curdate, @userid, @curdate, @userid)

-------------------------------------------------------------
-- Mechanical Engineering [construction - energy] September
-- Maskin [konstruktion - energi] september
-------------------------------------------------------------
insert into StudyType_TechnicalPackage
values (
26,
@studytypeversionid,
26,
@curdate, @userid, @curdate, @userid)

-------------------------------------------------------------
```

```
-- Mechanical Engineering [maritime - construction] February
-- Maskin [maritim - konstruktion] februar
-----------------------------------------------------------------------
insert into StudyType_TechnicalPackage
values (
27,
@studytypeversionid,
27,
@curdate, @userid, @curdate, @userid)

-----------------------------------------------------------------------
-- Mechanical Engineering [maritime - construction] September
-- Maskin [maritim - konstruktion] september
-----------------------------------------------------------------------
insert into StudyType_TechnicalPackage
values (
28,
@studytypeversionid,
28,
@curdate, @userid, @curdate, @userid)

-----------------------------------------------------------------
-- Mechanical Engineering [maritime] February
-- Maskin [maritim] februar
-----------------------------------------------------------------
insert into StudyType_TechnicalPackage
values (
29,
@studytypeversionid,
29,
@curdate, @userid, @curdate, @userid)

-----------------------------------------------------------------
-- Mechanical Engineering [maritime] September
-- Maskin [maritim] september
-----------------------------------------------------------------
insert into StudyType_TechnicalPackage
values (
30,
@studytypeversionid,
30,
@curdate, @userid, @curdate, @userid)

-----------------------------------------------------------------
-- Mechanical Engineering [maritime - energy] February
-- Maskin [maritim - energi] februar
-----------------------------------------------------------------
insert into StudyType_TechnicalPackage
values (
31,
@studytypeversionid,
31,
@curdate, @userid, @curdate, @userid)

-----------------------------------------------------------------
-- Mechanical Engineering [maritime - energy] September
-- Maskin [maritim - energi] september
-----------------------------------------------------------------
insert into StudyType_TechnicalPackage
values (
32,
@studytypeversionid,
32,
@curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------

-- Mechanical Engineering [production and management - process] February
-- Maskin [produktion og ledelse - proces] februar
-------------------------------------------------------------------------------

insert into StudyType_TechnicalPackage
values (
33,
@studytypeversionid,
33,
@curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------

-- Mechanical Engineering [production and management - process] September
-- Maskin [produktion og ledelse - proces] september
-------------------------------------------------------------------------------

insert into StudyType_TechnicalPackage
values (
34,
@studytypeversionid,
34,
@curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------

-- Mechanical Engineering [production and management - plastic] February
-- Maskin [produktion og ledelse - plast] februar
-------------------------------------------------------------------------------

insert into StudyType_TechnicalPackage
values (
```

```
35,
@studytypeversionid,
35,
@curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------

-- Mechanical Engineering [production and management - plastic] September
-- Maskin [produktion og ledelse - plast] september
------------------------------------------------------------------------------

insert into StudyType_TechnicalPackage
values (
36,
@studytypeversionid,
36,
@curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------------------

-- Mechanical Engineering [production and management - control and logistics] February
-- Maskin [produktion og ledelse - styring og logistik] februar
------------------------------------------------------------------------------------------

insert into StudyType_TechnicalPackage
values (
37,
@studytypeversionid,
37,
@curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------------------

-- Mechanical Engineering [production and management - control and logistics] September
-- Maskin [produktion og ledelse - styring og logistik] september
------------------------------------------------------------------------------------------

insert into StudyType_TechnicalPackage
values (
38,
@studytypeversionid,
38,
@curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------

-- Mechanical Engineering [production and management - organization] February
-- Maskin [produktion og ledelse - organisation] februar
------------------------------------------------------------------------------

insert into StudyType_TechnicalPackage
values (
39,
@studytypeversionid,
39,
@curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------

-- Mechanical Engineering [production and management - organization] September
-- Maskin [produktion og ledelse - organisation] september
------------------------------------------------------------------------------

insert into StudyType_TechnicalPackage
values (
40,
@studytypeversionid,
40,
@curdate, @userid, @curdate, @userid)
```

3.19 TechnicalField

```
------------------------------------------------------------------------------

--Common declarations
------------------------------------------------------------------------------

declare @userid uniqueidenti↓er
declare @curdate datetime
declare @da varchar(255), @en varchar(255)
declare @t↓d int, @tfvid int, @version int, @tfcid int
set @userid = 'D36DD218-AE59-4046-977C-659E4EB92C92'
set @curdate = getdate()
set @t↓d = 0
set @tfvid = 0
set @version = 0
set @tfcid = 0
------------------------------------------------------------------------------

--Data Security and Communication
------------------------------------------------------------------------------

set @t↓d = @t↓d + 1
set @da = 'Datasikkerhed og kommunikation'
```

```
set @en = 'Data Security and Communication'
insert into technical↓eld values (@t↓d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical↓eldversion
  values (@tfvid, @t↓d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '01227', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '01259', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '01425', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02230', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02240', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '34230', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '34240', @curdate, @userid, @curdate, @userid)
```

−−Operations Research

```
set @t↓d = @t↓d + 1
set @da = 'Operationsanalyse'
set @en = 'Operations Research'
insert into technical↓eld values (@t↓d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical↓eldversion
  values (@tfvid, @t↓d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02443', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02709', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02711', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02713', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02715', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02721', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02723', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '13430', @curdate, @userid, @curdate, @userid)
```

−−Dynamic Systems and non−linear Dynamics

```
set @t↓d = @t↓d + 1
set @da = 'Dynamiske systemer og ikke-lineær dynamik'
set @en = 'Dynamic Systems and non-linear Dynamics'
insert into technical↓eld values (@t↓d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical↓eldversion
  values (@tfvid, @t↓d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '01246', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '01449', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02417', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02421', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02443', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02613', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02623', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02645', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02655', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02685', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02687', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '10342', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '10344', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '28260', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '31310', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '31320', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '31360', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41223', @curdate, @userid, @curdate, @userid)
```

--Scientific Computing

```
set @t↓d = @t↓d + 1
set @da = 'Scientific computing'
set @en = 'Scientific Computing'
insert into technical↓eld values (@t↓d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical↓eldversion
  values (@tfvid, @t↓d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
```

```
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '01243', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '01246', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '01250', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '01456', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02581', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02611', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02613', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02623', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02645', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02685', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02715', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '10112', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '11105', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '11513', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '31430', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41124', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41223', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41320', @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------------------

--Data Analysis and Applied Statistics
------------------------------------------------------------------------------------------

set @t↓d = @t↓d + 1
set @da = 'Dataanalyse og anvendt statistik'
set @en = 'Data Analysis and Applied Statistics'
insert into technical↓eld values (@t↓d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical↓eldversion
  values (@tfvid, @t↓d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '01250', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '01227', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02407', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02409', @curdate, @userid, @curdate, @userid)
```

```
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02411', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02413', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02417', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02421', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02423', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02427', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02443', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02451', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02453', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02457', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02503', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02551', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02561', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02581', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '27511', @curdate, @userid, @curdate, @userid)
```

--Signal and Image Processing

```
set @t_d = @t_d + 1
set @da = 'Signal- og billedbehandling'
set @en = 'Signal and Image Processing'
insert into technical_eld values (@t_d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical_eldversion
  values (@tfvid, @t_d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '01243', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '01456', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02409', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02417', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02451', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02453', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02457', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02503', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
```

```
values(@tfcid, @tfvid, null, '02505', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02541', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02551', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02561', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02563', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02565', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '27011', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '31655', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '34320', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '34241', @curdate, @userid, @curdate, @userid)
```

−−Molecular Microbiology

```
set @t↓d = @t↓d + 1
set @da = 'Molekylær mikrobiologi'
set @en = 'Molecular Microbiology'
insert into technical↓eld values (@t↓d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical↓eldversion
  values (@tfvid, @t↓d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '27214', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '27217', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '27221', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '27231', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '27443', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '27251', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '27252', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '27405', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '27722', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '27723', @curdate, @userid, @curdate, @userid)
```

−−Environmental Biology and Bio−Diversity

```
set @t↓d = @t↓d + 1
set @da = 'Miljømikrobiologi og biodiversitet'
set @en = 'Environmental Biology and Bio-Diversity'
insert into technical↓eld values (@t↓d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1
```

```
insert into technical_eldversion
  values (@tfvid, @t_d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '27262', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '27443', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '27251', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '27252', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '27511', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '27550', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '27552', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '27582', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '27583', @curdate, @userid, @curdate, @userid)
```

--Bio-Chemistry and Nutrition

```
set @t_d = @t_d + 1
set @da = 'Biokemi og ernæring'
set @en = 'Bio-Chemistry and Nutrition'
insert into technical_eld values (@t_d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical_eldversion
  values (@tfvid, @t_d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '27301', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '27302', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '27311', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '27321', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '27333', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '27443', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '27732', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '27500', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '27733', @curdate, @userid, @curdate, @userid)
```

```
------------------------------------------------------------------------------------------------------------------------

--Bio-Informatics and System Biology
------------------------------------------------------------------------------------------------------------------------

set @t↓d = @t↓d + 1
set @da = 'Bioinformatik og system biologi'
set @en = 'Bio-Informatics and System Biology'
insert into technical↓eld values (@t↓d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical↓eldversion
  values (@tfvid, @t↓d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '27443', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '27251', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '27252', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '27311', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '27321', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '27403', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '27405', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '27442', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '27583', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '33232', @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------------------------------------------------

--Food-Biotechnology
------------------------------------------------------------------------------------------------------------------------

set @t↓d = @t↓d + 1
set @da = 'Levnedsmiddelbioteknologi'
set @en = 'Food-Biotechnology'
insert into technical↓eld values (@t↓d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical↓eldversion
  values (@tfvid, @t↓d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '27333', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '27500', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '27505', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '27520', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
```

```
values(@tfcid, @tfvid, null, '27580', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '27581', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '27582', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '27583', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '27717', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '27752', @curdate, @userid, @curdate, @userid)
```

--Process-Biotechnology

```
set @t_d = @t_d + 1
set @da = 'Procesbioteknologi'
set @en = 'Process-Biotechnology'
insert into technical_eld values (@t_d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical_eldversion
  values (@tfvid, @t_d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '27404', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '27406', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '27443', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '27442', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '27505', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '27733', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '27967', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '28231', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '28354', @curdate, @userid, @curdate, @userid)
```

--Medical Microbiology

```
set @t_d = @t_d + 1
set @da = 'Medicinsk mikrobiologi'
set @en = 'Medical Microbiology'
insert into technical_eld values (@t_d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical_eldversion
  values (@tfvid, @t_d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)
```

```
--No pointgiving courses!!
-------------------------------------------------------------------------------------------------

--Planning and Execution
-------------------------------------------------------------------------------------------------

set @t↓d = @t↓d + 1
set @da = 'Planlægning og udførelse'
set @en = 'Planning and Execution'
insert into technical↓eld values (@t↓d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical↓eldversion
  values (@tfvid, @t↓d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02541', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02543', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02551', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '11203', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '11205', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '11206', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '11207', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '11251', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '11252', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '11253', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '11254', @curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------------------------

--Building and Plant Construtions
-------------------------------------------------------------------------------------------------

set @t↓d = @t↓d + 1
set @da = 'Bygning- og anlægskontruktioner'
set @en = 'Building and Plant Constructions'
insert into technical↓eld values (@t↓d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical↓eldversion
  values (@tfvid, @t↓d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '11512', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '11513', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '11514', @curdate, @userid, @curdate, @userid)
```

```
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11521', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11522', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11523', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11531', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11532', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11541', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11542', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11544', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11551', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11552', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11553', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11601', @curdate, @userid, @curdate, @userid)
```

--Hydraulics, Geotechnology and Road Building

```
set @t_d = @t_d + 1
set @da = 'Vandbygning, geoteknik og vejbygning'
set @en = 'Hydraulics, Geotechnology and Road Building'
insert into technical_eld values (@t_d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical_eldversion
  values (@tfvid, @t_d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11413', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11414', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11412', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11415', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11423', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '12530', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '13310', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '13311', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '13320', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '12320', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '41110', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
```

```
values(@tfcid, @tfvid, null, '41112', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41120', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41121', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41122', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41123', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41124', @curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------------------------------

--Building Techniques and Building Physics
-------------------------------------------------------------------------------------------------------

set @t↓d = @t↓d + 1
set @da = 'Bygningsteknik og bygningsfysik'
set @en = 'Building Techniques and Building Physics'
insert into technical↓eld values (@t↓d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical↓eldversion
  values (@tfvid, @t↓d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '11102', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '11005', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '11106', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '11021', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '11022', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '27552', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '31200', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '31240', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '31275', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41722', @curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------------------------------

--Energy, Resource Economy and Town Planning
-------------------------------------------------------------------------------------------------------

set @t↓d = @t↓d + 1
set @da = 'Energi, ressourceøkonomi og byplanlægning'
set @en = 'Energy, Resource Economy and Town Planning'
insert into technical↓eld values (@t↓d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical↓eldversion
  values (@tfvid, @t↓d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
```

```
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11103', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11104', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11105', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11302', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11303', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11304', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11305', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11306', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11310', @curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------------------------

--Materials
-------------------------------------------------------------------------------------------------

set @t_d = @t_d + 1
set @da = 'Materialer'
set @en = 'Materials'
insert into technical_eld values (@t_d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical_eldversion
  values (@tfvid, @t_d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11502', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11503', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11505', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11506', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11546', @curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------------------------

--Acoustic Technology
-------------------------------------------------------------------------------------------------

set @t_d = @t_d + 1
set @da = 'Akustik teknologi'
set @en = 'Acoustic Technology'
insert into technical_eld values (@t_d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical_eldversion
  values (@tfvid, @t_d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
```

```
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '31220', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '31230', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '31240', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '31250', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '31260', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '31270', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '31275', @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------------------------------------------------

--Automation and Instrumentation
------------------------------------------------------------------------------------------------------------------------

set @t_d = @t_d + 1
set @da = 'Automation og instrumentering'
set @en = 'Automation and Instrumentation'
insert into technical_eld values (@t_d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical_eldversion
  values (@tfvid, @t_d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '31305', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '31310', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '31320', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '31340', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '31360', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '31361', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '31365', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '31370', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '31380', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '31385', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '31800', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '31820', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '31390', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '31825', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '31830', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '31840', @curdate, @userid, @curdate, @userid)
```

```
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '31860', @curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------------------

--Electromagnetic Systems
-------------------------------------------------------------------------------------------

set @t_d = @t_d + 1
set @da = 'Elektromagnetiske systemer'
set @en = 'Electromagnetic Systems'
insert into technical_eld values (@t_d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical_eldversion
  values (@tfvid, @t_d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '31415', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '31420', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '31425', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '31430', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '31435', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '31440', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '34030', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '34140', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '34230', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '34259', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '34349', @curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------------------

--Electronics and Signal Processing
-------------------------------------------------------------------------------------------

set @t_d = @t_d + 1
set @da = 'Elektronik og signalbehandling'
set @en = 'Electronics and Signal Processing'
insert into technical_eld values (@t_d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical_eldversion
  values (@tfvid, @t_d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02200', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02201', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
```

```
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02202', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02204', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02206', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02208', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02451', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02453', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02455', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02457', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02503', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '31610', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '31620', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '31625', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '31630', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '31640', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '31641', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '31650', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '31651', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '31655', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '31656', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '31661', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '31662', @curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------------------

--Electric Energy Technology
-------------------------------------------------------------------------------------------

set @t_d = @t_d + 1
set @da = 'El-energiteknik '
set @en = 'Electric Energy Technology'
insert into technical_eld values (@t_d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical_eldversion
  values (@tfvid, @t_d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '32070', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '32220', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '32230', @curdate, @userid, @curdate, @userid)
```

```
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '32240', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '32250', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '32260', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '32270', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '32405', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '32410', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '32420', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '32430', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '32440', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '32450', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '32620', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '32630', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '32640', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '32670', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '32820', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '32830', @curdate, @userid, @curdate, @userid)
-------------------------------------------------------------------------------------------

-- Building Energy
-------------------------------------------------------------------------------------------

set @t_d = @t_d + 1
set @da = 'Bygningsenergi'
set @en = 'Building Energy'
insert into technical_eld values (@t_d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical_eldversion
  values (@tfvid, @t_d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11102', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11103', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11104', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11105', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11106', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '41815', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '41814', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
```

```
values(@tfcid, @tfvid, null, '41721', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41722', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41420', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '32810', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '32820', @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------------------------

--Energy Technology
------------------------------------------------------------------------------------------------

set @t↓d = @t↓d + 1
set @da = 'Energiteknologi'
set @en = 'Energy Technology'
insert into technical↓eld values (@t↓d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical↓eldversion
  values (@tfvid, @t↓d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '28244', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41311', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41313', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41321', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41402', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41413', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41414', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41415', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41416', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41417', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41420', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41814', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41815', @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------------------------

--Electrical Power Supply
------------------------------------------------------------------------------------------------

set @t↓d = @t↓d + 1
set @da = 'El-forsyning'
set @en = 'Electrical Power Supply'
insert into technical↓eld values (@t↓d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical↓eldversion
  values (@tfvid, @t↓d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
```

```
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '32030', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '32076', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '32230', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '32240', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '32260', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '32270', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '32410', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '32430', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '32440', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '32640', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '32810', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '32820', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '32830', @curdate, @userid, @curdate, @userid)
```

--Renewable Energy Sources

```
set @t_d = @t_d + 1
set @da = 'Vedvarende energi'
set @en = 'Renewable Energy Sources'
insert into technical_eld values (@t_d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical_eldversion
  values (@tfvid, @t_d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '10474', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11103', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11104', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11310', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '28244', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '32076', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '32270', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
```

```
values(@tfcid, @tfvid, null, '41311', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '41313', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '41415', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '41420', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '41814', @curdate, @userid, @curdate, @userid)
```

--Manufacturing

```
set @t_d = @t_d + 1
set @da = 'Fremstilling'
set @en = 'Manufacturing'
insert into technical_eld values (@t_d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical_eldversion
  values (@tfvid, @t_d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '41812', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '42130', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '42155', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '42213', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '42215', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '42221', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '42222', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '42224', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '42230', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '42250', @curdate, @userid, @curdate, @userid)
```

--Production and Management Studies

```
set @t_d = @t_d + 1
set @da = 'Produktions- og virksomhedsledelse'
set @en = 'Production and Management Studies'
insert into technical_eld values (@t_d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical_eldversion
  values (@tfvid, @t_d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)
```

```
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '42371', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '42430', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '42435', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '42450', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '42455', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '42460', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '42465', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '42470', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '42531', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '42541', @curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------------------------

--Innovation and Viability
-------------------------------------------------------------------------------------------------

set @t_d = @t_d + 1
set @da = 'Innovation og bæredygtighed'
set @en = 'Innovation and Viability'
insert into technical_eld values (@t_d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical_eldversion
  values (@tfvid, @t_d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '41612', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '41629', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '42342', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '42374', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '42532', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '42540', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '42640', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '88312', @curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------------------------

--Global Computing
-------------------------------------------------------------------------------------------------

set @t_d = @t_d + 1
set @da = 'Globale computersystemer'
set @en = 'Global Computing'
insert into technical_eld values (@t_d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical_eldversion
  values (@tfvid, @t_d, @version,
  '<cultures>
```

```
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02220', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02222', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02224', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02225', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02226', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02230', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02232', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02443', @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------------------------------------------------

--Computer Systems Engineering
------------------------------------------------------------------------------------------------------------------------

set @t_d = @t_d + 1
set @da = 'Computersystemudvikling'
set @en = 'Computer Systems Engineering'
insert into technical_eld values (@t_d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical_eldversion
  values (@tfvid, @t_d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02200', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02201', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02202', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02204', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02206', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02208', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02220', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02224', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02225', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02226', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02228', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02260', @curdate, @userid, @curdate, @userid)
```

```
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02911', @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------------------

--Models of Computation
------------------------------------------------------------------------------------------

set @t_d = @t_d + 1
set @da = 'Computermodeller'
set @en = 'Models of Computation'
insert into technical_eld values (@t_d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical_eldversion
  values (@tfvid, @t_d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02240', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02242', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02248', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02260', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02261', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02262', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02264', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02266', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02268', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02280', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02282', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02284', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02286', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02288', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02913', @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------------------

--Data Analysis and Modelling
------------------------------------------------------------------------------------------

set @t_d = @t_d + 1
set @da = 'Dataanalyse og modellering'
set @en = 'Data Analysis and Modelling'
insert into technical_eld values (@t_d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical_eldversion
  values (@tfvid, @t_d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
```

```
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02501', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02561', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02563', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02565', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02503', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02507', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02453', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02455', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02457', @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------------------------------------

--Chemical Synthesis and Product Development
------------------------------------------------------------------------------------------------------------

set @t↓d = @t↓d + 1
set @da = 'Kemisk syntese og produktudvikling'
set @en = 'Chemical Synthesis and Product Development'
insert into technical↓eld values (@t↓d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical↓eldversion
  values (@tfvid, @t↓d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '26124', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '26126', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '26140', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '26142', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '26320', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '26416', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '26430', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '26510', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '27403', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '28212', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '28213', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '28310', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '28315', @curdate, @userid, @curdate, @userid)
```

```
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '28316', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '28414', @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------------------

--Chemical Process Design
------------------------------------------------------------------------------------------

set @t_d = @t_d + 1
set @da = 'Kemisk procesdesign'
set @en = 'Chemical Process Design'
insert into technical_eld values (@t_d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical_eldversion
  values (@tfvid, @t_d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '26128', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '27404', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '27406', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '28231', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '28240', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '28244', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '28250', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '28320', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '28350', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '28375', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '28376', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '28415', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '28416', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '28434', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '28443', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '28851', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '28852', @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------------------

--Analysis and Instrumentation
------------------------------------------------------------------------------------------

set @t_d = @t_d + 1
set @da = 'Analyse og instrumentering'
set @en = 'Analysis and Instrumentation'
insert into technical_eld values (@t_d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1
```

```
insert into technical↓eldversion
  values (@tfvid, @t↓d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '26240', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '26310', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '26312', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '26322', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '26330', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '26334', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '26418', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '27511', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '33232', @curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------------------

--Modelling of Chemical and Technical−Chemical Systems
-------------------------------------------------------------------------------------------

set @t↓d = @t↓d + 1
set @da = 'Modellering af kemiske og teknisk-kemiske systemer'
set @en = 'Modelling of Chemical and Technical-Chemical Systems'
insert into technical↓eld values (@t↓d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical↓eldversion
  values (@tfvid, @t↓d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '26235', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '26250', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '26230', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '28230', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '28241', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '28252', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '28423', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '28462', @curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------------------

--Construction and Product Development
-------------------------------------------------------------------------------------------
```

```
set @t↓d = @t↓d + 1
set @da = 'Konstruktion og produktudvikling'
set @en = 'Construction and Product Development'
insert into technical↓eld values (@t↓d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical↓eldversion
  values (@tfvid, @t↓d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41612', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41627', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41628', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41629', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '42374', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '42450', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '42460', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '42640', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41624', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41625', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41626', @curdate, @userid, @curdate, @userid)
```

−− Relative Strength and Dynamics

```
set @t↓d = @t↓d + 1
set @da = 'Styrkeforhold og dynamik'
set @en = 'Relative Strength and Dynamics'
insert into technical↓eld values (@t↓d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical↓eldversion
  values (@tfvid, @t↓d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41511', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41521', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41522', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41525', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41811', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
```

```
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41614', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41622', @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------------------------------------

--Aerodynamics and Industrial Flow Mechanics
------------------------------------------------------------------------------------------------------------

set @t↓d = @t↓d + 1
set @da = 'Aerodynamik og industriel strømningsmekanik'
set @en = 'Aerodynamics and Industrial Flow Mechanics'
insert into technical↓eld values (@t↓d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical↓eldversion
  values (@tfvid, @t↓d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41313', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41320', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41323', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41822', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41322', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41120', @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------------------------------------

--Maritime Technology
------------------------------------------------------------------------------------------------------------

set @t↓d = @t↓d + 1
set @da = 'Maritim teknik'
set @en = 'Maritime Technology'
insert into technical↓eld values (@t↓d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical↓eldversion
  values (@tfvid, @t↓d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41211', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41221', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41222', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41223', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41271', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41272', @curdate, @userid, @curdate, @userid)
```

```
------------------------------------------------------------------------------------------------------------

--Material Process Technology (Choice, Synthesis and Manufacturing)
------------------------------------------------------------------------------------------------------------

set @t↓d = @t↓d + 1
set @da = 'Materialeprocesteknologi (materialevalg, -syntese og forarbejdning)'
set @en = 'Material Process Technology (Choice, Synthesis and Manufacturing)'
insert into technical↓eld values (@t↓d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical↓eldversion
  values (@tfvid, @t↓d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '26140', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '26142', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '42155', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '42220', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '42222', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '42211', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '42230', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '42234', @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------------------------------------

--Material Characterization and Modelling
------------------------------------------------------------------------------------------------------------

set @t↓d = @t↓d + 1
set @da = 'Materialekarakterisering og -modellering'
set @en = 'Material Characterization and Modelling'
insert into technical↓eld values (@t↓d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical↓eldversion
  values (@tfvid, @t↓d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '42213', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '42260', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '26320', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '10312', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41522', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41502', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
```

```
values(@tfcid, @tfvid, null, '11506', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '42224', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '42150', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '42250', @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------------------------------------

--Structural Materials
------------------------------------------------------------------------------------------------------------

set @t↓d = @t↓d + 1
set @da = 'Strukturelle materialer'
set @en = 'Structural Materials'
insert into technical↓eld values (@t↓d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical↓eldversion
  values (@tfvid, @t↓d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '42130', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '42120', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41503', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '42135', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '11501', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '11502', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '11503', @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------------------------------------

--Functional Materials
------------------------------------------------------------------------------------------------------------

set @t↓d = @t↓d + 1
set @da = 'Funktionelle materialer'
set @en = 'Functional Materials'
insert into technical↓eld values (@t↓d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical↓eldversion
  values (@tfvid, @t↓d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '33441', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '33442', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '33470', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '33250', @curdate, @userid, @curdate, @userid)
```

```
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '33251', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '33253', @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------------------

--Micro- and Nano-Technology
------------------------------------------------------------------------------------------

set @t_d = @t_d + 1
set @da = 'Mikro- og nanoteknologi'
set @en = 'Micro- and Nano-Technology'
insert into technical_eld values (@t_d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical_eldversion
  values (@tfvid, @t_d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '33430', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '33355', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '33232', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '33320', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '33205', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '33471', @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------------------

--Water Resources
------------------------------------------------------------------------------------------

set @t_d = @t_d + 1
set @da = 'Vandressourcer'
set @en = 'Water Resources'
insert into technical_eld values (@t_d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical_eldversion
  values (@tfvid, @t_d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '12320', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '12321', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '12322', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '12323', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '12332', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '12333', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
```

```
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '12340', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '12341', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41122', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41123', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '41124', @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------------------------------------------------

--Natural Resources
------------------------------------------------------------------------------------------------------------------------

set @t↓d = @t↓d + 1
set @da = 'Naturlige ressourcer'
set @en = 'Natural Resources'
insert into technical↓eld values (@t↓d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical↓eldversion
  values (@tfvid, @t↓d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '11423', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '12420', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '12411', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '12421', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '12422', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '12440', @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------------------------------------------------

--Pollution of Air, Water and Ground
------------------------------------------------------------------------------------------------------------------------

set @t↓d = @t↓d + 1
set @da = 'Forurening af luft, vand og jord'
set @en = 'Pollution of Air, Water and Ground'
insert into technical↓eld values (@t↓d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical↓eldversion
  values (@tfvid, @t↓d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '11421', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '12220', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '12230', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
```

```
values(@tfcid, @tfvid, null, '12231', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '12232', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '12330', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '12331', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '41120', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '27262', @curdate, @userid, @curdate, @userid)
```

--Environmental Technology

```
set @t_d = @t_d + 1
set @da = 'Miljøteknologi'
set @en = 'Environmental Technology'
insert into technical_eld values (@t_d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical_eldversion
  values (@tfvid, @t_d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '12120', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '12121', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '12130', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '12131', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '12140', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '12242', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '12243', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '28244', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '28375', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '28376', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '28852', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '41413', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '41722', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '42342', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '42372', @curdate, @userid, @curdate, @userid)
```

--Planning and Management of Environment, Resources and Working Environment

---

```
set @t_d = @t_d + 1
set @da = 'Planlægning og ledelse af miljø, ressourcer og arbejdsmiljø'
set @en = 'Planning and Management of Environment, Resources and Working Environment'
insert into technical_eld values (@t_d, @curdate, @userid, @curdate, @userid)
```

```
set @tfvid = @tfvid + 1
set @version = 1

insert into technical↓eldversion
  values (@tfvid, @t↓d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '11310', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '11302', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '11303', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '12240', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '31250', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '42470', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '42521', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '42531', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '42540', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '42640', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '42644', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '42645', @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------------------------------------

--Methods and Models
------------------------------------------------------------------------------------------------------------

set @t↓d = @t↓d + 1
set @da = 'Metoder og modeller'
set @en = 'Methods and Models'
insert into technical↓eld values (@t↓d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical↓eldversion
  values (@tfvid, @t↓d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02344', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02405', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02406', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02411', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02413', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
```

```
values(@tfcid, @tfvid, null, '02417', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02423', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02431', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02443', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02531', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02541', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02543', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02551', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02561', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02563', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02565', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11202', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02711', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02713', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02715', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02725', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02731', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02735', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '34742', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '34743', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '34841', @curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------------------

--Traffic, Transportation and Town Plan
-------------------------------------------------------------------------------------------

set @t_d = @t_d + 1
set @da = 'Trafik, transport og byplan'
set @en = 'Traffic, Transportation and Town Plan'
insert into technical_eld values (@t_d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical_eldversion
  values (@tfvid, @t_d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11302', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11303', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11304', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
```

```
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11305', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11306', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '13110', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '13120', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '13140', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '13210', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '13230', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '13430', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '13310', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '13320', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '13340', @curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------------------

--Organization, Management and Economics
-------------------------------------------------------------------------------------------

set @t_d = @t_d + 1
set @da = 'Organisation, ledelse og økonomi'
set @en = 'Organization, Management and Economics'
insert into technical_eld values (@t_d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical_eldversion
  values (@tfvid, @t_d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11302', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11003', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11202', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11205', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11206', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11207', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11252', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11253', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11254', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11551', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '42371', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '42410', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '42415', @curdate, @userid, @curdate, @userid)
```

```
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '42420', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '42425', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '42430', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '42440', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '42450', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '42455', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '42460', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '42465', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '42470', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '42531', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '42540', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '42541', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '12240', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '42640', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '42642', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '42644', @curdate, @userid, @curdate, @userid)
```

--Nanoscale Materials Physics

```
set @t_d = @t_d + 1
set @da = 'Nanoskala materialefysik'
set @en = 'Nanoscale Materials Physics'
insert into technical_eld values (@t_d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical_eldversion
  values (@tfvid, @t_d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '10110', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '10111', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '10120', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '10112', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '10300', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '10302', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '10304', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
```

```
values(@tfcid, @tfvid, null, '10306', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '10308', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '10310', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '10312', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '10320', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '10322', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '10346', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '10467', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '10469', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '33205', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '33250', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '33251', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '33355', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '33441', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '33442', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '33471', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02613', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '26510', @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------------------

--Biophysics and Complex Systems
------------------------------------------------------------------------------------------

set @t↓d = @t↓d + 1
set @da = 'Biofysik og komplekse systemer'
set @en = 'Biophysics and Complex Systems'
insert into technical↓eld values (@t↓d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical↓eldversion
  values (@tfvid, @t↓d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '10110', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '10111', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '10120', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '10112', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '10302', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '10340', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
```

```
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '10342', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '10344', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '10346', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '10348', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '10350', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '10380', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '10467', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '10469', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '10477', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '33232', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '33430', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '01449', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02615', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02655', @curdate, @userid, @curdate, @userid)
```

−−Photonics

```
set @t↓d = @t↓d + 1
set @da = 'Fotonik'
set @en = 'Photonics'
insert into technical↓eld values (@t↓d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical↓eldversion
  values (@tfvid, @t↓d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '10110', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '10111', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '10120', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '10112', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '10300', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '10370', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '10372', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '10374', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '10376', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '10378', @curdate, @userid, @curdate, @userid)
```

```
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '10380', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '10392', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '10467', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '10469', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '34030', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '34040', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '34050', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '34049', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '34055', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '34059', @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------------------------------------------------

--Construction of Components and Software
------------------------------------------------------------------------------------------------------------------------

set @t↓d = @t↓d + 1
set @da = 'Konstruktion af komponenter og software'
set @en = 'Construction of Components and Software'
insert into technical↓eld values (@t↓d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical↓eldversion
  values (@tfvid, @t↓d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02202', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02220', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02222', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02224', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02266', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02445', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '10243', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '10247', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '31415', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '31420', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '31430', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '31641', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '34030', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
```

```
values(@tfcid, @tfvid, null, '34040', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '34049', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '34050', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '34055', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '34059', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '34140', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '34150', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '34151', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '34349', @curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------

--System Development and Methods
-------------------------------------------------------------------------------

set @t_d = @t_d + 1
set @da = 'Systemudvikling og metode'
set @en = 'System Development and Methods'
insert into technical_eld values (@t_d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical_eldversion
  values (@tfvid, @t_d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '01259', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '01425', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02222', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02501', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '02455', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '11201', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '34230', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '34240', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '34241', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '34259', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '34320', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '34340', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '34341', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '34350', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical_eld_course
values(@tfcid, @tfvid, null, '34351', @curdate, @userid, @curdate, @userid)
```

```
------------------------------------------------------------------------------------------------------------------------
--Applications
------------------------------------------------------------------------------------------------------------------------

set @t↓d = @t↓d + 1
set @da = 'Applikationer'
set @en = 'Applications'
insert into technical↓eld values (@t↓d, @curdate, @userid, @curdate, @userid)

set @tfvid = @tfvid + 1
set @version = 1

insert into technical↓eldversion
  values (@tfvid, @t↓d, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>',
@curdate, @userid, @curdate, @userid)

set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02264', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02266', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02335', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02561', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '02565', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '11201', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '31230', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '34531', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '34631', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '34641', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '34642', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '34643', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '34651', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '34741', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '34742', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '34743', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '34841', @curdate, @userid, @curdate, @userid)
set @tfcid = @tfcid + 1
insert into technical↓eld_course
values(@tfcid, @tfvid, null, '34842', @curdate, @userid, @curdate, @userid)
```

3.20 TechnicalLine

```
------------------------------------------------------------------------------------------------------------------------
--Common declarations
------------------------------------------------------------------------------------------------------------------------
declare @userid uniqueidenti↓er
declare @curdate datetime
declare @da varchar(255), @en varchar(255)
declare @tlid int, @tlvid int, @version int, @pid uniqueidenti↓er
declare @tlpcid int, @tpid int, @tlptpid int, @tlptpcid int
declare @tlsid int, @tlt↓d int, @sid int, @t↓d int
set @userid = 'D36DD218-AE59-4046-977C-659E4EB92C92'
```

```
set @curdate = getdate()
set @tlid = 0
set @tlvid = 0
set @tlpcid = 0
set @tpid = 0
set @tlptpid = 0
set @tlptpcid = 0
set @tlsid = 0
set @tlt␣d = 0
set @version = 1
-----------------------------------------------------------------------
set @pid = '4054b9a7-ae3a-40b6-b96e-14dc212afb81'
set @da = 'Anvendt matematik'
set @en = 'Applied Mathematics'

set @tlid = @tlid + 1
insert into technicalline values (@tlid, @curdate, @userid, @curdate, @userid)

set @tlvid = @tlvid + 1
insert into technicallineversion
  values (@tlvid, @tlid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>', @pid,
@curdate, @userid, @curdate, @userid)

set @sid = 1
while @sid < 7
begin
  set @tlsid = @tlsid + 1
  insert into technicalline_specialization
  values(@tlsid, @tlvid, @sid, @curdate, @userid, @curdate, @userid)
  set @sid = @sid + 1
end

set @t␣d = 1
while @t␣d < 7
begin
  set @tlt␣d = @tlt␣d + 1
  insert into technicalline_technical␣eld
  values(@tlt␣d, @tlvid, @t␣d, @curdate, @userid, @curdate, @userid)
  set @t␣d = @t␣d + 1
end

set @tpid = 1
set @tlptpid = @tlptpid + 1
insert into technicalline_prerequisitetechnicalpackage
values(@tlptpid, @tlvid, @tpid, @curdate, @userid, @curdate, @userid)
set @tlptpcid = @tlptpcid + 1
insert into technicalline_prerequisitetechnicalpackagecourse
values(@tlptpcid, @tlptpid, null, '01141', @curdate, @userid, @curdate, @userid)
set @tlptpcid = @tlptpcid + 1
insert into technicalline_prerequisitetechnicalpackagecourse
values(@tlptpcid, @tlptpid, null, '01142', @curdate, @userid, @curdate, @userid)
set @tlptpcid = @tlptpcid + 1
insert into technicalline_prerequisitetechnicalpackagecourse
values(@tlptpcid, @tlptpid, null, '02401', @curdate, @userid, @curdate, @userid)

set @tpid = 2
set @tlptpid = @tlptpid + 1
insert into technicalline_prerequisitetechnicalpackage
values(@tlptpid, @tlvid, @tpid, @curdate, @userid, @curdate, @userid)
set @tlptpcid = @tlptpcid + 1
insert into technicalline_prerequisitetechnicalpackagecourse
values(@tlptpcid, @tlptpid, null, '01141', @curdate, @userid, @curdate, @userid)
set @tlptpcid = @tlptpcid + 1
insert into technicalline_prerequisitetechnicalpackagecourse
values(@tlptpcid, @tlptpid, null, '01142', @curdate, @userid, @curdate, @userid)
set @tlptpcid = @tlptpcid + 1
insert into technicalline_prerequisitetechnicalpackagecourse
values(@tlptpcid, @tlptpid, null, '02401', @curdate, @userid, @curdate, @userid)

set @tpid = 3
set @tlptpid = @tlptpid + 1
insert into technicalline_prerequisitetechnicalpackage
values(@tlptpid, @tlvid, @tpid, @curdate, @userid, @curdate, @userid)
set @tlptpcid = @tlptpcid + 1
insert into technicalline_prerequisitetechnicalpackagecourse
values(@tlptpcid, @tlptpid, null, '01141', @curdate, @userid, @curdate, @userid)
set @tlptpcid = @tlptpcid + 1
insert into technicalline_prerequisitetechnicalpackagecourse
values(@tlptpcid, @tlptpid, null, '01142', @curdate, @userid, @curdate, @userid)
set @tlptpcid = @tlptpcid + 1
insert into technicalline_prerequisitetechnicalpackagecourse
values(@tlptpcid, @tlptpid, null, '02401', @curdate, @userid, @curdate, @userid)

set @tpid = 4
set @tlptpid = @tlptpid + 1
insert into technicalline_prerequisitetechnicalpackage
values(@tlptpid, @tlvid, @tpid, @curdate, @userid, @curdate, @userid)
set @tlptpcid = @tlptpcid + 1
insert into technicalline_prerequisitetechnicalpackagecourse
```

```
values(@tlptpcid, @tlptpid, null, '01141', @curdate, @userid, @curdate, @userid)
set @tlptpcid = @tlptpcid + 1
insert into technicalline_prerequisitetechnicalpackagecourse
values(@tlptpcid, @tlptpid, null, '01142', @curdate, @userid, @curdate, @userid)
set @tlptpcid = @tlptpcid + 1
insert into technicalline_prerequisitetechnicalpackagecourse
values(@tlptpcid, @tlptpid, null, '02401', @curdate, @userid, @curdate, @userid)

set @tpid = 5
set @tlptpid = @tlptpid + 1
insert into technicalline_prerequisitetechnicalpackage
values(@tlptpid, @tlvid, @tpid, @curdate, @userid, @curdate, @userid)
set @tlptpcid = @tlptpcid + 1
insert into technicalline_prerequisitetechnicalpackagecourse
values(@tlptpcid, @tlptpid, null, '01141', @curdate, @userid, @curdate, @userid)
set @tlptpcid = @tlptpcid + 1
insert into technicalline_prerequisitetechnicalpackagecourse
values(@tlptpcid, @tlptpid, null, '01142', @curdate, @userid, @curdate, @userid)
set @tlptpcid = @tlptpcid + 1
insert into technicalline_prerequisitetechnicalpackagecourse
values(@tlptpcid, @tlptpid, null, '02401', @curdate, @userid, @curdate, @userid)

set @tpid = 6 --Recommended
set @tlptpid = @tlptpid + 1
insert into technicalline_prerequisitetechnicalpackage
values(@tlptpid, @tlvid, @tpid, @curdate, @userid, @curdate, @userid)

set @tpid = 7
set @tlptpid = @tlptpid + 1
insert into technicalline_prerequisitetechnicalpackage
values(@tlptpid, @tlvid, @tpid, @curdate, @userid, @curdate, @userid)
set @tlptpcid = @tlptpcid + 1
insert into technicalline_prerequisitetechnicalpackagecourse
values(@tlptpcid, @tlptpid, null, '01141', @curdate, @userid, @curdate, @userid)
set @tlptpcid = @tlptpcid + 1
insert into technicalline_prerequisitetechnicalpackagecourse
values(@tlptpcid, @tlptpid, null, '01142', @curdate, @userid, @curdate, @userid)
set @tlptpcid = @tlptpcid + 1
insert into technicalline_prerequisitetechnicalpackagecourse
values(@tlptpcid, @tlptpid, null, '02401', @curdate, @userid, @curdate, @userid)

set @tpid = 8
set @tlptpid = @tlptpid + 1
insert into technicalline_prerequisitetechnicalpackage
values(@tlptpid, @tlvid, @tpid, @curdate, @userid, @curdate, @userid)
set @tlptpcid = @tlptpcid + 1
insert into technicalline_prerequisitetechnicalpackagecourse
values(@tlptpcid, @tlptpid, null, '01141', @curdate, @userid, @curdate, @userid)
set @tlptpcid = @tlptpcid + 1
insert into technicalline_prerequisitetechnicalpackagecourse
values(@tlptpcid, @tlptpid, null, '01142', @curdate, @userid, @curdate, @userid)
set @tlptpcid = @tlptpcid + 1
insert into technicalline_prerequisitetechnicalpackagecourse
values(@tlptpcid, @tlptpid, null, '02401', @curdate, @userid, @curdate, @userid)

set @tpid = 9
set @tlptpid = @tlptpid + 1
insert into technicalline_prerequisitetechnicalpackage
values(@tlptpid, @tlvid, @tpid, @curdate, @userid, @curdate, @userid)
set @tlptpcid = @tlptpcid + 1
insert into technicalline_prerequisitetechnicalpackagecourse
values(@tlptpcid, @tlptpid, null, '01141', @curdate, @userid, @curdate, @userid)
set @tlptpcid = @tlptpcid + 1
insert into technicalline_prerequisitetechnicalpackagecourse
values(@tlptpcid, @tlptpid, null, '01142', @curdate, @userid, @curdate, @userid)
set @tlptpcid = @tlptpcid + 1
insert into technicalline_prerequisitetechnicalpackagecourse
values(@tlptpcid, @tlptpid, null, '02401', @curdate, @userid, @curdate, @userid)

set @tpid = 10
set @tlptpid = @tlptpid + 1
insert into technicalline_prerequisitetechnicalpackage
values(@tlptpid, @tlvid, @tpid, @curdate, @userid, @curdate, @userid)
set @tlptpcid = @tlptpcid + 1
insert into technicalline_prerequisitetechnicalpackagecourse
values(@tlptpcid, @tlptpid, null, '01141', @curdate, @userid, @curdate, @userid)
set @tlptpcid = @tlptpcid + 1
insert into technicalline_prerequisitetechnicalpackagecourse
values(@tlptpcid, @tlptpid, null, '01142', @curdate, @userid, @curdate, @userid)
set @tlptpcid = @tlptpcid + 1
insert into technicalline_prerequisitetechnicalpackagecourse
values(@tlptpcid, @tlptpid, null, '02401', @curdate, @userid, @curdate, @userid)

set @tpid = 11
set @tlptpid = @tlptpid + 1
insert into technicalline_prerequisitetechnicalpackage
values(@tlptpid, @tlvid, @tpid, @curdate, @userid, @curdate, @userid)
set @tlptpcid = @tlptpcid + 1
insert into technicalline_prerequisitetechnicalpackagecourse
values(@tlptpcid, @tlptpid, null, '01141', @curdate, @userid, @curdate, @userid)
set @tlptpcid = @tlptpcid + 1
insert into technicalline_prerequisitetechnicalpackagecourse
values(@tlptpcid, @tlptpid, null, '01142', @curdate, @userid, @curdate, @userid)
set @tlptpcid = @tlptpcid + 1
insert into technicalline_prerequisitetechnicalpackagecourse
values(@tlptpcid, @tlptpid, null, '02401', @curdate, @userid, @curdate, @userid)
------------------------------------------------------------------------------
set @pid = '035e46a9-b98f-49cb-9ed0-30521bdb6dc0'
set @da = 'Bioteknologi'
```

```
set @en = 'Biotechnology'

set @tlid = @tlid + 1
insert into technicalline values (@tlid, @curdate, @userid, @curdate, @userid)

set @tlvid = @tlvid + 1
insert into technicallineversion
  values (@tlvid, @tlid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>', @pid,
@curdate, @userid, @curdate, @userid)

set @sid = 7
while @sid < 14
begin
  set @tlsid = @tlsid + 1
  insert into technicalline_specialization
  values(@tlsid, @tlvid, @sid, @curdate, @userid, @curdate, @userid)
  set @sid = @sid + 1
end

set @t↓d = 7
while @t↓d < 14
begin
  set @tlt↓d = @tlt↓d + 1
  insert into technicalline_technical↓eld
  values(@tlt↓d, @tlvid, @t↓d, @curdate, @userid, @curdate, @userid)
  set @t↓d = @t↓d + 1
end

set @tlpcid = @tlpcid + 1
insert into technicalline_prerequisitecourse
values(@tlpcid, @tlvid, null, '27021', @curdate, @userid, @curdate, @userid)
set @tlpcid = @tlpcid + 1
insert into technicalline_prerequisitecourse
values(@tlpcid, @tlvid, null, '27031', @curdate, @userid, @curdate, @userid)

set @tpid = 1
while @tpid < 12
begin
  set @tlptpid = @tlptpid + 1
  insert into technicalline_prerequisitetechnicalpackage
  values(@tlptpid, @tlvid, @tpid, @curdate, @userid, @curdate, @userid)
  set @tpid = @tpid + 1
end
------------------------------------------------------------------------
set @pid = '95877f88-e661-4657-a874-ff2065b7e653'
set @da = 'Bygning'
set @en = 'Civil engineering'

set @tlid = @tlid + 1
insert into technicalline values (@tlid, @curdate, @userid, @curdate, @userid)

set @tlvid = @tlvid + 1
insert into technicallineversion
  values (@tlvid, @tlid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>', @pid,
@curdate, @userid, @curdate, @userid)

set @sid = 14
while @sid < 18
begin
  set @tlsid = @tlsid + 1
  insert into technicalline_specialization
  values(@tlsid, @tlvid, @sid, @curdate, @userid, @curdate, @userid)
  set @sid = @sid + 1
end

set @t↓d = 14
while @t↓d < 20
begin
  set @tlt↓d = @tlt↓d + 1
  insert into technicalline_technical↓eld
  values(@tlt↓d, @tlvid, @t↓d, @curdate, @userid, @curdate, @userid)
  set @t↓d = @t↓d + 1
end

set @tpid = 2 --recommended
set @tlptpid = @tlptpid + 1
insert into technicalline_prerequisitetechnicalpackage
values(@tlptpid, @tlvid, @tpid, @curdate, @userid, @curdate, @userid)
set @tpid = 8
```

```
set @tlptpid = @tlptpid + 1
insert into technicalline_prerequisitetechnicalpackage
values(@tlptpid, @tlvid, @tpid, @curdate, @userid, @curdate, @userid)
set @tpid = 9
set @tlptpid = @tlptpid + 1
insert into technicalline_prerequisitetechnicalpackage
values(@tlptpid, @tlvid, @tpid, @curdate, @userid, @curdate, @userid)
------------------------------------------------------------------------
set @pid = 'bce9eddf-0ed6-4111-adef-88ac9f256895'
set @da = 'Elektro'
set @en = 'Electrical and Electronic Engineering'

set @tlid = @tlid + 1
insert into technicalline values (@tlid, @curdate, @userid, @curdate, @userid)

set @tlvid = @tlvid + 1
insert into technicallineversion
  values (@tlvid, @tlid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>', @pid,
@curdate, @userid, @curdate, @userid)

set @sid = 18
while @sid < 23
begin
  set @tlsid = @tlsid + 1
  insert into technicalline_specialization
  values(@tlsid, @tlvid, @sid, @curdate, @userid, @curdate, @userid)
  set @sid = @sid + 1
end

set @t↓d = 20
while @t↓d < 25
begin
  set @tlt↓d = @tlt↓d + 1
  insert into technicalline_technical↓eld
  values(@tlt↓d, @tlvid, @t↓d, @curdate, @userid, @curdate, @userid)
  set @t↓d = @t↓d + 1
end

set @tpid = 4 --recommended
set @tlptpid = @tlptpid + 1
insert into technicalline_prerequisitetechnicalpackage
values(@tlptpid, @tlvid, @tpid, @curdate, @userid, @curdate, @userid)
set @tpid = 6
set @tlptpid = @tlptpid + 1
insert into technicalline_prerequisitetechnicalpackage
values(@tlptpid, @tlvid, @tpid, @curdate, @userid, @curdate, @userid)
set @tpid = 10
set @tlptpid = @tlptpid + 1
insert into technicalline_prerequisitetechnicalpackage
values(@tlptpid, @tlvid, @tpid, @curdate, @userid, @curdate, @userid)
------------------------------------------------------------------------
set @pid = '720c1413-5fc1-48ac-bb67-ffa9f3f4cc3b'
set @da = 'Energi'
set @en = 'Energy'

set @tlid = @tlid + 1
insert into technicalline values (@tlid, @curdate, @userid, @curdate, @userid)

set @tlvid = @tlvid + 1
insert into technicallineversion
  values (@tlvid, @tlid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>', @pid,
@curdate, @userid, @curdate, @userid)

set @t↓d = 25
while @t↓d < 29
begin
  set @tlt↓d = @tlt↓d + 1
  insert into technicalline_technical↓eld
  values(@tlt↓d, @tlvid, @t↓d, @curdate, @userid, @curdate, @userid)
  set @t↓d = @t↓d + 1
end

--no specializations yet

set @tpid = 2
set @tlptpid = @tlptpid + 1
insert into technicalline_prerequisitetechnicalpackage
values(@tlptpid, @tlvid, @tpid, @curdate, @userid, @curdate, @userid)
set @tpid = 4
```

```
set @tlptpid = @tlptpid + 1
insert into technicalline_prerequisitetechnicalpackage
values(@tlptpid, @tlvid, @tpid, @curdate, @userid, @curdate, @userid)
set @tpid = 5
set @tlptpid = @tlptpid + 1
insert into technicalline_prerequisitetechnicalpackage
values(@tlptpid, @tlvid, @tpid, @curdate, @userid, @curdate, @userid)
set @tpid = 8
set @tlptpid = @tlptpid + 1
insert into technicalline_prerequisitetechnicalpackage
values(@tlptpid, @tlvid, @tpid, @curdate, @userid, @curdate, @userid)
set @tpid = 10
set @tlptpid = @tlptpid + 1
insert into technicalline_prerequisitetechnicalpackage
values(@tlptpid, @tlvid, @tpid, @curdate, @userid, @curdate, @userid)
------------------------------------------------------------------------
set @pid = 'f3ea98c4-1811-4c65-911a-e1217db87271'
set @da = 'Industriel produktion'
set @en = 'Manufacturing and Management'

set @tlid = @tlid + 1
insert into technicalline values (@tlid, @curdate, @userid, @curdate, @userid)

set @tlvid = @tlvid + 1
insert into technicallineversion
  values (@tlvid, @tlid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>', @pid,
@curdate, @userid, @curdate, @userid)

set @sid = 23
while @sid < 28
begin
  set @tlsid = @tlsid + 1
  insert into technicalline_specialization
  values(@tlsid, @tlvid, @sid, @curdate, @userid, @curdate, @userid)
  set @sid = @sid + 1
end

set @t↓d = 29
while @t↓d < 32
begin
  set @tlt↓d = @tlt↓d + 1
  insert into technicalline_technical↓eld
  values(@tlt↓d, @tlvid, @t↓d, @curdate, @userid, @curdate, @userid)
  set @t↓d = @t↓d + 1
end

set @tpid = 2
set @tlptpid = @tlptpid + 1
insert into technicalline_prerequisitetechnicalpackage
values(@tlptpid, @tlvid, @tpid, @curdate, @userid, @curdate, @userid)
set @tpid = 6
set @tlptpid = @tlptpid + 1
insert into technicalline_prerequisitetechnicalpackage
values(@tlptpid, @tlvid, @tpid, @curdate, @userid, @curdate, @userid)
set @tpid = 8 --recommended
set @tlptpid = @tlptpid + 1
insert into technicalline_prerequisitetechnicalpackage
values(@tlptpid, @tlvid, @tpid, @curdate, @userid, @curdate, @userid)
set @tpid = 9
set @tlptpid = @tlptpid + 1
insert into technicalline_prerequisitetechnicalpackage
values(@tlptpid, @tlvid, @tpid, @curdate, @userid, @curdate, @userid)
set @tpid = 10
set @tlptpid = @tlptpid + 1
insert into technicalline_prerequisitetechnicalpackage
values(@tlptpid, @tlvid, @tpid, @curdate, @userid, @curdate, @userid)
------------------------------------------------------------------------
set @pid = '575f214c-5ced-4d8e-a2d0-53ec109fc770'
set @da = 'Informatikretningen'
set @en = 'Informatics'

set @tlid = @tlid + 1
insert into technicalline values (@tlid, @curdate, @userid, @curdate, @userid)

set @tlvid = @tlvid + 1
insert into technicallineversion
  values (@tlvid, @tlid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>', @pid,
@curdate, @userid, @curdate, @userid)
```

```
set @sid = 28
while @sid < 31
begin
  set @tlsid = @tlsid + 1
  insert into technicalline_specialization
  values(@tlsid, @tlvid, @sid, @curdate, @userid, @curdate, @userid)
  set @sid = @sid + 1
end

set @t_d = 32
while @t_d < 36
begin
  set @tlt_d = @tlt_d + 1
  insert into technicalline_technical_eld
  values(@tlt_d, @tlvid, @t_d, @curdate, @userid, @curdate, @userid)
  set @t_d = @t_d + 1
end

set @tpid = 6
set @tlptpid = @tlptpid + 1
insert into technicalline_prerequisitetechnicalpackage
values(@tlptpid, @tlvid, @tpid, @curdate, @userid, @curdate, @userid)
set @tlptpcid = @tlptpcid + 1
insert into technicalline_prerequisitetechnicalpackagecourse
values(@tlptpcid, @tlptpid, null, '02130', @curdate, @userid, @curdate, @userid)
set @tlptpcid = @tlptpcid + 1
insert into technicalline_prerequisitetechnicalpackagecourse
values(@tlptpcid, @tlptpid, null, '02140', @curdate, @userid, @curdate, @userid)

set @tpid = 4
set @tlptpid = @tlptpid + 1
insert into technicalline_prerequisitetechnicalpackage
values(@tlptpid, @tlvid, @tpid, @curdate, @userid, @curdate, @userid)
set @tlptpcid = @tlptpcid + 1
insert into technicalline_prerequisitetechnicalpackagecourse
values(@tlptpcid, @tlptpid, null, '02130', @curdate, @userid, @curdate, @userid)
set @tlptpcid = @tlptpcid + 1
insert into technicalline_prerequisitetechnicalpackagecourse
values(@tlptpcid, @tlptpid, null, '02140', @curdate, @userid, @curdate, @userid)
set @tlptpcid = @tlptpcid + 1
insert into technicalline_prerequisitetechnicalpackagecourse
values(@tlptpcid, @tlptpid, null, '02100', @curdate, @userid, @curdate, @userid)
set @tlptpcid = @tlptpcid + 1
insert into technicalline_prerequisitetechnicalpackagecourse
values(@tlptpcid, @tlptpid, null, '02105', @curdate, @userid, @curdate, @userid)
set @tlptpcid = @tlptpcid + 1
insert into technicalline_prerequisitetechnicalpackagecourse
values(@tlptpcid, @tlptpid, null, '02110', @curdate, @userid, @curdate, @userid)
-------------------------------------------------------------------------------
set @pid = 'df49fff5-e6f2-4610-9f63-b6e5e8b0e522'
set @da = 'Kemi og kemiteknik'
set @en = 'Applied Chemistry and Chemical Engineering'

set @tlid = @tlid + 1
insert into technicalline values (@tlid, @curdate, @userid, @curdate, @userid)

set @tlvid = @tlvid + 1
insert into technicallineversion
  values (@tlvid, @tlid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>', @pid,
@curdate, @userid, @curdate, @userid)

set @sid = 31
while @sid < 35
begin
  set @tlsid = @tlsid + 1
  insert into technicalline_specialization
  values(@tlsid, @tlvid, @sid, @curdate, @userid, @curdate, @userid)
  set @sid = @sid + 1
end

set @t_d = 36
while @t_d < 40
begin
  set @tlt_d = @tlt_d + 1
  insert into technicalline_technical_eld
  values(@tlt_d, @tlvid, @t_d, @curdate, @userid, @curdate, @userid)
  set @t_d = @t_d + 1
end

set @tpid = 1
set @tlptpid = @tlptpid + 1
insert into technicalline_prerequisitetechnicalpackage
values(@tlptpid, @tlvid, @tpid, @curdate, @userid, @curdate, @userid)
set @tpid = 7
set @tlptpid = @tlptpid + 1
insert into technicalline_prerequisitetechnicalpackage
values(@tlptpid, @tlvid, @tpid, @curdate, @userid, @curdate, @userid)

set @tlpcid = @tlpcid + 1
```

```
insert into technicalline_prerequisitecourse
values(@tlpcid, @tlvid, null, '26220', @curdate, @userid, @curdate, @userid)
set @tlpcid = @tlpcid + 1
insert into technicalline_prerequisitecourse
values(@tlpcid, @tlvid, null, '26300', @curdate, @userid, @curdate, @userid)
set @tlpcid = @tlpcid + 1
insert into technicalline_prerequisitecourse
values(@tlpcid, @tlvid, null, '26410', @curdate, @userid, @curdate, @userid)
set @tlpcid = @tlpcid + 1
insert into technicalline_prerequisitecourse
values(@tlpcid, @tlvid, null, '28120', @curdate, @userid, @curdate, @userid)
set @tlpcid = @tlpcid + 1
insert into technicalline_prerequisitecourse
values(@tlpcid, @tlvid, null, '28260', @curdate, @userid, @curdate, @userid)
set @tlpcid = @tlpcid + 1
insert into technicalline_prerequisitecourse
values(@tlpcid, @tlvid, null, '42110', @curdate, @userid, @curdate, @userid)
------------------------------------------------------------------------
set @pid = '13afba8b-93a9-4e4f-b07b-4349a46515282'
set @da = 'Konstruktion og mekanik'
set @en = 'Engineering Design and Applied Mechanics'

set @tlid = @tlid + 1
insert into technicalline values (@tlid, @curdate, @userid, @curdate, @userid)

set @tlvid = @tlvid + 1
insert into technicallineversion
  values (@tlvid, @tlid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>', @pid,
@curdate, @userid, @curdate, @userid)

set @sid = 35
while @sid < 39
begin
  set @tlsid = @tlsid + 1
  insert into technicalline_specialization
  values(@tlsid, @tlvid, @sid, @curdate, @userid, @curdate, @userid)
  set @sid = @sid + 1
end

set @t⌊d = 40
while @t⌊d < 44
begin
  set @tlt⌊d = @tlt⌊d + 1
  insert into technicalline_technical⌊eld
  values(@tlt⌊d, @tlvid, @t⌊d, @curdate, @userid, @curdate, @userid)
  set @t⌊d = @t⌊d + 1
end

set @tpid = 2
set @tlptpid = @tlptpid + 1
insert into technicalline_prerequisitetechnicalpackage
values(@tlptpid, @tlvid, @tpid, @curdate, @userid, @curdate, @userid)
set @tpid = 4
set @tlptpid = @tlptpid + 1
insert into technicalline_prerequisitetechnicalpackage
values(@tlptpid, @tlvid, @tpid, @curdate, @userid, @curdate, @userid)
set @tpid = 5
set @tlptpid = @tlptpid + 1
insert into technicalline_prerequisitetechnicalpackage
values(@tlptpid, @tlvid, @tpid, @curdate, @userid, @curdate, @userid)
set @tpid = 8 --recommended
set @tlptpid = @tlptpid + 1
insert into technicalline_prerequisitetechnicalpackage
values(@tlptpid, @tlvid, @tpid, @curdate, @userid, @curdate, @userid)
set @tpid = 10
set @tlptpid = @tlptpid + 1
insert into technicalline_prerequisitetechnicalpackage
values(@tlptpid, @tlvid, @tpid, @curdate, @userid, @curdate, @userid)
------------------------------------------------------------------------
set @pid = '896e1da4-eb3f-471e-8938-ffa550731c9d'
set @da = 'Materialeteknologi'
set @en = 'Material Technology'

set @tlid = @tlid + 1
insert into technicalline values (@tlid, @curdate, @userid, @curdate, @userid)

set @tlvid = @tlvid + 1
insert into technicallineversion
  values (@tlvid, @tlid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>', @pid,
```

```
@curdate, @userid, @curdate, @userid)

set @t↓d = 44
while @t↓d < 49
begin
  set @tlt↓d = @tlt↓d + 1
  insert into technicalline_technical↓eld
  values(@tlt↓d, @tlvid, @t↓d, @curdate, @userid, @curdate, @userid)
  set @t↓d = @t↓d + 1
end

set @tpid = 2
set @tlptpid = @tlptpid + 1
insert into technicalline_prerequisitetechnicalpackage
values(@tlptpid, @tlvid, @tpid, @curdate, @userid, @curdate, @userid)
set @tpid = 7
set @tlptpid = @tlptpid + 1
insert into technicalline_prerequisitetechnicalpackage
values(@tlptpid, @tlvid, @tpid, @curdate, @userid, @curdate, @userid)
set @tpid = 8
set @tlptpid = @tlptpid + 1
insert into technicalline_prerequisitetechnicalpackage
values(@tlptpid, @tlvid, @tpid, @curdate, @userid, @curdate, @userid)
set @tpid = 10
set @tlptpid = @tlptpid + 1
insert into technicalline_prerequisitetechnicalpackage
values(@tlptpid, @tlvid, @tpid, @curdate, @userid, @curdate, @userid)

set @tlpcid = @tlpcid + 1
insert into technicalline_prerequisitecourse
values(@tlpcid, @tlvid, null, '42110', @curdate, @userid, @curdate, @userid)
------------------------------------------------------------------------
set @pid = '305a078e-6c54-40cc-a8b6-8202f2422cbb'
set @da = 'Miljø'
set @en = 'Environmental Engineering'

set @tlid = @tlid + 1
insert into technicalline values (@tlid, @curdate, @userid, @curdate, @userid)

set @tlvid = @tlvid + 1
insert into technicallineversion
  values (@tlvid, @tlid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>', @pid,
@curdate, @userid, @curdate, @userid)

set @sid = 39
while @sid < 44
begin
  set @tlsid = @tlsid + 1
  insert into technicalline_specialization
  values(@tlsid, @tlvid, @sid, @curdate, @userid, @curdate, @userid)
  set @sid = @sid + 1
end

set @t↓d = 49
while @t↓d < 54
begin
  set @tlt↓d = @tlt↓d + 1
  insert into technicalline_technical↓eld
  values(@tlt↓d, @tlvid, @t↓d, @curdate, @userid, @curdate, @userid)
  set @t↓d = @t↓d + 1
end

set @tpid = 1
set @tlptpid = @tlptpid + 1
insert into technicalline_prerequisitetechnicalpackage
values(@tlptpid, @tlvid, @tpid, @curdate, @userid, @curdate, @userid)
set @tpid = 2
set @tlptpid = @tlptpid + 1
insert into technicalline_prerequisitetechnicalpackage
values(@tlptpid, @tlvid, @tpid, @curdate, @userid, @curdate, @userid)
set @tpid = 7
set @tlptpid = @tlptpid + 1
insert into technicalline_prerequisitetechnicalpackage
values(@tlptpid, @tlvid, @tpid, @curdate, @userid, @curdate, @userid)
set @tpid = 9 --recommended
set @tlptpid = @tlptpid + 1
insert into technicalline_prerequisitetechnicalpackage
values(@tlptpid, @tlvid, @tpid, @curdate, @userid, @curdate, @userid)
------------------------------------------------------------------------
set @pid = 'a7a395ef-56b5-4753-9b70-9976cd9aafd0'
set @da = 'Planlægning og ledelse'
set @en = 'Planning and Management'

set @tlid = @tlid + 1
insert into technicalline values (@tlid, @curdate, @userid, @curdate, @userid)

set @tlvid = @tlvid + 1
insert into technicallineversion
  values (@tlvid, @tlid, @version,
```

```
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>', @pid,
@curdate, @userid, @curdate, @userid)

--links to specializations copied from environmental engineering but not updated!

set @t↓d = 54
while @t↓d < 57
begin
  set @tlt↓d = @tlt↓d + 1
  insert into technicalline_technical↓eld
  values(@tlt↓d, @tlvid, @t↓d, @curdate, @userid, @curdate, @userid)
  set @t↓d = @t↓d + 1
end

--prerequisites under construction
-------------------------------------------------------------------------------
set @pid = '2932da0b-621e-4f8c-b7da-b318f57d2ff2'
set @da = 'Teknisk fysik'
set @en = 'Engineering Physics'

set @tlid = @tlid + 1
insert into technicalline values (@tlid, @curdate, @userid, @curdate, @userid)

set @tlvid = @tlvid + 1
insert into technicallineversion
  values (@tlvid, @tlid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + '</value>
    </culture>
  </cultures>', @pid,
@curdate, @userid, @curdate, @userid)

set @sid = 44
while @sid < 47
begin
  set @tlsid = @tlsid + 1
  insert into technicalline_specialization
  values(@tlsid, @tlvid, @sid, @curdate, @userid, @curdate, @userid)
  set @sid = @sid + 1
end

set @t↓d = 57
while @t↓d < 60
begin
  set @tlt↓d = @tlt↓d + 1
  insert into technicalline_technical↓eld
  values(@tlt↓d, @tlvid, @t↓d, @curdate, @userid, @curdate, @userid)
  set @t↓d = @t↓d + 1
end

set @tpid = 10
set @tlptpid = @tlptpid + 1
insert into technicalline_prerequisitetechnicalpackage
values(@tlptpid, @tlvid, @tpid, @curdate, @userid, @curdate, @userid)
set @tlpcid = @tlpcid + 1
insert into technicalline_prerequisitecourse
values(@tlpcid, @tlvid, null, '10111', @curdate, @userid, @curdate, @userid)
set @tlpcid = @tlpcid + 1
insert into technicalline_prerequisitecourse
values(@tlpcid, @tlvid, null, '10120', @curdate, @userid, @curdate, @userid)
set @tlpcid = @tlpcid + 1
insert into technicalline_prerequisitecourse
values(@tlpcid, @tlvid, null, '10300', @curdate, @userid, @curdate, @userid)
set @tlpcid = @tlpcid + 1
insert into technicalline_prerequisitecourse
values(@tlpcid, @tlvid, null, '10370', @curdate, @userid, @curdate, @userid)
-------------------------------------------------------------------------------
set @pid = '5a3a437c-834e-41fd-8861-e867b74bf62c'
set @da = 'Telekommunikation'
set @en = 'Telecommunications'

set @tlid = @tlid + 1
insert into technicalline values (@tlid, @curdate, @userid, @curdate, @userid)

set @tlvid = @tlvid + 1
insert into technicallineversion
  values (@tlvid, @tlid, @version,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + '</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
```

```
      <value>' + @en + '</value>
    </culture>
  </cultures>', @pid,
@curdate, @userid, @curdate, @userid)

set @sid = 47
while @sid < 52
begin
  set @tlsid = @tlsid + 1
  insert into technicalline_specialization
  values(@tlsid, @tlvid, @sid, @curdate, @userid, @curdate, @userid)
  set @sid = @sid + 1
end

set @t↓d = 60
while @t↓d < 63
begin
  set @tlt↓d = @tlt↓d + 1
  insert into technicalline_technical↓eld
  values(@tlt↓d, @tlvid, @t↓d, @curdate, @userid, @curdate, @userid)
  set @t↓d = @t↓d + 1
end

set @tpid = 4
set @tlptpid = @tlptpid + 1
insert into technicalline_prerequisitetechnicalpackage
values(@tlptpid, @tlvid, @tpid, @curdate, @userid, @curdate, @userid)
set @tpid = 6
set @tlptpid = @tlptpid + 1
insert into technicalline_prerequisitetechnicalpackage
values(@tlptpid, @tlvid, @tpid, @curdate, @userid, @curdate, @userid)
set @tpid = 10
set @tlptpid = @tlptpid + 1
insert into technicalline_prerequisitetechnicalpackage
values(@tlptpid, @tlvid, @tpid, @curdate, @userid, @curdate, @userid)
```

3.21 TechnicalPackage

```
------------------------------------------------------------------------
--Common declarations
------------------------------------------------------------------------
declare @userid uniqueidenti↓er
declare @curdate datetime
declare @da varchar(255), @en varchar(255), @nm varchar(20)
declare @tpid int, @tpvid int, @version int, @endYear int, @year int
set @userid = 'D36DD218-AE59-4046-977C-659E4EB92C92'
set @curdate = getdate()
set @tpid = 0
set @tpvid = 0
set @version = 0
set @endYear = 0
set @year = 0
------------------------------------------------------------------------
--Biotechnology
------------------------------------------------------------------------
set @tpid = @tpid + 1
set @da = 'Bioteknologifagpakken'
set @en = 'The Biotechnology Technical Package'
set @nm = '98181'
insert into technicalpackage values (@tpid, @curdate, @userid, @curdate, @userid)

set @year = 2000
set @endYear = 2010

while @year <= @endYear
begin
  set @tpvid = @tpvid + 1
  set @version = @version + 1

  insert into technicalpackageversion
    values (@tpvid, @tpid, @version, @nm,
    '<cultures>
      <culture>
        <cultureID>da-DK</cultureID>
        <value>' + @da + ' ' + convert(varchar,@year) + '</value>
      </culture>
      <culture>
        <cultureID>en-GB</cultureID>
        <value>' + @en + ' ' + convert(varchar,@year) + '</value>
      </culture>
    </cultures>',
  @curdate, @userid, @curdate, @userid)
  set @year = @year + 1
end
------------------------------------------------------------------------
--Construction Engineering
------------------------------------------------------------------------
set @tpid = @tpid + 1
insert into technicalpackage values (@tpid, @curdate, @userid, @curdate, @userid)

set @tpvid = @tpvid + 1
set @version = 1
set @da = 'Bygningsfagpakken (Fysik- eller Kemi-A)'
set @en = 'The Construction Engineering Technical Package (Physics- or Chemistry-A)'
```

```
set @nm = '98023'

insert into technicalpackageversion
  values (@tpvid, @tpid, @version, @nm,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + ' 1999</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + ' 1999</value>
    </culture>
  </cultures>',
  @curdate, @userid, @curdate, @userid)

set @tpvid = @tpvid + 1
set @version = @version + 1
set @da = 'Bygningsfagpakken (Fysik-B)'
set @en = 'The Construction Engineering Technical Package (Physics-B)'
set @nm = '98021'

insert into technicalpackageversion
  values (@tpvid, @tpid, @version, @nm,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + ' 1999</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + ' 1999</value>
    </culture>
  </cultures>',
  @curdate, @userid, @curdate, @userid)

set @da = 'Bygningsfagpakken'
set @en = 'The Construction Engineering Technical Package'
set @nm = '98121'

set @year = 2000
set @endYear = 2010

while @year <= @endYear
begin
  set @tpvid = @tpvid + 1
  set @version = @version + 1

  insert into technicalpackageversion
    values (@tpvid, @tpid, @version, @nm,
    '<cultures>
      <culture>
        <cultureID>da-DK</cultureID>
        <value>' + @da + ' ' + convert(varchar,@year) + '</value>
      </culture>
      <culture>
        <cultureID>en-GB</cultureID>
        <value>' + @en + ' ' + convert(varchar,@year) + '</value>
      </culture>
    </cultures>',
  @curdate, @userid, @curdate, @userid)
  set @year = @year + 1
end
------------------------------------------------------------------------
--Design and innovation
------------------------------------------------------------------------
set @tpid = @tpid + 1
insert into technicalpackage values (@tpid, @curdate, @userid, @curdate, @userid)

set @version = 0
set @da = 'Design &amp; innovationsfagpakken'
set @en = 'The Design &amp; Innovation Technical Package'
set @nm = '98191'

set @year = 2002
set @endYear = 2010

while @year <= @endYear
begin
  set @tpvid = @tpvid + 1
  set @version = @version + 1

  insert into technicalpackageversion
    values (@tpvid, @tpid, @version, @nm,
    '<cultures>
      <culture>
        <cultureID>da-DK</cultureID>
        <value>' + @da + ' ' + convert(varchar,@year) + '</value>
      </culture>
      <culture>
        <cultureID>en-GB</cultureID>
        <value>' + @en + ' ' + convert(varchar,@year) + '</value>
      </culture>
    </cultures>',
  @curdate, @userid, @curdate, @userid)
  set @year = @year + 1
end
------------------------------------------------------------------------
--Electrical Engineering
```

```
------------------------------------------------------------------------
set @tpid = @tpid + 1
insert into technicalpackage values (@tpid, @curdate, @userid, @curdate, @userid)

set @tpvid = @tpvid + 1
set @version = 1
set @da = 'Elektrofagpakken'
set @en = 'The Electrical Engineering Technical Package'
set @nm = '98115'

insert into technicalpackageversion
  values (@tpvid, @tpid, @version, @nm,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + ' 1999</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + ' 1999</value>
    </culture>
  </cultures>',
  @curdate, @userid, @curdate, @userid)

set @nm = '98111'
set @year = 2000
set @endYear = 2010

while @year <= @endYear
begin
  set @tpvid = @tpvid + 1
  set @version = @version + 1

  insert into technicalpackageversion
    values (@tpvid, @tpid, @version, @nm,
    '<cultures>
      <culture>
        <cultureID>da-DK</cultureID>
        <value>' + @da + ' ' + convert(varchar,@year) + '</value>
      </culture>
      <culture>
        <cultureID>en-GB</cultureID>
        <value>' + @en + ' ' + convert(varchar,@year) + '</value>
      </culture>
    </cultures>',
  @curdate, @userid, @curdate, @userid)
  set @year = @year + 1
end
------------------------------------------------------------------------
--Energy
------------------------------------------------------------------------
set @tpid = @tpid + 1
insert into technicalpackage values (@tpid, @curdate, @userid, @curdate, @userid)

set @tpvid = @tpvid + 1
set @version = 1
set @da = 'Energifagpakken (Fysik- eller Kemi-A)'
set @en = 'The Energy Technical Package (Physics- or Chemistry-A)'
set @nm = '98053'

insert into technicalpackageversion
  values (@tpvid, @tpid, @version, @nm,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + ' 1999</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + ' 1999</value>
    </culture>
  </cultures>',
  @curdate, @userid, @curdate, @userid)

set @tpvid = @tpvid + 1
set @version = @version + 1
set @da = 'Energifagpakken (Fysik-B)'
set @en = 'The Energy Technical Package (Physics-B)'
set @nm = '98051'

insert into technicalpackageversion
  values (@tpvid, @tpid, @version, @nm,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + ' 1999</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + ' 1999</value>
    </culture>
  </cultures>',
  @curdate, @userid, @curdate, @userid)

set @da = 'Energifagpakken'
set @en = 'The Energy Technical Package'
set @nm = '98151'

set @year = 2000
```

```
set @endYear = 2010

while @year <= @endYear
begin
  set @tpvid = @tpvid + 1
  set @version = @version + 1

  insert into technicalpackageversion
    values (@tpvid, @tpid, @version, @nm,
    '<cultures>
      <culture>
        <cultureID>da-DK</cultureID>
        <value>' + @da + ' ' + convert(varchar,@year) + '</value>
      </culture>
      <culture>
        <cultureID>en-GB</cultureID>
        <value>' + @en + ' ' + convert(varchar,@year) + '</value>
      </culture>
    </cultures>',
  @curdate, @userid, @curdate, @userid)
  set @year = @year + 1
end
-------------------------------------------------------------------------------
--Information Technology
-------------------------------------------------------------------------------
set @tpid = @tpid + 1
insert into technicalpackage values (@tpid, @curdate, @userid, @curdate, @userid)

set @tpvid = @tpvid + 1
set @version = 1
set @da = 'Informatikfagpakken'
set @en = 'The Information Technology Technical Package'
set @nm = '98063'

insert into technicalpackageversion
  values (@tpvid, @tpid, @version, @nm,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + ' 1999</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + ' 1999</value>
    </culture>
  </cultures>',
  @curdate, @userid, @curdate, @userid)

set @nm = '98161'
set @year = 2000
set @endYear = 2010

while @year <= @endYear
begin
  set @tpvid = @tpvid + 1
  set @version = @version + 1

  insert into technicalpackageversion
    values (@tpvid, @tpid, @version, @nm,
    '<cultures>
      <culture>
        <cultureID>da-DK</cultureID>
        <value>' + @da + ' ' + convert(varchar,@year) + '</value>
      </culture>
      <culture>
        <cultureID>en-GB</cultureID>
        <value>' + @en + ' ' + convert(varchar,@year) + '</value>
      </culture>
    </cultures>',
  @curdate, @userid, @curdate, @userid)
  set @year = @year + 1
end
-------------------------------------------------------------------------------
--Chemical Engineering
-------------------------------------------------------------------------------
set @tpid = @tpid + 1
insert into technicalpackage values (@tpid, @curdate, @userid, @curdate, @userid)

set @tpvid = @tpvid + 1
set @version = 1
set @da = 'Kemifagpakken (Kemi-A)'
set @en = 'The Chemical Engineering Technical Package (Chemistry-A)'
set @nm = '98003'

insert into technicalpackageversion
  values (@tpvid, @tpid, @version, @nm,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + ' 1999</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + ' 1999</value>
    </culture>
  </cultures>',
  @curdate, @userid, @curdate, @userid)

set @tpvid = @tpvid + 1
```

```
set @version = @version + 1
set @da = 'Kemifagpakken (Kemi-B)'
set @en = 'The Chemical Engineering Technical Package (Chemistry-B)'
set @nm = '98001'

insert into technicalpackageversion
  values (@tpvid, @tpid, @version, @nm,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + ' 1999</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + ' 1999</value>
    </culture>
  </cultures>',
  @curdate, @userid, @curdate, @userid)

set @da = 'Kemifagpakken'
set @en = 'The Chemical Engineering Technical Package'
set @nm = '98101'

set @year = 2000
set @endYear = 2010

while @year <= @endYear
begin
  set @tpvid = @tpvid + 1
  set @version = @version + 1

  insert into technicalpackageversion
    values (@tpvid, @tpid, @version, @nm,
    '<cultures>
      <culture>
        <cultureID>da-DK</cultureID>
        <value>' + @da + ' ' + convert(varchar,@year) + '</value>
      </culture>
      <culture>
        <cultureID>en-GB</cultureID>
        <value>' + @en + ' ' + convert(varchar,@year) + '</value>
      </culture>
    </cultures>',
  @curdate, @userid, @curdate, @userid)
  set @year = @year + 1
end
------------------------------------------------------------------------
--Mechanical Engineering
------------------------------------------------------------------------
set @tpid = @tpid + 1
insert into technicalpackage values (@tpid, @curdate, @userid, @curdate, @userid)

set @tpvid = @tpvid + 1
set @version = 1
set @da = 'Maskinfagpakken (Fysik- eller Kemi-A)'
set @en = 'The Mechanical Engineering Technical Package (Physics- or Chemistry-A)'
set @nm = '98033'

insert into technicalpackageversion
  values (@tpvid, @tpid, @version, @nm,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + ' 1999</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + ' 1999</value>
    </culture>
  </cultures>',
  @curdate, @userid, @curdate, @userid)

set @tpvid = @tpvid + 1
set @version = @version + 1
set @da = 'Maskinfagpakken (Fysik-B)'
set @en = 'The Mechanical Engineering Technical Package (Physics-B)'
set @nm = '98031'

insert into technicalpackageversion
  values (@tpvid, @tpid, @version, @nm,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + ' 1999</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + ' 1999</value>
    </culture>
  </cultures>',
  @curdate, @userid, @curdate, @userid)

set @da = 'Maskinfagpakken'
set @en = 'The Mechanical Engineering Technical Package'
set @nm = '98131'

set @year = 2000
set @endYear = 2010
```

```
while @year <= @endYear
begin
  set @tpvid = @tpvid + 1
  set @version = @version + 1

  insert into technicalpackageversion
    values (@tpvid, @tpid, @version, @nm,
    '<cultures>
      <culture>
        <cultureID>da-DK</cultureID>
        <value>' + @da + ' ' + convert(varchar,@year) + '</value>
      </culture>
      <culture>
        <cultureID>en-GB</cultureID>
        <value>' + @en + ' ' + convert(varchar,@year) + '</value>
      </culture>
    </cultures>',
  @curdate, @userid, @curdate, @userid)
  set @year = @year + 1
end
-------------------------------------------------------------------------------
--Enviromental Engineering
-------------------------------------------------------------------------------
set @tpid = @tpid + 1
insert into technicalpackage values (@tpid, @curdate, @userid, @curdate, @userid)

set @tpvid = @tpvid + 1
set @version = 1
set @da = 'Miljøfagpakken (Kemi-B og -C)'
set @en = 'The Enviromental Engineering Technical Package (Chemistry-B and -C)'
set @nm = '98041'

insert into technicalpackageversion
  values (@tpvid, @tpid, @version, @nm,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + ' 1999</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + ' 1999</value>
    </culture>
  </cultures>',
  @curdate, @userid, @curdate, @userid)

set @tpvid = @tpvid + 1
set @version = @version + 1
set @da = 'Miljøfagpakken (Kemi-A)'
set @en = 'The Enviromental Engineering Technical Package (Chemistry-A)'
set @nm = '98043'

insert into technicalpackageversion
  values (@tpvid, @tpid, @version, @nm,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + ' 1999</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + ' 1999</value>
    </culture>
  </cultures>',
  @curdate, @userid, @curdate, @userid)

set @da = 'Miljøfagpakken'
set @en = 'The Enviromental Engineering Technical Package'
set @nm = '98141'

set @year = 2000
set @endYear = 2010

while @year <= @endYear
begin
  set @tpvid = @tpvid + 1
  set @version = @version + 1

  insert into technicalpackageversion
    values (@tpvid, @tpid, @version, @nm,
    '<cultures>
      <culture>
        <cultureID>da-DK</cultureID>
        <value>' + @da + ' ' + convert(varchar,@year) + '</value>
      </culture>
      <culture>
        <cultureID>en-GB</cultureID>
        <value>' + @en + ' ' + convert(varchar,@year) + '</value>
      </culture>
    </cultures>',
  @curdate, @userid, @curdate, @userid)
  set @year = @year + 1
end
-------------------------------------------------------------------------------
--Technical Physics
-------------------------------------------------------------------------------
set @tpid = @tpid + 1
insert into technicalpackage values (@tpid, @curdate, @userid, @curdate, @userid)
```

```
set @tpvid = @tpvid + 1
set @version = 1
set @da = 'Teknisk fysikfagpakken (Fysik- eller Kemi-A)'
set @en = 'The Technical Physics Technical Package (Physics- or Chemistry-A)'
set @nm = '98073'

insert into technicalpackageversion
  values (@tpvid, @tpid, @version, @nm,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + ' 1999</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + ' 1999</value>
    </culture>
  </cultures>',
  @curdate, @userid, @curdate, @userid)

set @tpvid = @tpvid + 1
set @version = @version + 1
set @da = 'Teknisk fysikfagpakken (Fysik-B)'
set @en = 'The Technical Physics Technical Package (Physics-B)'
set @nm = '98071'

insert into technicalpackageversion
  values (@tpvid, @tpid, @version, @nm,
  '<cultures>
    <culture>
      <cultureID>da-DK</cultureID>
      <value>' + @da + ' 1999</value>
    </culture>
    <culture>
      <cultureID>en-GB</cultureID>
      <value>' + @en + ' 1999</value>
    </culture>
  </cultures>',
  @curdate, @userid, @curdate, @userid)

set @da = 'Teknisk fysikfagpakken'
set @en = 'The Technical Physics Technical Package'
set @nm = '98171'

set @year = 2000
set @endYear = 2010

while @year <= @endYear
begin
  set @tpvid = @tpvid + 1
  set @version = @version + 1

  insert into technicalpackageversion
    values (@tpvid, @tpid, @version, @nm,
    '<cultures>
      <culture>
        <cultureID>da-DK</cultureID>
        <value>' + @da + ' ' + convert(varchar,@year) + '</value>
      </culture>
      <culture>
        <cultureID>en-GB</cultureID>
        <value>' + @en + ' ' + convert(varchar,@year) + '</value>
      </culture>
    </cultures>',
  @curdate, @userid, @curdate, @userid)
  set @year = @year + 1
end
------------------------------------------------------------------------
--Medication and Technology
------------------------------------------------------------------------
set @tpid = @tpid + 1
insert into technicalpackage values (@tpid, @curdate, @userid, @curdate, @userid)

set @version = 0
set @da = 'Medicin og teknologifagpakken'
set @en = 'The Medication and Technology Technical Package'
set @nm = '98000'

set @year = 2003
set @endYear = 2010

while @year <= @endYear
begin
  set @tpvid = @tpvid + 1
  set @version = @version + 1

  insert into technicalpackageversion
    values (@tpvid, @tpid, @version, @nm,
    '<cultures>
      <culture>
        <cultureID>da-DK</cultureID>
        <value>' + @da + ' ' + convert(varchar,@year) + '</value>
      </culture>
      <culture>
        <cultureID>en-GB</cultureID>
        <value>' + @en + ' ' + convert(varchar,@year) + '</value>
      </culture>
    </cultures>',
  @curdate, @userid, @curdate, @userid)
```

```
  set @year = @year + 1
end

-------------------------------------------------------------------------------
--Construction Engineering (Bachelor of Science)
-------------------------------------------------------------------------------
set @tpid = @tpid + 1
insert into technicalpackage values (@tpid, @curdate, @userid, @curdate, @userid)

set @version = 0
set @da = 'Bygning - februar'
set @en = 'Construction Engineering - February'
set @nm = ''

set @year = 2003
set @endYear = 2010

while @year <= @endYear
begin
  set @tpvid = @tpvid + 1
  set @version = @version + 1

  insert into technicalpackageversion
    values (@tpvid, @tpid, @version, @nm,
    '<cultures>
      <culture>
        <cultureID>da-DK</cultureID>
        <value>' + @da + ' ' + convert(varchar,@year) + '</value>
      </culture>
      <culture>
        <cultureID>en-GB</cultureID>
        <value>' + @en + ' ' + convert(varchar,@year) + '</value>
      </culture>
    </cultures>',
  @curdate, @userid, @curdate, @userid)
  set @year = @year + 1
end

set @tpid = @tpid + 1
insert into technicalpackage values (@tpid, @curdate, @userid, @curdate, @userid)

set @version = 0
set @da = 'Bygning - september'
set @en = 'Construction Engineering - September'
set @nm = ''

set @year = 2003
set @endYear = 2010

while @year <= @endYear
begin
  set @tpvid = @tpvid + 1
  set @version = @version + 1

  insert into technicalpackageversion
    values (@tpvid, @tpid, @version, @nm,
    '<cultures>
      <culture>
        <cultureID>da-DK</cultureID>
        <value>' + @da + ' ' + convert(varchar,@year) + '</value>
      </culture>
      <culture>
        <cultureID>en-GB</cultureID>
        <value>' + @en + ' ' + convert(varchar,@year) + '</value>
      </culture>
    </cultures>',
  @curdate, @userid, @curdate, @userid)
  set @year = @year + 1
end

-------------------------------------------------------------------------------
--Urban Planning and Construction (Bachelor of Science)
-------------------------------------------------------------------------------
set @tpid = @tpid + 1
insert into technicalpackage values (@tpid, @curdate, @userid, @curdate, @userid)

set @version = 0
set @da = 'By og bygning - september'
set @en = 'Urban Planning and Construction - September'
set @nm = ''

set @year = 2003
set @endYear = 2010

while @year <= @endYear
begin
  set @tpvid = @tpvid + 1
  set @version = @version + 1

  insert into technicalpackageversion
    values (@tpvid, @tpid, @version, @nm,
    '<cultures>
      <culture>
        <cultureID>da-DK</cultureID>
        <value>' + @da + ' ' + convert(varchar,@year) + '</value>
      </culture>
      <culture>
        <cultureID>en-GB</cultureID>
        <value>' + @en + ' ' + convert(varchar,@year) + '</value>
```

```
      </culture>
    </cultures>',
  @curdate, @userid, @curdate, @userid)
  set @year = @year + 1
end

-------------------------------------------------------------------------------
--Electrical Engineering (Bachelor of Science)
-------------------------------------------------------------------------------
set @tpid = @tpid + 1
insert into technicalpackage values (@tpid, @curdate, @userid, @curdate, @userid)

set @version = 0
set @da = 'Elektro - februar'
set @en = 'Electrical Engineering - February'
set @nm = ''

set @year = 2003
set @endYear = 2010

while @year <= @endYear
begin
  set @tpvid = @tpvid + 1
  set @version = @version + 1

  insert into technicalpackageversion
    values (@tpvid, @tpid, @version, @nm,
    '<cultures>
      <culture>
        <cultureID>da-DK</cultureID>
        <value>' + @da + ' ' + convert(varchar,@year) + '</value>
      </culture>
      <culture>
        <cultureID>en-GB</cultureID>
        <value>' + @en + ' ' + convert(varchar,@year) + '</value>
      </culture>
    </cultures>',
  @curdate, @userid, @curdate, @userid)
  set @year = @year + 1
end

set @tpid = @tpid + 1
insert into technicalpackage values (@tpid, @curdate, @userid, @curdate, @userid)

set @version = 0
set @da = 'Elektro - september'
set @en = 'Electrical Engineering - September'
set @nm = ''

set @year = 2003
set @endYear = 2010

while @year <= @endYear
begin
  set @tpvid = @tpvid + 1
  set @version = @version + 1

  insert into technicalpackageversion
    values (@tpvid, @tpid, @version, @nm,
    '<cultures>
      <culture>
        <cultureID>da-DK</cultureID>
        <value>' + @da + ' ' + convert(varchar,@year) + '</value>
      </culture>
      <culture>
        <cultureID>en-GB</cultureID>
        <value>' + @en + ' ' + convert(varchar,@year) + '</value>
      </culture>
    </cultures>',
  @curdate, @userid, @curdate, @userid)
  set @year = @year + 1
end

-------------------------------------------------------------------------------
--Chemical Engineering (Bachelor of Science)
-------------------------------------------------------------------------------
set @tpid = @tpid + 1
insert into technicalpackage values (@tpid, @curdate, @userid, @curdate, @userid)

set @version = 0
set @da = 'Kemi - februar'
set @en = 'Chemical Engineering - February'
set @nm = ''

set @year = 2003
set @endYear = 2010

while @year <= @endYear
begin
  set @tpvid = @tpvid + 1
  set @version = @version + 1

  insert into technicalpackageversion
    values (@tpvid, @tpid, @version, @nm,
    '<cultures>
      <culture>
        <cultureID>da-DK</cultureID>
        <value>' + @da + ' ' + convert(varchar,@year) + '</value>
      </culture>
```

```
      <culture>
        <cultureID>en-GB</cultureID>
        <value>' + @en + ' ' + convert(varchar,@year) + '</value>
      </culture>
    </cultures>',
  @curdate, @userid, @curdate, @userid)
  set @year = @year + 1
end

set @tpid = @tpid + 1
insert into technicalpackage values (@tpid, @curdate, @userid, @curdate, @userid)

set @version = 0
set @da = 'Kemi - september'
set @en = 'Chemical Engineering - September'
set @nm = ''

set @year = 2003
set @endYear = 2010

while @year <= @endYear
begin
  set @tpvid = @tpvid + 1
  set @version = @version + 1

  insert into technicalpackageversion
    values (@tpvid, @tpid, @version, @nm,
    '<cultures>
      <culture>
        <cultureID>da-DK</cultureID>
        <value>' + @da + ' ' + convert(varchar,@year) + '</value>
      </culture>
      <culture>
        <cultureID>en-GB</cultureID>
        <value>' + @en + ' ' + convert(varchar,@year) + '</value>
      </culture>
    </cultures>',
  @curdate, @userid, @curdate, @userid)
  set @year = @year + 1
end

------------------------------------------------------------------------
--Information Technology (Bachelor of Science)
------------------------------------------------------------------------
set @tpid = @tpid + 1
insert into technicalpackage values (@tpid, @curdate, @userid, @curdate, @userid)

set @version = 0
set @da = 'IT - februar'
set @en = 'IT - February'
set @nm = ''

set @year = 2003
set @endYear = 2010

while @year <= @endYear
begin
  set @tpvid = @tpvid + 1
  set @version = @version + 1

  insert into technicalpackageversion
    values (@tpvid, @tpid, @version, @nm,
    '<cultures>
      <culture>
        <cultureID>da-DK</cultureID>
        <value>' + @da + ' ' + convert(varchar,@year) + '</value>
      </culture>
      <culture>
        <cultureID>en-GB</cultureID>
        <value>' + @en + ' ' + convert(varchar,@year) + '</value>
      </culture>
    </cultures>',
  @curdate, @userid, @curdate, @userid)
  set @year = @year + 1
end

set @tpid = @tpid + 1
insert into technicalpackage values (@tpid, @curdate, @userid, @curdate, @userid)

set @version = 0
set @da = 'IT - september'
set @en = 'IT - September'
set @nm = ''

set @year = 2003
set @endYear = 2010

while @year <= @endYear
begin
  set @tpvid = @tpvid + 1
  set @version = @version + 1

  insert into technicalpackageversion
    values (@tpvid, @tpid, @version, @nm,
    '<cultures>
      <culture>
        <cultureID>da-DK</cultureID>
        <value>' + @da + ' ' + convert(varchar,@year) + '</value>
      </culture>
```

```
      <culture>
        <cultureID>en-GB</cultureID>
        <value>' + @en + ' ' + convert(varchar,@year) + '</value>
      </culture>
    </cultures>',
  @curdate, @userid, @curdate, @userid)
  set @year = @year + 1
end

-------------------------------------------------------------------------
--Mechanical Engineering 1 (Bachelor of Science)
-------------------------------------------------------------------------
set @tpid = @tpid + 1
insert into technicalpackage values (@tpid, @curdate, @userid, @curdate, @userid)

set @version = 0
set @da = 'Maskin (konstruktion) - februar'
set @en = 'Mechanical Engineering (construction) - February'
set @nm = ''

set @year = 2003
set @endYear = 2010

while @year <= @endYear
begin
  set @tpvid = @tpvid + 1
  set @version = @version + 1

  insert into technicalpackageversion
    values (@tpvid, @tpid, @version, @nm,
    '<cultures>
      <culture>
        <cultureID>da-DK</cultureID>
        <value>' + @da + ' ' + convert(varchar,@year) + '</value>
      </culture>
      <culture>
        <cultureID>en-GB</cultureID>
        <value>' + @en + ' ' + convert(varchar,@year) + '</value>
      </culture>
    </cultures>',
  @curdate, @userid, @curdate, @userid)
  set @year = @year + 1
end

set @tpid = @tpid + 1
insert into technicalpackage values (@tpid, @curdate, @userid, @curdate, @userid)

set @version = 0
set @da = 'Maskin (konstruktion) - september'
set @en = 'Mechanical Engineering (construction) - September'
set @nm = ''

set @year = 2003
set @endYear = 2010

while @year <= @endYear
begin
  set @tpvid = @tpvid + 1
  set @version = @version + 1

  insert into technicalpackageversion
    values (@tpvid, @tpid, @version, @nm,
    '<cultures>
      <culture>
        <cultureID>da-DK</cultureID>
        <value>' + @da + ' ' + convert(varchar,@year) + '</value>
      </culture>
      <culture>
        <cultureID>en-GB</cultureID>
        <value>' + @en + ' ' + convert(varchar,@year) + '</value>
      </culture>
    </cultures>',
  @curdate, @userid, @curdate, @userid)
  set @year = @year + 1
end

-------------------------------------------------------------------------
--Mechanical Engineering 2 (Bachelor of Science)
-------------------------------------------------------------------------
set @tpid = @tpid + 1
insert into technicalpackage values (@tpid, @curdate, @userid, @curdate, @userid)

set @version = 0
set @da = 'Maskin (konstruktion - skib) - februar'
set @en = 'Mechanical Engineering (construction - ship) - February'
set @nm = ''

set @year = 2003
set @endYear = 2010

while @year <= @endYear
begin
  set @tpvid = @tpvid + 1
  set @version = @version + 1

  insert into technicalpackageversion
    values (@tpvid, @tpid, @version, @nm,
    '<cultures>
      <culture>
```

```
        <cultureID>da-DK</cultureID>
        <value>' + @da + ' ' + convert(varchar,@year) + '</value>
      </culture>
      <culture>
        <cultureID>en-GB</cultureID>
        <value>' + @en + ' ' + convert(varchar,@year) + '</value>
      </culture>
    </cultures>',
  @curdate, @userid, @curdate, @userid)
  set @year = @year + 1
end

set @tpid = @tpid + 1
insert into technicalpackage values (@tpid, @curdate, @userid, @curdate, @userid)

set @version = 0
set @da = 'Maskin (konstruktion - skib) - september'
set @en = 'Mechanical Engineering (construction - ship) - September'
set @nm = ''

set @year = 2003
set @endYear = 2010

while @year <= @endYear
begin
  set @tpvid = @tpvid + 1
  set @version = @version + 1

  insert into technicalpackageversion
    values (@tpvid, @tpid, @version, @nm,
    '<cultures>
      <culture>
        <cultureID>da-DK</cultureID>
        <value>' + @da + ' ' + convert(varchar,@year) + '</value>
      </culture>
      <culture>
        <cultureID>en-GB</cultureID>
        <value>' + @en + ' ' + convert(varchar,@year) + '</value>
      </culture>
    </cultures>',
  @curdate, @userid, @curdate, @userid)
  set @year = @year + 1
end

-------------------------------------------------------------------------
--Mechanical Engineering 3 (Bachelor of Science)
-------------------------------------------------------------------------
set @tpid = @tpid + 1
insert into technicalpackage values (@tpid, @curdate, @userid, @curdate, @userid)

set @version = 0
set @da = 'Maskin (konstruktion - energi) - februar'
set @en = 'Mechanical Engineering (construction - energy) - February'
set @nm = ''

set @year = 2003
set @endYear = 2010

while @year <= @endYear
begin
  set @tpvid = @tpvid + 1
  set @version = @version + 1

  insert into technicalpackageversion
    values (@tpvid, @tpid, @version, @nm,
    '<cultures>
      <culture>
        <cultureID>da-DK</cultureID>
        <value>' + @da + ' ' + convert(varchar,@year) + '</value>
      </culture>
      <culture>
        <cultureID>en-GB</cultureID>
        <value>' + @en + ' ' + convert(varchar,@year) + '</value>
      </culture>
    </cultures>',
  @curdate, @userid, @curdate, @userid)
  set @year = @year + 1
end

set @tpid = @tpid + 1
insert into technicalpackage values (@tpid, @curdate, @userid, @curdate, @userid)

set @version = 0
set @da = 'Maskin (konstruktion - energi) - september'
set @en = 'Mechanical Engineering (construction - energy) - September'
set @nm = ''

set @year = 2003
set @endYear = 2010

while @year <= @endYear
begin
  set @tpvid = @tpvid + 1
  set @version = @version + 1

  insert into technicalpackageversion
    values (@tpvid, @tpid, @version, @nm,
    '<cultures>
      <culture>
```

```
        <cultureID>da-DK</cultureID>
        <value>' + @da + ' ' + convert(varchar,@year) + '</value>
      </culture>
      <culture>
        <cultureID>en-GB</cultureID>
        <value>' + @en + ' ' + convert(varchar,@year) + '</value>
      </culture>
    </cultures>',
  @curdate, @userid, @curdate, @userid)
  set @year = @year + 1
end

-------------------------------------------------------------------------
--Mechanical Engineering 4 (Bachelor of Science)
-------------------------------------------------------------------------
set @tpid = @tpid + 1
insert into technicalpackage values (@tpid, @curdate, @userid, @curdate, @userid)

set @version = 0
set @da = 'Maskin (maritim - konstruktion) - februar'
set @en = 'Mechanical Engineering (maritime - construction) - February'
set @nm = ''

set @year = 2003
set @endYear = 2010

while @year <= @endYear
begin
  set @tpvid = @tpvid + 1
  set @version = @version + 1

  insert into technicalpackageversion
    values (@tpvid, @tpid, @version, @nm,
    '<cultures>
      <culture>
        <cultureID>da-DK</cultureID>
        <value>' + @da + ' ' + convert(varchar,@year) + '</value>
      </culture>
      <culture>
        <cultureID>en-GB</cultureID>
        <value>' + @en + ' ' + convert(varchar,@year) + '</value>
      </culture>
    </cultures>',
  @curdate, @userid, @curdate, @userid)
  set @year = @year + 1
end

set @tpid = @tpid + 1
insert into technicalpackage values (@tpid, @curdate, @userid, @curdate, @userid)

set @version = 0
set @da = 'Maskin (maritim - konstruktion) - september'
set @en = 'Mechanical Engineering (maritime - construction) - September'
set @nm = ''

set @year = 2003
set @endYear = 2010

while @year <= @endYear
begin
  set @tpvid = @tpvid + 1
  set @version = @version + 1

  insert into technicalpackageversion
    values (@tpvid, @tpid, @version, @nm,
    '<cultures>
      <culture>
        <cultureID>da-DK</cultureID>
        <value>' + @da + ' ' + convert(varchar,@year) + '</value>
      </culture>
      <culture>
        <cultureID>en-GB</cultureID>
        <value>' + @en + ' ' + convert(varchar,@year) + '</value>
      </culture>
    </cultures>',
  @curdate, @userid, @curdate, @userid)
  set @year = @year + 1
end

-------------------------------------------------------------------------
--Mechanical Engineering 5 (Bachelor of Science)
-------------------------------------------------------------------------
set @tpid = @tpid + 1
insert into technicalpackage values (@tpid, @curdate, @userid, @curdate, @userid)

set @version = 0
set @da = 'Maskin (maritim) - februar'
set @en = 'Mechanical Engineering (maritime) - February'
set @nm = ''

set @year = 2003
set @endYear = 2010

while @year <= @endYear
begin
  set @tpvid = @tpvid + 1
  set @version = @version + 1

  insert into technicalpackageversion
```

```
    values (@tpvid, @tpid, @version, @nm,
    '<cultures>
      <culture>
        <cultureID>da-DK</cultureID>
        <value>' + @da + ' ' + convert(varchar,@year) + '</value>
      </culture>
      <culture>
        <cultureID>en-GB</cultureID>
        <value>' + @en + ' ' + convert(varchar,@year) + '</value>
      </culture>
    </cultures>',
  @curdate, @userid, @curdate, @userid)
  set @year = @year + 1
end

set @tpid = @tpid + 1
insert into technicalpackage values (@tpid, @curdate, @userid, @curdate, @userid)

set @version = 0
set @da = 'Maskin (maritim) - september'
set @en = 'Mechanical Engineering (maritime) - September'
set @nm = ''

set @year = 2003
set @endYear = 2010

while @year <= @endYear
begin
  set @tpvid = @tpvid + 1
  set @version = @version + 1

  insert into technicalpackageversion
    values (@tpvid, @tpid, @version, @nm,
    '<cultures>
      <culture>
        <cultureID>da-DK</cultureID>
        <value>' + @da + ' ' + convert(varchar,@year) + '</value>
      </culture>
      <culture>
        <cultureID>en-GB</cultureID>
        <value>' + @en + ' ' + convert(varchar,@year) + '</value>
      </culture>
    </cultures>',
  @curdate, @userid, @curdate, @userid)
  set @year = @year + 1
end

------------------------------------------------------------------------
--Mechanical Engineering 6 (Bachelor of Science)
------------------------------------------------------------------------
set @tpid = @tpid + 1
insert into technicalpackage values (@tpid, @curdate, @userid, @curdate, @userid)

set @version = 0
set @da = 'Maskin (maritim - energi) - februar'
set @en = 'Mechanical Engineering (maritime - energy) - February'
set @nm = ''

set @year = 2003
set @endYear = 2010

while @year <= @endYear
begin
  set @tpvid = @tpvid + 1
  set @version = @version + 1

  insert into technicalpackageversion
    values (@tpvid, @tpid, @version, @nm,
    '<cultures>
      <culture>
        <cultureID>da-DK</cultureID>
        <value>' + @da + ' ' + convert(varchar,@year) + '</value>
      </culture>
      <culture>
        <cultureID>en-GB</cultureID>
        <value>' + @en + ' ' + convert(varchar,@year) + '</value>
      </culture>
    </cultures>',
  @curdate, @userid, @curdate, @userid)
  set @year = @year + 1
end

set @tpid = @tpid + 1
insert into technicalpackage values (@tpid, @curdate, @userid, @curdate, @userid)

set @version = 0
set @da = 'Maskin (maritim - energi) - september'
set @en = 'Mechanical Engineering (maritime - energy) - September'
set @nm = ''

set @year = 2003
set @endYear = 2010

while @year <= @endYear
begin
  set @tpvid = @tpvid + 1
  set @version = @version + 1

  insert into technicalpackageversion
```

```
    values (@tpvid, @tpid, @version, @nm,
    '<cultures>
      <culture>
        <cultureID>da-DK</cultureID>
        <value>' + @da + ' ' + convert(varchar,@year) + '</value>
      </culture>
      <culture>
        <cultureID>en-GB</cultureID>
        <value>' + @en + ' ' + convert(varchar,@year) + '</value>
      </culture>
    </cultures>',
  @curdate, @userid, @curdate, @userid)
  set @year = @year + 1
end

-------------------------------------------------------------------------------
--Mechanical Engineering 7 (Bachelor of Science)
-------------------------------------------------------------------------------
set @tpid = @tpid + 1
insert into technicalpackage values (@tpid, @curdate, @userid, @curdate, @userid)

set @version = 0
set @da = 'Maskin (produktion og ledelse - proces) - februar'
set @en = 'Mechanical Engineering (production and management - process) - February'
set @nm = ''

set @year = 2003
set @endYear = 2010

while @year <= @endYear
begin
  set @tpvid = @tpvid + 1
  set @version = @version + 1

  insert into technicalpackageversion
    values (@tpvid, @tpid, @version, @nm,
    '<cultures>
      <culture>
        <cultureID>da-DK</cultureID>
        <value>' + @da + ' ' + convert(varchar,@year) + '</value>
      </culture>
      <culture>
        <cultureID>en-GB</cultureID>
        <value>' + @en + ' ' + convert(varchar,@year) + '</value>
      </culture>
    </cultures>',
  @curdate, @userid, @curdate, @userid)
  set @year = @year + 1
end

set @tpid = @tpid + 1
insert into technicalpackage values (@tpid, @curdate, @userid, @curdate, @userid)

set @version = 0
set @da = 'Maskin (produktion og ledelse - proces) - september'
set @en = 'Mechanical Engineering (production and management - process) - September'
set @nm = ''

set @year = 2003
set @endYear = 2010

while @year <= @endYear
begin
  set @tpvid = @tpvid + 1
  set @version = @version + 1

  insert into technicalpackageversion
    values (@tpvid, @tpid, @version, @nm,
    '<cultures>
      <culture>
        <cultureID>da-DK</cultureID>
        <value>' + @da + ' ' + convert(varchar,@year) + '</value>
      </culture>
      <culture>
        <cultureID>en-GB</cultureID>
        <value>' + @en + ' ' + convert(varchar,@year) + '</value>
      </culture>
    </cultures>',
  @curdate, @userid, @curdate, @userid)
  set @year = @year + 1
end

-------------------------------------------------------------------------------
--Mechanical Engineering 8 (Bachelor of Science)
-------------------------------------------------------------------------------
set @tpid = @tpid + 1
insert into technicalpackage values (@tpid, @curdate, @userid, @curdate, @userid)

set @version = 0
set @da = 'Maskin (produktion og ledelse - plast) - februar'
set @en = 'Mechanical Engineering (production and management - plastic) - February'
set @nm = ''

set @year = 2003
set @endYear = 2010

while @year <= @endYear
begin
  set @tpvid = @tpvid + 1
```

```
  set @version = @version + 1

  insert into technicalpackageversion
    values (@tpvid, @tpid, @version, @nm,
    '<cultures>
      <culture>
        <cultureID>da-DK</cultureID>
        <value>' + @da + ' ' + convert(varchar,@year) + '</value>
      </culture>
      <culture>
        <cultureID>en-GB</cultureID>
        <value>' + @en + ' ' + convert(varchar,@year) + '</value>
      </culture>
    </cultures>',
  @curdate, @userid, @curdate, @userid)
  set @year = @year + 1
end

set @tpid = @tpid + 1
insert into technicalpackage values (@tpid, @curdate, @userid, @curdate, @userid)

set @version = 0
set @da = 'Maskin (produktion og ledelse - plast) - september'
set @en = 'Mechanical Engineering (production and management - plastic) - September'
set @nm = ''

set @year = 2003
set @endYear = 2010

while @year <= @endYear
begin
  set @tpvid = @tpvid + 1
  set @version = @version + 1

  insert into technicalpackageversion
    values (@tpvid, @tpid, @version, @nm,
    '<cultures>
      <culture>
        <cultureID>da-DK</cultureID>
        <value>' + @da + ' ' + convert(varchar,@year) + '</value>
      </culture>
      <culture>
        <cultureID>en-GB</cultureID>
        <value>' + @en + ' ' + convert(varchar,@year) + '</value>
      </culture>
    </cultures>',
  @curdate, @userid, @curdate, @userid)
  set @year = @year + 1
end

------------------------------------------------------------------------------
--Mechanical Engineering 9 (Bachelor of Science)
------------------------------------------------------------------------------
set @tpid = @tpid + 1
insert into technicalpackage values (@tpid, @curdate, @userid, @curdate, @userid)

set @version = 0
set @da = 'Maskin (produktion og ledelse - styring og logistik) - februar'
set @en = 'Mechanical Engineering (production and management - control and logistics) - February'
set @nm = ''

set @year = 2003
set @endYear = 2010

while @year <= @endYear
begin
  set @tpvid = @tpvid + 1
  set @version = @version + 1

  insert into technicalpackageversion
    values (@tpvid, @tpid, @version, @nm,
    '<cultures>
      <culture>
        <cultureID>da-DK</cultureID>
        <value>' + @da + ' ' + convert(varchar,@year) + '</value>
      </culture>
      <culture>
        <cultureID>en-GB</cultureID>
        <value>' + @en + ' ' + convert(varchar,@year) + '</value>
      </culture>
    </cultures>',
  @curdate, @userid, @curdate, @userid)
  set @year = @year + 1
end

set @tpid = @tpid + 1
insert into technicalpackage values (@tpid, @curdate, @userid, @curdate, @userid)

set @version = 0
set @da = 'Maskin (produktion og ledelse - styring og logistik) - september'
set @en = 'Mechanical Engineering (production and management - control and logistics) - September'
set @nm = ''

set @year = 2003
set @endYear = 2010

while @year <= @endYear
begin
  set @tpvid = @tpvid + 1
```

```
  set @version = @version + 1

  insert into technicalpackageversion
    values (@tpvid, @tpid, @version, @nm,
    '<cultures>
      <culture>
        <cultureID>da-DK</cultureID>
        <value>' + @da + ' ' + convert(varchar,@year) + '</value>
      </culture>
      <culture>
        <cultureID>en-GB</cultureID>
        <value>' + @en + ' ' + convert(varchar,@year) + '</value>
      </culture>
    </cultures>',
  @curdate, @userid, @curdate, @userid)
  set @year = @year + 1
end

-------------------------------------------------------------------------------
--Mechanical Engineering 10 (Bachelor of Science)
-------------------------------------------------------------------------------
set @tpid = @tpid + 1
insert into technicalpackage values (@tpid, @curdate, @userid, @curdate, @userid)

set @version = 0
set @da = 'Maskin (produktion og ledelse - organisation) - februar'
set @en = 'Mechanical Engineering (production and management - organization) - February'
set @nm = ''

set @year = 2003
set @endYear = 2010

while @year <= @endYear
begin
  set @tpvid = @tpvid + 1
  set @version = @version + 1

  insert into technicalpackageversion
    values (@tpvid, @tpid, @version, @nm,
    '<cultures>
      <culture>
        <cultureID>da-DK</cultureID>
        <value>' + @da + ' ' + convert(varchar,@year) + '</value>
      </culture>
      <culture>
        <cultureID>en-GB</cultureID>
        <value>' + @en + ' ' + convert(varchar,@year) + '</value>
      </culture>
    </cultures>',
  @curdate, @userid, @curdate, @userid)
  set @year = @year + 1
end

set @tpid = @tpid + 1
insert into technicalpackage values (@tpid, @curdate, @userid, @curdate, @userid)

set @version = 0
set @da = 'Maskin (produktion og ledelse - organisation) - september'
set @en = 'Mechanical Engineering (production and management - organization) - September'
set @nm = ''

set @year = 2003
set @endYear = 2010

while @year <= @endYear
begin
  set @tpvid = @tpvid + 1
  set @version = @version + 1

  insert into technicalpackageversion
    values (@tpvid, @tpid, @version, @nm,
    '<cultures>
      <culture>
        <cultureID>da-DK</cultureID>
        <value>' + @da + ' ' + convert(varchar,@year) + '</value>
      </culture>
      <culture>
        <cultureID>en-GB</cultureID>
        <value>' + @en + ' ' + convert(varchar,@year) + '</value>
      </culture>
    </cultures>',
  @curdate, @userid, @curdate, @userid)
  set @year = @year + 1
end
```

3.21.1 FundamentalCourse

```
declare @userid uniqueidenti↓er
declare @curdate datetime
declare @tpfcid int
declare @tpvid int
declare @tpfciid int
declare @iteration int

set @userid = 'D36DD218-AE59-4046-977C-659E4EB92C92'
set @curdate = getdate()
```

```
set @tpfcid = 0
set @tpfciid = 0

-------------------------------------------------------------------------------
--Biotechnology 2000
-------------------------------------------------------------------------------
set @tpvid = 1

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10011', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10013', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '36260', @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '01030', @curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------
--Biotechnology 2001
-------------------------------------------------------------------------------
set @tpvid = 2

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '01030', @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '28260', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10010', @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10011', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10013', @curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------
--Biotechnology 2002
-------------------------------------------------------------------------------
set @tpvid = 3

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '01030', @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '28260', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10010', @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10011', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10013', @curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------
--Biotechnology 2003 - 2010
-------------------------------------------------------------------------------
```

```
set @iteration = 1

while @iteration < 9
begin
  set @tpvid = @tpvid + 1

  set @tpfcid = @tpfcid + 1
  insert into technicalpackage_fundamentalcourse
  values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
  set @tpfciid = @tpfciid + 1
  insert into technicalpackage_fundamentalcourseitem
  values(@tpfciid, @tpfcid, null, '01030', @curdate, @userid, @curdate, @userid)
  set @tpfciid = @tpfciid + 1
  insert into technicalpackage_fundamentalcourseitem
  values(@tpfciid, @tpfcid, null, '28260', @curdate, @userid, @curdate, @userid)

  set @tpfcid = @tpfcid + 1
  insert into technicalpackage_fundamentalcourse
  values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
  set @tpfciid = @tpfciid + 1
  insert into technicalpackage_fundamentalcourseitem
  values(@tpfciid, @tpfcid, null, '10010', @curdate, @userid, @curdate, @userid)
  set @tpfciid = @tpfciid + 1
  insert into technicalpackage_fundamentalcourseitem
  values(@tpfciid, @tpfcid, null, '10011', @curdate, @userid, @curdate, @userid)

  set @tpfcid = @tpfcid + 1
  insert into technicalpackage_fundamentalcourse
  values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
  set @tpfciid = @tpfciid + 1
  insert into technicalpackage_fundamentalcourseitem
  values(@tpfciid, @tpfcid, null, '10013', @curdate, @userid, @curdate, @userid)

  set @iteration = @iteration + 1
end

-------------------------------------------------------------------------------
--Contruction Engineering 1999 - 1
-------------------------------------------------------------------------------
set @tpvid = 12

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '01020', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10012', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10013', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '26010', @curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------
--Contruction Engineering 1999 - 2
-------------------------------------------------------------------------------
set @tpvid = 13

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '01020', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10012', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10013', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
```

```
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '20010', @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------
--Contruction Engineering 2000
------------------------------------------------------------------------------
set @tpvid = 14

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10012', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10013', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '21010', @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------
--Contruction Engineering 2001
------------------------------------------------------------------------------
set @tpvid = 15

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10012', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10013', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '26026', @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------
--Contruction Engineering 2002
------------------------------------------------------------------------------
set @tpvid = 16

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10012', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10013', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '26026', @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------
--Contruction Engineering 2003 - 2010
------------------------------------------------------------------------------

set @iteration = 1

while @iteration < 9
begin
  set @tpvid = @tpvid + 1
  set @tpfcid = @tpfcid + 1
  insert into technicalpackage_fundamentalcourse
  values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
  set @tpfciid = @tpfciid + 1
  insert into technicalpackage_fundamentalcourseitem
  values(@tpfciid, @tpfcid, null, '10012', @curdate, @userid, @curdate, @userid)

  set @tpfcid = @tpfcid + 1
```

```
  insert into technicalpackage_fundamentalcourse
  values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
  set @tpfciid = @tpfciid + 1
  insert into technicalpackage_fundamentalcourseitem
  values(@tpfciid, @tpfcid, null, '10013', @curdate, @userid, @curdate, @userid)

  set @tpfcid = @tpfcid + 1
  insert into technicalpackage_fundamentalcourse
  values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
  set @tpfciid = @tpfciid + 1
  insert into technicalpackage_fundamentalcourseitem
  values(@tpfciid, @tpfcid, null, '26026', @curdate, @userid, @curdate, @userid)

  set @iteration = @iteration + 1
end

-------------------------------------------------------------------------------
--Design and Innovation 2002
-------------------------------------------------------------------------------
set @tpvid = 25

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)

--Kernefag mangler!!!

-------------------------------------------------------------------------------
--Design and Innovation 2003 - 2010
-------------------------------------------------------------------------------

set @iteration = 1

while @iteration < 9
begin
  set @tpvid = @tpvid + 1
  set @tpfcid = @tpfcid + 1
  insert into technicalpackage_fundamentalcourse
  values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)

--Kernefag mangler!!

  set @iteration = @iteration + 1
end

-------------------------------------------------------------------------------
--Electrical Engineering 1999
-------------------------------------------------------------------------------
set @tpvid = 34

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10012', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '21010', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10010', @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10011', @curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------
--Electrical Engineering 2000
-------------------------------------------------------------------------------
set @tpvid = 35

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10012', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '21010', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10010', @curdate, @userid, @curdate, @userid)
```

```
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10011', @curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------
--Electrical Engineering 2001
-------------------------------------------------------------------------------
set @tpvid = 36

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10012', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '26026', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10010', @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10011', @curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------
--Electrical Engineering 2002
-------------------------------------------------------------------------------
set @tpvid = 37

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10012', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '26026', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10010', @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10011', @curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------
--Electrical Engineering 2003 - 2010
-------------------------------------------------------------------------------
set @iteration = 1

while @iteration < 9
begin
  set @tpvid = @tpvid + 1
  set @tpfcid = @tpfcid + 1
  insert into technicalpackage_fundamentalcourse
  values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)

  set @tpfcid = @tpfcid + 1
  insert into technicalpackage_fundamentalcourse
  values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
  set @tpfciid = @tpfciid + 1
  insert into technicalpackage_fundamentalcourseitem
  values(@tpfciid, @tpfcid, null, '10012', @curdate, @userid, @curdate, @userid)

  set @tpfcid = @tpfcid + 1
  insert into technicalpackage_fundamentalcourse
  values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
  set @tpfciid = @tpfciid + 1
  insert into technicalpackage_fundamentalcourseitem
  values(@tpfciid, @tpfcid, null, '26026', @curdate, @userid, @curdate, @userid)

  set @tpfcid = @tpfcid + 1
  insert into technicalpackage_fundamentalcourse
  values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
  set @tpfciid = @tpfciid + 1
  insert into technicalpackage_fundamentalcourseitem
  values(@tpfciid, @tpfcid, null, '10010', @curdate, @userid, @curdate, @userid)
  set @tpfciid = @tpfciid + 1
  insert into technicalpackage_fundamentalcourseitem
  values(@tpfciid, @tpfcid, null, '10011', @curdate, @userid, @curdate, @userid)

  set @iteration = @iteration + 1
```

```
end

-------------------------------------------------------------------------------
--Energy Engineering 1999 - 1
-------------------------------------------------------------------------------
set @tpvid = 46

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '01030', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10032', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10013', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '21010', @curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------
--Energy Engineering 1999 - 2
-------------------------------------------------------------------------------
set @tpvid = 47

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '01030', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10032', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10013', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '21010', @curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------
--Energy Engineering 2000
-------------------------------------------------------------------------------
set @tpvid = 48

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '01030', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '21010', @curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------
--Energy Engineering 2001
-------------------------------------------------------------------------------
set @tpvid = 49

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '01030', @curdate, @userid, @curdate, @userid)
```

```
set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '26026', @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------
--Energy Engineering 2002
------------------------------------------------------------------------------
set @tpvid = 50

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '01030', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '26026', @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------
--Energy Engineering 2003 - 2010
------------------------------------------------------------------------------

set @iteration = 1

while @iteration < 9
begin
  set @tpvid = @tpvid + 1
  set @tpfcid = @tpfcid + 1
  insert into technicalpackage_fundamentalcourse
  values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)

  set @tpfciid = @tpfciid + 1
  insert into technicalpackage_fundamentalcourseitem
  values(@tpfciid, @tpfcid, null, '01030', @curdate, @userid, @curdate, @userid)
  set @tpfciid = @tpfciid + 1
  insert into technicalpackage_fundamentalcourseitem
  values(@tpfciid, @tpfcid, null, '26026', @curdate, @userid, @curdate, @userid)

  set @iteration = @iteration + 1
end

------------------------------------------------------------------------------
--Information Technology 1999
------------------------------------------------------------------------------
set @tpvid = 59

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10010', @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10011', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10012', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10013', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '21010', @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------
--Information Technology 2000
------------------------------------------------------------------------------
set @tpvid = 60

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
```

```
values(@tpfciid, @tpfcid, null, '10010', @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10011', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10012', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10013', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '21010', @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------
--Information Technology 2001
------------------------------------------------------------------------------
set @tpvid = 61

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10010', @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10011', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10012', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10013', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '26026', @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------
--Information Technology 2002
------------------------------------------------------------------------------
set @tpvid = 62

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10010', @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10011', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10012', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10013', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '26026', @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------
```

```
--Information Technology 2003 - 2010
-----------------------------------------------------------------------------------------
set @iteration = 1

while @iteration < 9
begin
  set @tpvid = @tpvid + 1
  set @tpfcid = @tpfcid + 1
  insert into technicalpackage_fundamentalcourse
  values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
  set @tpfciid = @tpfciid + 1
  insert into technicalpackage_fundamentalcourseitem
  values(@tpfciid, @tpfcid, null, '10010', @curdate, @userid, @curdate, @userid)
  set @tpfciid = @tpfciid + 1
  insert into technicalpackage_fundamentalcourseitem
  values(@tpfciid, @tpfcid, null, '10011', @curdate, @userid, @curdate, @userid)

  set @tpfcid = @tpfcid + 1
  insert into technicalpackage_fundamentalcourse
  values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
  set @tpfciid = @tpfciid + 1
  insert into technicalpackage_fundamentalcourseitem
  values(@tpfciid, @tpfcid, null, '10012', @curdate, @userid, @curdate, @userid)

  set @tpfcid = @tpfcid + 1
  insert into technicalpackage_fundamentalcourse
  values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
  set @tpfciid = @tpfciid + 1
  insert into technicalpackage_fundamentalcourseitem
  values(@tpfciid, @tpfcid, null, '10013', @curdate, @userid, @curdate, @userid)

  set @tpfcid = @tpfcid + 1
  insert into technicalpackage_fundamentalcourse
  values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
  set @tpfciid = @tpfciid + 1
  insert into technicalpackage_fundamentalcourseitem
  values(@tpfciid, @tpfcid, null, '26026', @curdate, @userid, @curdate, @userid)

  set @iteration = @iteration + 1
end

-----------------------------------------------------------------------------------------
--Chemical Engineering 1999 - 1
-----------------------------------------------------------------------------------------
set @tpvid = 71

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '01030', @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '36260', @curdate, @userid, @curdate, @userid)

-----------------------------------------------------------------------------------------
--Chemical Engineering 1999 - 2
-----------------------------------------------------------------------------------------
set @tpvid = 72

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '01030', @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '36260', @curdate, @userid, @curdate, @userid)

-----------------------------------------------------------------------------------------
--Chemical Engineering 2000
-----------------------------------------------------------------------------------------
set @tpvid = 73

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10013', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '36260', @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '01030', @curdate, @userid, @curdate, @userid)

-----------------------------------------------------------------------------------------
--Chemical Engineering 2001
-----------------------------------------------------------------------------------------
set @tpvid = 74

set @tpfcid = @tpfcid + 1
```

```
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10013', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '28260', @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '01030', @curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------
--Chemical Engineering 2002
-------------------------------------------------------------------------------
set @tpvid = 75

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10013', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '28260', @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '01030', @curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------
--Chemical Engineering 2003 - 2010
-------------------------------------------------------------------------------
set @iteration = 1

while @iteration < 9
begin
  set @tpvid = @tpvid + 1
  set @tpfcid = @tpfcid + 1
  insert into technicalpackage_fundamentalcourse
  values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
  set @tpfciid = @tpfciid + 1
  insert into technicalpackage_fundamentalcourseitem
  values(@tpfciid, @tpfcid, null, '10013', @curdate, @userid, @curdate, @userid)

  set @tpfcid = @tpfcid + 1
  insert into technicalpackage_fundamentalcourse
  values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
  set @tpfciid = @tpfciid + 1
  insert into technicalpackage_fundamentalcourseitem
  values(@tpfciid, @tpfcid, null, '28260', @curdate, @userid, @curdate, @userid)
  set @tpfciid = @tpfciid + 1
  insert into technicalpackage_fundamentalcourseitem
  values(@tpfciid, @tpfcid, null, '01030', @curdate, @userid, @curdate, @userid)

  set @iteration = @iteration + 1
end

-------------------------------------------------------------------------------
--Mechanical Engineering 1999 - 1
-------------------------------------------------------------------------------
set @tpvid = 84

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10013', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '01030', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '21010', @curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------
--Mechanical Engineering 1999 - 2
-------------------------------------------------------------------------------
set @tpvid = 85

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
```

```
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10013', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '01030', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '21010', @curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------
--Mechanical Engineering 2000
-------------------------------------------------------------------------------
set @tpvid = 86

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10013', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '01030', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '21021', @curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------
--Mechanical Engineering 2001
-------------------------------------------------------------------------------
set @tpvid = 87

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10013', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '01030', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '26026', @curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------
--Mechanical Engineering 2002
-------------------------------------------------------------------------------
set @tpvid = 88

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10013', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '01030', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '26026', @curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------
--Mechanical Engineering 2003 - 2010
```

```
------------------------------------------------------------------------------

set @iteration = 1

while @iteration < 9
begin
  set @tpvid = @tpvid + 1
  set @tpfcid = @tpfcid + 1
  insert into technicalpackage_fundamentalcourse
  values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
  set @tpfciid = @tpfciid + 1
  insert into technicalpackage_fundamentalcourseitem
  values(@tpfciid, @tpfcid, null, '10013', @curdate, @userid, @curdate, @userid)

  set @tpfcid = @tpfcid + 1
  insert into technicalpackage_fundamentalcourse
  values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
  set @tpfciid = @tpfciid + 1
  insert into technicalpackage_fundamentalcourseitem
  values(@tpfciid, @tpfcid, null, '01030', @curdate, @userid, @curdate, @userid)

  set @tpfcid = @tpfcid + 1
  insert into technicalpackage_fundamentalcourse
  values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
  set @tpfciid = @tpfciid + 1
  insert into technicalpackage_fundamentalcourseitem
  values(@tpfciid, @tpfcid, null, '26026', @curdate, @userid, @curdate, @userid)

  set @iteration = @iteration + 1
end

------------------------------------------------------------------------------
--Environmental Engineering 1999 - 1
------------------------------------------------------------------------------
set @tpvid = 97

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '01030', @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '36260', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10032', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10013', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '21061', @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------
--Environmental Engineering 1999 - 2
------------------------------------------------------------------------------
set @tpvid = 98

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '01030', @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '36260', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10032', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10013', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
```

```
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '21061', @curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------
--Environmental Engineering 2000
-------------------------------------------------------------------------------
set @tpvid = 99

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10013', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '01030', @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '36260', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10010', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10012', @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '21061', @curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------
--Environmental Engineering 2001
-------------------------------------------------------------------------------
set @tpvid = 100

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10013', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '01030', @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '28260', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10010', @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10011', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10012', @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '26201', @curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------
--Environmental Engineering 2002
-------------------------------------------------------------------------------
set @tpvid = 101

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10013', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
```

```
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '01030', @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '28260', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10010', @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10011', @curdate, @userid, @curdate, @userid)

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '10012', @curdate, @userid, @curdate, @userid)
set @tpfciid = @tpfciid + 1
insert into technicalpackage_fundamentalcourseitem
values(@tpfciid, @tpfcid, null, '26201', @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------
--Environmental Engineering 2003 - 2010
------------------------------------------------------------------------------
set @iteration = 1

while @iteration < 9
begin
  set @tpvid = @tpvid + 1
  set @tpfcid = @tpfcid + 1
  insert into technicalpackage_fundamentalcourse
  values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
  set @tpfciid = @tpfciid + 1
  insert into technicalpackage_fundamentalcourseitem
  values(@tpfciid, @tpfcid, null, '10013', @curdate, @userid, @curdate, @userid)

  set @tpfcid = @tpfcid + 1
  insert into technicalpackage_fundamentalcourse
  values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
  set @tpfciid = @tpfciid + 1
  insert into technicalpackage_fundamentalcourseitem
  values(@tpfciid, @tpfcid, null, '01030', @curdate, @userid, @curdate, @userid)
  set @tpfciid = @tpfciid + 1
  insert into technicalpackage_fundamentalcourseitem
  values(@tpfciid, @tpfcid, null, '28260', @curdate, @userid, @curdate, @userid)

  set @tpfcid = @tpfcid + 1
  insert into technicalpackage_fundamentalcourse
  values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
  set @tpfciid = @tpfciid + 1
  insert into technicalpackage_fundamentalcourseitem
  values(@tpfciid, @tpfcid, null, '10010', @curdate, @userid, @curdate, @userid)
  set @tpfciid = @tpfciid + 1
  insert into technicalpackage_fundamentalcourseitem
  values(@tpfciid, @tpfcid, null, '10011', @curdate, @userid, @curdate, @userid)

  set @tpfcid = @tpfcid + 1
  insert into technicalpackage_fundamentalcourse
  values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
  set @tpfciid = @tpfciid + 1
  insert into technicalpackage_fundamentalcourseitem
  values(@tpfciid, @tpfcid, null, '10012', @curdate, @userid, @curdate, @userid)
  set @tpfciid = @tpfciid + 1
  insert into technicalpackage_fundamentalcourseitem
  values(@tpfciid, @tpfcid, null, '26201', @curdate, @userid, @curdate, @userid)
  set @iteration = @iteration + 1
end

------------------------------------------------------------------------------
--Technical Physics 1999 - 1
------------------------------------------------------------------------------
set @tpvid = 110

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------
--Technical Physics 1999 - 2
------------------------------------------------------------------------------
set @tpvid = 111

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)

------------------------------------------------------------------------------
--Technical Physics 2000
------------------------------------------------------------------------------
set @tpvid = 112

set @tpfcid = @tpfcid + 1
```

```
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------
--Technical Physics 2001
-------------------------------------------------------------------------------
set @tpvid = 113

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------
--Technical Physics 2002
-------------------------------------------------------------------------------
set @tpvid = 114

set @tpfcid = @tpfcid + 1
insert into technicalpackage_fundamentalcourse
values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)

-------------------------------------------------------------------------------
--Technical Physics 2003 - 2010
-------------------------------------------------------------------------------

set @iteration = 1

while @iteration < 9
begin
  set @tpvid = @tpvid + 1
  set @tpfcid = @tpfcid + 1
  insert into technicalpackage_fundamentalcourse
  values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)

  set @iteration = @iteration + 1
end

-------------------------------------------------------------------------------
--Medication and Technology 2003 - 2010
-------------------------------------------------------------------------------

set @iteration = 1

while @iteration < 9
begin
  set @tpvid = @tpvid + 1
  set @tpfcid = @tpfcid + 1
  insert into technicalpackage_fundamentalcourse
  values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
  set @tpfciid = @tpfciid + 1
  insert into technicalpackage_fundamentalcourseitem
  values(@tpfciid, @tpfcid, null, '10013', @curdate, @userid, @curdate, @userid)

  set @tpfcid = @tpfcid + 1
  insert into technicalpackage_fundamentalcourse
  values(@tpfcid, @tpvid, @curdate, @userid, @curdate, @userid)
  set @tpfciid = @tpfciid + 1
  insert into technicalpackage_fundamentalcourseitem
  values(@tpfciid, @tpfcid, null, '26026', @curdate, @userid, @curdate, @userid)

  set @iteration = @iteration + 1
end
```

3.21.2 Period

```
-------------------------------------------------------------------------------------------------

--NOTICE NOTICE NOTICE NOTICE NOTICE NOTICE NOTICE NOTICE NOTICE NOTICE NOTICE
-------------------------------------------------------------------------------------------------

--This SQL uses the period table, so the sql for creating periods must be run
--prior to running the sql in this file.
-------------------------------------------------------------------------------------------------

--Common declarations
-------------------------------------------------------------------------------------------------

declare @tppid int, @tpvid int, @tpid int
declare @startdate datetime

set @tppid = 0
set @tpvid = 0

-------------------------------------------------------------------------------------------------

--Biotechnology
-------------------------------------------------------------------------------------------------

set @startdate = '2000-08-01'
set @tpid = 1

declare tpv_cursor cursor for
select technicalpackageversion_id
from technicalpackageversion
where technicalpackage_id = @tpid
order by technicalpackageversion_id
```

```
for read only
open tpv_cursor

fetch next from tpv_cursor into @tpvid
while @@fetch_status = 0
begin
  execute sp_createtpp_master @startdate, @tpvid, @tppid, @tppid output
  set @startdate = dateadd(year, 1, @startdate)
  fetch next from tpv_cursor into @tpvid
end

close tpv_cursor
deallocate tpv_cursor
------------------------------------------------------------------------------------------

--Construction
------------------------------------------------------------------------------------------

set @startdate = '1999-08-01'
set @tpid = 2

declare tpv_cursor cursor for
select technicalpackageversion_id
from technicalpackageversion
where technicalpackage_id = @tpid
order by technicalpackageversion_id
for read only
open tpv_cursor

fetch next from tpv_cursor into @tpvid
while @@fetch_status = 0
begin
  execute sp_createtpp_master @startdate, @tpvid, @tppid, @tppid output
  if @tpvid > 12
    set @startdate = dateadd(year, 1, @startdate)
  fetch next from tpv_cursor into @tpvid
end

close tpv_cursor
deallocate tpv_cursor
------------------------------------------------------------------------------------------

--Design and Innovation
------------------------------------------------------------------------------------------

set @startdate = '2002-08-01'
set @tpid = 3

declare tpv_cursor cursor for
select technicalpackageversion_id
from technicalpackageversion
where technicalpackage_id = @tpid
order by technicalpackageversion_id
for read only
open tpv_cursor

fetch next from tpv_cursor into @tpvid
while @@fetch_status = 0
begin
  execute sp_createtpp_master @startdate, @tpvid, @tppid, @tppid output
  set @startdate = dateadd(year, 1, @startdate)
  fetch next from tpv_cursor into @tpvid
end

close tpv_cursor
deallocate tpv_cursor
------------------------------------------------------------------------------------------

--Electrical Engineering
------------------------------------------------------------------------------------------

set @startdate = '1999-08-01'
set @tpid = 4

declare tpv_cursor cursor for
select technicalpackageversion_id
from technicalpackageversion
where technicalpackage_id = @tpid
order by technicalpackageversion_id
for read only
open tpv_cursor

fetch next from tpv_cursor into @tpvid
while @@fetch_status = 0
begin
  execute sp_createtpp_master @startdate, @tpvid, @tppid, @tppid output
  set @startdate = dateadd(year, 1, @startdate)
  fetch next from tpv_cursor into @tpvid
end

close tpv_cursor
deallocate tpv_cursor
------------------------------------------------------------------------------------------

--Energy Engineering
------------------------------------------------------------------------------------------

set @startdate = '1999-08-01'
set @tpid = 5
```

```
declare tpv_cursor cursor for
select technicalpackageversion_id
from technicalpackageversion
where technicalpackage_id = @tpid
order by technicalpackageversion_id
for read only
open tpv_cursor

fetch next from tpv_cursor into @tpvid
while @@fetch_status = 0
begin
  execute sp_createtpp_master @startdate, @tpvid, @tppid, @tppid output
  if @tpvid > 46
    set @startdate = dateadd(year, 1, @startdate)
  fetch next from tpv_cursor into @tpvid
end

close tpv_cursor
deallocate tpv_cursor
-------------------------------------------------------------------------------------------------

--Information Technology
-------------------------------------------------------------------------------------------------

set @startdate = '1999-08-01'
set @tpid = 6

declare tpv_cursor cursor for
select technicalpackageversion_id
from technicalpackageversion
where technicalpackage_id = @tpid
order by technicalpackageversion_id
for read only
open tpv_cursor

fetch next from tpv_cursor into @tpvid
while @@fetch_status = 0
begin
  execute sp_createtpp_master @startdate, @tpvid, @tppid, @tppid output
  set @startdate = dateadd(year, 1, @startdate)
  fetch next from tpv_cursor into @tpvid
end

close tpv_cursor
deallocate tpv_cursor
-------------------------------------------------------------------------------------------------

--Chemical Engineering
-------------------------------------------------------------------------------------------------

set @startdate = '1999-08-01'
set @tpid = 7

declare tpv_cursor cursor for
select technicalpackageversion_id
from technicalpackageversion
where technicalpackage_id = @tpid
order by technicalpackageversion_id
for read only
open tpv_cursor

fetch next from tpv_cursor into @tpvid
while @@fetch_status = 0
begin
  execute sp_createtpp_master @startdate, @tpvid, @tppid, @tppid output
  if @tpvid > 71
    set @startdate = dateadd(year, 1, @startdate)
  fetch next from tpv_cursor into @tpvid
end

close tpv_cursor
deallocate tpv_cursor
-------------------------------------------------------------------------------------------------

--Mechanical Engineering
-------------------------------------------------------------------------------------------------

set @startdate = '1999-08-01'
set @tpid = 8

declare tpv_cursor cursor for
select technicalpackageversion_id
from technicalpackageversion
where technicalpackage_id = @tpid
order by technicalpackageversion_id
for read only
open tpv_cursor

fetch next from tpv_cursor into @tpvid
while @@fetch_status = 0
begin
  execute sp_createtpp_master @startdate, @tpvid, @tppid, @tppid output
  if @tpvid > 84
    set @startdate = dateadd(year, 1, @startdate)
  fetch next from tpv_cursor into @tpvid
end

close tpv_cursor
```

```
deallocate tpv_cursor
------------------------------------------------------------------------------------------------------------------------

--Environmental Engineering
------------------------------------------------------------------------------------------------------------------------

set @startdate = '1999-08-01'
set @tpid = 9

declare tpv_cursor cursor for
select technicalpackageversion_id
from technicalpackageversion
where technicalpackage_id = @tpid
order by technicalpackageversion_id
for read only
open tpv_cursor

fetch next from tpv_cursor into @tpvid
while @@fetch_status = 0
begin
  execute sp_createtpp_master @startdate, @tpvid, @tppid, @tppid output
  if @tpvid > 97
    set @startdate = dateadd(year, 1, @startdate)
  fetch next from tpv_cursor into @tpvid
end

close tpv_cursor
deallocate tpv_cursor
------------------------------------------------------------------------------------------------------------------------

--Technical Physics
------------------------------------------------------------------------------------------------------------------------

set @startdate = '1999-08-01'
set @tpid = 10

declare tpv_cursor cursor for
select technicalpackageversion_id
from technicalpackageversion
where technicalpackage_id = @tpid
order by technicalpackageversion_id
for read only
open tpv_cursor

fetch next from tpv_cursor into @tpvid
while @@fetch_status = 0
begin
  execute sp_createtpp_master @startdate, @tpvid, @tppid, @tppid output
  if @tpvid > 110
    set @startdate = dateadd(year, 1, @startdate)
  fetch next from tpv_cursor into @tpvid
end

close tpv_cursor
deallocate tpv_cursor
------------------------------------------------------------------------------------------------------------------------

--Medication and Technology 2003 - 2010
------------------------------------------------------------------------------------------------------------------------

set @startdate = '2003-08-01'
set @tpid = 11

declare tpv_cursor cursor for
select technicalpackageversion_id
from technicalpackageversion
where technicalpackage_id = @tpid
order by technicalpackageversion_id
for read only
open tpv_cursor

fetch next from tpv_cursor into @tpvid
while @@fetch_status = 0
begin
  execute sp_createtpp_master @startdate, @tpvid, @tppid, @tppid output
  set @startdate = dateadd(year, 1, @startdate)
  fetch next from tpv_cursor into @tpvid
end

close tpv_cursor
deallocate tpv_cursor

------------------------------------------------------------------------------------------------------------------------

--Construction Engineering (Bachelor of Science) 2003 - 2010
------------------------------------------------------------------------------------------------------------------------

set @startdate = '2003-02-01'
set @tpid = 12

declare tpv_cursor cursor for
select technicalpackageversion_id
from technicalpackageversion
where technicalpackage_id = @tpid
order by technicalpackageversion_id
for read only
open tpv_cursor

fetch next from tpv_cursor into @tpvid
```

```
while @@fetch_status = 0
begin
  execute sp_createtpp_bachelor @startdate, @tpvid, @tppid, @tppid output
  set @startdate = dateadd(year, 1, @startdate)
  fetch next from tpv_cursor into @tpvid
end

close tpv_cursor
deallocate tpv_cursor

set @startdate = '2003-08-01'
set @tpid = 13

declare tpv_cursor cursor for
select technicalpackageversion_id
from technicalpackageversion
where technicalpackage_id = @tpid
order by technicalpackageversion_id
for read only
open tpv_cursor

fetch next from tpv_cursor into @tpvid
while @@fetch_status = 0
begin
  execute sp_createtpp_bachelor @startdate, @tpvid, @tppid, @tppid output
  set @startdate = dateadd(year, 1, @startdate)
  fetch next from tpv_cursor into @tpvid
end

close tpv_cursor
deallocate tpv_cursor
-------------------------------------------------------------------------------------------------

--Urban Planning and Contruction Engineering (Bachelor of Science) 2003 - 2010
-------------------------------------------------------------------------------------------------

set @startdate = '2003-08-01'
set @tpid = 14

declare tpv_cursor cursor for
select technicalpackageversion_id
from technicalpackageversion
where technicalpackage_id = @tpid
order by technicalpackageversion_id
for read only
open tpv_cursor

fetch next from tpv_cursor into @tpvid
while @@fetch_status = 0
begin
  execute sp_createtpp_bachelor @startdate, @tpvid, @tppid, @tppid output
  set @startdate = dateadd(year, 1, @startdate)
  fetch next from tpv_cursor into @tpvid
end

close tpv_cursor
deallocate tpv_cursor

-------------------------------------------------------------------------------------------------

--Electrical Engineering (Bachelor of Science) 2003 - 2010
-------------------------------------------------------------------------------------------------

set @startdate = '2003-02-01'
set @tpid = 15

declare tpv_cursor cursor for
select technicalpackageversion_id
from technicalpackageversion
where technicalpackage_id = @tpid
order by technicalpackageversion_id
for read only
open tpv_cursor

fetch next from tpv_cursor into @tpvid
while @@fetch_status = 0
begin
  execute sp_createtpp_bachelor @startdate, @tpvid, @tppid, @tppid output
  set @startdate = dateadd(year, 1, @startdate)
  fetch next from tpv_cursor into @tpvid
end

close tpv_cursor
deallocate tpv_cursor

set @startdate = '2003-08-01'
set @tpid = 16

declare tpv_cursor cursor for
select technicalpackageversion_id
from technicalpackageversion
where technicalpackage_id = @tpid
order by technicalpackageversion_id
for read only
open tpv_cursor

fetch next from tpv_cursor into @tpvid
while @@fetch_status = 0
begin
```

```
  execute sp_createtpp_bachelor @startdate, @tpvid, @tppid, @tppid output
  set @startdate = dateadd(year, 1, @startdate)
  fetch next from tpv_cursor into @tpvid
end

close tpv_cursor
deallocate tpv_cursor

------------------------------------------------------------------------------------------------------------------------

--Chemical Engineering (Bachelor of Science) 2003 - 2010
------------------------------------------------------------------------------------------------------------------------

set @startdate = '2003-02-01'
set @tpid = 17

declare tpv_cursor cursor for
select technicalpackageversion_id
from technicalpackageversion
where technicalpackage_id = @tpid
order by technicalpackageversion_id
for read only
open tpv_cursor

fetch next from tpv_cursor into @tpvid
while @@fetch_status = 0
begin
  execute sp_createtpp_bachelor @startdate, @tpvid, @tppid, @tppid output
  set @startdate = dateadd(year, 1, @startdate)
  fetch next from tpv_cursor into @tpvid
end

close tpv_cursor
deallocate tpv_cursor

set @startdate = '2003-08-01'
set @tpid = 18

declare tpv_cursor cursor for
select technicalpackageversion_id
from technicalpackageversion
where technicalpackage_id = @tpid
order by technicalpackageversion_id
for read only
open tpv_cursor

fetch next from tpv_cursor into @tpvid
while @@fetch_status = 0
begin
  execute sp_createtpp_bachelor @startdate, @tpvid, @tppid, @tppid output
  set @startdate = dateadd(year, 1, @startdate)
  fetch next from tpv_cursor into @tpvid
end

close tpv_cursor
deallocate tpv_cursor

------------------------------------------------------------------------------------------------------------------------

--Information Technology (Bachelor of Science) 2003 - 2010
------------------------------------------------------------------------------------------------------------------------

set @startdate = '2003-02-01'
set @tpid = 19

declare tpv_cursor cursor for
select technicalpackageversion_id
from technicalpackageversion
where technicalpackage_id = @tpid
order by technicalpackageversion_id
for read only
open tpv_cursor

fetch next from tpv_cursor into @tpvid
while @@fetch_status = 0
begin
  execute sp_createtpp_bachelor @startdate, @tpvid, @tppid, @tppid output
  set @startdate = dateadd(year, 1, @startdate)
  fetch next from tpv_cursor into @tpvid
end

close tpv_cursor
deallocate tpv_cursor

set @startdate = '2003-08-01'
set @tpid = 20

declare tpv_cursor cursor for
select technicalpackageversion_id
from technicalpackageversion
where technicalpackage_id = @tpid
order by technicalpackageversion_id
for read only
open tpv_cursor

fetch next from tpv_cursor into @tpvid
while @@fetch_status = 0
begin
  execute sp_createtpp_bachelor @startdate, @tpvid, @tppid, @tppid output
```

```
  set @startdate = dateadd(year, 1, @startdate)
  fetch next from tpv_cursor into @tpvid
end

close tpv_cursor
deallocate tpv_cursor

-------------------------------------------------------------------------------------------------

--Mechanical Engineering (Bachelor of Science) 2003 - 2010
-------------------------------------------------------------------------------------------------

set @tpid = 21

while @tpid < 41
begin
  set @startdate = '2003-02-01'
  declare tpv_cursor cursor for
  select technicalpackageversion_id
  from technicalpackageversion
  where technicalpackage_id = @tpid
  order by technicalpackageversion_id
  for read only
  open tpv_cursor

  fetch next from tpv_cursor into @tpvid
  while @@fetch_status = 0
  begin
    execute sp_createtpp_bachelor @startdate, @tpvid, @tppid, @tppid output
    set @startdate = dateadd(year, 1, @startdate)
    fetch next from tpv_cursor into @tpvid
  end

  close tpv_cursor
  deallocate tpv_cursor

  set @tpid = @tpid + 2
end

set @tpid = 22

while @tpid < 41
begin
  set @startdate = '2003-08-01'
  declare tpv_cursor cursor for
  select technicalpackageversion_id
  from technicalpackageversion
  where technicalpackage_id = @tpid
  order by technicalpackageversion_id
  for read only
  open tpv_cursor

  fetch next from tpv_cursor into @tpvid
  while @@fetch_status = 0
  begin
    execute sp_createtpp_bachelor @startdate, @tpvid, @tppid, @tppid output
    set @startdate = dateadd(year, 1, @startdate)
    fetch next from tpv_cursor into @tpvid
  end

  close tpv_cursor
  deallocate tpv_cursor

  set @tpid = @tpid + 2
end
```

3.21.3 PeriodCourse

```
delete from technicalpackage_periodcourse
delete from technicalpackage_periodcourseitem
delete from technicalpackage_periodoptionalcourse

declare @userid uniqueidenti↓er
declare @curdate datetime
declare @tppcid int
declare @tppid int
declare @tpvid int
declare @tppciid int
declare @tppocid int
declare @iteration int

set @userid = 'D36DD218-AE59-4046-977C-659E4EB92C92'
set @curdate = getdate()
set @tppcid = 0
set @tppid = 0
set @tppciid = 0
set @tppocid = 0

-------------------------------------------------------------------------------
--Biotechnology 2000
-------------------------------------------------------------------------------
set @tpvid = 1

declare tpp_cursor cursor for
select top 8 technicalpackage_period_id
from technicalpackage_period
```

```
where technicalpackageversion_id = @tpvid
order by technicalpackage_period_id
for read only
open tpp_cursor

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '21001', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '23110', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '30000', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '30000', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '21001', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '23110', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 3, '30000', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 4, '30000', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
```

```
  values(@tppciid, @tppcid, null, 1, '21004', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '36100', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '27021', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '01030', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10010', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '36100', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '85100', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10013', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26406', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '27031', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '27031', @curdate, @userid, @curdate, @userid)
end

close tpp_cursor
deallocate tpp_cursor
------------------------------------------------------------------------------
--Biotechnology 2001
------------------------------------------------------------------------------

set @tpvid = 2

declare tpp_cursor cursor for
select top 8 technicalpackage_period_id
from technicalpackage_period
where technicalpackageversion_id = @tpvid
order by technicalpackage_period_id
for read only
open tpp_cursor

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '26001', @curdate, @userid, @curdate, @userid)
```

```
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '27000', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '27000', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '26001', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 3, '27000', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '27011', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26300', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 4, '27000', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '26010', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '28120', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '27021', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '01030', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10010', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
```

```
  values(@tppciid, @tppcid, null, 2, '26010', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '28120', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '42510', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10013', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26406', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '27031', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '27031', @curdate, @userid, @curdate, @userid)
end

close tpp_cursor
deallocate tpp_cursor
------------------------------------------------------------------------------
--Biotechnology 2002
------------------------------------------------------------------------------
set @tpvid = 3

declare tpp_cursor cursor for
select top 8 technicalpackage_period_id
from technicalpackage_period
where technicalpackageversion_id = @tpvid
order by technicalpackage_period_id
for read only
open tpp_cursor

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '26001', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '27000', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '27000', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
```

```
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '26001', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 3, '27000', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '27011', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26300', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 4, '27000', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '26010', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '28120', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '27021', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '01030', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10010', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '26010', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '28120', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '42510', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10013', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26406', @curdate, @userid, @curdate, @userid)
```

```
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '27031', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '27031', @curdate, @userid, @curdate, @userid)
end

close tpp_cursor
deallocate tpp_cursor
-------------------------------------------------------------------------------
--Biotechnology 2003 - 2010
-------------------------------------------------------------------------------

set @iteration = 1
while @iteration < 9
begin
  set @tpvid = @tpvid + 1

  declare tpp_cursor cursor for
  select top 8 technicalpackage_period_id
  from technicalpackage_period
  where technicalpackageversion_id = @tpvid
  order by technicalpackage_period_id
  for read only
  open tpp_cursor

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '01005', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '26001', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '27000', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '27000', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '01005', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '26001', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 3, '27000', @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '27011', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
```

```
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '26300', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 4, '27000', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '26010', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '28120', @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '27021', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '01030', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '10010', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '26010', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '28120', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42510', @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '10013', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '26406', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '27031', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 2, '27031', @curdate, @userid, @curdate, @userid)
  end
  close tpp_cursor
  deallocate tpp_cursor
  set @iteration = @iteration + 1
end
```

```
-------------------------------------------------------------------------------
set @tpvid = 12

declare tpp_cursor cursor for
select top 8 technicalpackage_period_id
from technicalpackage_period
where technicalpackageversion_id = @tpvid
order by technicalpackage_period_id
for read only
open tpp_cursor

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01010', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01012', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '59000', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '64000', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '67100', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '67161', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '67121', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10010', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '67111', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
```

```
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '49104', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '63010', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '57003', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '49105', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01034', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, 'xx800', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

close tpp_cursor
deallocate tpp_cursor
------------------------------------------------------------------------------------------
--Construction Engineering 1999 - 2
------------------------------------------------------------------------------------------
set @tpvid = 13

declare tpp_cursor cursor for
select top 8 technicalpackage_period_id
from technicalpackage_period
where technicalpackageversion_id = @tpvid
order by technicalpackage_period_id
for read only
open tpp_cursor

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01010', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01012', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '59000', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
```

```
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '64000', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '67100', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '67161', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '67121', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01013', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10010', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '67111', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '49104', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '57002', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '49105', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
```

```
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01034', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, 'xx800', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

close tpp_cursor
deallocate tpp_cursor
------------------------------------------------------------------------------
--Construction Engineering 2000
------------------------------------------------------------------------------
set @tpvid = 14

declare tpp_cursor cursor for
select top 8 technicalpackage_period_id
from technicalpackage_period
where technicalpackageversion_id = @tpvid
order by technicalpackage_period_id
for read only
open tpp_cursor

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10011', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '59001', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '64000', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '67161', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '67000', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
```

```
  values(@tppciid, @tppcid, null, 2, '10011', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '59106', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '56002', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '67121', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '67271', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '85221', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '04120', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '56452', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '67151', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '67462', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '88383', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '49104', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '57002', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '13000', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11511', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02531', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26026', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10012', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10013', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '49105', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
```

```
  values(@tppciid, @tppcid, null, 1, '01034', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, 'xx800', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11101', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11501', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11301', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11003', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11411', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11202', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10012', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10013', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26026', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11412', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02531', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02661', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11204', @curdate, @userid, @curdate, @userid)
end

close tpp_cursor
deallocate tpp_cursor
------------------------------------------------------------------------------
--Construction Engineering 2001
------------------------------------------------------------------------------
set @tpvid = 15

declare tpp_cursor cursor for
select top 8 technicalpackage_period_id
from technicalpackage_period
where technicalpackageversion_id = @tpvid
order by technicalpackage_period_id
for read only
open tpp_cursor

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10011', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '11001', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
```

```
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '11200', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '11510', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '11002', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '10011', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11101', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11301', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11401', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11501', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02531', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11204', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02661', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11402', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '02197', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '12300', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '13000', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
```

```
  values(@tppocid, @tppid, null, 1, '11511', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02531', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26026', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10012', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10013', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '02197', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01034', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '11690', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11101', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11501', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11301', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11003', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11411', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11202', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10012', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10013', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26026', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11412', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02531', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02661', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11204', @curdate, @userid, @curdate, @userid)
end

close tpp_cursor
deallocate tpp_cursor
------------------------------------------------------------------------------
--Construction Engineering 2002
------------------------------------------------------------------------------
set @tpvid = 16
```

```
declare tpp_cursor cursor for
select top 8 technicalpackage_period_id
from technicalpackage_period
where technicalpackageversion_id = @tpvid
order by technicalpackage_period_id
for read only
open tpp_cursor

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10011', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '11001', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '11200', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '11510', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '11002', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '10011', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11101', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11301', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11202', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11501', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '12400', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
```

```
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02532', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11204', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02661', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '12401', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '02197', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '12300', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02401', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02534', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02701', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10012', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10013', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11511', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '13000', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26026', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '02197', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01034', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '11690', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02401', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02701', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10012', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10013', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
```

```
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11003', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11101', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11202', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11301', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11411', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11501', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11512', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '12100', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '13210', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26026', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02532', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02661', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11204', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11412', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '13001', @curdate, @userid, @curdate, @userid)
end

close tpp_cursor
deallocate tpp_cursor
------------------------------------------------------------------------------------------
--Construction Engineering 2003 - 2010
------------------------------------------------------------------------------------------

set @iteration = 1
while @iteration < 9
begin
  set @tpvid = @tpvid + 1

  declare tpp_cursor cursor for
  select top 8 technicalpackage_period_id
  from technicalpackage_period
  where technicalpackageversion_id = @tpvid
  order by technicalpackage_period_id
  for read only
  open tpp_cursor

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '01005', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '10011', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '11001', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
```

```
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '11200', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '11510', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '11002', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '01005', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '10011', @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '11101', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '11301', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '11202', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '11501', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '12400', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '02532', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '11204', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '02661', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '12401', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02197', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '12300', @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
```

```
  values(@tppocid, @tppid, null, 1, '02402', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02534', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02701', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10012', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10013', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11511', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '13000', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26026', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '02197', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01034', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '11690', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02402', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02701', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10012', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10013', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11003', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11101', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11202', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11301', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11411', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11501', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11512', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '12102', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '13210', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26026', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
```

```
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '02532', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '02661', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '11204', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '11412', @curdate, @userid, @curdate, @userid)
  end

  close tpp_cursor
  deallocate tpp_cursor
  set @iteration = @iteration + 1
end

-- ------------------------------------------------------------------------------
-- --Design and Innovation 2002
-- ------------------------------------------------------------------------------

set @tpvid = 25

declare tpp_cursor cursor for
select top 8 technicalpackage_period_id
from technicalpackage_period
where technicalpackageversion_id = @tpvid
order by technicalpackage_period_id
for read only
open tpp_cursor

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41010', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41012', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '42011', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41015', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '41010', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '41012', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '42020', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
```

```
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '42021', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01007', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41025', @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41026', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '42020', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41030', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41031', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41035', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '01007', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41040', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '42041', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41045', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
```

```
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41025', @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41026', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

close tpp_cursor
deallocate tpp_cursor
------------------------------------------------------------------------------
--Design and Innovation 2003 - 2010
------------------------------------------------------------------------------

set @iteration = 1
while @iteration < 9
begin
  set @tpvid = @tpvid + 1

  declare tpp_cursor cursor for
  select top 8 technicalpackage_period_id
  from technicalpackage_period
  where technicalpackageversion_id = @tpvid
  order by technicalpackage_period_id
  for read only
  open tpp_cursor

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41010', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41012', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42011', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41015', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41010', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41012', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42020', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
```

```
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '42021', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01007', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41025', @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41026', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '42020', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41030', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41031', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41035', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '01007', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41040', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '42041', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41045', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
```

```
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41025', @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41026', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  end

  close tpp_cursor
  deallocate tpp_cursor
  set @iteration = @iteration + 1
end
-------------------------------------------------------------------------------
--Electrical Engineering 1999
-------------------------------------------------------------------------------

set @tpvid = 34

declare tpp_cursor cursor for
select top 8 technicalpackage_period_id
from technicalpackage_period
where technicalpackageversion_id = @tpvid
order by technicalpackage_period_id
for read only
open tpp_cursor

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01010', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01012', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '49102', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '49104', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '85101', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '49105', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01020', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
```

```
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '49108', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '46150', @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '49111', @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '51150', @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '52150', @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '53150', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '85111', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '46150', @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '49111', @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '51150', @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '52150', @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '53150', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01032', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '49116', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '01032', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '48000', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
```

```
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

close tpp_cursor
deallocate tpp_cursor
------------------------------------------------------------------------------
--Electrical Engineering 2000
------------------------------------------------------------------------------

set @tpvid = 35

declare tpp_cursor cursor for
select top 8 technicalpackage_period_id
from technicalpackage_period
where technicalpackageversion_id = @tpvid
order by technicalpackage_period_id
for read only
open tpp_cursor

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '49102', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '49104', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '49105', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '49108', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '49116', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '?????', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
```

```
  values(@tppciid, @tppcid, null, 2, '?????', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '48000', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '85100', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02401', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02701', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10011', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10100', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '31050', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '31230', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02601', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10010', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10012', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26026', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '31200', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '34320', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '32200', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '32210', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01032', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02643', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10010', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10012', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10370', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26026', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
```

```
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '31220', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '31300', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '31650', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02701', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '10011', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10100', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10405', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '31030', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '31240', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '31405', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '33253', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '31030', @curdate, @userid, @curdate, @userid)
end

close tpp_cursor
deallocate tpp_cursor
-------------------------------------------------------------------------------
--Electrical Engineering 2001
-------------------------------------------------------------------------------

set @tpvid = 36

declare tpp_cursor cursor for
select top 8 technicalpackage_period_id
from technicalpackage_period
where technicalpackageversion_id = @tpvid
order by technicalpackage_period_id
for read only
open tpp_cursor

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '02199', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '31000', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '02199', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
```

```
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '31000', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '31010', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '01016', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02601', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10010', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10011', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '31010', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '31400', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '42510', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02401', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02701', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '10011', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10100', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '31050', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '31230', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02601', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10010', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10012', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26026', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '31200', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
```

```
  values(@tppocid, @tppid, null, 1, '34320', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '32200', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '32210', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01032', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02643', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10010', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10012', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10370', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26026', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '31220', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '31300', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '31650', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02701', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10011', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10100', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10405', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '31030', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '31240', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '31405', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '33253', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '31030', @curdate, @userid, @curdate, @userid)
end

close tpp_cursor
deallocate tpp_cursor
------------------------------------------------------------------------------
--Electrical Engineering 2002
------------------------------------------------------------------------------

set @tpvid = 37

declare tpp_cursor cursor for
select top 8 technicalpackage_period_id
```

```
from technicalpackage_period
where technicalpackageversion_id = @tpvid
order by technicalpackage_period_id
for read only
open tpp_cursor

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '02100', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '31000', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '02100', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '31000', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '31010', @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '34200', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '01016', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02601', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '31010', @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '34200', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
```

```
  values(@tppciid, @tppcid, null, 1, '31400', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '42510', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02401', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02701', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '10011', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10100', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '31050', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '31230', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02601', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10010', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10012', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26026', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '31200', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '34320', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01032', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02643', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10010', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10012', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10370', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '31030', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '31240', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '31405', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '32020', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02701', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10011', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10100', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
```

```
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26026', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '31220', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '31300', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '31650', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '33253', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '31030', @curdate, @userid, @curdate, @userid)
end

close tpp_cursor
deallocate tpp_cursor
-- ------------------------------------------------------------------------------------------
-- --Electrical Engineering 2003 - 2010
-- ------------------------------------------------------------------------------------------

set @iteration = 1
while @iteration < 9
begin
set @tpvid = @tpvid + 1

  declare tpp_cursor cursor for
  select top 8 technicalpackage_period_id
  from technicalpackage_period
  where technicalpackageversion_id = @tpvid
  order by technicalpackage_period_id
  for read only
  open tpp_cursor

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '01005', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02199', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '31000', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '02199', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '01005', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '31000', @curdate, @userid, @curdate, @userid)
```

```
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '31011', @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '34300', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '01016', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02601', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '31011', @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '34300', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '31400', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '42510', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02402', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02701', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '10011', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10100', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '31050', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '31230', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02601', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10010', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10012', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26026', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '31200', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '34320', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
```

```
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '01032', @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '02643', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '10010', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '10012', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '10370', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '26026', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '31030', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '31240', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '31405', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '33253', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '02701', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '10011', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '10100', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '26026', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '31220', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '31300', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '31650', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 2, '31030', @curdate, @userid, @curdate, @userid)
  end

  close tpp_cursor
  deallocate tpp_cursor
  set @iteration = @iteration + 1
end

-------------------------------------------------------------------------------
--Energy Engineering 1999 - 1
-------------------------------------------------------------------------------

set @tpvid = 46

declare tpp_cursor cursor for
select top 8 technicalpackage_period_id
from technicalpackage_period
where technicalpackageversion_id = @tpvid
order by technicalpackage_period_id
for read only
open tpp_cursor

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01010', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
```

```
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01012', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '63001', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '64080', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '64040', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '64082', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01020', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '?????', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '64070', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01012', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '53013', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
```

```
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '64090', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '77141', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

close tpp_cursor
deallocate tpp_cursor
------------------------------------------------------------------------------
--Energy Engineering 1999 - 2
------------------------------------------------------------------------------

set @tpvid = 47

declare tpp_cursor cursor for
select top 8 technicalpackage_period_id
from technicalpackage_period
where technicalpackageversion_id = @tpvid
order by technicalpackage_period_id
for read only
open tpp_cursor

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01010', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01012', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '63001', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '64080', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10016', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '64082', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01020', @curdate, @userid, @curdate, @userid)
```

```
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '?????', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '64070', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01012', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '53013', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '64040', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '64090', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '77141', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

close tpp_cursor
deallocate tpp_cursor
------------------------------------------------------------------------------
--Energy Engineering 2000
------------------------------------------------------------------------------

set @tpvid = 48

declare tpp_cursor cursor for
select top 8 technicalpackage_period_id
from technicalpackage_period
where technicalpackageversion_id = @tpvid
order by technicalpackage_period_id
for read only
open tpp_cursor
```

```
fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10011', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '49104', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '53003', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '64040', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '49105', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '10011', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '53004', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '53004', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10012', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
```

```
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10013', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '53013', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02701', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '12411', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '32200', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '12210', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '28010', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '32210', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '41201', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '77141', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '85100', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '?????', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02401', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11104', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '31600', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10100', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11501', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '32070', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '32610', @curdate, @userid, @curdate, @userid)
end
```

```
close tpp_cursor
deallocate tpp_cursor
------------------------------------------------------------------------------
--Energy Engineering 2001
------------------------------------------------------------------------------

set @tpvid = 49

declare tpp_cursor cursor for
select top 8 technicalpackage_period_id
from technicalpackage_period
where technicalpackageversion_id = @tpvid
order by technicalpackage_period_id
for read only
open tpp_cursor

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10011', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41403', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '02197', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '02197', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '10011', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '11101', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11310', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
```

```
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10012', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10013', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '32060', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02701', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '12411', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '32200', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '12210', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '28010', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '32210', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '41201', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '32065', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41401', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '42510', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02401', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10100', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11104', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11501', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '31600', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
```

```
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '32070', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '32610', @curdate, @userid, @curdate, @userid)
end

close tpp_cursor
deallocate tpp_cursor
------------------------------------------------------------------------------
--Energy Engineering 2002
------------------------------------------------------------------------------

set @tpvid = 50

declare tpp_cursor cursor for
select top 8 technicalpackage_period_id
from technicalpackage_period
where technicalpackageversion_id = @tpvid
order by technicalpackage_period_id
for read only
open tpp_cursor

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '02197', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10011', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41403', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '02197', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '10011', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '11101', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42412', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
```

```
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10012', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10013', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '32060', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10100', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '12411', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '28010', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '41501', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11510', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26026', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '41201', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '12210', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '32220', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '32210', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '32065', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41401', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '42510', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '01030', @curdate, @userid, @curdate, @userid)
```

```
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02701', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11310', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '31600', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '41402', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42110', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02601', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11104', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11501', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '32070', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '41502', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '32610', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42521', @curdate, @userid, @curdate, @userid)
end

close tpp_cursor
deallocate tpp_cursor

------------------------------------------------------------------------------
--Energy Engineering 2003 - 2010
------------------------------------------------------------------------------

set @iteration = 1
while @iteration < 9
begin
  set @tpvid = @tpvid + 1
  declare tpp_cursor cursor for
  select top 8 technicalpackage_period_id
  from technicalpackage_period
  where technicalpackageversion_id = @tpvid
  order by technicalpackage_period_id
  for read only
  open tpp_cursor

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '01005', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02197', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '10011', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41403', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
```

```
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '02197', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '10011', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '11101', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42412', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10012', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10013', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '31760', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10100', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '12411', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '28010', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '41501', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11510', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26026', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '41201', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
```

```
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '12210', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41410', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '31765', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41401', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42510', @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '01030', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '02701', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '11310', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '31600', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41502', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '02601', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '11104', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '11501', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41402', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '42110', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '42521', @curdate, @userid, @curdate, @userid)
  end

  close tpp_cursor
  deallocate tpp_cursor
  set @iteration = @iteration + 1
end

-------------------------------------------------------------------------------
--Information Technology 1999
-------------------------------------------------------------------------------

set @tpvid = 59

declare tpp_cursor cursor for
select top 8 technicalpackage_period_id
from technicalpackage_period
where technicalpackageversion_id = @tpvid
order by technicalpackage_period_id
for read only
open tpp_cursor

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
```

```
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01010', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01012', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01016', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '49135', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '49136', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '49137', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01020', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '04010', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '49142', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '04010', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01030', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
```

```
  values(@tppciid, @tppcid, null, 1, '04040', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '49156', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '04030', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '49156', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '85100', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

close tpp_cursor
deallocate tpp_cursor
------------------------------------------------------------------------------
--Information Technology 2000
------------------------------------------------------------------------------

set @tpvid = 60

declare tpp_cursor cursor for
select top 8 technicalpackage_period_id
from technicalpackage_period
where technicalpackageversion_id = @tpvid
order by technicalpackage_period_id
for read only
open tpp_cursor

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01016', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '49135', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
```

```
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '49137', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '04010', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '49142', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01030', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '04040', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '49156', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '01227', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '01141', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02140', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02501', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '01248', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '01425', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '31000', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02661', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10010', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
```

```
  values(@tppocid, @tppid, null, 1, '02441', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '49156', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '04030', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '85100', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02621', @curdate, @userid, @curdate, @userid)
end

close tpp_cursor
deallocate tpp_cursor
------------------------------------------------------------------------------
--Information Technology 2001
------------------------------------------------------------------------------

set @tpvid = 61

declare tpp_cursor cursor for
select top 8 technicalpackage_period_id
from technicalpackage_period
where technicalpackageversion_id = @tpvid
order by technicalpackage_period_id
for read only
open tpp_cursor

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01016', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '02100', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '02110', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
```

```
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '02110', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '02601', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10010', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26026', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02621', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01030', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '02120', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '02401', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '01227', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '01141', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02140', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02501', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '01248', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '01425', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '31000', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02661', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10010', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
```

```
  values(@tppocid, @tppid, null, 1, '02441', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '02120', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '02701', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '42510', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '31000', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26026', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '01248', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02643', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10013', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02417', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '01250', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02130', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '01248', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02441', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

close tpp_cursor
deallocate tpp_cursor
------------------------------------------------------------------------------
--Information Techology 2002
------------------------------------------------------------------------------

set @tpvid = 62

declare tpp_cursor cursor for
select top 8 technicalpackage_period_id
from technicalpackage_period
where technicalpackageversion_id = @tpvid
order by technicalpackage_period_id
for read only
open tpp_cursor

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01016', @curdate, @userid, @curdate, @userid)
```

```
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '02100', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '02105', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '02100', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '02110', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '02601', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10010', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26026', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02621', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01030', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '02120', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '02402', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
```

```
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '02405', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '01227', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02140', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '31000', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '01425', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10010', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02441', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02661', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '02120', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '02701', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '42510', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '01250', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02411', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02643', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26026', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02130', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02417', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10013', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '31000', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

close tpp_cursor
deallocate tpp_cursor
```

```
-------------------------------------------------------------------------------

set @iteration = 1
while @iteration < 9
begin
  set @tpvid = @tpvid + 1
  declare tpp_cursor cursor for
  select top 8 technicalpackage_period_id
  from technicalpackage_period
  where technicalpackageversion_id = @tpvid
  order by technicalpackage_period_id
  for read only
  open tpp_cursor

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '01005', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '01016', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02100', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02105', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '02100', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '01005', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02110', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02601', @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '10010', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '26026', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
```

```
  values(@tppocid, @tppid, null, 1, '02621', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01030', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '02120', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '02402', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '02405', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '01227', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02140', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '31000', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '01425', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10010', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02441', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02661', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '02120', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '02701', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '42510', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '01250', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02411', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02643', @curdate, @userid, @curdate, @userid)
```

```
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '26026', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '02130', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '02417', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '10013', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 2, '31000', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  end

  close tpp_cursor
  deallocate tpp_cursor
  set @iteration = @iteration + 1
end

-------------------------------------------------------------------------------
--Chemical Engineering 1999 - 1
-------------------------------------------------------------------------------

set @tpvid = 71

declare tpp_cursor cursor for
select top 8 technicalpackage_period_id
from technicalpackage_period
where technicalpackageversion_id = @tpvid
order by technicalpackage_period_id
for read only
open tpp_cursor

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01000', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01012', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '21000', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '21101', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '21210', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '21211', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '85141', @curdate, @userid, @curdate, @userid)
```

```
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01002', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '21061', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '23110', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '85142', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '21102', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10032', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '21103', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10013', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

close tpp_cursor
deallocate tpp_cursor
------------------------------------------------------------------------------
--Chemical Engineering 1999 - 2
------------------------------------------------------------------------------

set @tpvid = 72
```

```
declare tpp_cursor cursor for
select top 8 technicalpackage_period_id
from technicalpackage_period
where technicalpackageversion_id = @tpvid
order by technicalpackage_period_id
for read only
open tpp_cursor

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01000', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '21000', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '21101', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '21210', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '21211', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '21240', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '85141', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01013', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '21061', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '23110', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '85142', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
```

```
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '21002', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01002', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10032', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '21103', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10013', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

close tpp_cursor
deallocate tpp_cursor
------------------------------------------------------------------------------
--Chemical Engineering 2000
------------------------------------------------------------------------------

set @tpvid = 73

declare tpp_cursor cursor for
select top 8 technicalpackage_period_id
from technicalpackage_period
where technicalpackageversion_id = @tpvid
order by technicalpackage_period_id
for read only
open tpp_cursor

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '21001', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
```

```
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '36001', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '23110', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '21002', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '21001', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '36001', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '23110', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '21004', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '21004', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10011', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26406', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26410', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26220', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '28220', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
```

```
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '28120', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '85100', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '26220', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '28220', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '28120', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26203', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26406', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26500', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

close tpp_cursor
deallocate tpp_cursor
------------------------------------------------------------------------------
--Chemical Engineering 2001
------------------------------------------------------------------------------

set @tpvid = 74

declare tpp_cursor cursor for
select top 8 technicalpackage_period_id
from technicalpackage_period
where technicalpackageversion_id = @tpvid
order by technicalpackage_period_id
for read only
open tpp_cursor

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '26001', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '28010', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
```

```
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '26002', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '26001', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '28010', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26300', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26010', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10010', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '26010', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26406', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26410', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26220', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '28220', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '28120', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '42510', @curdate, @userid, @curdate, @userid)
```

```
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '26220', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '28220', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '28120', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26203', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26500', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26406', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

close tpp_cursor
deallocate tpp_cursor

------------------------------------------------------------------------------
--Chemical Engineering 2002
------------------------------------------------------------------------------

set @tpvid = 75

declare tpp_cursor cursor for
select top 8 technicalpackage_period_id
from technicalpackage_period
where technicalpackageversion_id = @tpvid
order by technicalpackage_period_id
for read only
open tpp_cursor

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '26001', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '28010', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '26002', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '26001', @curdate, @userid, @curdate, @userid)
```

```
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '28010', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26300', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26010', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10010', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '26010', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26406', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26410', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26220', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '28220', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '28120', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42110', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '42510', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '26220', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '28220', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '28120', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26203', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26500', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26406', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
```

```
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42110', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

close tpp_cursor
deallocate tpp_cursor
------------------------------------------------------------------------------
--Chemical Engineering 2003 - 2010
------------------------------------------------------------------------------

set @iteration = 1
while @iteration < 9
begin
  set @tpvid = @tpvid + 1
  declare tpp_cursor cursor for
  select top 8 technicalpackage_period_id
  from technicalpackage_period
  where technicalpackageversion_id = @tpvid
  order by technicalpackage_period_id
  for read only
  open tpp_cursor

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '01005', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '26001', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '28010', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '26002', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '01005', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '26001', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '28010', @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '26300', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
```

```
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '26010', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '10010', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '26010', @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '26222', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '26410', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '42110', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '26406', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '28120', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42510', @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '26406', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '28120', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '42110', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '26500', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '28221', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  end

  close tpp_cursor
  deallocate tpp_cursor
  set @iteration = @iteration + 1
end

-------------------------------------------------------------------------------
--Mechanical Engineering 1999 - 1
-------------------------------------------------------------------------------

set @tpvid = 84

declare tpp_cursor cursor for
select top 8 technicalpackage_period_id
```

```
from technicalpackage_period
where technicalpackageversion_id = @tpvid
order by technicalpackage_period_id
for read only
open tpp_cursor

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01010', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01012', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '72001', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '72101', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '77001', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '80001', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '67163', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01020', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10010', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '80002', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
```

```
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10012', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '70101', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '72102', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '85151', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '72102', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '77141', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '77142', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

close tpp_cursor
deallocate tpp_cursor
------------------------------------------------------------------------------
-- Mechanical Engineering 1999 - 2
------------------------------------------------------------------------------

set @tpvid = 85

declare tpp_cursor cursor for
select top 8 technicalpackage_period_id
from technicalpackage_period
where technicalpackageversion_id = @tpvid
order by technicalpackage_period_id
for read only
open tpp_cursor

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01010', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
```

```
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10016', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '72001', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '72101', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '77001', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '80001', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '67163', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01013', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01020', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10010', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '80002', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10012', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '77101', @curdate, @userid, @curdate, @userid)
```

```
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '77102', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '85151', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '72102', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '77141', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '77142', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

close tpp_cursor
deallocate tpp_cursor
------------------------------------------------------------------------------
--Mechanical Engineering 2000
------------------------------------------------------------------------------

set @tpvid = 86

declare tpp_cursor cursor for
select top 8 technicalpackage_period_id
from technicalpackage_period
where technicalpackageversion_id = @tpvid
order by technicalpackage_period_id
for read only
open tpp_cursor

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10011', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '72103', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
```

```
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '80001', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '72103', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '10011', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '80002', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10012', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '77101', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42222', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42405', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42520', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42220', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '41201', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42503', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42110', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42211', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
```

```
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42341', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '42222', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42250', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42120', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02661', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '72102', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '77141', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '77142', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42520', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '41502', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '41611', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42110', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '41512', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '51271', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '41271', @curdate, @userid, @curdate, @userid)
end

close tpp_cursor
deallocate tpp_cursor
------------------------------------------------------------------------------
--Mechanical Engineering 2001
------------------------------------------------------------------------------

set @tpvid = 87

declare tpp_cursor cursor for
select top 8 technicalpackage_period_id
from technicalpackage_period
where technicalpackageversion_id = @tpvid
order by technicalpackage_period_id
for read only
open tpp_cursor

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
```

```
  values(@tppciid, @tppcid, null, 1, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10011', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41601', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '42301', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '41601', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '10011', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '42302', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42110', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42520', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42521', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42341', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02661', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10012', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
```

```
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41501', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42222', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42405', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42520', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42220', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '41201', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '41503', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42110', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42211', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42341', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '42222', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42250', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42120', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02661', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41401', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41602', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41861', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42520', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '41502', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '41611', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42415', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42110', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '41512', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '41271', @curdate, @userid, @curdate, @userid)
```

```
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '41271', @curdate, @userid, @curdate, @userid)
end

close tpp_cursor
deallocate tpp_cursor

------------------------------------------------------------------------------
--Mechanical Engineering 2002
------------------------------------------------------------------------------

set @tpvid = 88

declare tpp_cursor cursor for
select top 8 technicalpackage_period_id
from technicalpackage_period
where technicalpackageversion_id = @tpvid
order by technicalpackage_period_id
for read only
open tpp_cursor

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10011', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41601', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '42301', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '41601', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '10011', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '42302', @curdate, @userid, @curdate, @userid)
```

```
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42110', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42412', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02661', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42341', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42120', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42521', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10012', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41501', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '41201', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '41503', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42211', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42222', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42412', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '41260', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42110', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42220', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42405', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42415', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02661', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '42222', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42341', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42120', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
```

```
  values(@tppocid, @tppid, null, 1, '42250', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41401', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41602', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41861', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '41260', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '41271', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42110', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42415', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '41261', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '41502', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42412', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02661', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42226', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42120', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42521', @curdate, @userid, @curdate, @userid)
end

close tpp_cursor
deallocate tpp_cursor
------------------------------------------------------------------------------
--Mechanical Engineering 2003 - 2010
------------------------------------------------------------------------------

set @iteration = 1
while @iteration < 9
begin
  set @tpvid = @tpvid + 1
  declare tpp_cursor cursor for
  select top 8 technicalpackage_period_id
  from technicalpackage_period
  where technicalpackageversion_id = @tpvid
  order by technicalpackage_period_id
  for read only
  open tpp_cursor

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '01005', @curdate, @userid, @curdate, @userid)
```

```
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '10011', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41601', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42302', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41601', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '01005', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '10011', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42301', @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '42262', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '42412', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '42110', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '02661', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '42341', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '42120', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '42521', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '10012', @curdate, @userid, @curdate, @userid)
```

```
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41501', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '41260', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42110', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42222', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42412', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '41503', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42220', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42405', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42415', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02661', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '42222', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42341', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42120', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42250', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41401', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41602', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41861', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '41262', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '41502', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42412', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '41271', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42110', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
```

```
    values(@tppocid, @tppid, null, 1, '42415', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '02661', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '42226', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '42120', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '42521', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 2, '41262', @curdate, @userid, @curdate, @userid)
  end

  close tpp_cursor
  deallocate tpp_cursor
  set @iteration = @iteration + 1
end

------------------------------------------------------------------------------
--Environmental Engineering 1999 - 1
------------------------------------------------------------------------------

set @tpvid = 97

declare tpp_cursor cursor for
select top 8 technicalpackage_period_id
from technicalpackage_period
where technicalpackageversion_id = @tpvid
order by technicalpackage_period_id
for read only
open tpp_cursor

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01010', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '21240', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '56061', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '64080', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '56062', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01013', @curdate, @userid, @curdate, @userid)
```

```
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '21000', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '56002', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '63190', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '?????', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01020', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '85131', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '64070', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '64090', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

close tpp_cursor
deallocate tpp_cursor
------------------------------------------------------------------------------
--Environmental Engineering 1999 - 2
------------------------------------------------------------------------------

set @tpvid = 98

declare tpp_cursor cursor for
```

```
select top 8 technicalpackage_period_id
from technicalpackage_period
where technicalpackageversion_id = @tpvid
order by technicalpackage_period_id
for read only
open tpp_cursor

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01010', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '21000', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '56059', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '63001', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '64080', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '56060', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01013', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '56002', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '63190', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '?????', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
```

```
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01020', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '85131', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '64070', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '64090', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

close tpp_cursor
deallocate tpp_cursor
------------------------------------------------------------------------------
--Environmental Engineering 2000
------------------------------------------------------------------------------

set @tpvid = 99

declare tpp_cursor cursor for
select top 8 technicalpackage_period_id
from technicalpackage_period
where technicalpackageversion_id = @tpvid
order by technicalpackage_period_id
for read only
open tpp_cursor

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '21000', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '21003', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '56059', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
```

```
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '63001', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '56059', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '56002', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '85100', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '57002', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '01030', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02199', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02501', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10010', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11302', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '12420', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26201', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '27721', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42372', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '88890', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '88995', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02197', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
```

```
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02401', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02531', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10011', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '12411', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26100', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26400', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '31200', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42412', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '88894', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '88996', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '02199', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42630', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '12210', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '88383', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '02197', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '63130', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '?????', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '01030', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10010', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10012', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11304', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '26400', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '27931', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '88890', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '88996', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02401', @curdate, @userid, @curdate, @userid)
```

```
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '10011', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10013', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26201', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26470', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42412', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '88995', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02531', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '11304', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '26470', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42521', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02661', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '12422', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '31250', @curdate, @userid, @curdate, @userid)
end

close tpp_cursor
deallocate tpp_cursor
------------------------------------------------------------------------------------------
--Enviromental Engineering 2001
------------------------------------------------------------------------------------------

set @tpvid = 100

declare tpp_cursor cursor for
select top 8 technicalpackage_period_id
from technicalpackage_period
where technicalpackageversion_id = @tpvid
order by technicalpackage_period_id
for read only
open tpp_cursor

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '12200', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '26000', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '26020', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
```

```
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '12200', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '12400', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02391', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11301', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26201', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26372', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '88890', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '88996', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02401', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '12211', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26370', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '88888', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '88995', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02531', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11402', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '12300', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '42510', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '01030', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
```

```
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02199', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02501', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10010', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11302', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '12420', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26201', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '27721', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42372', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '88890', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '88995', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02197', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02401', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '02531', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10011', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '12411', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26100', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26400', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '31200', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42412', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '88894', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '88996', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '02199', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '02197', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42630', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '12210', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '88383', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '12102', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
```

```
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '12201', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '01030', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10010', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10012', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11304', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '26400', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '27931', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '88890', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '88996', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02401', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '10011', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10013', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26201', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26470', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42412', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '88995', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02531', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '11304', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '26470', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42521', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02661', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '12422', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '31250', @curdate, @userid, @curdate, @userid)
end

close tpp_cursor
deallocate tpp_cursor

------------------------------------------------------------------------------
--Environmental Engineering 2002
------------------------------------------------------------------------------

set @tpvid = 101

declare tpp_cursor cursor for
select top 8 technicalpackage_period_id
from technicalpackage_period
where technicalpackageversion_id = @tpvid
order by technicalpackage_period_id
for read only
open tpp_cursor
```

```
fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '12200', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '26000', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '26020', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '12200', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '12400', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02391', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02405', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26201', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26372', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '88890', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '88996', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02402', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11301', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26370', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '27931', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '88995', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
```

```
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02531', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11402', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '12300', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '42510', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '01030', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02199', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02405', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10010', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11302', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '12420', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26100', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26400', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '31200', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42412', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '88890', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '88995', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02197', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02402', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02501', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10011', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '12411', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '27931', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26201', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '27721', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42372', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42533', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '88894', @curdate, @userid, @curdate, @userid)
```

```
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '88996', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '02199', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '02197', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42630', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '12210', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '88383', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '12102', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '12201', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '01030', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02402', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02405', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10010', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '10011', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10012', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10013', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11304', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26201', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '26400', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26470', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '27931', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42412', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42533', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '88890', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '88995', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '88996', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
end
```

```
fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02661', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '12422', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '31250', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '11304', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '26470', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42521', @curdate, @userid, @curdate, @userid)
end

close tpp_cursor
deallocate tpp_cursor
------------------------------------------------------------------------------
--Environmental Engineering 2003 - 2010
------------------------------------------------------------------------------

set @iteration = 1
while @iteration < 9
begin
  set @tpvid = @tpvid + 1
  declare tpp_cursor cursor for
  select top 8 technicalpackage_period_id
  from technicalpackage_period
  where technicalpackageversion_id = @tpvid
  order by technicalpackage_period_id
  for read only
  open tpp_cursor

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '01005', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '12200', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '26000', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '26020', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '12200', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '01005', @curdate, @userid, @curdate, @userid)
```

```
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '12400', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02312', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02402', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02405', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26201', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26372', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '88890', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '88996', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11301', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26370', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '27931', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '88995', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '12102', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '02312', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '12401', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02532', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '12300', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '42510', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '01030', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02197', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02199', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02402', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02405', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02501', @curdate, @userid, @curdate, @userid)
```

```
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '10010', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '10011', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '11302', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '11422', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '12411', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '12420', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '26201', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '26400', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '26470', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '27001', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '27721', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '27931', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '31200', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '42533', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '42412', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '42533', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '88890', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '88995', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '88996', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 2, '02197', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 2, '02199', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 2, '11422', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '12210', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '26002', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 2, '26470', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 2, '27001', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '42630', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '88383', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
```

```
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '12102', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '12201', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '01030', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02401', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10010', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '10011', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10012', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10013', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11304', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '11421', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 3, '11422', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '12400', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26201', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '26400', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '26470', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 3, '27001', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '27931', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '31240', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42412', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42533', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '88890', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '88995', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '88996', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02661', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '11304', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 4, '11422', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '12401', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
```

```
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 2, '26470', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '31250', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '42521', @curdate, @userid, @curdate, @userid)
  end

  close tpp_cursor
  deallocate tpp_cursor
  set @iteration = @iteration + 1
end

------------------------------------------------------------------------------
-- Technical Physics 1999 - 1
------------------------------------------------------------------------------

set @tpvid = 110

declare tpp_cursor cursor for
select top 8 technicalpackage_period_id
from technicalpackage_period
where technicalpackageversion_id = @tpvid
order by technicalpackage_period_id
for read only
open tpp_cursor

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01010', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01012', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10001', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10004', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '85112', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '10001', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01020', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10002', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
```

```
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10005', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01030', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10014', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10061', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10100', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10015', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10062', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

close tpp_cursor
deallocate tpp_cursor
------------------------------------------------------------------------------
-- Technical Physics 1999 - 2
------------------------------------------------------------------------------

set @tpvid = 111

declare tpp_cursor cursor for
select top 8 technicalpackage_period_id
from technicalpackage_period
where technicalpackageversion_id = @tpvid
order by technicalpackage_period_id
for read only
open tpp_cursor

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
```

```
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01010', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01012', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10003', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10004', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10016', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10018', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '85112', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '10003', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '10018', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01020', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10010', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10019', @curdate, @userid, @curdate, @userid)
end
```

```
fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01030', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10014', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10061', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10100', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10015', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10062', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

close tpp_cursor
deallocate tpp_cursor
------------------------------------------------------------------------------
-- Technical Physics 2000
------------------------------------------------------------------------------

set @tpvid = 112

declare tpp_cursor cursor for
select top 8 technicalpackage_period_id
from technicalpackage_period
where technicalpackageversion_id = @tpvid
order by technicalpackage_period_id
for read only
open tpp_cursor

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
```

```
  values(@tppciid, @tppcid, null, 1, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10001', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10004', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '85100', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '10001', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10002', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10005', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01030', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10014', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10061', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02601', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '01016', @curdate, @userid, @curdate, @userid)
```

```
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42520', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42301', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02661', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42302', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10100', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10015', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10062', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02199', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '01141', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '01142', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02402', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '02199', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '01257', @curdate, @userid, @curdate, @userid)
end

close tpp_cursor
deallocate tpp_cursor
------------------------------------------------------------------------------
-- Technical Physics 2001
------------------------------------------------------------------------------

set @tpvid = 113

declare tpp_cursor cursor for
select top 8 technicalpackage_period_id
from technicalpackage_period
where technicalpackageversion_id = @tpvid
order by technicalpackage_period_id
for read only
open tpp_cursor

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
```

```
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10001', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10004', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '10001', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 3, '10001', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '10004', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02601', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '01016', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42520', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42301', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02661', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42302', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01030', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10014', @curdate, @userid, @curdate, @userid)
```

```
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10061', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02199', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '01141', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '01142', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '01402', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '02199', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '01257', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '10061', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10100', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '42510', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '41501', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10370', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '33253', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10454', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02643', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10467', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '33430', @curdate, @userid, @curdate, @userid)
end

close tpp_cursor
deallocate tpp_cursor
```

```
set @tpvid = 114

declare tpp_cursor cursor for
select top 8 technicalpackage_period_id
from technicalpackage_period
where technicalpackageversion_id = @tpvid
order by technicalpackage_period_id
for read only
open tpp_cursor

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10001', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10004', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '10001', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '01005', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 3, '10001', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '10004', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02601', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '01016', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42412', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02661', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42302', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
```

```
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01030', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10009', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10061', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02199', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '01142', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02405', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10405', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '42301', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '01141', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02402', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10341', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '41501', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 2, '02199', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '01257', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '10061', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10100', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '42510', @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '02643', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10370', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
```

```
  values(@tppocid, @tppid, null, 1, '33253', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10545', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '32450', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '10467', @curdate, @userid, @curdate, @userid)
  set @tppocid = @tppocid + 1
  insert into technicalpackage_periodoptionalcourse
  values(@tppocid, @tppid, null, 1, '33430', @curdate, @userid, @curdate, @userid)
end

close tpp_cursor
deallocate tpp_cursor
------------------------------------------------------------------------------
--Technical Physics 2003 - 2010
------------------------------------------------------------------------------

set @iteration = 1
while @iteration < 9
begin
  set @tpvid = @tpvid + 1
  declare tpp_cursor cursor for
  select top 8 technicalpackage_period_id
  from technicalpackage_period
  where technicalpackageversion_id = @tpvid
  order by technicalpackage_period_id
  for read only
  open tpp_cursor

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '01005', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '10001', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '10004', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '10001', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '01005', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 3, '10001', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
```

```
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '10004', @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '01016', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '02601', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '10454', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '42412', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '02661', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '42302', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '01030', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '10009', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '10061', @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '01141', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '02199', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '02402', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '02405', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '10405', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41501', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '42301', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 2, '02199', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '01257', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
```

```
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '10061', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '10100', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42510', @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '02643', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '10370', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '10454', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '33253', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '10467', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '33430', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '33470', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '33471', @curdate, @userid, @curdate, @userid)
  end

  close tpp_cursor
  deallocate tpp_cursor
  set @iteration = @iteration + 1
end

-------------------------------------------------------------------------------
--Medication and Technology 2003 - 2010
-------------------------------------------------------------------------------

set @tpvid = 122

set @iteration = 1
while @iteration < 9
begin
  set @tpvid = @tpvid + 1
  declare tpp_cursor cursor for
  select top 8 technicalpackage_period_id
  from technicalpackage_period
  where technicalpackageversion_id = @tpvid
  order by technicalpackage_period_id
  for read only
  open tpp_cursor

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '01005', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '31654', @curdate, @userid, @curdate, @userid)
--KU kurser....
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
```

```
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--KU kurser....
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '01005', @curdate, @userid, @curdate, @userid)
--KU kurser....
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '10010', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '31600', @curdate, @userid, @curdate, @userid)
--KU kurser....
 end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '01032', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '31650', @curdate, @userid, @curdate, @userid)
--KU kurser....
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--KU kurser....
  end

  close tpp_cursor
  deallocate tpp_cursor
  set @iteration = @iteration + 1
end

-------------------------------------------------------------------------------
--Construction Engineering (Bachelor of Science) 2003 - 2010
-------------------------------------------------------------------------------

set @tpvid = 130

set @iteration = 1
while @iteration < 9
begin
  set @tpvid = @tpvid + 1
  declare tpp_cursor cursor for
```

```
select top 14 technicalpackage_period_id
from technicalpackage_period
where technicalpackageversion_id = @tpvid
order by technicalpackage_period_id
for read only
open tpp_cursor

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '11713', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10914', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01905', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '11714', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '11715', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10921', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '12700', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '11721', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '02535', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '11731', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
```

```
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '11732', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '11733', @curdate, @userid, @curdate, @userid)
 end

 fetch next from tpp_cursor into @tppid
 if @@fetch_status = 0
 begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '11733', @curdate, @userid, @curdate, @userid)
 end

 fetch next from tpp_cursor into @tppid
 if @@fetch_status = 0
 begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '11742', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '11743', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '11744', @curdate, @userid, @curdate, @userid)
 end

 fetch next from tpp_cursor into @tppid
 if @@fetch_status = 0
 begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '11745', @curdate, @userid, @curdate, @userid)
 end

 fetch next from tpp_cursor into @tppid
 if @@fetch_status = 0
 begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--project
 end

 fetch next from tpp_cursor into @tppid
 if @@fetch_status = 0
 begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--project
 end

 fetch next from tpp_cursor into @tppid
 if @@fetch_status = 0
 begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '11761', @curdate, @userid, @curdate, @userid)
 end

 fetch next from tpp_cursor into @tppid
 if @@fetch_status = 0
 begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
```

```
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '11762', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '11799', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '11799', @curdate, @userid, @curdate, @userid)
  end

  close tpp_cursor
  deallocate tpp_cursor
  set @iteration = @iteration + 1
end

---

set @tpvid = 138

set @iteration = 1
while @iteration < 9
begin
  set @tpvid = @tpvid + 1
  declare tpp_cursor cursor for
  select top 14 technicalpackage_period_id
  from technicalpackage_period
  where technicalpackageversion_id = @tpvid
  order by technicalpackage_period_id
  for read only
  open tpp_cursor

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '11713', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '10914', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '01905', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '11714', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '11715', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
```

```
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '10921', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '12700', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '11721', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02535', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '11731', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '11732', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '11733', @curdate, @userid, @curdate, @userid)
end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '11733', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '11742', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '11743', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '11744', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
```

```
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '11745', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--project
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--project
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '11761', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '11762', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '11799', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '11799', @curdate, @userid, @curdate, @userid)
  end

  close tpp_cursor
  deallocate tpp_cursor
  set @iteration = @iteration + 1
end

-------------------------------------------------------------------------------
--Architectural Engineering (Bachelor of Science) 2003 - 2010
-------------------------------------------------------------------------------

set @tpvid = 146

set @iteration = 1
while @iteration < 9
begin
  set @tpvid = @tpvid + 1
  declare tpp_cursor cursor for
  select top 14 technicalpackage_period_id
  from technicalpackage_period
  where technicalpackageversion_id = @tpvid
  order by technicalpackage_period_id
  for read only
  open tpp_cursor

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
```

```
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01931', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '11911', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '11913', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '11915', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '11912', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '11912', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01932', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '11921', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '11913', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '11915', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '11924', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '11924', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
```

```
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10931', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '11931', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '11937', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '11933', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '11935', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '11932', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '11932', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '11946', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '11941', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '11947', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '11933', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '11935', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
```

```
    values(@tppciid, @tppcid, null, 1, '11944', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '11944', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--project
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--project
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '11969', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '1xxxx', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '11969', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '11979', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '11979', @curdate, @userid, @curdate, @userid)
  end

  close tpp_cursor
  deallocate tpp_cursor
  set @iteration = @iteration + 1
end

------------------------------------------------------------------------------
--Electrical Engineering (Bachelor of Science) 2003 - 2010
------------------------------------------------------------------------------

set @tpvid = 154

set @iteration = 1
```

```
while @iteration < 9
begin
  set @tpvid = @tpvid + 1
  declare tpp_cursor cursor for
  select top 14 technicalpackage_period_id
  from technicalpackage_period
  where technicalpackageversion_id = @tpvid
  order by technicalpackage_period_id
  for read only
  open tpp_cursor

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '01905', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02318', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '31020', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '31020', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '31025', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '31022', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '31027', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '31027', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '01962', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
```

```
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02323', @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '01963', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '32020', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '31028', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '31024', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--no courses
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '31300', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '31029', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '31023', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '31023', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '31030', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '31030', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
```

```
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--project
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--project
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--project
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--project
  end

  close tpp_cursor
  deallocate tpp_cursor
  set @iteration = @iteration + 1
end

--

set @tpvid = 162

set @iteration = 1
while @iteration < 9
begin
  set @tpvid = @tpvid + 1
  declare tpp_cursor cursor for
  select top 14 technicalpackage_period_id
  from technicalpackage_period
  where technicalpackageversion_id = @tpvid
  order by technicalpackage_period_id
  for read only
  open tpp_cursor

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '01905', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02318', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '31020', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '31020', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
```

```
    values(@tppciid, @tppcid, null, 1, '01962', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '31022', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '31027', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '31027', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '31025', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02323', @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '01963', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '31029', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '31023', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '31024', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '31350', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '31300', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '31028', @curdate, @userid, @curdate, @userid)
```

```
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '31030', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '31030', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--no courses
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--no courses
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--project
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--project
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--project
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--project
  end

  close tpp_cursor
  deallocate tpp_cursor
  set @iteration = @iteration + 1
end
-------------------------------------------------------------------
--Chemical Engineering (Bachelor of Science) 2003 - 2010
-------------------------------------------------------------------

set @tpvid = 170

set @iteration = 1
while @iteration < 9
begin
  set @tpvid = @tpvid + 1
  declare tpp_cursor cursor for
  select top 14 technicalpackage_period_id
  from technicalpackage_period
  where technicalpackageversion_id = @tpvid
  order by technicalpackage_period_id
  for read only
  open tpp_cursor

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
```

```
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10911', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01905', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '26170', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '26370', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '28011', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '28011', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01902', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '26270', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '26372', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '26470', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '26172', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '26470', @curdate, @userid, @curdate, @userid)
end
```

```
  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '26272', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '26472', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '27931', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '28021', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '28021', @curdate, @userid, @curdate, @userid)

  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '10941', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02591', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42891', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '27942', @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '28153', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '27942', @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '28153', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
```

```
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '28171', @curdate, @userid, @curdate, @userid)
--project
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--project
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '28341', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '28351', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '28321', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '27961', @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42983', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--nocourses
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '28381', @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '28382', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '28381', @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '28382', @curdate, @userid, @curdate, @userid)
  end

  close tpp_cursor
  deallocate tpp_cursor
  set @iteration = @iteration + 1
end

--

set @tpvid = 178

set @iteration = 1
while @iteration < 9
begin
```

```
set @tpvid = @tpvid + 1
declare tpp_cursor cursor for
select top 14 technicalpackage_period_id
from technicalpackage_period
where technicalpackageversion_id = @tpvid
order by technicalpackage_period_id
for read only
open tpp_cursor

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '10911', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01905', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '26170', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '26370', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '28011', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '28011', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01902', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '26270', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '26372', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '26470', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '26172', @curdate, @userid, @curdate, @userid)
end
```

```
  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '26470', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '26272', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '26472', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '27931', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '28021', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '28021', @curdate, @userid, @curdate, @userid)

  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '10941', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02591', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42891', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '27942', @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '28153', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
```

```
    values(@tppciid, @tppcid, null, 2, '27942', @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '28153', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '28171', @curdate, @userid, @curdate, @userid)
--project
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--project
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '28341', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '28351', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '28321', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '27961', @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42983', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--nocourses
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '28381', @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '28382', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '28381', @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '28382', @curdate, @userid, @curdate, @userid)
  end
```

```
  close tpp_cursor
  deallocate tpp_cursor
  set @iteration = @iteration + 1
end

-------------------------------------------------------------------------------
--Information Technology (Bachelor of Science) 2003 - 2010
-------------------------------------------------------------------------------

set @tpvid = 186

set @iteration = 1
while @iteration < 9
begin
  set @tpvid = @tpvid + 1
  declare tpp_cursor cursor for
  select top 14 technicalpackage_period_id
  from technicalpackage_period
  where technicalpackageversion_id = @tpvid
  order by technicalpackage_period_id
  for read only
  open tpp_cursor

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '01961', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '31021', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02311', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02312', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '02312', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '01016', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02323', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02322', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02321', @curdate, @userid, @curdate, @userid)
  end
```

```
  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02329', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02335', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02333', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02330', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '02330', @curdate, @userid, @curdate, @userid)

  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02340', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02348', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02344', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '02344', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02355', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
```

```
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '02355', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--project
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--project
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02371', @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02372', @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02373', @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02374', @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02375', @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02376', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--project
  end

  close tpp_cursor
  deallocate tpp_cursor
  set @iteration = @iteration + 1
end

--

set @tpvid = 194

set @iteration = 1
while @iteration < 9
begin
  set @tpvid = @tpvid + 1
  declare tpp_cursor cursor for
  select top 14 technicalpackage_period_id
  from technicalpackage_period
  where technicalpackageversion_id = @tpvid
  order by technicalpackage_period_id
  for read only
  open tpp_cursor

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '01961', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '31021', @curdate, @userid, @curdate, @userid)
```

```
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02311', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02312', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '02312', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '01016', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02323', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02322', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02321', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02329', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02335', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02333', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02330', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
```

```
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '02330', @curdate, @userid, @curdate, @userid)

  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02340', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02348', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02344', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '02344', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02355', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '02355', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--project
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--project
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02371', @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02372', @curdate, @userid, @curdate, @userid)
```

```
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02373', @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02374', @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02375', @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02376', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--project
  end

  close tpp_cursor
  deallocate tpp_cursor
  set @iteration = @iteration + 1
end

-------------------------------------------------------------------------------
--Mechanical Engineering 1 (Bachelor of Science) 2003 - 2010
-------------------------------------------------------------------------------

set @tpvid = 202

set @iteration = 1
while @iteration < 9
begin
  set @tpvid = @tpvid + 1
  declare tpp_cursor cursor for
  select top 14 technicalpackage_period_id
  from technicalpackage_period
  where technicalpackageversion_id = @tpvid
  order by technicalpackage_period_id
  for read only
  open tpp_cursor

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '01905', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41630', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41532', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42302', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41533', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41431', @curdate, @userid, @curdate, @userid)
```

```
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42301', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42110', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02323', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41431', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41535', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42201', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41640', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41841', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41640', @curdate, @userid, @curdate, @userid)

  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41636', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41534', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
```

```
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41639', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42953', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41639', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--praktik
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--praktik
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41660', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41625', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41511', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41312', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41812', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41661', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41614', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
```

```
    values(@tppocid, @tppid, null, 1, '41611', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41560', @curdate, @userid, @curdate, @userid)
--project
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 2, '41812', @curdate, @userid, @curdate, @userid)
--project
  end

  close tpp_cursor
  deallocate tpp_cursor
  set @iteration = @iteration + 1
end

--

set @tpvid = 210

set @iteration = 1
while @iteration < 9
begin
  set @tpvid = @tpvid + 1
  declare tpp_cursor cursor for
  select top 14 technicalpackage_period_id
  from technicalpackage_period
  where technicalpackageversion_id = @tpvid
  order by technicalpackage_period_id
  for read only
  open tpp_cursor

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '01905', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41630', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41532', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42302', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41533', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41431', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
```

```
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42301', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42110', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02323', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41431', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41636', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41534', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41639', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41639', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41535', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42201', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41640', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41841', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
```

```
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41640', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--praktik
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--praktik
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41812', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41661', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41614', @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41611', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41560', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41812', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41660', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41625', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41511', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41312', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41814', @curdate, @userid, @curdate, @userid)
--project
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
```

```
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--project
  end

  close tpp_cursor
  deallocate tpp_cursor
  set @iteration = @iteration + 1
end

-------------------------------------------------------------------------------
--Mechanical Engineering 2 (Bachelor of Science) 2003 - 2010
-------------------------------------------------------------------------------

set @tpvid = 218

set @iteration = 1
while @iteration < 9
begin
  set @tpvid = @tpvid + 1
  declare tpp_cursor cursor for
  select top 14 technicalpackage_period_id
  from technicalpackage_period
  where technicalpackageversion_id = @tpvid
  order by technicalpackage_period_id
  for read only
  open tpp_cursor

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '01905', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41630', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41532', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42302', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41533', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41431', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42301', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42110', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
```

```
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02323', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41431', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41535', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42201', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41640', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41841', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41640', @curdate, @userid, @curdate, @userid)

  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41636', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41534', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41639', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42953', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
```

```
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41639', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--praktik
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--praktik
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41212', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41511', @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41313', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41260', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41272', @curdate, @userid, @curdate, @userid)

  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41271', @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41812', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41813', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41213', @curdate, @userid, @curdate, @userid)
--project
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 2, '41812', @curdate, @userid, @curdate, @userid)
--project
  end

  close tpp_cursor
  deallocate tpp_cursor
```

```
  set @iteration = @iteration + 1
end

--

set @tpvid = 226

set @iteration = 1
while @iteration < 9
begin
  set @tpvid = @tpvid + 1
  declare tpp_cursor cursor for
  select top 14 technicalpackage_period_id
  from technicalpackage_period
  where technicalpackageversion_id = @tpvid
  order by technicalpackage_period_id
  for read only
  open tpp_cursor

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '01905', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41630', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41532', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42302', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41533', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41431', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42301', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42110', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02323', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
```

```
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41431', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41636', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41534', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41639', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41639', @curdate, @userid, @curdate, @userid)

  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41535', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42201', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41640', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41841', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41640', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--praktik
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
```

```
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--praktik
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41271', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41812', @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41813', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41213', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41812', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41212', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41511', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41313', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41260', @curdate, @userid, @curdate, @userid)
--project
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--project
  end

  close tpp_cursor
  deallocate tpp_cursor
  set @iteration = @iteration + 1
end

-------------------------------------------------------------------------------
--Mechanical Engineering 3 (Bachelor of Science) 2003 - 2010
-------------------------------------------------------------------------------

set @tpvid = 234

set @iteration = 1
while @iteration < 9
begin
  set @tpvid = @tpvid + 1
  declare tpp_cursor cursor for
  select top 14 technicalpackage_period_id
  from technicalpackage_period
  where technicalpackageversion_id = @tpvid
  order by technicalpackage_period_id
  for read only
  open tpp_cursor

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
```

```
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01905', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41630', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41532', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '42302', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41533', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41431', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '42301', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '42110', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '02323', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '41431', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41535', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
```

```
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42201', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41640', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41841', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41640', @curdate, @userid, @curdate, @userid)

end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41636', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41534', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41639', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42953', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41639', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--praktik
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--praktik
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
```

```
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41814', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41312', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41313', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41410', @curdate, @userid, @curdate, @userid)

  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41415', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41813', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41815', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41323', @curdate, @userid, @curdate, @userid)
--project
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--project
  end

  close tpp_cursor
  deallocate tpp_cursor
  set @iteration = @iteration + 1
end

--

set @tpvid = 242

set @iteration = 1
while @iteration < 9
begin
  set @tpvid = @tpvid + 1
  declare tpp_cursor cursor for
  select top 14 technicalpackage_period_id
  from technicalpackage_period
  where technicalpackageversion_id = @tpvid
  order by technicalpackage_period_id
  for read only
  open tpp_cursor

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '01905', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41630', @curdate, @userid, @curdate, @userid)
```

```
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41532', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42302', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41533', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41431', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42301', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42110', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02323', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41431', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41636', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41534', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41639', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
```

```
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41639', @curdate, @userid, @curdate, @userid)

  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41535', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42201', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41640', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41841', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41640', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--praktik
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--praktik
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41415', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41813', @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41815', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41323', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
```

```
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41814', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41312', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41313', @curdate, @userid, @curdate, @userid)
--project
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41410', @curdate, @userid, @curdate, @userid)
--project
  end

  close tpp_cursor
  deallocate tpp_cursor
  set @iteration = @iteration + 1
end

-------------------------------------------------------------------------------
--Mechanical Engineering 4 (Bachelor of Science) 2003 - 2010
-------------------------------------------------------------------------------

set @tpvid = 250

set @iteration = 1
while @iteration < 9
begin
  set @tpvid = @tpvid + 1
  declare tpp_cursor cursor for
  select top 14 technicalpackage_period_id
  from technicalpackage_period
  where technicalpackageversion_id = @tpvid
  order by technicalpackage_period_id
  for read only
  open tpp_cursor

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '01905', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41630', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41532', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42302', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
```

```
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41533', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41431', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '42301', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '42110', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '02323', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '41431', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41535', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '42201', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41640', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41841', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '41640', @curdate, @userid, @curdate, @userid)

end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
```

```
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41636', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41534', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41639', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42953', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41639', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--praktik
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--praktik
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41660', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41312', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41625', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41511', @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41814', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  end
```

```
  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41812', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41661', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41614', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41611', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41560', @curdate, @userid, @curdate, @userid)
--project
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 2, '41812', @curdate, @userid, @curdate, @userid)
--project
  end

  close tpp_cursor
  deallocate tpp_cursor
  set @iteration = @iteration + 1
end

--

set @tpvid = 258

set @iteration = 1
while @iteration < 9
begin
  set @tpvid = @tpvid + 1
  declare tpp_cursor cursor for
  select top 14 technicalpackage_period_id
  from technicalpackage_period
  where technicalpackageversion_id = @tpvid
  order by technicalpackage_period_id
  for read only
  open tpp_cursor

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '01905', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41630', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41532', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42302', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
```

```
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41533', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41431', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42301', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42110', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02323', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41431', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41636', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41534', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41639', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42953', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41639', @curdate, @userid, @curdate, @userid)

  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
```

```
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41535', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42201', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41262', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41841', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41262', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--praktik
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--praktik
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41812', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41611', @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41661', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41614', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41560', @curdate, @userid, @curdate, @userid)

  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41812', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
```

```
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41660', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41312', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41625', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41511', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41814', @curdate, @userid, @curdate, @userid)
--project
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--project
  end

  close tpp_cursor
  deallocate tpp_cursor
  set @iteration = @iteration + 1
end

-------------------------------------------------------------------------------
--Mechanical Engineering 5 (Bachelor of Science) 2003 - 2010
-------------------------------------------------------------------------------

set @tpvid = 266

set @iteration = 1
while @iteration < 9
begin
  set @tpvid = @tpvid + 1
  declare tpp_cursor cursor for
  select top 14 technicalpackage_period_id
  from technicalpackage_period
  where technicalpackageversion_id = @tpvid
  order by technicalpackage_period_id
  for read only
  open tpp_cursor

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '01905', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41630', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41532', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42302', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41533', @curdate, @userid, @curdate, @userid)
```

```
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41431', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42301', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42110', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02323', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41431', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41535', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42201', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41262', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41841', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41262', @curdate, @userid, @curdate, @userid)

  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41636', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
```

```
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41534', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41639', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42953', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41639', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--praktik
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--praktik
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41212', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41511', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41313', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41260', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41813', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
```

```
    values(@tppocid, @tppid, null, 1, '41812', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41213', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41212', @curdate, @userid, @curdate, @userid)
--project
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 2, '41812', @curdate, @userid, @curdate, @userid)
--project
  end

  close tpp_cursor
  deallocate tpp_cursor
  set @iteration = @iteration + 1
end

--

set @tpvid = 274

set @iteration = 1
while @iteration < 9
begin
  set @tpvid = @tpvid + 1
  declare tpp_cursor cursor for
  select top 14 technicalpackage_period_id
  from technicalpackage_period
  where technicalpackageversion_id = @tpvid
  order by technicalpackage_period_id
  for read only
  open tpp_cursor

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '01905', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41630', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41532', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42302', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41533', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41431', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
```

```
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42301', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42110', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02323', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41431', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41636', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41534', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41639', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42953', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41639', @curdate, @userid, @curdate, @userid)

  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41535', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42201', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
```

```
    values(@tppciid, @tppcid, null, 1, '41262', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41841', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41262', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--praktik
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--praktik
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41813', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41812', @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41213', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41212', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41812', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41212', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41511', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41313', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41260', @curdate, @userid, @curdate, @userid)
--project
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
```

```
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--project
  end

  close tpp_cursor
  deallocate tpp_cursor
  set @iteration = @iteration + 1
end

-------------------------------------------------------------------------------
--Mechanical Engineering 6 (Bachelor of Science) 2003 - 2010
-------------------------------------------------------------------------------

set @tpvid = 282

set @iteration = 1
while @iteration < 9
begin
  set @tpvid = @tpvid + 1
  declare tpp_cursor cursor for
  select top 14 technicalpackage_period_id
  from technicalpackage_period
  where technicalpackageversion_id = @tpvid
  order by technicalpackage_period_id
  for read only
  open tpp_cursor

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '01905', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41630', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41532', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42302', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41533', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41431', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42301', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42110', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
```

```
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02323', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41431', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41535', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42201', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41262', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41841', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41262', @curdate, @userid, @curdate, @userid)

  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41636', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41534', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41639', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42953', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
```

```
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41639', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--praktik
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--praktik
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41814', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41312', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41313', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41410', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41415', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41813', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41815', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41323', @curdate, @userid, @curdate, @userid)
--project
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--project
  end

  close tpp_cursor
  deallocate tpp_cursor
  set @iteration = @iteration + 1
end

--
```

```
set @tpvid = 290

set @iteration = 1
while @iteration < 9
begin
  set @tpvid = @tpvid + 1
  declare tpp_cursor cursor for
  select top 14 technicalpackage_period_id
  from technicalpackage_period
  where technicalpackageversion_id = @tpvid
  order by technicalpackage_period_id
  for read only
  open tpp_cursor

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '01905', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41630', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41532', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42302', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41533', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41431', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42301', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42110', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02323', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
```

```
    values(@tppciid, @tppcid, null, 2, '41431', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41636', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41534', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41639', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42953', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41639', @curdate, @userid, @curdate, @userid)

  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41535', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42201', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41262', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41841', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41262', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--praktik
  end
```

```
  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--praktik
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41415', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41813', @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41815', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41323', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41814', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41312', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41313', @curdate, @userid, @curdate, @userid)
--project
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--project
  end

  close tpp_cursor
  deallocate tpp_cursor
  set @iteration = @iteration + 1
end

-------------------------------------------------------------------------------
--Mechanical Engineering 7 (Bachelor of Science) 2003 - 2010
-------------------------------------------------------------------------------

set @tpvid = 298

set @iteration = 1
while @iteration < 9
begin
  set @tpvid = @tpvid + 1
  declare tpp_cursor cursor for
  select top 14 technicalpackage_period_id
  from technicalpackage_period
  where technicalpackageversion_id = @tpvid
  order by technicalpackage_period_id
  for read only
  open tpp_cursor

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
```

```
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01905', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41630', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41532', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '42302', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41533', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41431', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '42301', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '42110', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '02323', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '41431', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '42954', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '42201', @curdate, @userid, @curdate, @userid)
```

```
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41841', @curdate, @userid, @curdate, @userid)
 end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '42594', @curdate, @userid, @curdate, @userid)

  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41636', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42422', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41639', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42953', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41639', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--praktik
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--praktik
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42213', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42222', @curdate, @userid, @curdate, @userid)
  end
```

```
  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '42222', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '42250', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '42260', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--project
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42226', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '42260', @curdate, @userid, @curdate, @userid)
--project
  end

  close tpp_cursor
  deallocate tpp_cursor
  set @iteration = @iteration + 1
end

--

set @tpvid = 306

set @iteration = 1
while @iteration < 9
begin
  set @tpvid = @tpvid + 1
  declare tpp_cursor cursor for
  select top 14 technicalpackage_period_id
  from technicalpackage_period
  where technicalpackageversion_id = @tpvid
  order by technicalpackage_period_id
  for read only
  open tpp_cursor

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '01905', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41630', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41532', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42302', @curdate, @userid, @curdate, @userid)
  end
```

```
  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41533', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41431', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42301', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42110', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02323', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41431', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41636', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41534', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41639', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42953', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41639', @curdate, @userid, @curdate, @userid)

  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
```

```
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41535', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42201', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41262', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41841', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41262', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--praktik
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--praktik
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41415', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41813', @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41815', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41323', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
```

```
    values(@tppocid, @tppid, null, 1, '41814', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41312', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '41313', @curdate, @userid, @curdate, @userid)
--project
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--project
  end

  close tpp_cursor
  deallocate tpp_cursor
  set @iteration = @iteration + 1
end

-------------------------------------------------------------------------------
--Mechanical Engineering 8 (Bachelor of Science) 2003 - 2010
-------------------------------------------------------------------------------

set @tpvid = 314

set @iteration = 1
while @iteration < 9
begin
  set @tpvid = @tpvid + 1
  declare tpp_cursor cursor for
  select top 14 technicalpackage_period_id
  from technicalpackage_period
  where technicalpackageversion_id = @tpvid
  order by technicalpackage_period_id
  for read only
  open tpp_cursor

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '01905', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41630', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41532', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42302', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41533', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41431', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
```

```
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42301', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42110', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02323', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41431', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42954', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42201', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41841', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '42594', @curdate, @userid, @curdate, @userid)

  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41636', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42932', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41639', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42953', @curdate, @userid, @curdate, @userid)
  end
```

```
  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41639', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--praktik
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--praktik
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42230', @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '42936', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42232', @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '42937', @curdate, @userid, @curdate, @userid)
--project
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '42234', @curdate, @userid, @curdate, @userid)
--project
  end

  close tpp_cursor
  deallocate tpp_cursor
  set @iteration = @iteration + 1
end

--

set @tpvid = 322

set @iteration = 1
while @iteration < 9
begin
  set @tpvid = @tpvid + 1
  declare tpp_cursor cursor for
  select top 14 technicalpackage_period_id
  from technicalpackage_period
  where technicalpackageversion_id = @tpvid
  order by technicalpackage_period_id
```

```
for read only
open tpp_cursor

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '01905', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41630', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41532', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '42302', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41533', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '41431', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '42301', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '42110', @curdate, @userid, @curdate, @userid)

  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 1, '02323', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
  values(@tppciid, @tppcid, null, 2, '41431', @curdate, @userid, @curdate, @userid)
end

fetch next from tpp_cursor into @tppid
if @@fetch_status = 0
begin
  set @tppcid = @tppcid + 1
  insert into technicalpackage_periodcourse
  values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  set @tppciid = @tppciid + 1
  insert into technicalpackage_periodcourseitem
```

```
    values(@tppciid, @tppcid, null, 1, '41636', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42932', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41639', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42953', @curdate, @userid, @curdate, @userid)
 end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41639', @curdate, @userid, @curdate, @userid)

  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42954', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42201', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41841', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--praktik
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--praktik
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42232', @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
```

```
    values(@tppocid, @tppid, null, 1, '42937', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '42234', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42230', @curdate, @userid, @curdate, @userid)
--project
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--project
  end

  close tpp_cursor
  deallocate tpp_cursor
  set @iteration = @iteration + 1
end

-------------------------------------------------------------------------------
--Mechanical Engineering 9 (Bachelor of Science) 2003 - 2010
-------------------------------------------------------------------------------

set @tpvid = 330

set @iteration = 1
while @iteration < 9
begin
  set @tpvid = @tpvid + 1
  declare tpp_cursor cursor for
  select top 14 technicalpackage_period_id
  from technicalpackage_period
  where technicalpackageversion_id = @tpvid
  order by technicalpackage_period_id
  for read only
  open tpp_cursor

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '01905', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41630', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41532', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42302', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
```

```
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41533', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41431', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42301', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42110', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02323', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41431', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42954', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42201', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41841', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '42594', @curdate, @userid, @curdate, @userid)

  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41636', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
```

```
    values(@tppciid, @tppcid, null, 1, '42422', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41639', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42953', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41639', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--praktik
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--praktik
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42968', @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '42960', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '42240', @curdate, @userid, @curdate, @userid)

  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42342', @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '42967', @curdate, @userid, @curdate, @userid)
--project
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--project
  end

  close tpp_cursor
```

```
  deallocate tpp_cursor
  set @iteration = @iteration + 1
end

--

set @tpvid = 338

set @iteration = 1
while @iteration < 9
begin
  set @tpvid = @tpvid + 1
  declare tpp_cursor cursor for
  select top 14 technicalpackage_period_id
  from technicalpackage_period
  where technicalpackageversion_id = @tpvid
  order by technicalpackage_period_id
  for read only
  open tpp_cursor

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '01905', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41630', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41532', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42302', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41533', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41431', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42301', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42110', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02323', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
```

```
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41431', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41636', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42422', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41639', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42953', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41639', @curdate, @userid, @curdate, @userid)

  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42954', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42201', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41841', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--praktik
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
```

```
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--praktik
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42342', @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '42960', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '42967', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42968', @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '42440', @curdate, @userid, @curdate, @userid)
--project
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--project
  end

  close tpp_cursor
  deallocate tpp_cursor
  set @iteration = @iteration + 1
end

-------------------------------------------------------------------------------
--Mechanical Engineering 10 (Bachelor of Science) 2003 - 2010
-------------------------------------------------------------------------------

set @tpvid = 346

set @iteration = 1
while @iteration < 9
begin
  set @tpvid = @tpvid + 1
  declare tpp_cursor cursor for
  select top 14 technicalpackage_period_id
  from technicalpackage_period
  where technicalpackageversion_id = @tpvid
  order by technicalpackage_period_id
  for read only
  open tpp_cursor

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '01905', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41630', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
```

```
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41532', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42302', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41533', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41431', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42301', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42110', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02323', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41431', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42954', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42201', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41841', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
```

```
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '42594', @curdate, @userid, @curdate, @userid)

  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41636', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42422', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41639', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42953', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41639', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--praktik
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--praktik
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42642', @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '42440', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '42470', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '42966', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '42642', @curdate, @userid, @curdate, @userid)
  end
```

```
  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42531', @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '42430', @curdate, @userid, @curdate, @userid)
--project
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--project
  end

  close tpp_cursor
  deallocate tpp_cursor
  set @iteration = @iteration + 1
end

--

set @tpvid = 354

set @iteration = 1
while @iteration < 9
begin
  set @tpvid = @tpvid + 1
  declare tpp_cursor cursor for
  select top 14 technicalpackage_period_id
  from technicalpackage_period
  where technicalpackageversion_id = @tpvid
  order by technicalpackage_period_id
  for read only
  open tpp_cursor

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '01905', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41630', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41532', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42302', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41533', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41431', @curdate, @userid, @curdate, @userid)
```

```
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42301', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42110', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '02323', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)

    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41431', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41636', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42422', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41639', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42953', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '41639', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42954', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42201', @curdate, @userid, @curdate, @userid)

    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
```

```
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '41841', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--praktik
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
--praktik
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42531', @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '42430', @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 1, '42642', @curdate, @userid, @curdate, @userid)

    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '42440', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '42470', @curdate, @userid, @curdate, @userid)
    set @tppocid = @tppocid + 1
    insert into technicalpackage_periodoptionalcourse
    values(@tppocid, @tppid, null, 1, '42966', @curdate, @userid, @curdate, @userid)
--project
  end

  fetch next from tpp_cursor into @tppid
  if @@fetch_status = 0
  begin
    set @tppcid = @tppcid + 1
    insert into technicalpackage_periodcourse
    values(@tppcid, @tppid, @curdate, @userid, @curdate, @userid)
    set @tppciid = @tppciid + 1
    insert into technicalpackage_periodcourseitem
    values(@tppciid, @tppcid, null, 2, '42642', @curdate, @userid, @curdate, @userid)
--project
  end

  close tpp_cursor
  deallocate tpp_cursor
  set @iteration = @iteration + 1
end
```

3.21.4 Project

```
declare @userid uniqueidenti↓er
```

```
declare @curdate datetime

set @userid = '{BC61A572-A19A-41FB-97B8-9B0B463F2DBB}'
set @curdate = getdate()

declare @tpproj table (
  tp_id int,
  projnumber varchar(20),
  proj_id uniqueidentifier,
  tpversmin int,
  tpversmax int
)

-------------------------------------------------------------
-- Construction Engineering (Bygning)
-------------------------------------------------------------

insert into @tpproj values (12, '11751', '{C1AB722E-01F3-473a-AFB8-DFCEC7DB493B}', 131, 138)
insert into @tpproj values (13, '11751', '{C1AB722E-01F3-473a-AFB8-DFCEC7DB493B}', 139, 146)

-------------------------------------------------------------
-- Urban Planning and Construction (By og bygning)
-------------------------------------------------------------

insert into @tpproj values (14, '11951', '{06F28C4C-E7C1-4b95-987B-DE09838A7AE6}', 147, 154)

-------------------------------------------------------------
-- Electrical Engineering (Elektro)
-------------------------------------------------------------

insert into @tpproj values (15, '31031', '{32B5D8AA-16FF-4c88-93B8-02B29372F3E9}', 155, 162)
insert into @tpproj values (16, '31031', '{32B5D8AA-16FF-4c88-93B8-02B29372F3E9}', 163, 170)

-------------------------------------------------------------
-- Chemical Engineering (Kemi)
-------------------------------------------------------------

insert into @tpproj values (17, '28181', '{AF585023-3773-44fe-B3E8-22BFC9718E8B}', 171, 178)
insert into @tpproj values (18, '28181', '{AF585023-3773-44fe-B3E8-22BFC9718E8B}', 179, 186)

-------------------------------------------------------------
-- IT (IT)
-------------------------------------------------------------

insert into @tpproj values (19, '02365', '{870899CF-B9E0-49ae-9800-094EBAFEC157}', 187, 194)
insert into @tpproj values (20, '02365', '{870899CF-B9E0-49ae-9800-094EBAFEC157}', 195, 202)

----------------------------------------------------------------------------------

-- Mechanical Engineering (construction ship/energy, maritime energy/construction)
-- Maskin (konstruktion skib/energi, maritim energi/konstruktion)
----------------------------------------------------------------------------------


insert into @tpproj values (21, '41845', '{22E27B35-4609-4124-933D-563FCD350E24}', 203, 210)
insert into @tpproj values (22, '41845', '{22E27B35-4609-4124-933D-563FCD350E24}', 211, 218)
insert into @tpproj values (23, '41845', '{22E27B35-4609-4124-933D-563FCD350E24}', 219, 226)
insert into @tpproj values (24, '41845', '{22E27B35-4609-4124-933D-563FCD350E24}', 227, 234)
insert into @tpproj values (25, '41845', '{22E27B35-4609-4124-933D-563FCD350E24}', 235, 242)
insert into @tpproj values (26, '41845', '{22E27B35-4609-4124-933D-563FCD350E24}', 243, 250)
insert into @tpproj values (27, '41845', '{22E27B35-4609-4124-933D-563FCD350E24}', 251, 258)
insert into @tpproj values (28, '41845', '{22E27B35-4609-4124-933D-563FCD350E24}', 259, 266)
insert into @tpproj values (29, '41845', '{22E27B35-4609-4124-933D-563FCD350E24}', 267, 274)
insert into @tpproj values (30, '41845', '{22E27B35-4609-4124-933D-563FCD350E24}', 275, 282)
insert into @tpproj values (31, '41845', '{22E27B35-4609-4124-933D-563FCD350E24}', 283, 290)
insert into @tpproj values (32, '41845', '{22E27B35-4609-4124-933D-563FCD350E24}', 291, 298)

----------------------------------------------------------------------------------------------------

-- Mechanical Engineering (production and management process/plastic/control and logistics/organization)
-- Maskin (produktion og ledelse proces/plast/styring og logistik/organisation)
----------------------------------------------------------------------------------------------------


insert into @tpproj values (33, '42845', '{40BC7783-154F-4c07-8BF7-20E2BFC6A7D0}', 299, 306)
insert into @tpproj values (34, '42845', '{40BC7783-154F-4c07-8BF7-20E2BFC6A7D0}', 307, 314)
insert into @tpproj values (35, '42845', '{40BC7783-154F-4c07-8BF7-20E2BFC6A7D0}', 315, 322)
insert into @tpproj values (36, '42845', '{40BC7783-154F-4c07-8BF7-20E2BFC6A7D0}', 323, 330)
insert into @tpproj values (37, '42845', '{40BC7783-154F-4c07-8BF7-20E2BFC6A7D0}', 331, 338)
insert into @tpproj values (38, '42845', '{40BC7783-154F-4c07-8BF7-20E2BFC6A7D0}', 339, 346)
insert into @tpproj values (39, '42845', '{40BC7783-154F-4c07-8BF7-20E2BFC6A7D0}', 347, 354)
insert into @tpproj values (40, '42845', '{40BC7783-154F-4c07-8BF7-20E2BFC6A7D0}', 355, 362)

--
  - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
--
  - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

-- declare @tpproj table (
-- tp_id int,
-- projnumber varchar(20),
-- proj_id uniqueidentifier,
-- tpversmin int,
-- tpversmax int
-- )

declare @technicalpackageversion_id int
declare @projnum varchar(20)
declare @proj_id uniqueidentifier
```

```
declare @tpversmin int
declare @tpversmax int

declare @primarykey int
set @primarykey = 1

declare @startdate datetime
declare @enddate datetime

declare @period_id uniqueidenti↓er

declare @periodtype_id int
set @periodtype_id = 3

declare proj_cursor cursor for
select tp_id, projnumber, proj_id, tpversmin, tpversmax
from @tpproj
for read only
open proj_cursor

fetch next from proj_cursor into @technicalpackageversion_id, @projnum, @proj_id, @tpversmin, @tpversmax
while @@fetch_status = 0
begin
  declare @curtpvers int
  set @curtpvers = @tpversmin
  while (@curtpvers <= @tpversmax)
  begin

    select @startdate=per.StartDate, @enddate=per.EndDate
    from
      TechnicalPackage_Period p,
      TechnicalPackage_PeriodCourse pc,
      TechnicalPackage_PeriodCourseItem pci,
      Period per
    where
      pci.Number=@projnum
      and pci.TechnicalPackage_PeriodCourse_ID = pc.TechnicalPackage_PeriodCourse_ID
      and pc.TechnicalPackage_Period_ID = p.TechnicalPackage_Period_ID
      and TechnicalPackageVersion_ID=@curtpvers
      and pci.Part = 1
      and per.Period_ID = p.Period_ID

    declare @parts int
    select @parts=Parts from CourseVersion where number=@projnum

    if (@parts > 1)
    begin
      select @enddate=per.EndDate
      from TechnicalPackage_Period p, TechnicalPackage_PeriodCourse pc, TechnicalPackage_PeriodCourseItem pci, Period per
      where
        pci.Number=@projnum
        and pci.TechnicalPackage_PeriodCourse_ID = pc.TechnicalPackage_PeriodCourse_ID
        and pc.TechnicalPackage_Period_ID = p.TechnicalPackage_Period_ID
        and TechnicalPackageVersion_ID=@curtpvers
        and pci.Part = (@parts)
        and per.Period_ID = p.Period_ID
    end

    set @period_id = newid()
    insert into Period (
      Period_ID,
      PeriodType_ID,
      Name,
      StartDate,
      EndDate,
      Created, CreatedBy, Updated, UpdatedBy
    ) values (
      @period_id,
      @periodtype_id,
      '<cultures>
        <culture>
          <cultureID>da-DK</cultureID>
          <value>Individuel periode for projekt ' + @projnum + ' og fagpakkeversion ' + CAST(@curtpvers AS VARCHAR)+ '</value>
        </culture>
        <culture>
          <cultureID>en</cultureID>
          <value>Individual period for project ' + @projnum + ' and technical package version ' + CAST(@curtpvers AS VARCHAR) +
            '</value>
        </culture>
      </cultures>',
      @startdate,
      @enddate,
      @curdate, @userid, @curdate, @userid
    )

    insert into technicalpackage_project (
      TechnicalPackage_Project_ID,
      TechnicalPackageVersion_ID,
      Project_ID,
      Period_ID,
      Created, CreatedBy, Updated, UpdatedBy
    ) values (
      @primarykey,
      @curtpvers,
      @proj_id,
      @period_id,
      @curdate, @userid, @curdate, @userid
    )
    set @primarykey = @primarykey + 1
```

```
    set @curtpvers = @curtpvers + 1
  end
  fetch next from proj_cursor into @technicalpackageversion_id, @projnum, @proj_id, @tpversmin, @tpversmax
end

close proj_cursor
deallocate proj_cursor
```

3.22 Text

```
declare @userid uniqueidenti↓er
set @userid = '{BC61A572-A19A-41FB-97B8-9B0B463F2DBB}'

declare @curdate datetime
set @curdate = getdate()

insert into Text
values (5231, 523, '4', 'en', 'Validation error on property @@V1.', NULL, @curdate, @userid, @curdate, @userid)

insert into Text
values (5232, 523, '8', 'en', 'Row does not exists.', NULL, @curdate, @userid, @curdate, @userid)

insert into Text
values (5233, 523, '12', 'en', 'Row already exists. Violation of primary key constraint.', NULL, @curdate, @userid, @curdate, @userid)

insert into Text
values (5234, 523, '16', 'en', 'A severe error has occurred.', NULL, @curdate, @userid, @curdate, @userid)

insert into Text
values (5331, 710, '4_A', 'en', 'An error was thrown in the data access layer.', NULL, @curdate, @userid, @curdate, @userid)

insert into Text
values (5332, 710, '4_B', 'en', 'Data Access Layer: Error on parameter @@V1 using value @@V2.', NULL, @curdate, @userid, @curdate, @userid)

insert into Text
values (5333, 710, '4_C', 'en', 'Data Access Layer: Error on parameter @@V1.', NULL, @curdate, @userid, @curdate, @userid)
```

3.23 TextGroup

```
declare @userid uniqueidenti↓er
set @userid = '{BC61A572-A19A-41FB-97B8-9B0B463F2DBB}'

declare @curdate datetime
set @curdate = getdate()

declare @descr varchar(600)

set @descr = 'Texts used in the data tier should be in this group or in one of its sub-groups.'
insert into TextGroup
values (500, 'Data Access Layer', @descr, NULL, @curdate, @userid, @curdate, @userid)

insert into TextGroup
values (510, 'StudyPlanning.DAL.Common.dll', NULL, 500, @curdate, @userid, @curdate, @userid)

  insert into TextGroup
  values (511, 'DbObject.cs', NULL, 510, @curdate, @userid, @curdate, @userid)

  insert into TextGroup
  values (512, 'DataLog.cs', NULL, 510, @curdate, @userid, @curdate, @userid)

  insert into TextGroup
  values (513, 'TextItem.cs', NULL, 510, @curdate, @userid, @curdate, @userid)

  insert into TextGroup
  values (520, 'DalType.cs', NULL, 510, @curdate, @userid, @curdate, @userid)

    insert into TextGroup
    values (521, 'DalBool.cs', NULL, 520, @curdate, @userid, @curdate, @userid)

    insert into TextGroup
    values (522, 'DalDateTime.cs', NULL, 520, @curdate, @userid, @curdate, @userid)

    insert into TextGroup
    values (523, 'DalException.cs', NULL, 520, @curdate, @userid, @curdate, @userid)

    insert into TextGroup
    values (524, 'DalGuid.cs', NULL, 520, @curdate, @userid, @curdate, @userid)

    insert into TextGroup
    values (525, 'DalInt.cs', NULL, 520, @curdate, @userid, @curdate, @userid)

    insert into TextGroup
    values (526, 'DalString.cs', NULL, 520, @curdate, @userid, @curdate, @userid)

    insert into TextGroup
    values (527, 'DalStringLocalizable.cs', NULL, 520, @curdate, @userid, @curdate, @userid)

set @descr = 'Texts used in the business services layer should be in this group or in one of its sub-groups.'
insert into TextGroup
```

```
values (700, 'Business Services Layer', @descr, NULL, @curdate, @userid, @curdate, @userid)

insert into TextGroup
values (710, 'BizException.cs', NULL, 700, @curdate, @userid, @curdate, @userid)

-----------------------------------------------------------------------------------------------------------------------------------------------------------------


set @descr = 'Texts used in the presentation layer should be in this group or in one of its sub-groups.'
insert into TextGroup
values (200, 'Presentation Layer', @descr, NULL, @curdate, @userid, @curdate, @userid)

insert into TextGroup
values (210, 'authenticate.aspx', NULL, 200, @curdate, @userid, @curdate, @userid)

-- studyplanning
---- default.aspx
---- studyplancriteria.aspx
---- studyplans.aspx

insert into TextGroup
values (230, 'studyplanning', NULL, 200, @curdate, @userid, @curdate, @userid)

  insert into TextGroup
  values (235, 'default.aspx', NULL, 230, @curdate, @userid, @curdate, @userid)

  insert into TextGroup
  values (240, 'studyplancriteria.aspx', NULL, 230, @curdate, @userid, @curdate, @userid)

  insert into TextGroup
  values (245, 'studyplans.aspx', NULL, 230, @curdate, @userid, @curdate, @userid)

-- studyinfo
---- default.aspx
---- lines.aspx
---- periods.aspx
---- technicalpackages.aspx

insert into TextGroup
values (280, 'studyinfo', NULL, 200, @curdate, @userid, @curdate, @userid)

  insert into TextGroup
  values (285, 'default.aspx', NULL, 280, @curdate, @userid, @curdate, @userid)

  insert into TextGroup
  values (290, 'lines.aspx', NULL, 280, @curdate, @userid, @curdate, @userid)

  insert into TextGroup
  values (295, 'periods.aspx', NULL, 280, @curdate, @userid, @curdate, @userid)

  insert into TextGroup
  values (300, 'technicalpackages.aspx', NULL, 280, @curdate, @userid, @curdate, @userid)

-- coursebase
---- default.aspx
---- search.aspx
---- alfabetical.aspx
---- institute.aspx

insert into TextGroup
values (330, 'coursebase', NULL, 200, @curdate, @userid, @curdate, @userid)

  insert into TextGroup
  values (335, 'default.aspx', NULL, 330, @curdate, @userid, @curdate, @userid)

  insert into TextGroup
  values (340, 'search.aspx', NULL, 330, @curdate, @userid, @curdate, @userid)

  insert into TextGroup
  values (345, 'alfabetical.aspx', NULL, 330, @curdate, @userid, @curdate, @userid)

  insert into TextGroup
  values (350, 'institute.aspx', NULL, 330, @curdate, @userid, @curdate, @userid)

-- administration
---- default.aspx
---- departments
------ default.aspx
---- users
------ default.aspx

insert into TextGroup
values (380, 'administration', NULL, 200, @curdate, @userid, @curdate, @userid)

  insert into TextGroup
  values (385, 'default.aspx', NULL, 380, @curdate, @userid, @curdate, @userid)

  insert into TextGroup
  values (390, 'departments', NULL, 380, @curdate, @userid, @curdate, @userid)

    insert into TextGroup
    values (391, 'default.aspx', NULL, 390, @curdate, @userid, @curdate, @userid)

  insert into TextGroup
  values (395, 'users', NULL, 380, @curdate, @userid, @curdate, @userid)

    insert into TextGroup
    values (396, 'default.aspx', NULL, 395, @curdate, @userid, @curdate, @userid)
```

```
−−controlpanel
−−−−default.aspx
−−−−changePassword.aspx
−−−−userProfile.aspx

insert into TextGroup
values (420, 'controlpanel', NULL, 200, @curdate, @userid, @curdate, @userid)

  insert into TextGroup
  values (425, 'default.aspx', NULL, 420, @curdate, @userid, @curdate, @userid)

  insert into TextGroup
  values (430, 'changePassword.aspx', NULL, 420, @curdate, @userid, @curdate, @userid)

  insert into TextGroup
  values (435, 'userProfile.aspx', NULL, 420, @curdate, @userid, @curdate, @userid)
```

3.24 User_LoginType

```
declare @userid uniqueidenti↓er
set @userid = '{BC61A572-A19A-41FB-97B8-9B0B463F2DBB}'

declare @curdate datetime
set @curdate = getdate()

insert into User_LoginType
values (1,
        'Rejection. The user has been rejected authentication and thus refused access to the system.',
        @curdate, @userid, @curdate, @userid)

insert into User_LoginType
values (2,
        'Approval. The user is authenticated and thus granted access to the system.',
        @curdate, @userid, @curdate, @userid)
```

3.25 Weekday

```
declare @userid uniqueidenti↓er
set @userid = '{BC61A572-A19A-41FB-97B8-9B0B463F2DBB}'

declare @curdate datetime
set @curdate = getdate()

insert into Weekday
values (1,
'<cultures>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>Monday</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>Mandag</value>
  </culture>
</cultures>',
@curdate, @userid, @curdate, @userid)

insert into Weekday
values (2,
'<cultures>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>Tuesday</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>Tirsdag</value>
  </culture>
</cultures>',
@curdate, @userid, @curdate, @userid)

insert into Weekday
values (3,
'<cultures>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>Wednesday</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>Onsdag</value>
  </culture>
</cultures>',
@curdate, @userid, @curdate, @userid)

insert into Weekday
values (4,
'<cultures>
  <culture>
```

```
    <cultureID>en-GB</cultureID>
    <value>Thursday</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>Torsdag</value>
  </culture>
</cultures>',
@curdate, @userid, @curdate, @userid)

insert into Weekday
values (5,
'<cultures>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>Friday</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>Fredag</value>
  </culture>
</cultures>',
@curdate, @userid, @curdate, @userid)

insert into Weekday
values (6,
'<cultures>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>Saturday</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>Lørdag</value>
  </culture>
</cultures>',
@curdate, @userid, @curdate, @userid)

insert into Weekday
values (7,
'<cultures>
  <culture>
    <cultureID>en-GB</cultureID>
    <value>Sunday</value>
  </culture>
  <culture>
    <cultureID>da-DK</cultureID>
    <value>Søndag</value>
  </culture>
</cultures>',
@curdate, @userid, @curdate, @userid)
```

Chapter 4

Manual Handling of Grabbed Data

4.1 Course_Period and Module

4.1.1 Automatic

```
declare @courseversion_id uniqueidenti¦er
declare @course_period_id uniqueidenti¦er
declare @course_period_module_id uniqueidenti¦er

declare @number varchar(20)
declare @schedule varchar(200)
declare @duration varchar(200)
declare @module_id int
declare @period_id uniqueidenti¦er
declare @module varchar(30)
declare @userid uniqueidenti¦er
declare @curdate datetime
declare @done int
declare @periodtype_id int
declare @part int

declare @modules table (module varchar(30), module_id int, startmonth int, periodtype_id int, part int)
insert into @modules values('<value>F1A<br/', 1, 1, 1, 1)
insert into @modules values('<value>F1B<br/', 2, 1, 1, 1)
insert into @modules values('<value>F2A<br/', 3, 1, 1, 1)
insert into @modules values('<value>F2B<br/', 4, 1, 1, 1)
insert into @modules values('<value>F3A<br/', 5, 1, 1, 1)
insert into @modules values('<value>F3B<br/', 6, 1, 1, 1)
insert into @modules values('<value>F4A<br/', 7, 1, 1, 1)
insert into @modules values('<value>F4B<br/', 8, 1, 1, 1)
insert into @modules values('<value>F5A<br/', 9, 1, 1, 1)
insert into @modules values('<value>F5B<br/', 10, 1, 1, 1)
insert into @modules values('<value>F1A<br/', 1, 2, 1, 1)
insert into @modules values('<value>F1B<br/', 2, 2, 1, 1)
insert into @modules values('<value>F2A<br/', 3, 2, 1, 1)
insert into @modules values('<value>F2B<br/', 4, 2, 1, 1)
insert into @modules values('<value>F3A<br/', 5, 2, 1, 1)
insert into @modules values('<value>F3B<br/', 6, 2, 1, 1)
insert into @modules values('<value>F4A<br/', 7, 2, 1, 1)
insert into @modules values('<value>F4B<br/', 8, 2, 1, 1)
insert into @modules values('<value>F5A<br/', 9, 2, 1, 1)
insert into @modules values('<value>F5B<br/', 10, 2, 1, 1)

insert into @modules values('<value>E1A<br/', 1, 8, 1, 1)
insert into @modules values('<value>E1B<br/', 2, 8, 1, 1)
insert into @modules values('<value>E2A<br/', 3, 8, 1, 1)
insert into @modules values('<value>E2B<br/', 4, 8, 1, 1)
insert into @modules values('<value>E3A<br/', 5, 8, 1, 1)
insert into @modules values('<value>E3B<br/', 6, 8, 1, 1)
insert into @modules values('<value>E4A<br/', 7, 8, 1, 1)
insert into @modules values('<value>E4B<br/', 8, 8, 1, 1)
insert into @modules values('<value>E5A<br/', 9, 8, 1, 1)
insert into @modules values('<value>E5B<br/', 10, 8, 1, 1)
insert into @modules values('<value>E1A<br/', 1, 9, 1, 1)
insert into @modules values('<value>E1B<br/', 2, 9, 1, 1)
insert into @modules values('<value>E2A<br/', 3, 9, 1, 1)
insert into @modules values('<value>E2B<br/', 4, 9, 1, 1)
insert into @modules values('<value>E3A<br/', 5, 9, 1, 1)
insert into @modules values('<value>E3B<br/', 6, 9, 1, 1)
insert into @modules values('<value>E4A<br/', 7, 9, 1, 1)
insert into @modules values('<value>E4B<br/', 8, 9, 1, 1)
insert into @modules values('<value>E5A<br/', 9, 9, 1, 1)
insert into @modules values('<value>E5B<br/', 10, 9, 1, 1)

insert into @modules values('<value>F1<br/>', 1, 1, 1, 1)
insert into @modules values('<value>F1<br/>', 2, 2, 1, 1)
insert into @modules values('<value>F2<br/>', 3, 1, 1, 1)
insert into @modules values('<value>F2<br/>', 4, 2, 1, 1)
insert into @modules values('<value>F3<br/>', 5, 1, 1, 1)
insert into @modules values('<value>F3<br/>', 6, 2, 1, 1)
insert into @modules values('<value>F4<br/>', 7, 1, 1, 1)
insert into @modules values('<value>F4<br/>', 8, 2, 1, 1)
insert into @modules values('<value>F5<br/>', 9, 1, 1, 1)
insert into @modules values('<value>F5<br/>', 10, 2, 1, 1)

insert into @modules values('<value>E1<br/>', 1, 8, 1, 1)
insert into @modules values('<value>E1<br/>', 2, 9, 1, 1)
insert into @modules values('<value>E2<br/>', 3, 8, 1, 1)
insert into @modules values('<value>E2<br/>', 4, 9, 1, 1)
insert into @modules values('<value>E2<br/>', 5, 8, 1, 1)
insert into @modules values('<value>E3<br/>', 6, 9, 1, 1)
insert into @modules values('<value>E4<br/>', 7, 8, 1, 1)
insert into @modules values('<value>E4<br/>', 8, 9, 1, 1)
insert into @modules values('<value>E5<br/>', 9, 8, 1, 1)
insert into @modules values('<value>E5<br/>', 10, 9, 1, 1)

insert into @modules values('<value>E1A and E1B<br/>', 1, 8, 1, 1)
insert into @modules values('<value>E1A and E1B<br/>', 2, 9, 1, 1)
insert into @modules values('<value>E2A and E2B<br/>', 3, 8, 1, 1)
insert into @modules values('<value>E2A and E2B<br/>', 4, 9, 1, 1)
insert into @modules values('<value>E3A and E3B<br/>', 5, 8, 1, 1)
insert into @modules values('<value>E3A and E3B<br/>', 6, 9, 1, 1)
insert into @modules values('<value>E4A and E4B<br/>', 7, 8, 1, 1)
insert into @modules values('<value>E4A and E4B<br/>', 8, 9, 1, 1)
insert into @modules values('<value>E5A and E5B<br/>', 9, 8, 1, 1)
insert into @modules values('<value>E5A and E5B<br/>', 10, 9, 1, 1)
```

```
insert into @modules values('<value>F1A and F1B<br/>', 1, 8, 1, 1)
insert into @modules values('<value>F1A and F1B<br/>', 2, 9, 1, 1)
insert into @modules values('<value>F2A and F2B<br/>', 3, 8, 1, 1)
insert into @modules values('<value>F2A and F2B<br/>', 4, 9, 1, 1)
insert into @modules values('<value>F3A and F3B<br/>', 5, 8, 1, 1)
insert into @modules values('<value>F3A and F3B<br/>', 6, 9, 1, 1)
insert into @modules values('<value>F4A and F4B<br/>', 7, 8, 1, 1)
insert into @modules values('<value>F4A and F4B<br/>', 8, 9, 1, 1)
insert into @modules values('<value>F5A and F5B<br/>', 9, 8, 1, 1)
insert into @modules values('<value>F5A and F5B<br/>', 10, 9, 1, 1)

insert into @modules values('<value>E1A and January<br/>', 1, 8, 1, 1)
insert into @modules values('<value>E1A and January<br/>', 1, 9, 1, 1)
insert into @modules values('<value>E1A and January<br/>', 0, 1, 2, 2)
insert into @modules values('<value>E1B and January<br/>', 2, 8, 1, 1)
insert into @modules values('<value>E1B and January<br/>', 2, 9, 1, 1)
insert into @modules values('<value>E1B and January<br/>', 0, 1, 2, 2)
insert into @modules values('<value>E2A and January<br/>', 3, 8, 1, 1)
insert into @modules values('<value>E2A and January<br/>', 3, 9, 1, 1)
insert into @modules values('<value>E2A and January<br/>', 0, 1, 2, 2)
insert into @modules values('<value>E2B and January<br/>', 4, 8, 1, 1)
insert into @modules values('<value>E2B and January<br/>', 4, 9, 1, 1)
insert into @modules values('<value>E2B and January<br/>', 0, 1, 2, 2)
insert into @modules values('<value>E3A and January<br/>', 5, 8, 1, 1)
insert into @modules values('<value>E3A and January<br/>', 5, 9, 1, 1)
insert into @modules values('<value>E3A and January<br/>', 0, 1, 2, 2)
insert into @modules values('<value>E3B and January<br/>', 6, 8, 1, 1)
insert into @modules values('<value>E3B and January<br/>', 6, 9, 1, 1)
insert into @modules values('<value>E3B and January<br/>', 0, 1, 2, 2)
insert into @modules values('<value>E4A and January<br/>', 7, 8, 1, 1)
insert into @modules values('<value>E4A and January<br/>', 7, 9, 1, 1)
insert into @modules values('<value>E4A and January<br/>', 0, 1, 2, 2)
insert into @modules values('<value>E4B and January<br/>', 8, 8, 1, 1)
insert into @modules values('<value>E4B and January<br/>', 8, 9, 1, 1)
insert into @modules values('<value>E4B and January<br/>', 0, 1, 2, 2)
insert into @modules values('<value>E5A and January<br/>', 9, 8, 1, 1)
insert into @modules values('<value>E5A and January<br/>', 9, 9, 1, 1)
insert into @modules values('<value>E5A and January<br/>', 0, 1, 2, 2)
insert into @modules values('<value>E5B and January<br/>', 10, 8, 1, 1)
insert into @modules values('<value>E5B and January<br/>', 10, 9, 1, 1)
insert into @modules values('<value>E5B and January<br/>', 0, 1, 2, 2)

insert into @modules values('<value>F1A and June<br/>', 1, 1, 1, 1)
insert into @modules values('<value>F1A and June<br/>', 1, 2, 1, 1)
insert into @modules values('<value>F1A and June<br/>', 0, 6, 2, 2)
insert into @modules values('<value>F1B and June<br/>', 2, 1, 1, 1)
insert into @modules values('<value>F1B and June<br/>', 2, 2, 1, 1)
insert into @modules values('<value>F1B and June<br/>', 0, 6, 2, 2)
insert into @modules values('<value>F2A and June<br/>', 3, 1, 1, 1)
insert into @modules values('<value>F2A and June<br/>', 3, 2, 1, 1)
insert into @modules values('<value>F2A and June<br/>', 0, 6, 2, 2)
insert into @modules values('<value>F2B and June<br/>', 4, 1, 1, 1)
insert into @modules values('<value>F2B and June<br/>', 4, 2, 1, 1)
insert into @modules values('<value>F2B and June<br/>', 0, 6, 2, 2)
insert into @modules values('<value>F3A and June<br/>', 5, 1, 1, 1)
insert into @modules values('<value>F3A and June<br/>', 5, 2, 1, 1)
insert into @modules values('<value>F3A and June<br/>', 0, 6, 2, 2)
insert into @modules values('<value>F3B and June<br/>', 6, 1, 1, 1)
insert into @modules values('<value>F3B and June<br/>', 6, 2, 1, 1)
insert into @modules values('<value>F3B and June<br/>', 0, 6, 2, 2)
insert into @modules values('<value>F4A and June<br/>', 7, 1, 1, 1)
insert into @modules values('<value>F4A and June<br/>', 7, 2, 1, 1)
insert into @modules values('<value>F4A and June<br/>', 0, 6, 2, 2)
insert into @modules values('<value>F4B and June<br/>', 8, 1, 1, 1)
insert into @modules values('<value>F4B and June<br/>', 8, 2, 1, 1)
insert into @modules values('<value>F4B and June<br/>', 0, 6, 2, 2)
insert into @modules values('<value>F5A and June<br/>', 9, 1, 1, 1)
insert into @modules values('<value>F5A and June<br/>', 9, 2, 1, 1)
insert into @modules values('<value>F5A and June<br/>', 0, 6, 2, 2)
insert into @modules values('<value>F5B and June<br/>', 10, 1, 1, 1)
insert into @modules values('<value>F5B and June<br/>', 10, 2, 1, 1)
insert into @modules values('<value>F5B and June<br/>', 0, 6, 2, 2)

insert into @modules values('<value>June<br/>', 0, 6, 2, 1)
insert into @modules values('<value>January<br/>', 0, 1, 2, 1)

set @userid = '{40AA0D66-B5A3-4E42-9B13-BCDA0905EB9D}'
set @curdate = getdate()
-----------------------------------------------------------------------------
declare grab_cursor cursor for
select courseversion_id, number, schedule, duration
from coursegrab a
where ((duration like '1 semester') or (duration like '3-ugers periode'))
and schedule is not null
and created = (select max(created) from coursegrab where number = a.number)
order by number
for read only
open grab_cursor

fetch next from grab_cursor into @courseversion_id, @number, @schedule, @duration
while @@fetch_status = 0
begin
  set @done = 0
  set @module = substring(@schedule,46,30)
  if charindex ('and January', @module) > 0 and @done = 0
  begin
    set @module = substring(@schedule,46,27)
    set @done = 1
```

```
  end
  if charindex ('and June', @module) > 0 and @done = 0
  begin
    set @module = substring(@schedule,46,24)
    set @done = 1
  end
  if charindex('and', @module) > 0 and @done = 0
  begin
    set @module = substring(@schedule,46,23)
    set @done = 1
  end

  if @done = 0
  begin
    set @module = substring(@schedule,46,14)
  end

  if @module = '<value>January' and @done = 0
  begin
    set @module = substring(@schedule,46,19)
    set @done = 1
  end
  if @module = '<value>June<br' and @done = 0
  begin
    set @module = substring(@schedule,46,16)
    set @done = 1
  end

  if exists (select module from @modules where module = @module)
  begin
    declare period_cursor cursor for
    select period_id, periodtype_id
    from period
    where periodtype_id in (select distinct periodtype_id from @modules where module = @module)
    and month(startdate) in (select startmonth from @modules where module = @module)
    for read only
    open period_cursor

    fetch next from period_cursor into @period_id, @periodtype_id
    while @@fetch_status = 0
    begin
      set @part = (select top 1 part from @modules
                   where module = @module
                   and periodtype_id = @periodtype_id)
      set @course_period_id = newid()
      insert into course_period
      values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

      declare module_cursor cursor for
      select module_id
      from @modules
      where module = @module
      and module_id <> 0
      and periodtype_id = @periodtype_id
      group by module_id
      for read only
      open module_cursor

      fetch next from module_cursor into @module_id
      while @@fetch_status = 0
      begin
        set @course_period_module_id = newid()
        insert into course_period_module
        values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

        insert into course_period_moduleitem
        values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

        fetch next from module_cursor into @module_id
      end
      close module_cursor
      deallocate module_cursor

      fetch next from period_cursor into @period_id, @periodtype_id
    end

    close period_cursor
    deallocate period_cursor
  end
  fetch next from grab_cursor into @courseversion_id, @number, @schedule, @duration
end

close grab_cursor
deallocate grab_cursor
```

4.1.2 Manual

```
declare @courseversion_id uniqueidenti↓er
declare @course_period_id uniqueidenti↓er
declare @course_period_module_id uniqueidenti↓er

declare @number varchar(20)
declare @part int
declare @module_id int
declare @period_id uniqueidenti↓er
declare @startdate datetime
```

```
declare @userid uniqueidenti↓er
declare @curdate datetime

set @userid = '{40AA0D66-B5A3-4E42-9B13-BCDA0905EB9D}'
set @curdate = getdate()
------------------------------------------------------------------------------

--All courses which are taught in the same modules in all 13-week periods

declare period_cursor cursor for
select period_id, startdate
from period
where periodtype_id = 1
for read only
open period_cursor

fetch next from period_cursor into @period_id, @startdate
while @@fetch_status = 0
begin
  set @number = '01016'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 2
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '01030'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 1
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '01031'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 1
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '02311'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '02318'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
```

```
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '02321'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 3
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '02322'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '02335'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '02365'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 1
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 2
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
```

```
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 3
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 4
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 5
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 6
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 7
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 8
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 9
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 10
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
set @number = '02374'
set @part = 1

set @course_period_id = newid()
set @courseversion_id = (select top 1 courseversion_id from coursegrab
                         where number = @number order by created desc)
insert into course_period
values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 1
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 2
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 3
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)
```

```
  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 6
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 10
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '02376'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 1
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 2
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 3
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
```

```
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 6
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 7
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 8
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 9
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 10
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
set @number = '02906'
set @part = 1

set @course_period_id = newid()
set @courseversion_id = (select top 1 courseversion_id from coursegrab
                         where number = @number order by created desc)
insert into course_period
values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 7
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
set @number = '02591'
set @part = 1

set @course_period_id = newid()
set @courseversion_id = (select top 1 courseversion_id from coursegrab
                         where number = @number order by created desc)
insert into course_period
values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 1
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 2
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
set @number = '10010'
set @part = 1

set @course_period_id = newid()
set @courseversion_id = (select top 1 courseversion_id from coursegrab
                         where number = @number order by created desc)
insert into course_period
values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 3
```

```
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '10012'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '10013'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '10501'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '10504'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '11742'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)
```

```
  set @module_id = 10
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
– –
  set @number = '11751'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 1
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 2
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 3
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 6
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 10
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
– –
  set @number = '11761'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
```

```
                        where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '11771'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                        where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 10
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '11801'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                        where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '11802'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                        where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 10
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '11803'
  set @part = 1

  set @course_period_id = newid()
```

```
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 6
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '11804'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 1
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 3
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '11805'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 1
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 2
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 3
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)
```

```
  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 6
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 10
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '11913'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '12201'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 10
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '12243'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)
```

```
set @module_id = 1
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 2
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 3
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 4
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 5
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 6
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 7
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 8
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 9
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 10
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
set @number = '27443'
set @part = 1

set @course_period_id = newid()
set @courseversion_id = (select top 1 courseversion_id from coursegrab
                         where number = @number order by created desc)
insert into course_period
values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 1
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 2
insert into course_period_moduleitem
```

```
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '27706'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 1
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 2
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 3
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 6
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 10
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '27931'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
```

```
  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '27961'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 10
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '28021'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 6
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '28120'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '28171'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 6
  insert into course_period_moduleitem
```

```
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '28181'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 1
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 2
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 3
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 6
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 10
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '28321'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
```

```
  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '28341'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 2
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '28351'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 1
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '28381'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 3
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 6
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '31031'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 1
  insert into course_period_moduleitem
```

```
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 2
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 3
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 6
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 10
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '31361'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '34290'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
```

```
  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '34390'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '41845'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 1
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 2
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 3
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 6
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
```

```
  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 10
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '41859'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 6
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '42110'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 2
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '42645'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)
```

```
  set @module_id = 6
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '42845'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 1
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 2
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 3
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 6
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 10
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '42859'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
```

```
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 1
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 2
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 3
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '42983'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '26172'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '26272'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 1
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '26472'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)
```

```
  set @module_id = 3
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 10
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '26500'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 3
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '31020'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 3
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '31021'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '31022'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)
```

```
  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '31027'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 1
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 2
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '41212'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '41430'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '41431'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '41531'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
```

```
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
– –
  set @number = '41532'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
– –
  set @number = '41533'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 1
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
– –
  set @number = '41630'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 10
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
– –
  set @number = '41637'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
– –
  set @number = '41638'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
```

```
--
  set @number = '41816'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 3
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '42215'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 10
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '42301'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '42302'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 10
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '42305'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
```

```
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 9
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 10
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
set @number = '42309'
set @part = 1

set @course_period_id = newid()
set @courseversion_id = (select top 1 courseversion_id from coursegrab
                         where number = @number order by created desc)
insert into course_period
values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 9
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 10
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
set @number = '42410'
set @part = 1

set @course_period_id = newid()
set @courseversion_id = (select top 1 courseversion_id from coursegrab
                         where number = @number order by created desc)
insert into course_period
values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 5
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
set @number = '42412'
set @part = 1

set @course_period_id = newid()
set @courseversion_id = (select top 1 courseversion_id from coursegrab
                         where number = @number order by created desc)
insert into course_period
values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 2
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
set @number = '42510'
set @part = 1

set @course_period_id = newid()
set @courseversion_id = (select top 1 courseversion_id from coursegrab
                         where number = @number order by created desc)
insert into course_period
values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 8
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
set @number = '42533'
set @part = 1

set @course_period_id = newid()
set @courseversion_id = (select top 1 courseversion_id from coursegrab
                         where number = @number order by created desc)
insert into course_period
values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
```

```
  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '42931'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '42952'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '88890'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '88312'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 2
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 3
  insert into course_period_moduleitem
```

```
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '42981'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '02393'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 12
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '33522'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '33534'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '33556'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab
                           where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  fetch next from period_cursor into @period_id, @startdate
end

close period_cursor
deallocate period_cursor
```

4.1.3 Manual – Autumn periods only

```
declare @courseversion_id uniqueidenti↓er
```

```
declare @course_period_id uniqueidenti↓er
declare @course_period_module_id uniqueidenti↓er

declare @number varchar(20)
declare @part int
declare @module_id int
declare @period_id uniqueidenti↓er
declare @startdate datetime

declare @userid uniqueidenti↓er
declare @curdate datetime

set @userid = '{40AA0D66-B5A3-4E42-9B13-BCDA0905EB9D}'
set @curdate = getdate()
-----------------------------------------------------------------------------------------------------------------------

declare period_cursor cursor for
select period_id, startdate
from period
where periodtype_id = 1
and month(startdate) in (8,9)
for read only
open period_cursor

fetch next from period_cursor into @period_id, @startdate
while @@fetch_status = 0
begin
  set @number = '01005'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 1
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 3
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '01007'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 3
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '01902'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
```

```
  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '01905'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 1
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 3
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 2
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '02120'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '02323'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '02355'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
```

```
--
  set @number = '02402'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '02405'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 6
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '02411'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 3
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '02537'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '02601'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '02701'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 3
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '02904'
  set @part = 1
```

```
  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 3
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '02907'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 2
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '02908'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 1
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '10001'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 10
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '10004'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 3
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 6
```

```
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 10
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '10011'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '10061'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '10472'
  set @part = 1

  if year(@startdate) <> 2003
  begin
    set @course_period_id = newid()
    set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
    insert into course_period
    values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

    set @course_period_module_id = newid()
    insert into course_period_module
    values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

    set @module_id = 4
    insert into course_period_moduleitem
    values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
  end
--
  set @number = '10502'
  set @part = 1

  if (year(@startdate) % 2) = 0
  begin
    set @course_period_id = newid()
    set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
    insert into course_period
    values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

    set @course_period_module_id = newid()
    insert into course_period_module
    values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

    set @module_id = 3
    insert into course_period_moduleitem
    values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
  end
--
  set @number = '10911'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
```

```
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 1
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '10914'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '10921'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '10941'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '11208'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 2
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '11305'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 2
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 3
```

```
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '11424'
  set @part = 3

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 3
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '11713'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '11714'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '11731'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 10
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '11732'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
```

```
  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 1
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 2
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '11733'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '11734'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '11743'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '11744'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 3
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '11912'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)
```

```
  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 10
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '11915'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 2
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '11931'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 10
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '11932'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '11933'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '11935'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 2
  insert into course_period_moduleitem
```

```
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '12140'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '12200'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 2
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 10
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '12700'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '26010'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
```

```
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 10
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '26906'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 3
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '26909'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 10
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '26912'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '26930'
  set @part = 1

  if (year(@startdate) % 2) = 0
  begin
    set @course_period_id = newid()
    set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
    insert into course_period
    values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

    set @course_period_module_id = newid()
    insert into course_period_module
    values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

    set @module_id = 4
    insert into course_period_moduleitem
    values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
  end
--
  set @number = '26936'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '27021'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
```

```
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 2
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 6
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 7
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 8
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
set @number = '27711'
set @part = 1

set @course_period_id = newid()
set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
insert into course_period
values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 7
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 8
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 9
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 10
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
set @number = '27721'
set @part = 1

set @course_period_id = newid()
set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
insert into course_period
values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 3
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 4
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)
```

```
  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 10
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '27731'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 1
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 10
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '27752'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 1
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 2
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 3
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '28010'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
```

```
  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '28011'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '28153'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 3
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 10
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '31625'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 10
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '34059'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 1
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 2
  insert into course_period_moduleitem
```

```
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '42936'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '88001'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 1
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 3
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '12423'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '26001'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 6
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '26004'
```

```
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '26025'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 3
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '26026'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '26170'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '26270'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '26370'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)
```

```
  set @module_id = 1
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 6
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '26372'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '26400'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 6
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '26406'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 2
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 3
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 10
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
```

```
--
  set @number = '26470'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 1
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '27000'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '27001'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '27253'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '27942'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 6
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 10
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '31000'
  set @part = 1
```

```
  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '31300'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '31630'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 1
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '31840'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 1
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 2
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 3
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '41010'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 2
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
```

```
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 10
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '41120'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '41601'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 1
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 3
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '41632'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 1
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 2
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '41639'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)
```

```
  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '42222'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '42415'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 3
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '42642'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '42930'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '42932'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 3
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 6
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
```

```
--
  set @number = '42960'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 6
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '28863'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '42425'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '42813'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '01950'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '11422'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
```

```
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 1
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 2
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 3
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 4
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 5
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 6
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 7
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 8
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 9
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 10
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
set @number = '02115'
set @part = 1

set @course_period_id = newid()
set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
insert into course_period
values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

set @course_period_module_id = newid()
insert into course_period_module
values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

set @module_id = 13
insert into course_period_moduleitem
values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
set @number = '42966'
set @part = 1

set @course_period_id = newid()
set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
```

```
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 11
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '27485'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 3
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 10
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 2
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 6
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  fetch next from period_cursor into @period_id, @startdate
end

close period_cursor
deallocate period_cursor
```

4.1.4 Manual – Spring periods only

```
declare @courseversion_id uniqueidenti↓er
declare @course_period_id uniqueidenti↓er
declare @course_period_module_id uniqueidenti↓er

declare @number varchar(20)
declare @part int
declare @module_id int
declare @period_id uniqueidenti↓er
declare @startdate datetime

declare @userid uniqueidenti↓er
declare @curdate datetime

set @userid = '{40AA0D66-B5A3-4E42-9B13-BCDA0905EB9D}'
set @curdate = getdate()
----------------------------------------------------------------------------------------------------------------------------------------------------------------


declare period_cursor cursor for
select period_id, startdate
from period
where periodtype_id = 1
and month(startdate) in (1,2)
for read only
open period_cursor

fetch next from period_cursor into @period_id, @startdate
while @@fetch_status = 0
```

```
begin
  set @number = '01005'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 1
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 3
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '01007'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 3
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '01902'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 6
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '01905'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 1
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
```

```
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 2
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '02120'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '02312'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '02323'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '02355'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '02402'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '02405'
```

```
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '02411'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 1
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '02453'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '02537'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '02601'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 2
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '02701'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '02902'
  set @part = 1

  if (year(@startdate) % 2) = 0
```

```
  begin
    set @course_period_id = newid()
    set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
    insert into course_period
    values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

    set @course_period_module_id = newid()
    insert into course_period_module
    values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

    set @module_id = 4
    insert into course_period_moduleitem
    values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
  end
- -
  set @number = '02905'
  set @part = 1

  if (year(@startdate) % 2) <> 0
  begin
    set @course_period_id = newid()
    set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
    insert into course_period
    values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

    set @course_period_module_id = newid()
    insert into course_period_module
    values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

    set @module_id = 1
    insert into course_period_moduleitem
    values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
  end
- -
  set @number = '02909'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '02913'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '10001'
  set @part = 3

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 10
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '10004'
```

```
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 1
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '10011'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '10061'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 10
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '10468'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '10469'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '10911'
  set @part = 1

  set @course_period_id = newid()
```

```
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '10914'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 2
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '10921'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '10941'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '11305'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 2
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 3
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '11424'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)
```

```
  set @module_id = 3
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '11713'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 1
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '11714'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '11722'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '11724'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 3
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '11731'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
```

```
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '11732'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 3
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '11733'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '11734'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '11743'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 3
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '11915'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 2
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '11744'
```

```
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '11784'
  set @part = 1

  if year(@startdate) <> 2004
  begin
    set @course_period_id = newid()
    set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
    insert into course_period
    values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

    set @course_period_module_id = newid()
    insert into course_period_module
    values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

    set @module_id = 6
    insert into course_period_moduleitem
    values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
  end
--
  set @number = '11924'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 10
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '11933'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '11935'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 2
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '12700'
```

```
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 1
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 2
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '26900'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 2
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '26906'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 3
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '26926'
  set @part = 1

  if (year(@startdate) % 2) <> 0
  begin
    set @course_period_id = newid()
    set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
    insert into course_period
    values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

    set @course_period_module_id = newid()
    insert into course_period_module
    values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

    set @module_id = 7
    insert into course_period_moduleitem
    values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
  end
- -
  set @number = '26936'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 6
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '28010'
```

```
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '28011'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '28153'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 6
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '88001'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 1
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 3
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '26001'
  set @part = 2

  set @course_period_id = newid()
```

```
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '26025'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '26026'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '26170'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '26270'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '26300'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)
```

```
  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 2
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 10
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 6
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '26301'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 2
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 10
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 6
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '26370'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 1
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 2
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '26372'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '26400'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
```

```
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '26406'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 2
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 3
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 10
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '26470'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 1
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '27000'
  set @part = 3

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '27001'
  set @part = 3

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
```

```
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '27031'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 10
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '27253'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '27942'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 6
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 10
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
  set @number = '31000'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
--
```

```
  set @number = '31011'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '31300'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 10
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '31630'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 1
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '41040'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 6
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '41120'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 10
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '41262'
  set @part = 1
```

```
  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 6
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '41640'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 2
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '41812'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 2
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '42415'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '42930'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '42954'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 3
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
```

```
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '42960'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '42020'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 10
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '10503'
  set @part = 1

  if (year(@startdate) % 2) = 0
  begin
    set @course_period_id = newid()
    set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
    insert into course_period
    values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

    set @course_period_module_id = newid()
    insert into course_period_module
    values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

    set @module_id = 4
    insert into course_period_moduleitem
    values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
  end
- -
  set @number = '28863'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '42425'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
```

```
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '42813'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 1
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '42813'
  set @part = 3

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 3
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '01950'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '01950'
  set @part = 3

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
- -
  set @number = '11422'
  set @part = 3

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 1
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 2
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 3
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
```

```
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 4
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 5
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 6
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)
  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 7
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 8
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 9
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  set @course_period_module_id = newid()
  insert into course_period_module
  values(@course_period_module_id, @course_period_id, @curdate, @userid, @curdate, @userid)

  set @module_id = 10
  insert into course_period_moduleitem
  values(newid(), @course_period_module_id, @module_id, @curdate, @userid, @curdate, @userid)

  fetch next from period_cursor into @period_id, @startdate
end

close period_cursor
deallocate period_cursor
```

4.1.5 Manual – January periods only

```
declare @courseversion_id uniqueidenti↓er
declare @course_period_id uniqueidenti↓er
declare @course_period_module_id uniqueidenti↓er

declare @number varchar(20)
declare @part int
declare @module_id int
declare @period_id uniqueidenti↓er
declare @startdate datetime

declare @userid uniqueidenti↓er
declare @curdate datetime

set @userid = '{40AA0D66-B5A3-4E42-9B13-BCDA0905EB9D}'
set @curdate = getdate()
-------------------------------------------------------------------------------------------------------------------------------


declare period_cursor cursor for
select period_id
from period
where periodtype_id = 2
and month(startdate) = 1
for read only
open period_cursor

fetch next from period_cursor into @period_id
while @@fetch_status = 0
begin
  set @number = '02312'
  set @part = 2

  set @course_period_id = newid()
```

```
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '02329'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '02355'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '02365'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '02661'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '10001'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '10322'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '10468'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '10469'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '11424'
  set @part = 4

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '11715'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '11733'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '11745'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
```

```
--
  set @number = '11751'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
--
  set @number = '11762'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
--
  set @number = '11805'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
--
  set @number = '11912'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
--
  set @number = '11932'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
--
  set @number = '12200'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
--
  set @number = '12243'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
--
  set @number = '28011'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
--
  set @number = '28181'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
--
  set @number = '31031'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
--
  set @number = '34049'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
--
  set @number = '41845'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
--
  set @number = '42845'
  set @part = 2
```

```
  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '26470'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '27000'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '27001'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '27942'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '31020'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '31027'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '31070'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '31305'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '31385'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '31793'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '33430'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '33443'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
```

```
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '33470'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '33471'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '41010'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '41123'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '41431'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '41601'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '41632'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '41638'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '41639'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '42120'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '42222'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '42260'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
```

```
  set @number = '42302'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
--
  set @number = '42305'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
--
  set @number = '42341'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
--
  set @number = '42642'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
--
  set @number = '42981'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
--
  set @number = '88383'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
--
  set @number = '11422'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
--
  set @number = '42646'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  fetch next from period_cursor into @period_id
end

close period_cursor
deallocate period_cursor
```

4.1.6 Manual – June periods only

```
declare @courseversion_id uniqueidenti↓er
declare @course_period_id uniqueidenti↓er
declare @course_period_module_id uniqueidenti↓er

declare @number varchar(20)
declare @part int
declare @module_id int
declare @period_id uniqueidenti↓er
declare @startdate datetime

declare @userid uniqueidenti↓er
declare @curdate datetime

set @userid = '{40AA0D66-B5A3-4E42-9B13-BCDA0905EB9D}'
set @curdate = getdate()
-------------------------------------------------------------------------------------------------------------------------

declare period_cursor cursor for
select period_id, startdate
from period
where periodtype_id = 2
and month(startdate) = 6
for read only
```

```
open period_cursor

fetch next from period_cursor into @period_id, @startdate
while @@fetch_status = 0
begin
  set @number = '02312'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '02329'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '02355'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '02365'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '02661'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '10322'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '10472'
  set @part = 2

  if exists (select period_id from period where period_id = @period_id and year(startdate) <> 2003)
  begin
    set @course_period_id = newid()
    set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
    insert into course_period
    values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
  end
- -
  set @number = '11424'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '11715'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '11733'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '11745'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '11751'
  set @part = 2
```

```
  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '11762'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '11805'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '11924'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '12243'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '26010'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '26940'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '27801'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '27802'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '28021'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '28181'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '31031'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '34049'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
```

```
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '41845'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '42845'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '12423'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '26470'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '27000'
  set @part = 4

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '27031'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '27942'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '31011'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '31020'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '31027'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '31070'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '31305'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '31385'
```

```
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
--
  set @number = '31793'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
--
  set @number = '33430'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
--
  set @number = '33443'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
--
  set @number = '33470'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
--
  set @number = '33471'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
--
  set @number = '41040'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
--
  set @number = '41123'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
--
  set @number = '41262'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
--
  set @number = '41431'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
--
  set @number = '41638'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
--
  set @number = '41640'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
--
  set @number = '41812'
  set @part = 2

  set @course_period_id = newid()
```

```
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '42120'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '42260'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '42302'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '42305'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '42341'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '42954'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '42981'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '88383'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '42020'
  set @part = 2

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '11422'
  set @part = 4

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '41390'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
- -
  set @number = '02536'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
```

```
--
  set @number = '12325'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
--
  set @number = '12801'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
--
  set @number = '28902'
  set @part = 1

  if (year(@startdate) % 2) <> 0
  begin
    set @course_period_id = newid()
    set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
    insert into course_period
    values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
  end
--
  set @number = '28903'
  set @part = 1

  if (year(@startdate) % 2) <> 0
  begin
    set @course_period_id = newid()
    set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
    insert into course_period
    values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
  end
--
  set @number = '02535'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
--
  set @number = '27033'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
--
  set @number = '31275'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)
--
  set @number = '41245'
  set @part = 1

  set @course_period_id = newid()
  set @courseversion_id = (select top 1 courseversion_id from coursegrab where number = @number order by created desc)
  insert into course_period
  values(@course_period_id, @courseversion_id, @period_id, @part, @curdate, @userid, @curdate, @userid)

  fetch next from period_cursor into @period_id, @startdate

end

close period_cursor
deallocate period_cursor
```

4.2 Course Relations

4.2.1 Desirable Prerequisites

```
declare @number varchar(20)
declare @relnumber varchar(20)
declare @course_relationcourse_id uniqueidenti↓er
declare @courseversion_id uniqueidenti↓er
declare @course_id uniqueidenti↓er

declare @userid uniqueidenti↓er
declare @curdate datetime

set @userid = '{40AA0D66-B5A3-4E42-9B13-BCDA0905EB9D}'
```

```
set @curdate = getdate()
-----------------------------------------------------------------------------------------------------------------------------------

set @number = '01032'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '10010'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

--Desirable prerequisites for course 01243 cannot be represented
--Some programming experience

--Desirable prerequisites for course 02323 cannot be represented
--No English text specified. Danish text is: Matematik svarende til 1. halvårs pensum

--Desirable prerequisites for course 02407 cannot be represented
--Knowledge of programming, e.g. Matlab.

set @number = '02421'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '02417'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

--

set @number = '02427'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '02409'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

--

set @number = '02431'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '02701'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '02735'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '02407'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

--Desirable prerequisites for course 02561 cannot be represented
--Programming experinence

set @number = '02565'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '11202'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '11913'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '41816'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '34641'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)
```

```
set @relnumber = '34642'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '34643'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

--

set @number = '02611'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '02401'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '02402'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

--

set @number = '02613'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '02220'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '02333'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

--

set @number = '02623'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '02601'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

--

set @number = '02701'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '01005'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '02323'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '02405'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

--

set @number = '02725'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '02701'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '04030'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)
```

```
set @number = '02725'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '02405'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

--Desirable prerequisites for course 10454 cannot be represented
--No English text specified. Danish text is: Påbegyndt Teknisk Fysik Fagpakke

--Desirable prerequisites for course 11005 cannot be represented
--No English text specified. Danish text is: Så mange B-faglige kurser som muligt

set @number = '11411'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '11801'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '12401'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '12420'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @number = '11411'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '02323'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '02402'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

--

set @number = '11412'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '12400'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '11401'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '56002'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

--

set @number = '11413'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '11801'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '12420'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @number = '11413'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
```

```
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '10001'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '10010'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '10011'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

--

set @number = '11416'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '02601'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

--

set @number = '11421'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '21000'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '21001'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '26000'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '26001'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '12200'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

--Desirable prerequisites for course 11422 cannot be represented
--An advanced BYG.DTU, IMM course or other relevant prerequisites upon agreement with the contact persons.

--Desirable prerequisites for course 11424 cannot be represented
--An advanced BYG.DTU course or other relevant prerequisites upon agreement with the contact person

--Desirable prerequisites for course 12240 cannot be represented
--As many environmental engineering courses as possible. In addition, environmental management courses may be relevant.

set @number = '12332'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '12322'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '12325'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @number = '12332'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '13121'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)
```

```
--

set @number = '12423'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '12400'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

--Desirable prerequisites for course 13003 cannot be represented
--Knowledge on programming

set @number = '26126'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '25211'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

--

set @number = '26500'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '36100'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

--

set @number = '26510'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '28241'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '10300'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '26230'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

--

set @number = '26920'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '23330'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

--

set @number = '27031'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '27021'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

--

set @number = '27311'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '27021'
```

```
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

--

set @number = '27321'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '25111'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '27021'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '27311'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '27731'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

--

set @number = '27405'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '27011'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

--

set @number = '27511'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '02423'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @number = '27511'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '02402'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @number = '27511'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '02591'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

--Desirable prerequisites for course 27552 cannot be represented
--At least one advanced course in biotechnology, building energy technology or indoor climate

set @number = '28241'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '28120'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

--Desirable prerequisites for course 28252 cannot be represented
--Mathematical and Process technology background

--Desirable prerequisites for course 28252 cannot be represented
```

```
--English text not specified. Danish text is: Godt kendskab til organisk, uorganisk, biologisk, fysisk kemi og teknisk kemi, herunder procesteknik,
  enhedsoperationer og reaktionsteknik

set @number = '28416'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '36100'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

--Desirable prerequisites for course 28443 cannot be represented
--Experience in numerical solutions of mathematic models

set @number = '28811'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '41015'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '42110'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '42981'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

--

set @number = '31358'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '31353'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '32630'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '92908'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '92907'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

--Desirable prerequisites for course 31365 not specified

set @number = '31405'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '02323'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '02405'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @number = '31405'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '02322'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

--

set @number = '31440'
set @course_relationcourse_id = newid()
```

```
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '02322'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

--Desirable prerequisites for course 31653 cannot be represented
--Experience with programming

--Desirable prerequisites for course 31825 cannot be represented
--Experience with laboratory work from practical exercises

--Desirable prerequisites for course 33441 cannot be represented
--Some experience with numerical methods, e.g., from Technical Physics Fagpakke

set @number = '41212'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '41225'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

--

set @number = '41324'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '41320'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '41232'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

--

set @number = '41390'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '28835'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '41312'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '42822'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

--

set @number = '41420'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '41815'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '41814'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

--

set @number = '41525'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '70205'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
```

```
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '41512'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

--

set @number = '41612'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '72111'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

--

set @number = '41627'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '41612'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

--

set @number = '41629'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '41612'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

--

set @number = '42215'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '42302'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

--Desirable prerequisites for course 42222 cannot be represented
--Danish text: Fagpakkeprojekt

--Desirable prerequisites for course 42250 cannot be represented
--Computer user knowledge

set @number = '42371'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '42450'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '80284'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '83284'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

--

set @number = '42405'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '42415'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)
```

```
--

set @number = '42422'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '42405'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '80171'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '83171'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

--

set @number = '42450'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '42420'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '86174'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '83174'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

--

set @number = '42465'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '42455'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '83288'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '80288'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '42450'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '80287'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '83287'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '42460'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '83284'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '80284'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)
```

```
set @relnumber = '83290'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '80290'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '42371'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '83489'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '80489'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '42430'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '83486'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '80486'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

- -

set @number = '42521'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '42510'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

- -

set @number = '42645'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '42644'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '85401'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @relnumber = '85403'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

- -

set @number = '42983'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '26172'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)

set @number = '42983'
set @course_relationcourse_id = newid()
set @courseversion_id = (select courseversion_id from courseversion where number = @number)
insert into course_relationcourse
values(@course_relationcourse_id, @courseversion_id, 3, @curdate, @userid, @curdate, @userid)

set @relnumber = '28212'
set @course_id = (select course_id from courseversion where number = @relnumber)
insert into course_relationcourseitem
values(newid(), @course_relationcourse_id, @course_id, @relnumber, @curdate, @userid, @curdate, @userid)
```

4.2.2 Mandatory Prerequisites

```
declare @number varchar(20)

declare @manpreq table (
  number varchar(20),
  manpreqcourse varchar(20),
  relationcourse_id uniqueidentifier
)
-----------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------

set @number = '02225'
insert into @manpreq values (@number,'02130','{D9883192-D267-427c-8702-3F3B0E014530}')

set @number = '02266'
insert into @manpreq values (@number,'02264','{AA459B36-FE75-4d29-A16C-9C0818554A8D}')

-- course 02365: mandatory requirements can not be handled.

-- course 02374: mandatory requirements can not be handled.

-- course 02376: mandatory requirements can not be handled.

-- course 10502: mandatory requirements can not be handled.

-- course 11771: some parts of the mandatory requirements can not be handled.
set @number = '11771'
insert into @manpreq values (@number,'11742','{11A14B3D-8C21-4da7-9EAF-5219DA761B5E}')

set @number = '11931'
insert into @manpreq values (@number,'11911','{CF57031D-B864-4bd7-B452-6BD83B9CD262}')
insert into @manpreq values (@number,'11921','{CF57031D-B864-4bd7-B452-6BD83B9CD262}')
insert into @manpreq values (@number,'01931','{CF57031D-B864-4bd7-B452-6BD83B9CD262}')
insert into @manpreq values (@number,'01932','{CF57031D-B864-4bd7-B452-6BD83B9CD262}')

set @number = '11933'
insert into @manpreq values (@number,'11913','{FF776303-3198-4c09-BE35-EDCAE7CEA3AE}')

-- course 12200: mandatory requirements can not be handled.

-- course 12210: mandatory requirements can not be handled.

-- course 12801: mandatory requirements can not be handled.

set @number = '13003'
insert into @manpreq values (@number,'13000','{9D0AA677-98E7-4354-928A-FDEF3DA157B9}')

set @number = '13123'
insert into @manpreq values (@number,'13000','{2A50E4D7-F972-453b-B4FD-42F38439E143}')

-- course 26002: some parts of the mandatory requirements can not be handled.
set @number = '26002'
insert into @manpreq values (@number,'21001','{50368702-BCD6-4ec8-BB82-ACD6AEB8A299}')
insert into @manpreq values (@number,'26001','{50368702-BCD6-4ec8-BB82-ACD6AEB8A299}')

set @number = '26120'
insert into @manpreq values (@number,'21245','{0E375006-1E78-4546-B131-6421556B994F}')
insert into @manpreq values (@number,'26300','{0E375006-1E78-4546-B131-6421556B994F}')
insert into @manpreq values (@number,'56059','{0E375006-1E78-4546-B131-6421556B994F}')
insert into @manpreq values (@number,'56060','{0E375006-1E78-4546-B131-6421556B994F}')

set @number = '26122'
insert into @manpreq values (@number,'21245','{D40CD911-E50B-4a9d-85BD-B43D4A986A1D}')
insert into @manpreq values (@number,'26300','{D40CD911-E50B-4a9d-85BD-B43D4A986A1D}')

set @number = '26300'
insert into @manpreq values (@number,'21211','{479B71AF-F37C-454e-AA8B-262F6B5FC5C8}')
insert into @manpreq values (@number,'21001','{479B71AF-F37C-454e-AA8B-262F6B5FC5C8}')
insert into @manpreq values (@number,'26000','{479B71AF-F37C-454e-AA8B-262F6B5FC5C8}')
insert into @manpreq values (@number,'26001','{479B71AF-F37C-454e-AA8B-262F6B5FC5C8}')

set @number = '26301'
insert into @manpreq values (@number,'21211','{28818981-F5DF-4681-8749-C5059E787AFC}')
insert into @manpreq values (@number,'21001','{28818981-F5DF-4681-8749-C5059E787AFC}')
insert into @manpreq values (@number,'26000','{28818981-F5DF-4681-8749-C5059E787AFC}')
insert into @manpreq values (@number,'26001','{28818981-F5DF-4681-8749-C5059E787AFC}')

set @number = '26312'
insert into @manpreq values (@number,'2107','{37676230-E74C-421b-B322-F4115D58EB48}')
insert into @manpreq values (@number,'21245','{37676230-E74C-421b-B322-F4115D58EB48}')
insert into @manpreq values (@number,'26300','{37676230-E74C-421b-B322-F4115D58EB48}')
-- and
insert into @manpreq values (@number,'8860','{FD238FD0-7D70-4c37-9561-6B73CCB4D548}')
insert into @manpreq values (@number,'56060','{FD238FD0-7D70-4c37-9561-6B73CCB4D548}')
insert into @manpreq values (@number,'8859','{FD238FD0-7D70-4c37-9561-6B73CCB4D548}')
insert into @manpreq values (@number,'56059','{FD238FD0-7D70-4c37-9561-6B73CCB4D548}')

set @number = '26330'
insert into @manpreq values (@number,'21245','{BD65923C-25CD-4872-8F3E-C3639FFFADD2}')
insert into @manpreq values (@number,'56060','{BD65923C-25CD-4872-8F3E-C3639FFFADD2}')
insert into @manpreq values (@number,'26300','{BD65923C-25CD-4872-8F3E-C3639FFFADD2}')

set @number = '26372'
insert into @manpreq values (@number,'91312','{DB817669-BCB0-4fbf-A8EF-F770EC8636C0}')
```

```
insert into @manpreq values (@number,'26370','{DB817669-BCB0-4fbf-A8EF-F770EC8636C0}')

set @number = '26378'
insert into @manpreq values (@number,'26376','{0DAB6A75-6888-4f99-A551-C42B1F099094}')
insert into @manpreq values (@number,'91316','{0DAB6A75-6888-4f99-A551-C42B1F099094}')

set @number = '26406'
insert into @manpreq values (@number,'26001','{EF29AFDD-0D13-43c5-9559-F626BE4E5775}')
insert into @manpreq values (@number,'26400','{EF29AFDD-0D13-43c5-9559-F626BE4E5775}')
insert into @manpreq values (@number,'26401','{EF29AFDD-0D13-43c5-9559-F626BE4E5775}')
insert into @manpreq values (@number,'23110','{EF29AFDD-0D13-43c5-9559-F626BE4E5775}')
-- and
insert into @manpreq values (@number,'21001','{5D27709D-5C13-4901-A0B0-C86E806B936E}')
insert into @manpreq values (@number,'21210','{5D27709D-5C13-4901-A0B0-C86E806B936E}')
-- and
insert into @manpreq values (@number,'21211','{39DF1419-8A03-4196-9E35-1425220D5E1A}')

set @number = '26416'
insert into @manpreq values (@number,'26406','{F7E57BA6-3E6D-4374-81B2-75B3F1CF1428}')
insert into @manpreq values (@number,'23117','{F7E57BA6-3E6D-4374-81B2-75B3F1CF1428}')

set @number = '26418'
insert into @manpreq values (@number,'26406','{DE2FA4AD-AE6F-4e14-B593-B08D281589EC}')
insert into @manpreq values (@number,'23117','{DE2FA4AD-AE6F-4e14-B593-B08D281589EC}')

set @number = '26472'
insert into @manpreq values (@number,'26372','{EA6E35A3-35AF-48f0-B046-E99746E06255}')
-- and
insert into @manpreq values (@number,'26470','{2BAFC68E-00CE-4720-9355-84B7B34A18AC}')

set @number = '26474'
insert into @manpreq values (@number,'91323','{AA5CBE86-65AF-4e1a-A796-9632B1B1791A}')
insert into @manpreq values (@number,'26472','{AA5CBE86-65AF-4e1a-A796-9632B1B1791A}')

set @number = '26500'
insert into @manpreq values (@number,'21210','{3B3E93E3-440B-4d6b-B44A-77AD018AB802}')
insert into @manpreq values (@number,'26000','{3B3E93E3-440B-4d6b-B44A-77AD018AB802}')
insert into @manpreq values (@number,'26001','{3B3E93E3-440B-4d6b-B44A-77AD018AB802}')

-- course 26920: mandatory requirements can not be handled.

-- course 26926: mandatory requirements can not be handled.

-- course 26930: mandatory requirements can not be handled.

-- course 27081: mandatory requirements can not be handled.

set @number = '27931'
insert into @manpreq values (@number,'26170','{6ADB0EA3-8598-473a-9AA7-3A883B19B4FD}')
insert into @manpreq values (@number,'26370','{6ADB0EA3-8598-473a-9AA7-3A883B19B4FD}')
insert into @manpreq values (@number,'26372','{6ADB0EA3-8598-473a-9AA7-3A883B19B4FD}')

set @number = '27942'
insert into @manpreq values (@number,'27931','{07477A46-F4A9-4415-B356-269A7D6C95A5}')

set @number = '27961'
insert into @manpreq values (@number,'27931','{7493E32D-5432-41ac-80C9-F13D81C3F85A}')
insert into @manpreq values (@number,'27942','{7493E32D-5432-41ac-80C9-F13D81C3F85A}')

set @number = '28121'
insert into @manpreq values (@number,'27931','{347F2884-5BAA-4d88-8738-649BEAD60C40}')
insert into @manpreq values (@number,'27942','{347F2884-5BAA-4d88-8738-649BEAD60C40}')

-- course 28171: mandatory requirements can not be handled.

-- course 28172: mandatory requirements can not be handled.

-- course 28181: mandatory requirements can not be handled.

-- course 28375: some parts of the mandatory requirements can not be handled.
set @number = '28375'
insert into @manpreq values (@number,'28341','{674B4BCB-ADD1-41fa-B8CC-7BE984724D12}')

set @number = '28376'
insert into @manpreq values (@number,'28375','{EF840551-0E9A-436f-B55D-E96F9B337C30}')

-- course 28381: mandatory requirements can not be handled.

set @number = '28845'
insert into @manpreq values (@number,'28341','{5B4E637C-2921-48b7-9D5D-EF2344B93ED9}')

-- course 28901: mandatory requirements can not be handled.

-- course 28902: mandatory requirements can not be handled.

-- course 28905: mandatory requirements can not be handled.

-- course 31031: mandatory requirements can not be handled.

set @number = '31236'
insert into @manpreq values (@number,'31200','{A1D8BF83-B6EB-401d-971F-26C91B8197B7}')

set @number = '31356'
insert into @manpreq values (@number,'31352','{0653EB6E-64BF-40d2-84B7-1259372239A5}')

-- course 31365: mandatory requirements can not be handled.

set @number = '31620'
insert into @manpreq values (@number,'31000','{7E15A182-FA3E-4ce4-BABE-D8958ED9A9B0}')
```

```
-- and
insert into @manpreq values (@number,'01032','{9CB1A5F5-EC44-46f8-B9C7-C2830C766100}')

set @number = '31635'
insert into @manpreq values (@number,'31630','{E1AF4AA7-4DB3-4dc1-9906-AD6ED69D4565}')

set @number = '31636'
insert into @manpreq values (@number,'31630','{5F7B7E9B-591B-4a66-AB88-31BA1C0E21E7}')

set @number = '31656'
insert into @manpreq values (@number,'31650','{05EED5C4-EE1C-4e12-9B4B-ADBB5B5EA4BE}')
insert into @manpreq values (@number,'31657','{05EED5C4-EE1C-4e12-9B4B-ADBB5B5EA4BE}')
insert into @manpreq values (@number,'31658','{05EED5C4-EE1C-4e12-9B4B-ADBB5B5EA4BE}')

set @number = '41030'
insert into @manpreq values (@number,'42040','{BE5E792F-B3EB-478f-B34A-69DF3CBCFC76}')

set @number = '41035'
insert into @manpreq values (@number,'41015','{D6025922-52C9-4046-8756-115F5E50AE5B}')
-- and
insert into @manpreq values (@number,'01007','{7611BDB1-1CA7-4f1c-B337-A3A0D9FAA9B1}')

set @number = '41040'
insert into @manpreq values (@number,'41010','{C6EEA025-5B6D-4dc0-96A2-F85F84E2E112}')
-- and
insert into @manpreq values (@number,'42020','{701D0939-235F-4283-AFEA-E19AD029E051}')
-- and
insert into @manpreq values (@number,'41030','{F6186FD0-DE96-4032-84E7-1430F6EFBCAC}')

set @number = '41110'
insert into @manpreq values (@number,'12300','{1DB79E75-FAB6-4789-802E-41028F2BAEB3}')

set @number = '41212'
insert into @manpreq values (@number,'02401','{0FD490CE-054F-41f7-B727-12805DB030A1}')

set @number = '41322'
insert into @manpreq values (@number,'41313','{895DB9CE-4278-4fc9-9D18-BC229F80A102}')

set @number = '41638'
insert into @manpreq values (@number,'42930','{32B72532-12DF-4f9d-BC0A-87AF5E7E3B9A}')
insert into @manpreq values (@number,'94405','{32B72532-12DF-4f9d-BC0A-87AF5E7E3B9A}')
-- and
insert into @manpreq values (@number,'42902','{E4CA4B65-7EA8-4897-A8FD-751C9C743DAC}')
insert into @manpreq values (@number,'94410','{E4CA4B65-7EA8-4897-A8FD-751C9C743DAC}')
-- and
insert into @manpreq values (@number,'41633','{E4238949-B252-4143-AC7A-99AD4B369212}')
-- and
insert into @manpreq values (@number,'42951','{853D8606-D8E8-42e0-B1B4-B81EA65768EB}')

-- course 41845: mandatory requirements can not be handled.

set @number = '42150'
insert into @manpreq values (@number,'42110','{FE231AEE-22A3-4c3b-9C80-51A9B1277106}')
insert into @manpreq values (@number,'80251','{FE231AEE-22A3-4c3b-9C80-51A9B1277106}')

-- course 42845: mandatory requirements can not be handled.

-- course 42859: mandatory requirements can not be handled.

-- course 42952: some parts of the mandatory requirements can not be handled.
set @number = '42952'
insert into @manpreq values (@number,'42951','{1C606B7D-CD1C-4c48-B2BE-83CD44E646E7}')
insert into @manpreq values (@number,'94500','{1C606B7D-CD1C-4c48-B2BE-83CD44E646E7}')
-- and
insert into @manpreq values (@number,'42959','{F5A0988D-A757-4b5f-8FCD-3EB09652E20D}')
-- and
insert into @manpreq values (@number,'94920','{66569239-8979-486d-A432-D8BC2B9A456F}')

set @number = '42954'
insert into @manpreq values (@number,'02323','{FA42B5F8-625F-4c06-B917-AB757231ED34}') -- Probability and Statistics

set @number = '42965'
insert into @manpreq values (@number,'02323','{A0FFAF0F-8595-4cf0-BC90-CF465E49E8C8}')

set @number = '42967'
insert into @manpreq values (@number,'41830','{4DD7B624-A8FB-4477-8EF7-8BF0B16041D3}')

set @number = '42968'
insert into @manpreq values (@number,'42954','{71A5A21B-01AF-4c3f-BC6D-88BEBDF610ED}')

-------------------------------------------------------------------------------------------------------------------------------------------------------------

-- ********************************************************************************************** --
-------------------------------------------------------------------------------------------------------------------------------------------------------------


declare @userid uniqueidenti↓er
set @userid = '{40AA0D66-B5A3-4E42-9B13-BCDA0905EB9D}'

declare @curdate datetime
set @curdate = getdate()

declare @RelationCourseType_ID int
set @RelationCourseType_ID = 1

declare @courseversion_id uniqueidenti↓er

declare man_cursor cursor for
select distinct relationcourse_id, number
```

```
from @manpreq
order by number
for read only
open man_cursor

declare @coursenumber varchar(20)
declare @relationcourse_id uniqueidenti↓er

fetch next from man_cursor into @relationcourse_id, @coursenumber
while @@fetch_status = 0
begin
  if exists (select courseversion_id from courseversion where number=@coursenumber)
  begin
    select @courseversion_id=courseversion_id from courseversion where number=@coursenumber
    insert into Course_RelationCourse
    (Course_RelationCourse_ID,
     CourseVersion_ID,
     Course_RelationCourseType_ID,
     Created,
     CreatedBy,
     Updated,
     UpdatedBy)
    values
    (@relationcourse_id,
     @courseversion_id,
     @RelationCourseType_ID,
     @curdate,
     @userid,
     @curdate,
     @userid)
  end
  fetch next from man_cursor into @relationcourse_id, @coursenumber
end

close man_cursor
deallocate man_cursor

------------------------------------------------------------

declare items_cursor cursor for
select number, manpreqcourse, relationcourse_id
from @manpreq
order by number, relationcourse_id
for read only
open items_cursor

declare @manpreqcourse varchar(20)
declare @mpc_relationcourse_id uniqueidenti↓er

fetch next from items_cursor into @coursenumber, @manpreqcourse, @mpc_relationcourse_id
while @@fetch_status = 0
begin
  if exists (select courseversion_id from courseversion where number=@coursenumber)
  begin
    insert into Course_RelationCourseItem
    (Course_RelationCourseItem_ID,
     Course_RelationCourse_ID,
     Course_ID,
     CourseNumber,
     Created,
     CreatedBy,
     Updated,
     UpdatedBy)
    values
    (newid(),
     @mpc_relationcourse_id,
     null,
     @manpreqcourse,
     @curdate,
     @userid,
     @curdate,
     @userid)
  end
  fetch next from items_cursor into @coursenumber, @manpreqcourse, @mpc_relationcourse_id
end

close items_cursor
deallocate items_cursor

------------------------------------------------------------

declare items_cursor cursor for
select course_relationcourseitem_id, coursenumber
from course_relationcourseitem
for read only
open items_cursor

declare @course_relationcourseitem_id uniqueidenti↓er
declare @course_id uniqueidenti↓er

fetch next from items_cursor into @course_relationcourseitem_id, @coursenumber
while @@fetch_status = 0
begin
  if exists (select course_id from courseversion where number=@coursenumber)
  begin
    select @course_id=course_id from courseversion where number=@coursenumber

    update Course_RelationCourseItem
    set course_id=@course_id
```

```
    where course_relationcourseitem_id=@course_relationcourseitem_id
  end
  fetch next from items_cursor into @course_relationcourseitem_id, @coursenumber
end

close items_cursor
deallocate items_cursor
```

4.2.3 Technical Prerequisites

```
declare @number varchar(20)

declare @techpreq table (
  number varchar(20),
  techpreqcourse varchar(20),
  relationcourse_id uniqueidenti¦er
)

------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------


set @number = '01030'
insert into @techpreq values (@number,'01005','{4E2E4BBA-BB6C-4b1b-8646-5AB1E4893ED6}')

set @number = '01031'
insert into @techpreq values (@number,'01005','{B451D3FF-170C-4296-B16D-2ACA092CC5E1}')

set @number = '01032'
insert into @techpreq values (@number,'31000','{86CBE388-8A74-4590-8F98-02ED245D44C4}')
--and
insert into @techpreq values (@number,'01010','{7F469063-9BEB-4ed9-A155-9D56D70BBE72}')
insert into @techpreq values (@number,'01011','{7F469063-9BEB-4ed9-A155-9D56D70BBE72}')
--and
insert into @techpreq values (@number,'01012','{DD045020-12F2-47f4-B9E4-A4D50338AEE0}')
insert into @techpreq values (@number,'01014','{DD045020-12F2-47f4-B9E4-A4D50338AEE0}')

set @number = '01034'
insert into @techpreq values (@number,'01010','{F0F133C8-7C04-44fd-B1A6-4AA0AF052BB0}')
insert into @techpreq values (@number,'01011','{F0F133C8-7C04-44fd-B1A6-4AA0AF052BB0}')
--and
insert into @techpreq values (@number,'01012','{8777C90A-C4B7-43f3-9F53-CA9857A9C43B}')
insert into @techpreq values (@number,'01014','{8777C90A-C4B7-43f3-9F53-CA9857A9C43B}')
insert into @techpreq values (@number,'01005','{8777C90A-C4B7-43f3-9F53-CA9857A9C43B}')
--and
insert into @techpreq values (@number,'10010','{AB345E5D-E8B2-4447-9BAE-9EBEC2399B3C}')

set @number = '01141'
insert into @techpreq values (@number,'01005','{AECFE006-4C47-4ea9-8E1D-DF4B64E894B8}')
--and
insert into @techpreq values (@number,'01030','{E9688C83-456E-436b-8FBA-1153275B6403}')
insert into @techpreq values (@number,'01031','{E9688C83-456E-436b-8FBA-1153275B6403}')
insert into @techpreq values (@number,'01032','{E9688C83-456E-436b-8FBA-1153275B6403}')
insert into @techpreq values (@number,'01034','{E9688C83-456E-436b-8FBA-1153275B6403}')
insert into @techpreq values (@number,'28260','{E9688C83-456E-436b-8FBA-1153275B6403}')
insert into @techpreq values (@number,'36260','{E9688C83-456E-436b-8FBA-1153275B6403}')

set @number = '01227'
insert into @techpreq values (@number,'01010','{5281E194-8DAE-41c0-B22E-C11B12E6485E}')
insert into @techpreq values (@number,'01011','{5281E194-8DAE-41c0-B22E-C11B12E6485E}')
insert into @techpreq values (@number,'01000','{5281E194-8DAE-41c0-B22E-C11B12E6485E}')
insert into @techpreq values (@number,'01001','{5281E194-8DAE-41c0-B22E-C11B12E6485E}')
--and
insert into @techpreq values (@number,'01012','{BC251F70-E328-45ae-BBFC-4D35FF4FEC94}')
insert into @techpreq values (@number,'01013','{BC251F70-E328-45ae-BBFC-4D35FF4FEC94}')
insert into @techpreq values (@number,'01014','{BC251F70-E328-45ae-BBFC-4D35FF4FEC94}')
--and
insert into @techpreq values (@number,'01005','{D6C3F866-E221-4a4e-95BA-7205F0237C83}')
insert into @techpreq values (@number,'01015','{D6C3F866-E221-4a4e-95BA-7205F0237C83}')
insert into @techpreq values (@number,'01016','{D6C3F866-E221-4a4e-95BA-7205F0237C83}')

set @number = '01243'
insert into @techpreq values (@number,'01012','{9ACCAA6C-9C91-43e1-B2B6-B0A9E347AD80}')
insert into @techpreq values (@number,'01013','{9ACCAA6C-9C91-43e1-B2B6-B0A9E347AD80}')
insert into @techpreq values (@number,'01014','{9ACCAA6C-9C91-43e1-B2B6-B0A9E347AD80}')
--and
insert into @techpreq values (@number,'01020','{84103BBB-0E2D-43bf-958D-934EA05BFE0F}')
insert into @techpreq values (@number,'01021','{84103BBB-0E2D-43bf-958D-934EA05BFE0F}')
insert into @techpreq values (@number,'01002','{84103BBB-0E2D-43bf-958D-934EA05BFE0F}')
insert into @techpreq values (@number,'01003','{84103BBB-0E2D-43bf-958D-934EA05BFE0F}')
insert into @techpreq values (@number,'01005','{84103BBB-0E2D-43bf-958D-934EA05BFE0F}')

set @number = '01246'
insert into @techpreq values (@number,'01030','{F98B903E-60CA-4582-9012-71B6E244FEC0}')
insert into @techpreq values (@number,'01031','{F98B903E-60CA-4582-9012-71B6E244FEC0}')
insert into @techpreq values (@number,'01032','{F98B903E-60CA-4582-9012-71B6E244FEC0}')
insert into @techpreq values (@number,'01034','{F98B903E-60CA-4582-9012-71B6E244FEC0}')
insert into @techpreq values (@number,'28260','{F98B903E-60CA-4582-9012-71B6E244FEC0}')

set @number = '01250'
insert into @techpreq values (@number,'01030','{1BC2222E-319D-4a7d-9633-D855820552AA}')
insert into @techpreq values (@number,'01031','{1BC2222E-319D-4a7d-9633-D855820552AA}')
insert into @techpreq values (@number,'01032','{1BC2222E-319D-4a7d-9633-D855820552AA}')
insert into @techpreq values (@number,'36260','{1BC2222E-319D-4a7d-9633-D855820552AA}')
```

```
insert into @techpreq values (@number,'01034','{1BC2222E-319D-4a7d-9633-D855820552AA}')

set @number = '01257'
insert into @techpreq values (@number,'01141','{6070D791-8D75-4423-95B7-8EC9FD504DF8}')
insert into @techpreq values (@number,'02643','{6070D791-8D75-4423-95B7-8EC9FD504DF8}')
insert into @techpreq values (@number,'01246','{6070D791-8D75-4423-95B7-8EC9FD504DF8}')
insert into @techpreq values (@number,'01258','{6070D791-8D75-4423-95B7-8EC9FD504DF8}')

set @number = '01259'
insert into @techpreq values (@number,'01005','{426B77EE-EF21-4f1c-9512-96C2AA49D5FE}')
insert into @techpreq values (@number,'01016','{426B77EE-EF21-4f1c-9512-96C2AA49D5FE}')

set @number = '01425'
insert into @techpreq values (@number,'01015','{F14C293F-48B3-4e5b-9619-FC57B4B8673D}')
insert into @techpreq values (@number,'01016','{F14C293F-48B3-4e5b-9619-FC57B4B8673D}')
-- and
insert into @techpreq values (@number,'01005','{0D8172A7-D84B-4ba2-B306-46F7E671EF7A}')

set @number = '01449'
insert into @techpreq values (@number,'01248','{C928B7C5-0EF9-411a-902C-DC5A5AF0EF43}')
insert into @techpreq values (@number,'02345','{C928B7C5-0EF9-411a-902C-DC5A5AF0EF43}')
insert into @techpreq values (@number,'10340','{C928B7C5-0EF9-411a-902C-DC5A5AF0EF43}')

set @number = '01902'
insert into @techpreq values (@number,'01901','{83573EF9-DF60-45de-93A9-A629990F9778}')

set @number = '01912'
insert into @techpreq values (@number,'01911','{B496E185-0B53-4650-A25C-A1A34F8F43B3}')

set @number = '01922'
insert into @techpreq values (@number,'01921','{AB66CBE4-AC03-4f22-B8D6-358F1C928022}')

set @number = '01932'
insert into @techpreq values (@number,'01931','{23A9E0A6-7795-485a-A58D-688186E1AC79}')

set @number = '02105'
insert into @techpreq values (@number,'02100','{0A86665D-D4EB-4ed6-BDE9-CF9FF8C58D4B}')
insert into @techpreq values (@number,'02115','{0A86665D-D4EB-4ed6-BDE9-CF9FF8C58D4B}')
insert into @techpreq values (@number,'02197','{0A86665D-D4EB-4ed6-BDE9-CF9FF8C58D4B}')
insert into @techpreq values (@number,'02199','{0A86665D-D4EB-4ed6-BDE9-CF9FF8C58D4B}')
insert into @techpreq values (@number,'02312','{0A86665D-D4EB-4ed6-BDE9-CF9FF8C58D4B}')
insert into @techpreq values (@number,'02318','{0A86665D-D4EB-4ed6-BDE9-CF9FF8C58D4B}')

set @number = '02110'
insert into @techpreq values (@number,'02105','{32201AC6-7188-421f-9E70-7FCE5BD3ADAE}')

set @number = '02120'
insert into @techpreq values (@number,'02100','{9B4FE7AF-4CF3-412c-9118-6B81B1558306}')
-- and
insert into @techpreq values (@number,'02110','{CE65B796-D30C-4379-B682-1C6B99FBE6CF}')

set @number = '02130'
insert into @techpreq values (@number,'02100','{C079F9EB-90CC-412e-8EA5-B9D916E05D44}')
insert into @techpreq values (@number,'02199','{C079F9EB-90CC-412e-8EA5-B9D916E05D44}')
insert into @techpreq values (@number,'02115','{C079F9EB-90CC-412e-8EA5-B9D916E05D44}')

set @number = '02140'
insert into @techpreq values (@number,'02110','{326FA7A8-C41A-4868-8236-CCF2962A8327}')
-- and
insert into @techpreq values (@number,'01016','{326FA7A8-C41A-4868-8236-CCF2962A8327}')

set @number = '02170'
insert into @techpreq values (@number,'02100','{0CD3A4C5-7F64-421c-B2DF-1B24A8D6B789}')
-- and
insert into @techpreq values (@number,'02105','{D2B59D36-31E7-483b-A6E3-A38FB2BF8C55}')
-- and
insert into @techpreq values (@number,'01016','{04CCA0A7-733F-4a72-982A-55EE4E0C5AB0}')

set @number = '02200'
insert into @techpreq values (@number,'02130','{547D86C6-2260-4f62-A3C4-508E1984E84A}')
insert into @techpreq values (@number,'3100','{547D86C6-2260-4f62-A3C4-508E1984E84A}')
-- and
insert into @techpreq values (@number,'02199','{A1377087-BC63-4017-AB37-9DAA1BC0DD78}')

set @number = '02201'
insert into @techpreq values (@number,'02200','{84E51EAC-144F-4c09-B7ED-911803C8459D}')

set @number = '02202'
insert into @techpreq values (@number,'02200','{5DD66013-0BED-4a44-8887-E1B505DF594E}')

set @number = '02204'
insert into @techpreq values (@number,'02200','{AE59A768-6CE4-4fb5-995E-63A5413A736F}')

set @number = '02206'
insert into @techpreq values (@number,'02200','{7EABAE08-FE17-430e-AC17-E14122F2AF56}')
insert into @techpreq values (@number,'31028','{7EABAE08-FE17-430e-AC17-E14122F2AF56}')

set @number = '02208'
insert into @techpreq values (@number,'02200','{3AF43A4C-5F0E-40ff-8331-5AD3FA50D853}')
insert into @techpreq values (@number,'02206','{3AF43A4C-5F0E-40ff-8331-5AD3FA50D853}')

set @number = '02220'
insert into @techpreq values (@number,'02130','{361695FC-1551-4e91-96B0-C9124C6A89AD}')
-- and
insert into @techpreq values (@number,'02140','{83D34617-C781-40c5-A1AD-BA0B6A200C20}')

set @number = '02222'
insert into @techpreq values (@number,'02220','{DCBD5617-369A-47b8-9789-BDF8B3059B90}')
```

```
set @number = '02224'
insert into @techpreq values (@number,'02220','{5606A9D7-2056-4601-B962-4B5D5BA32A6B}')

-- prerequisites are primarily specified in free text
set @number = '02225'
insert into @techpreq values (@number,'02130','{F8843B28-9EB1-4455-9EBE-3514214B687D}')
-- and
insert into @techpreq values (@number,'02230','{3802B4EB-77E9-4386-9F67-5031EC8BBE17}')

set @number = '02226'
insert into @techpreq values (@number,'02220','{D03EC635-8B0A-4740-A87B-2330DAAE2728}')

set @number = '02230'
insert into @techpreq values (@number,'02100','{F1F1B561-CA23-408c-807E-08F6BD44C036}')
insert into @techpreq values (@number,'02199','{F1F1B561-CA23-408c-807E-08F6BD44C036}')
insert into @techpreq values (@number,'02115','{F1F1B561-CA23-408c-807E-08F6BD44C036}')
-- and
insert into @techpreq values (@number,'02110','{BEA7D17C-140F-447b-8F93-BBD47ACF5F76}')
-- and
insert into @techpreq values (@number,'02130','{8E302C1A-969A-4c9d-8536-414079029BD6}')

set @number = '02232'
insert into @techpreq values (@number,'02230','{57E2FBEE-902A-445a-9AD0-1AE7349507C7}')

set @number = '02233'
insert into @techpreq values (@number,'02230','{92C998E7-797A-45d7-81CE-2C675650F1D0}')

set @number = '02240'
insert into @techpreq values (@number,'02140','{DC658AEE-823A-483c-8A86-8A75737E1108}')

set @number = '02242'
insert into @techpreq values (@number,'02140','{E632E319-069D-420f-B3EC-217A0EA111CE}')

set @number = '02260'
insert into @techpreq values (@number,'02120','{3ED6BB7B-1927-47da-AB92-56566EAC3ED4}')
-- and
insert into @techpreq values (@number,'02130','{16C3E5AE-31B7-4772-B55B-C72140AA1EC8}')

set @number = '02261'
insert into @techpreq values (@number,'98161','{3D8D87C8-FC9B-4104-8B31-2998E957F098}')

set @number = '02262'
insert into @techpreq values (@number,'02140','{CC985D93-6402-4281-8627-05E82A96D2EB}')

set @number = '02268'
insert into @techpreq values (@number,'02262','{6D454BC6-600B-45db-B150-4A15938EA254}')

set @number = '02280'
insert into @techpreq values (@number,'02140','{137FD67B-99FE-4667-A26B-435BA26B4CC2}')

set @number = '02284'
insert into @techpreq values (@number,'02280','{FFAC0E31-67F7-43b9-A86B-622781BB00B8}')

set @number = '02286'
insert into @techpreq values (@number,'02282','{AA6E1DF6-09A2-41b7-91E0-81399EF87650}')

set @number = '02288'
insert into @techpreq values (@number,'02280','{E91DAC3C-4273-48c6-BE28-F3836DEE4B9D}')

set @number = '02321'
insert into @techpreq values (@number,'02311','{EC5FA131-CF96-4db5-B5D5-AFFB439DA06B}')
-- and
insert into @techpreq values (@number,'02312','{52080288-BD6F-48a5-9531-DA260A727E9D}')

set @number = '02322'
insert into @techpreq values (@number,'01961','{E759635F-9F41-4b47-82FF-3F22DB782279}')
insert into @techpreq values (@number,'01981','{E759635F-9F41-4b47-82FF-3F22DB782279}')

set @number = '02329'
insert into @techpreq values (@number,'02311','{F880D3F8-12FA-4ef1-865F-0ED7FAB4BDB4}')
-- and
insert into @techpreq values (@number,'02130','{42E80BDA-5B84-46d2-A031-A356978A0F22}')

set @number = '02330'
insert into @techpreq values (@number,'02321','{32781CDE-367D-4d7a-B2F1-2DF0E608E408}')
-- and
insert into @techpreq values (@number,'31021','{60FE3D98-F250-478d-A895-D769285CD31A}')

set @number = '02333'
insert into @techpreq values (@number,'02321','{97EF53DC-E4CF-40c4-B705-512ED9617A45}')

set @number = '02340'
insert into @techpreq values (@number,'02321','{32AB909A-2604-485e-8368-AEFA19E7715A}')

set @number = '02344'
insert into @techpreq values (@number,'02320','{82542F16-4F34-44ae-8D99-FE66A9CB2AD3}')

-- (some parts of the) technical prerequisites on course 02350 are not clear
set @number = '02350'
insert into @techpreq values (@number,'02312','{82615EC1-F7F3-4bd1-A274-6A262E6C60C6}')

set @number = '02355'
insert into @techpreq values (@number,'02320','{9B95975D-17BC-4f6d-9838-3D0CA3D4FC8E}')
-- and
insert into @techpreq values (@number,'02344','{3B9F1922-6572-4ebc-9FD4-7ED739D9456A}')
-- and
insert into @techpreq values (@number,'02335','{BCD6BA8E-625F-4b65-9D5F-1EEF08F0DC51}')

set @number = '02356'
```

```
insert into @techpreq values (@number,'02355','{A758F7D0-B4B3-4557-B26D-7F42FFB9E6AE}')

-- technical prerequisites on course 02393 are described in free text with no
-- course numbers specified.

-- technical prerequisites on course 02402 are described in free text with no
-- course numbers specified.

set @number = '02407'
insert into @techpreq values (@number,'02401','{C0099738-32EB-4a74-9A26-E608CE3471C3}')
insert into @techpreq values (@number,'02402','{C0099738-32EB-4a74-9A26-E608CE3471C3}')
-- and
insert into @techpreq values (@number,'02405','{15B5F3C6-7726-409f-8B17-54C1F1A42887}')

set @number = '02409'
insert into @techpreq values (@number,'02402','{72166C00-57CC-4cd7-9FB2-38232384574A}')

set @number = '02411'
insert into @techpreq values (@number,'02402','{5FDF1668-6520-4750-BBB9-C6EFF75782C1}')

set @number = '02413'
insert into @techpreq values (@number,'02402','{07A2C492-5A6E-428c-84E5-CB6F8794E6AE}')

set @number = '02417'
insert into @techpreq values (@number,'02402','{2179525E-80C0-40c6-B311-86088C2EE2C8}')

set @number = '02421'
insert into @techpreq values (@number,'02401','{082A7071-43EF-4cdc-AB4C-5141705FDDC8}')
insert into @techpreq values (@number,'02402','{082A7071-43EF-4cdc-AB4C-5141705FDDC8}')
-- and
insert into @techpreq values (@number,'31300','{1DB7F8C1-BA95-4366-A832-A5640A771C55}')
insert into @techpreq values (@number,'28250','{1DB7F8C1-BA95-4366-A832-A5640A771C55}')

set @number = '02423'
insert into @techpreq values (@number,'02402','{BB34C43F-5648-46dd-BEA4-4084CAD86C35}')

set @number = '02427'
insert into @techpreq values (@number,'02417','{D0B9DEC4-7AA9-423f-9C79-460B4E48C1D5}')

-- technical prerequisites on course 02402 are described in free text specifying
-- required qulifications/skills which is not representable (only course numbers can
-- be specified)

set @number = '02441'
insert into @techpreq values (@number,'02401','{4218145E-47A1-4b8c-8196-C47C00311DEB}')
insert into @techpreq values (@number,'04041','{4218145E-47A1-4b8c-8196-C47C00311DEB}')
insert into @techpreq values (@number,'04040','{4218145E-47A1-4b8c-8196-C47C00311DEB}')

set @number = '02443'
insert into @techpreq values (@number,'02402','{BB3595B3-21E5-435d-866F-6666C3FFE1EA}')

set @number = '02451'
insert into @techpreq values (@number,'01030','{6FF6A59F-AD6C-4e22-B602-8B5146F32313}')
insert into @techpreq values (@number,'01032','{6FF6A59F-AD6C-4e22-B602-8B5146F32313}')
-- and
insert into @techpreq values (@number,'02401','{FE3AF855-DF0E-4c24-A4B0-5263951B1669}')
insert into @techpreq values (@number,'02402','{FE3AF855-DF0E-4c24-A4B0-5263951B1669}')

set @number = '02453'
insert into @techpreq values (@number,'02451','{DD8F7D3D-FAA5-489e-BBEF-C178F6F44CA0}')

set @number = '02455'
insert into @techpreq values (@number,'02451','{B7D3FDDA-B379-463d-872A-AA9D8C7D0B0B}')

set @number = '02457'
insert into @techpreq values (@number,'02451','{961E274C-B558-47ba-86BF-D1407D0CFF4E}')
insert into @techpreq values (@number,'04361','{961E274C-B558-47ba-86BF-D1407D0CFF4E}')
insert into @techpreq values (@number,'04362','{961E274C-B558-47ba-86BF-D1407D0CFF4E}')

-- technical prerequisites on course 02501 are described in free text specifying
-- required qulifications/skills which is not representable (only course numbers can
-- be specified)

set @number = '02503'
insert into @techpreq values (@number,'02501','{725AF6A9-2419-4ba2-868F-D52E5981043E}')
insert into @techpreq values (@number,'04250','{725AF6A9-2419-4ba2-868F-D52E5981043E}')
-- and
insert into @techpreq values (@number,'02409','{5DF47614-83CA-4344-9296-A092D5AB08F2}')
insert into @techpreq values (@number,'04241','{5DF47614-83CA-4344-9296-A092D5AB08F2}')

set @number = '02505'
insert into @techpreq values (@number,'02402','{88086A7E-ED86-4ed4-81B3-437FB7AC12F5}')
-- and
insert into @techpreq values (@number,'02501','{2F9F410F-92B0-4436-88F3-86AC5B6AB37A}')

-- technical prerequisites on course 02507 are described in free text specifying
-- required qulifications/skills which is not representable (only course numbers can
-- be specified)

set @number = '02543'
insert into @techpreq values (@number,'02531','{33FC6E18-D8E4-4036-8286-ED60FB78E990}')
insert into @techpreq values (@number,'02532','{33FC6E18-D8E4-4036-8286-ED60FB78E990}')
-- and
insert into @techpreq values (@number,'02534','{AD6753EA-7D98-4059-8504-0B6351280D77}')

set @number = '02561'
insert into @techpreq values (@number,'02501','{C4073B96-A8C0-429c-9787-9C357019BF83}')
insert into @techpreq values (@number,'04251','{C4073B96-A8C0-429c-9787-9C357019BF83}')
insert into @techpreq values (@number,'67277','{C4073B96-A8C0-429c-9787-9C357019BF83}')
```

```
set @number = '02563'
insert into @techpreq values (@number,'02501','{C30214B2-324D-4c2a-A634-8F7F9E1B4EFD}')
insert into @techpreq values (@number,'04251','{C30214B2-324D-4c2a-A634-8F7F9E1B4EFD}')
insert into @techpreq values (@number,'67277','{C30214B2-324D-4c2a-A634-8F7F9E1B4EFD}')

set @number = '02565'
insert into @techpreq values (@number,'02501','{70C4CA1E-E464-42ac-8D4B-34EF71755FF2}')
insert into @techpreq values (@number,'04251','{70C4CA1E-E464-42ac-8D4B-34EF71755FF2}')
insert into @techpreq values (@number,'67277','{70C4CA1E-E464-42ac-8D4B-34EF71755FF2}')

set @number = '02567'
insert into @techpreq values (@number,'11201','{698838DA-47FC-45d6-ACE1-D8A6B4E6AB9C}')

set @number = '02581'
insert into @techpreq values (@number,'02402','{00250604-202D-45b8-BF79-54D244217373}')

set @number = '02591'
insert into @techpreq values (@number,'01905','{06BF8077-D46B-4701-9491-0B1DC7B9C7C2}')
insert into @techpreq values (@number,'01911','{06BF8077-D46B-4701-9491-0B1DC7B9C7C2}')

-- technical prerequisites on course 02601 are described in free text specifying
-- required qulifications/skills which is not representable (only course numbers can
-- be specified)

set @number = '02611'
insert into @techpreq values (@number,'02601','{4879F73E-20B7-48cc-B939-F8D9D945E028}')

-- technical prerequisites on course 02613 are described in free text specifying
-- required qulifications/skills which is not representable (only course numbers can
-- be specified)

-- some parts of the technical prerequisites on course 02621 are specified as
-- required qulifications/skills which is not representable (only course numbers can
-- be specified)
set @number = '02621'
insert into @techpreq values (@number,'02601','{7EAF9531-67D9-4a16-990D-81DC4E288404}')

-- technical prerequisites on course 02623 are described in free text specifying
-- required qulifications/skills which is not representable (only course numbers can
-- be specified)

-- course 02643: technical prerequisites are described in free text specifying
-- required qulifications/skills which is not representable (only course numbers can
-- be specified)

set @number = '02647'
insert into @techpreq values (@number,'01005','{D9822B57-D9E0-4d4e-8504-EB177F3633B2}')
-- and
insert into @techpreq values (@number,'01030','{84568B7F-8949-4475-8D2C-C6A50CC786EF}')
-- and
insert into @techpreq values (@number,'10001','{69722426-B647-4bae-9033-583ACC580974}')
insert into @techpreq values (@number,'10002','{69722426-B647-4bae-9033-583ACC580974}')
-- and
insert into @techpreq values (@number,'10014','{E37123E5-43B3-4139-A6CC-6E56AA7DAE5E}')
insert into @techpreq values (@number,'10015','{E37123E5-43B3-4139-A6CC-6E56AA7DAE5E}')
-- and
insert into @techpreq values (@number,'10100','{850EFEB5-8729-4439-8BD9-E51AB0E87B4C}')

-- course 02655: technical prerequisites are described in free text specifying
-- required qulifications/skills which is not representable (only course numbers can
-- be specified)

-- course 02661: technical prerequisites are described in free text specifying
-- required qulifications/skills which is not representable (only course numbers can
-- be specified)

set @number = '02685'
insert into @techpreq values (@number,'02601','{11F6D99C-9151-4db6-A572-F3FEF8C50D57}')
-- and
insert into @techpreq values (@number,'02643','{A3E0C200-3EAA-4fc1-B0AE-73590D813CD9}')

-- course 02687: technical prerequisites are described in free text specifying
-- required qulifications/skills which is not representable (only course numbers can
-- be specified)

set @number = '02709'
insert into @techpreq values (@number,'02701','{36AA2629-0E24-4ebb-BD2E-B18EE2344F3B}')

set @number = '02711'
insert into @techpreq values (@number,'02701','{AA36D79C-3FD4-45e4-B5BC-DD5C7D98290E}')

set @number = '02713'
insert into @techpreq values (@number,'02701','{FE75B801-DAA5-43eb-9885-E8B98C0702A7}')
insert into @techpreq values (@number,'04030','{FE75B801-DAA5-43eb-9885-E8B98C0702A7}')

-- course 02715: some parts of the technical prerequisites are specified as
-- required qulifications/skills which is not representable (only course numbers can
-- be specified)
set @number = '02715'
insert into @techpreq values (@number,'02701','{5C9E76E7-1D5A-4945-976D-7AE49FFDB5B6}')
insert into @techpreq values (@number,'04030','{5C9E76E7-1D5A-4945-976D-7AE49FFDB5B6}')

set @number = '02721'
insert into @techpreq values (@number,'02713','{BAEDE09A-893A-4cfb-9020-9130601E66A5}')
insert into @techpreq values (@number,'04232','{BAEDE09A-893A-4cfb-9020-9130601E66A5}')

-- course 02731: some parts of the technical prerequisites are specified as
-- required qulifications/skills which is not representable (only course numbers can
```

```
-- be specified)
set @number = '02731'
insert into @techpreq values (@number,'02701','{508B76B2-134B-4ce5-B3CF-94963B7F287C}')
insert into @techpreq values (@number,'02402','{508B76B2-134B-4ce5-B3CF-94963B7F287C}')

-- course 02735: some parts of the technical prerequisites are specified as
-- required qulifications/skills which is not representable (only course numbers can
-- be specified)
set @number = '02735'
insert into @techpreq values (@number,'02701','{CAF12EC6-B5CB-4d9c-BD31-0A3FFDE1C44D}')
insert into @techpreq values (@number,'04030','{CAF12EC6-B5CB-4d9c-BD31-0A3FFDE1C44D}')

-- course 02906: technical prerequisites are described in free text specifying
-- required qulifications/skills which is not representable (only course numbers can
-- be specified)

-- course 02907: technical prerequisites does not make sense

set @number = '02913'
insert into @techpreq values (@number,'02220','{0E527DE4-C82C-440d-B9EB-123A593391D1}')
-- and
insert into @techpreq values (@number,'02230','{3B205396-1907-4e87-BC11-BAD1D2B04D27}')
-- and
insert into @techpreq values (@number,'02240','{46ECEB48-0287-4083-8561-566E906529D7}')
-- and
insert into @techpreq values (@number,'02242','{E9E437E2-0F27-4d54-B405-8EFB37BF591D}')

set @number = '10001'
insert into @techpreq values (@number,'01005','{2EE2D367-92BE-4220-8C88-74AC9598448E}')

set @number = '10004'
insert into @techpreq values (@number,'01005','{011314B1-C4DD-4c77-9E2E-3F2452C33F37}')

set @number = '10009'
insert into @techpreq values (@number,'10001','{82E22098-3D0A-47b0-A20A-7756BCEAE2CA}')
insert into @techpreq values (@number,'10002','{82E22098-3D0A-47b0-A20A-7756BCEAE2CA}')
insert into @techpreq values (@number,'10010','{82E22098-3D0A-47b0-A20A-7756BCEAE2CA}')
insert into @techpreq values (@number,'10032','{82E22098-3D0A-47b0-A20A-7756BCEAE2CA}')
-- and
insert into @techpreq values (@number,'01005','{7060BAA4-C5C5-4914-94ED-861707734DB6}')

set @number = '10010'
insert into @techpreq values (@number,'01005','{F3D8C5D4-DDC3-4c7f-8042-A6A15FD19177}')
insert into @techpreq values (@number,'01010','{F3D8C5D4-DDC3-4c7f-8042-A6A15FD19177}')

set @number = '10011'
insert into @techpreq values (@number,'01005','{BF1F8E54-F81F-4ced-8EB3-6E2B22587830}')

set @number = '10012'
insert into @techpreq values (@number,'10010','{EFC4B30B-A2D3-45b3-BCEE-0BC80B14A264}')
insert into @techpreq values (@number,'10011','{EFC4B30B-A2D3-45b3-BCEE-0BC80B14A264}')

set @number = '10013'
insert into @techpreq values (@number,'10011','{1D8509C5-83DA-4179-B129-31B0EB04A4E5}')
insert into @techpreq values (@number,'10032','{1D8509C5-83DA-4179-B129-31B0EB04A4E5}')
insert into @techpreq values (@number,'10004','{1D8509C5-83DA-4179-B129-31B0EB04A4E5}')
-- and
insert into @techpreq values (@number,'01005','{2A6AF047-23EF-41db-9630-B2AB0B8875FE}')

set @number = '10061'
insert into @techpreq values (@number,'10009','{170A784F-3772-4a4a-A9B3-DBE54D18659E}')
insert into @techpreq values (@number,'10014','{170A784F-3772-4a4a-A9B3-DBE54D18659E}')
-- and
insert into @techpreq values (@number,'10015','{0D805CD2-B8D8-405b-B3CA-8D4EA186D78A}')

set @number = '10100'
insert into @techpreq values (@number,'10001','{FE6A31E9-2906-4635-98EF-5D21DFEB567D}')
insert into @techpreq values (@number,'10002','{FE6A31E9-2906-4635-98EF-5D21DFEB567D}')
insert into @techpreq values (@number,'10010','{FE6A31E9-2906-4635-98EF-5D21DFEB567D}')
insert into @techpreq values (@number,'10011','{FE6A31E9-2906-4635-98EF-5D21DFEB567D}')
-- and
insert into @techpreq values (@number,'10009','{B08BEC90-F5B1-4b05-9953-25638F3C4BE0}')
insert into @techpreq values (@number,'10013','{B08BEC90-F5B1-4b05-9953-25638F3C4BE0}')
insert into @techpreq values (@number,'10015','{B08BEC90-F5B1-4b05-9953-25638F3C4BE0}')
insert into @techpreq values (@number,'31400','{B08BEC90-F5B1-4b05-9953-25638F3C4BE0}')
insert into @techpreq values (@number,'48000','{B08BEC90-F5B1-4b05-9953-25638F3C4BE0}')

set @number = '10111'
insert into @techpreq values (@number,'01030','{7FDA741E-DDBC-4140-8269-B99E987AD969}')
insert into @techpreq values (@number,'01031','{7FDA741E-DDBC-4140-8269-B99E987AD969}')
-- and
insert into @techpreq values (@number,'10100','{32F3D59A-0694-4c92-8B84-703A78A7B64F}')

set @number = '10112'
insert into @techpreq values (@number,'10110','{F9E3415A-2304-42f1-89DD-EB21C32A4E19}')
insert into @techpreq values (@number,'10201','{F9E3415A-2304-42f1-89DD-EB21C32A4E19}')
insert into @techpreq values (@number,'10111','{F9E3415A-2304-42f1-89DD-EB21C32A4E19}')
-- and
insert into @techpreq values (@number,'10120','{4193B94E-3798-49c2-99BA-EB2FFF7A56BB}')

set @number = '10120'
insert into @techpreq values (@number,'10100','{B22AD708-B7FB-4f6f-A433-7F233B270143}')
-- and
insert into @techpreq values (@number,'01030','{9B6F5F91-0B25-4e8f-A221-4572DF7003DB}')
insert into @techpreq values (@number,'01031','{9B6F5F91-0B25-4e8f-A221-4572DF7003DB}')

set @number = '10300'
insert into @techpreq values (@number,'10201','{DFFAC31E-9BC9-4816-8BD9-C9D05A0E120D}')
insert into @techpreq values (@number,'21270','{DFFAC31E-9BC9-4816-8BD9-C9D05A0E120D}')
```

```
insert into @techpreq values (@number,'10111','{DFFAC31E-9BC9-4816-8BD9-C9D05A0E120D}')

set @number = '10302'
insert into @techpreq values (@number,'10112','{CCE8820B-4A99-4db8-8FFB-1E089BEA1F73}')
insert into @techpreq values (@number,'10221','{CCE8820B-4A99-4db8-8FFB-1E089BEA1F73}')
-- and
insert into @techpreq values (@number,'10203','{6405BBFA-6E6E-406e-A177-E3FCC6466A48}')
insert into @techpreq values (@number,'10300','{6405BBFA-6E6E-406e-A177-E3FCC6466A48}')

set @number = '10304'
insert into @techpreq values (@number,'10300','{08A4DCB1-3624-4e5f-B8A4-AA4EC23F765A}')
insert into @techpreq values (@number,'10203','{08A4DCB1-3624-4e5f-B8A4-AA4EC23F765A}')

set @number = '10306'
insert into @techpreq values (@number,'10111','{276C0A68-7956-48f8-99E6-D3266D7E746B}')
insert into @techpreq values (@number,'10201','{276C0A68-7956-48f8-99E6-D3266D7E746B}')
-- and
insert into @techpreq values (@number,'10120','{5F9E6AAD-1133-4214-A73C-5142B1ECB93E}')
insert into @techpreq values (@number,'10239','{5F9E6AAD-1133-4214-A73C-5142B1ECB93E}')
-- and
insert into @techpreq values (@number,'10203','{2973B8BF-45CD-4431-8A0F-13FCCDB02E11}')
insert into @techpreq values (@number,'10300','{2973B8BF-45CD-4431-8A0F-13FCCDB02E11}')

set @number = '10308'
insert into @techpreq values (@number,'10201','{1EE44D64-EC50-46c7-9912-BF9A26250D00}')
insert into @techpreq values (@number,'10110','{1EE44D64-EC50-46c7-9912-BF9A26250D00}')
insert into @techpreq values (@number,'10111','{1EE44D64-EC50-46c7-9912-BF9A26250D00}')
-- and
insert into @techpreq values (@number,'10203','{2B77B185-0BA0-4782-AFFE-F3FA393EDAFC}')
insert into @techpreq values (@number,'10300','{2B77B185-0BA0-4782-AFFE-F3FA393EDAFC}')

set @number = '10310'
insert into @techpreq values (@number,'10201','{86B4701F-7AF1-42d2-95E9-56671313B572}')
insert into @techpreq values (@number,'10110','{86B4701F-7AF1-42d2-95E9-56671313B572}')
insert into @techpreq values (@number,'10111','{86B4701F-7AF1-42d2-95E9-56671313B572}')
-- and
insert into @techpreq values (@number,'10300','{DE928AE2-2D61-47de-97CC-BE93C422BCAA}')

set @number = '10322'
insert into @techpreq values (@number,'10100','{70B535E9-C000-4d09-AD18-AAAECE3C4357}')

set @number = '10343'
insert into @techpreq values (@number,'10340','{15C93708-134A-472d-8F36-9080253F4A64}')
insert into @techpreq values (@number,'10341','{15C93708-134A-472d-8F36-9080253F4A64}')
insert into @techpreq values (@number,'10231','{15C93708-134A-472d-8F36-9080253F4A64}')

set @number = '10344'
insert into @techpreq values (@number,'10342','{90E95B32-75A8-4eca-A683-694BD3B982AC}')
insert into @techpreq values (@number,'10233','{90E95B32-75A8-4eca-A683-694BD3B982AC}')

set @number = '10345'
insert into @techpreq values (@number,'01030','{A6D97F5D-CF4F-4fbe-A402-EC739CBB62CB}')

set @number = '10347'
insert into @techpreq values (@number,'10100','{31152F29-5127-4865-9218-B8516E3400E1}')

set @number = '10350'
insert into @techpreq values (@number,'10257','{AF6E0676-1A88-4187-BE66-9D6399DAF4CE}')
insert into @techpreq values (@number,'10340','{AF6E0676-1A88-4187-BE66-9D6399DAF4CE}')
-- and
insert into @techpreq values (@number,'10111','{64D3A5F9-B443-4b8f-AB53-D2531D90F1A7}')
insert into @techpreq values (@number,'10110','{64D3A5F9-B443-4b8f-AB53-D2531D90F1A7}')
insert into @techpreq values (@number,'10201','{64D3A5F9-B443-4b8f-AB53-D2531D90F1A7}')

set @number = '10370'
insert into @techpreq values (@number,'10009','{BD0D093B-816F-4215-BA18-A40D3B66C817}')
insert into @techpreq values (@number,'10013','{BD0D093B-816F-4215-BA18-A40D3B66C817}')
insert into @techpreq values (@number,'10015','{BD0D093B-816F-4215-BA18-A40D3B66C817}')
insert into @techpreq values (@number,'31400','{BD0D093B-816F-4215-BA18-A40D3B66C817}')

set @number = '10370'
insert into @techpreq values (@number,'10009','{B2BDEE08-CA0F-4e72-AE42-AE7E46D9DD19}')
insert into @techpreq values (@number,'10013','{B2BDEE08-CA0F-4e72-AE42-AE7E46D9DD19}')
insert into @techpreq values (@number,'10015','{B2BDEE08-CA0F-4e72-AE42-AE7E46D9DD19}')
insert into @techpreq values (@number,'31400','{B2BDEE08-CA0F-4e72-AE42-AE7E46D9DD19}')

set @number = '10372'
insert into @techpreq values (@number,'10211','{61D41D41-4A1E-4971-8AFD-D612D0F5A562}')
insert into @techpreq values (@number,'10370','{61D41D41-4A1E-4971-8AFD-D612D0F5A562}')

set @number = '10374'
insert into @techpreq values (@number,'10100','{9785ECDD-1EE6-4464-B260-1F39D578977E}')
-- and
insert into @techpreq values (@number,'10211','{811C5802-80EF-4c3d-9DD2-9E62C231FA3E}')
insert into @techpreq values (@number,'10370','{811C5802-80EF-4c3d-9DD2-9E62C231FA3E}')

set @number = '10376'
insert into @techpreq values (@number,'10370','{60058237-9B59-4ed0-8C9C-27F1B44808F6}')
insert into @techpreq values (@number,'10211','{60058237-9B59-4ed0-8C9C-27F1B44808F6}')

set @number = '10378'
insert into @techpreq values (@number,'10110','{CA02A5E2-B3FA-4f43-9143-1A693C5E588D}')
insert into @techpreq values (@number,'10111','{CA02A5E2-B3FA-4f43-9143-1A693C5E588D}')
-- and
insert into @techpreq values (@number,'10370','{7CD57177-EDCD-4fc4-9E15-1D2A91EB7D2A}')

-- course 10381: technical prerequisites are described in free text specifying
-- required qulifications/skills which is not representable (only course numbers can
-- be specified)
```

```
set @number = '10392'
insert into @techpreq values (@number,'10211','{E15DF63D-29DB-415d-B900-4A317025B068}')
insert into @techpreq values (@number,'10370','{E15DF63D-29DB-415d-B900-4A317025B068}')

set @number = '10405'
insert into @techpreq values (@number,'10009','{E8BD013F-23ED-4e0e-B7BC-34E28D1A69C2}')
insert into @techpreq values (@number,'10013','{E8BD013F-23ED-4e0e-B7BC-34E28D1A69C2}')
insert into @techpreq values (@number,'10015','{E8BD013F-23ED-4e0e-B7BC-34E28D1A69C2}')
insert into @techpreq values (@number,'48000','{E8BD013F-23ED-4e0e-B7BC-34E28D1A69C2}')

set @number = '10412'
insert into @techpreq values (@number,'10001','{715781EF-AE8B-4ad0-86CB-356181D0BA9C}')
insert into @techpreq values (@number,'10010','{715781EF-AE8B-4ad0-86CB-356181D0BA9C}')
insert into @techpreq values (@number,'10011','{715781EF-AE8B-4ad0-86CB-356181D0BA9C}')

set @number = '10467'
insert into @techpreq values (@number,'10009','{F823B68F-B176-4835-A49E-E77CE0C149FD}')
insert into @techpreq values (@number,'10013','{F823B68F-B176-4835-A49E-E77CE0C149FD}')
insert into @techpreq values (@number,'31400','{F823B68F-B176-4835-A49E-E77CE0C149FD}')

set @number = '10468'
insert into @techpreq values (@number,'10110','{54909061-E82A-406f-B0AF-4586B445F8F0}')
insert into @techpreq values (@number,'10201','{54909061-E82A-406f-B0AF-4586B445F8F0}')
insert into @techpreq values (@number,'10111','{54909061-E82A-406f-B0AF-4586B445F8F0}')

set @number = '10469'
insert into @techpreq values (@number,'10110','{FB28F088-AA1E-4e67-BA53-8472BB252B2F}')
insert into @techpreq values (@number,'10201','{FB28F088-AA1E-4e67-BA53-8472BB252B2F}')
insert into @techpreq values (@number,'10111','{FB28F088-AA1E-4e67-BA53-8472BB252B2F}')

set @number = '10472'
insert into @techpreq values (@number,'01901','{5FC5DEC5-69CA-4993-81D5-E3EF23547A8A}')
insert into @techpreq values (@number,'01902','{5FC5DEC5-69CA-4993-81D5-E3EF23547A8A}')
insert into @techpreq values (@number,'01005','{5FC5DEC5-69CA-4993-81D5-E3EF23547A8A}')

set @number = '10477'
insert into @techpreq values (@number,'10100','{99931F15-1155-4f73-B277-689FB429518C}')

-- course 10503: technical prerequisites are described in free text specifying
-- required qulifications/skills which is not representable (only course numbers can
-- be specified)

set @number = '10504'
insert into @techpreq values (@number,'10111','{C79EDD74-C443-4add-8813-8A2DC1CE108D}')
insert into @techpreq values (@number,'10201','{C79EDD74-C443-4add-8813-8A2DC1CE108D}')
-- and
insert into @techpreq values (@number,'10203','{097DA04B-E5BB-4316-A27D-0FE9C253B32B}')
insert into @techpreq values (@number,'10300','{097DA04B-E5BB-4316-A27D-0FE9C253B32B}')

-- course 10911: technical prerequisites are described in free text specifying
-- required qulifications/skills which is not representable (only course numbers can
-- be specified)

-- course 10912: technical prerequisites are described in free text specifying
-- required qulifications/skills which is not representable (only course numbers can
-- be specified)

set @number = '10921'
insert into @techpreq values (@number,'10913','{A5DD56DE-88DC-468d-AAF7-7A8D7213B456}')

set @number = '10941'
insert into @techpreq values (@number,'01901','{6F676412-3839-41f5-BCA7-0B092DB6EB19}')
insert into @techpreq values (@number,'01902','{6F676412-3839-41f5-BCA7-0B092DB6EB19}')
insert into @techpreq values (@number,'01005','{6F676412-3839-41f5-BCA7-0B092DB6EB19}')

set @number = '11001'
insert into @techpreq values (@number,'11200','{12E96586-7579-4aa0-A06E-15B3C5AF50F5}')
-- and
insert into @techpreq values (@number,'11510','{8ADB9CCE-653C-48bb-A4D3-8EB471D3F05B}')

set @number = '11003'
insert into @techpreq values (@number,'11001','{FA436667-4124-48a3-A96A-C6022DF902FC}')
-- and
insert into @techpreq values (@number,'11511','{094FA654-5DB8-44b0-8E11-895710AE5B5F}')

set @number = '11005'
insert into @techpreq values (@number,'11003','{9D568F4D-26E4-4bad-9D82-542367C19B64}')
insert into @techpreq values (@number,'64004','{9D568F4D-26E4-4bad-9D82-542367C19B64}')

set @number = '11021'
insert into @techpreq values (@number,'11003','{AA8204A3-8CB8-416a-8ECD-6F078010D9A2}')
-- and
insert into @techpreq values (@number,'11521','{7FA3EE83-F17C-4f11-BB08-2D227A552BAA}')
-- and
insert into @techpreq values (@number,'11501','{2823DDCA-F075-4104-94F4-659AE8AE48D8}')

set @number = '11102'
insert into @techpreq values (@number,'11101','{7210C128-6E5B-4d7b-8B47-4EB34F2ACBB8}')
insert into @techpreq values (@number,'11732','{7210C128-6E5B-4d7b-8B47-4EB34F2ACBB8}')

set @number = '11103'
insert into @techpreq values (@number,'64044','{A2AAF0F5-A2FB-4930-BB1F-9777FC93A805}')
insert into @techpreq values (@number,'11102','{A2AAF0F5-A2FB-4930-BB1F-9777FC93A805}')

set @number = '11104'
insert into @techpreq values (@number,'64040','{4F57EA96-834B-4154-BBF7-C90FD5D279F1}')
insert into @techpreq values (@number,'11101','{4F57EA96-834B-4154-BBF7-C90FD5D279F1}')
```

```
set @number = '11105'
insert into @techpreq values (@number,'11102','{F0242EEE-7E7B-4fef-BCBE-C06D045BF063}')
-- and
insert into @techpreq values (@number,'02601','{70A39426-9DCC-4b53-B520-CC95C6DBE26D}')

set @number = '11106'
insert into @techpreq values (@number,'64040','{AC487404-9ACB-4835-96F7-C28345FE7DA8}')
insert into @techpreq values (@number,'11101','{AC487404-9ACB-4835-96F7-C28345FE7DA8}')

set @number = '11202'
insert into @techpreq values (@number,'67161','{A41B2911-3B77-414d-9CE0-F4D3BF060CF8}')
insert into @techpreq values (@number,'11200','{A41B2911-3B77-414d-9CE0-F4D3BF060CF8}')

set @number = '11204'
insert into @techpreq values (@number,'11200','{02C09353-C2F0-4db7-B8ED-AA61E8E29608}')
insert into @techpreq values (@number,'11201','{02C09353-C2F0-4db7-B8ED-AA61E8E29608}')
insert into @techpreq values (@number,'41601','{02C09353-C2F0-4db7-B8ED-AA61E8E29608}')

set @number = '11251'
insert into @techpreq values (@number,'11002','{70B87776-6563-447b-8248-0998C5471986}')

set @number = '11253'
insert into @techpreq values (@number,'11002 ','{B3B0EF98-B37F-4570-849C-26D63CF1552A}')

set @number = '11254'
insert into @techpreq values (@number,'11002','{6AAE4881-17E9-469d-BDFB-079B648F40D1}')

set @number = '11302'
insert into @techpreq values (@number,'11301','{6BF74F04-614E-4def-9A62-23D5D5CF079A}')
insert into @techpreq values (@number,'11712','{6BF74F04-614E-4def-9A62-23D5D5CF079A}')
insert into @techpreq values (@number,'67121','{6BF74F04-614E-4def-9A62-23D5D5CF079A}')
insert into @techpreq values (@number,'64070','{6BF74F04-614E-4def-9A62-23D5D5CF079A}')

set @number = '11303'
insert into @techpreq values (@number,'11301','{5B36860E-18CC-49ee-985F-BE76EAA79688}')
insert into @techpreq values (@number,'11712','{5B36860E-18CC-49ee-985F-BE76EAA79688}')
-- and
insert into @techpreq values (@number,'11302','{85EE69FD-C1AE-4eac-80EE-AFD4CBAA190A}')

set @number = '11304'
insert into @techpreq values (@number,'11302','{F10362F1-1F10-4241-A944-1A11155A2564}')
insert into @techpreq values (@number,'11303','{F10362F1-1F10-4241-A944-1A11155A2564}')

set @number = '11305'
insert into @techpreq values (@number,'11301','{E120E409-0F48-4f10-B790-B153D2A04853}')
insert into @techpreq values (@number,'11712','{E120E409-0F48-4f10-B790-B153D2A04853}')
-- and
insert into @techpreq values (@number,'11303','{5919D1EC-B8DD-4189-9979-5AAD3C53966D}')

set @number = '11306'
insert into @techpreq values (@number,'11301','{10E96F4E-1F29-4586-96A5-2B23C468C74D}')
insert into @techpreq values (@number,'11302','{10E96F4E-1F29-4586-96A5-2B23C468C74D}')
insert into @techpreq values (@number,'11303','{10E96F4E-1F29-4586-96A5-2B23C468C74D}')
insert into @techpreq values (@number,'11712','{10E96F4E-1F29-4586-96A5-2B23C468C74D}')

set @number = '11412'
insert into @techpreq values (@number,'11411','{9A85A1EE-2A64-4939-88A6-65E32B0357B0}')
insert into @techpreq values (@number,'56131','{9A85A1EE-2A64-4939-88A6-65E32B0357B0}')

set @number = '11413'
insert into @techpreq values (@number,'56131','{BAD32483-2326-4724-8058-FB9E6C5A1902}')
insert into @techpreq values (@number,'11411','{BAD32483-2326-4724-8058-FB9E6C5A1902}')
insert into @techpreq values (@number,'93603','{BAD32483-2326-4724-8058-FB9E6C5A1902}')
insert into @techpreq values (@number,'11743','{BAD32483-2326-4724-8058-FB9E6C5A1902}')

set @number = '11414'
insert into @techpreq values (@number,'56141','{728479E3-A842-4941-BC29-C525FC522833}')
insert into @techpreq values (@number,'11413','{728479E3-A842-4941-BC29-C525FC522833}')

-- course 11415: technical prerequisites are described in free text specifying
-- required qulifications/skills which is not representable (only course numbers can
-- be specified)

set @number = '11416'
insert into @techpreq values (@number,'11413','{C2E1810B-AFB2-4cf9-9C50-A17C72AEBDD7}')

set @number = '11417'
insert into @techpreq values (@number,'11413','{69B946DF-F836-4be2-B02A-E696A647BB1A}')
insert into @techpreq values (@number,'56141','{69B946DF-F836-4be2-B02A-E696A647BB1A}')
-- and
insert into @techpreq values (@number,'11423','{90418554-EFB8-4c67-97A4-AC8EC2B8A017}')
insert into @techpreq values (@number,'56204','{90418554-EFB8-4c67-97A4-AC8EC2B8A017}')

set @number = '11423'
insert into @techpreq values (@number,'11401','{753C375B-250E-4c36-BCD4-6AB073CE4A3D}')
insert into @techpreq values (@number,'12400','{753C375B-250E-4c36-BCD4-6AB073CE4A3D}')

set @number = '11424'
insert into @techpreq values (@number,'11422','{AFD8EF16-CF57-478b-96BA-D72A64911F63}')

set @number = '11501'
insert into @techpreq values (@number,'64000','{960D5D73-02B9-4e93-B5A5-3A60F13DE4D0}')
insert into @techpreq values (@number,'11001','{960D5D73-02B9-4e93-B5A5-3A60F13DE4D0}')

set @number = '11502'
insert into @techpreq values (@number,'59106','{105863A4-F9EB-4b07-A558-6068CC29FA6C}')
insert into @techpreq values (@number,'11501','{105863A4-F9EB-4b07-A558-6068CC29FA6C}')

set @number = '11503'
```

```
insert into @techpreq values (@number,'59106','{66AD9068-A992-44ce-9F39-9006EF159B5F}')
insert into @techpreq values (@number,'11501','{66AD9068-A992-44ce-9F39-9006EF159B5F}')

set @number = '11505'
insert into @techpreq values (@number,'10013','{E4398556-4A82-42f3-B738-AE5AD540D6F6}')
-- and
insert into @techpreq values (@number,'11511','{F85770C0-E66C-4320-8EF9-2D13B399929B}')

set @number = '11506'
insert into @techpreq values (@number,'11501','{A2FBAE7D-6245-468c-B2A7-26FDC6CD5C59}')
-- and
insert into @techpreq values (@number,'11511','{B0A48D67-E75D-490c-9F47-1A9F7CD7CD28}')

set @number = '11511'
insert into @techpreq values (@number,'01005','{7CC6F9BE-83A1-4143-976C-BC074BB94110}')
-- and
insert into @techpreq values (@number,'11510','{56465B85-1CFC-4272-9750-68401F1C9F25}')

set @number = '11512'
insert into @techpreq values (@number,'11511','{9C2E8A00-E891-4529-BEDF-204234D435CD}')
insert into @techpreq values (@number,'59107','{9C2E8A00-E891-4529-BEDF-204234D435CD}')

set @number = '11513'
insert into @techpreq values (@number,'11511','{FB163D81-5BE4-4488-B199-06306E69A5BB}')

-- course 11514: some parts of the technical prerequisites can not be represented.
set @number = '11514'
insert into @techpreq values (@number,'11513','{41E47E2A-D522-4f39-B1CE-E4FF28E61CF2}')

set @number = '11521'
insert into @techpreq values (@number,'59107','{BBA8ECC4-E775-4439-A842-0D46E3CDA672}')
insert into @techpreq values (@number,'11511','{BBA8ECC4-E775-4439-A842-0D46E3CDA672}')

set @number = '11522'
insert into @techpreq values (@number,'11521','{8BE7C8D0-EDC2-4cc6-94F4-CF0B1BD6D6C3}')
-- and
insert into @techpreq values (@number,'11512','{D05A3FC8-5050-4662-AE8B-DB5B763A8D90}')

set @number = '11523'
insert into @techpreq values (@number,'59331','{EA1C8F2D-21EF-4c80-AA89-5EF33CEBD738}')
-- and
insert into @techpreq values (@number,'11512','{B534A354-3A26-4e1c-9B70-6F52D6EA1697}')

set @number = '11531'
insert into @techpreq values (@number,'11511','{9350EAEA-61A6-412d-8F22-DD80CE9A055B}')

set @number = '11532'
insert into @techpreq values (@number,'11531','{712D61F9-B1D7-4c79-ABB2-BDD4B0D3E346}')

set @number = '11541'
insert into @techpreq values (@number,'11521','{1C9A3E9D-B72F-4ca3-A056-A1B39809ED19}')
-- and
insert into @techpreq values (@number,'11531','{97A9ED79-CE96-4af9-9A6E-F00E540212B9}')
-- and
insert into @techpreq values (@number,'11512','{FAD99934-0837-4314-89A3-D51C613E8810}')

set @number = '11542'
insert into @techpreq values (@number,'11512','{5C9FB429-3C36-4582-8261-C017945892D3}')

set @number = '11544'
insert into @techpreq values (@number,'11511','{4156EDD6-7890-419a-9ADB-CBBC0D9DDD81}')
insert into @techpreq values (@number,'59107','{6A9B5E3B-CD94-4932-AE01-B9A06EBE5C78}')

set @number = '11546'
insert into @techpreq values (@number,'11501','{323A2472-0153-4465-B31C-91AAB72C5393}')
-- and
insert into @techpreq values (@number,'11511','{10EC83CB-B86F-44e4-80C7-74FA61D19C46}')

set @number = '11601'
insert into @techpreq values (@number,'11521','{4E450FFA-D57F-464e-8C05-5D344C7190C7}')
-- and
insert into @techpreq values (@number,'11531','{18854EEC-7B5F-467a-B348-084075AF7E40}')
-- and
insert into @techpreq values (@number,'11544','{38A55B8C-2F92-49bb-B60D-EC818ADEAB36}')

set @number = '11721'
insert into @techpreq values (@number,'11711','{971B6145-0A67-441e-A348-D1D1787466FC}')
insert into @techpreq values (@number,'93101','{971B6145-0A67-441e-A348-D1D1787466FC}')
-- and
insert into @techpreq values (@number,'93111','{B4BDE924-02BD-4251-8D7B-EFAC946555C1}')
-- and
insert into @techpreq values (@number,'93152','{2187A1CA-66DA-4bbf-B550-202BF55B2B55}')
-- and
insert into @techpreq values (@number,'93251','{51393DC1-9A21-471d-8AAC-4C9C5276A284}')

set @number = '11721'
insert into @techpreq values (@number,'11711','{86978E02-5D8E-4e71-89C8-801599FE6163}')
insert into @techpreq values (@number,'93101','{86978E02-5D8E-4e71-89C8-801599FE6163}')
-- and
insert into @techpreq values (@number,'93111','{B826186E-E1F4-4a70-9A8D-B673E09BA577}')
-- and
insert into @techpreq values (@number,'93152','{C32FE29B-C08B-47f6-9F28-C90F1B90EA9B}')
-- and
insert into @techpreq values (@number,'93251','{59E90EA8-2150-415a-9CB8-567D128A511B}')

set @number = '11722'
insert into @techpreq values (@number,'11714','{221AB170-AF22-4334-A0B4-584F5835AFAA}')
insert into @techpreq values (@number,'11711','{221AB170-AF22-4334-A0B4-584F5835AFAA}')
-- and
```

```
insert into @techpreq values (@number,'01905','{AA690302-17CA-4cc2-91C2-9E0B0559A712}')
insert into @techpreq values (@number,'10913','{AA690302-17CA-4cc2-91C2-9E0B0559A712}')

set @number = '11724'
insert into @techpreq values (@number,'11714','{EEDF9EA9-AA6D-49ea-B02E-C51227B9E623}')
-- and
insert into @techpreq values (@number,'01905','{7D04C18E-DA27-4b77-8FFB-6CDA343BBE8E}')

set @number = '11731'
insert into @techpreq values (@number,'11721','{6CBF035D-8C66-43e0-B2E8-279102898B85}')

set @number = '11732'
insert into @techpreq values (@number,'12700','{E66D7920-629B-4f62-A9F8-822ABB08F712}')

set @number = '11734'
insert into @techpreq values (@number,'11724','{EC1781AE-B00F-4d34-A602-0528412510A4}')

set @number = '11743'
insert into @techpreq values (@number,'11734','{58069539-575E-49da-80E9-79168CE04D5D}')

set @number = '11744'
insert into @techpreq values (@number,'11731','{864618FB-AD55-4acc-8924-505747EFFF64}')

set @number = '11745'
insert into @techpreq values (@number,'11744','{8B898602-7BF0-4325-A150-76C12BCEDDAD}')

-- course 11751: technical prerequisites can not be handled.

-- course 11761: technical prerequisites does not make sense.

-- course 11762: technical prerequisites can not be handled.

set @number = '11784'
insert into @techpreq values (@number,'11712','{1CC8565D-92A5-4bfa-B321-87A0EA246A32}')
insert into @techpreq values (@number,'93121','{1CC8565D-92A5-4bfa-B321-87A0EA246A32}')
insert into @techpreq values (@number,'67121','{1CC8565D-92A5-4bfa-B321-87A0EA246A32}')
insert into @techpreq values (@number,'11301','{1CC8565D-92A5-4bfa-B321-87A0EA246A32}')

set @number = '11785'
insert into @techpreq values (@number,'11733','{113829A1-4F77-40e8-8E3D-79E4BEA55AF6}')
insert into @techpreq values (@number,'11705','{113829A1-4F77-40e8-8E3D-79E4BEA55AF6}')
insert into @techpreq values (@number,'93453','{113829A1-4F77-40e8-8E3D-79E4BEA55AF6}')

set @number = '11787'
insert into @techpreq values (@number,'11732','{D74CC3E0-0968-4613-8417-FEBC04F95C9C}')

set @number = '11804'
insert into @techpreq values (@number,'11802','{61CA18B4-A609-467b-8FC6-2D7D1A07367D}')

-- course 11805: technical prerequisites can not be handled.

-- course 11924: technical prerequisites can not be handled.

-- course 11932: technical prerequisites can not be handled.

-- course 11935: technical prerequisites can not be handled.

-- course 11961: technical prerequisites can not be handled.

set @number = '12121'
insert into @techpreq values (@number,'12100','{08368AE1-BAD0-45cb-BF5F-E59CAF1F6FD4}')
-- and
insert into @techpreq values (@number,'12200','{7404CFC5-A5DC-467b-82D9-EB003537315C}')
-- and
insert into @techpreq values (@number,'12300','{304AE357-480D-465a-9554-65BD03CF6E68}')

set @number = '12122'
insert into @techpreq values (@number,'12300','{69488C77-E512-47ca-A1C5-D6EC1D12AA36}')
insert into @techpreq values (@number,'12700','{69488C77-E512-47ca-A1C5-D6EC1D12AA36}')

set @number = '12130'
insert into @techpreq values (@number,'12100','{B551C3ED-0C47-4433-8CE5-EB1338662CBB}')
insert into @techpreq values (@number,'12101','{B551C3ED-0C47-4433-8CE5-EB1338662CBB}')

set @number = '12131'
insert into @techpreq values (@number,'12100','{6AE8CD60-B191-41c2-8F73-15A4950643A4}')

-- course 12140: technical prerequisites are described in free text specifying
-- required qulifications/skills which is not representable (only course numbers can
-- be specified)

set @number = '12201'
insert into @techpreq values (@number,'12100','{C1394CF3-E52C-4d7d-AF2A-86507BC4A7CA}')
-- and
insert into @techpreq values (@number,'12200','{58FA7AF2-7562-43d2-AA2B-41F115A89A90}')
-- and
insert into @techpreq values (@number,'12300','{0A0A5FC1-25F8-4740-858B-4F9924DA3F07}')

-- course 12210: technical prerequisites do not make sense.

-- course 12225: some parts of the technical prerequisites can not be handled.
set @number = '12225'
insert into @techpreq values (@number,'63130','{986E6169-D9B4-4746-8455-E2010B15E429}')
-- and
insert into @techpreq values (@number,'12100','{821537AE-0BC6-459f-97D2-CD281845209D}')

set @number = '12230'
insert into @techpreq values (@number,'12100','{802AB13D-BC89-481f-A3C5-64F429B4DCDA}')
insert into @techpreq values (@number,'12101','{802AB13D-BC89-481f-A3C5-64F429B4DCDA}')
```

```
set @number = '12231'
insert into @techpreq values (@number,'12102','{370A05C8-3130-4b5a-96EA-1C23D7E48A99}')
insert into @techpreq values (@number,'12103','{370A05C8-3130-4b5a-96EA-1C23D7E48A99}')

set @number = '12233'
insert into @techpreq values (@number,'12100','{7E388537-9E40-4f72-8A69-ACCB6E9EF706}')
insert into @techpreq values (@number,'12101','{7E388537-9E40-4f72-8A69-ACCB6E9EF706}')

set @number = '12242'
insert into @techpreq values (@number,'12210','{EA403530-A755-41fa-A917-EC850875285B}')
-- and
insert into @techpreq values (@number,'12121','{86A68FCA-ED9B-4b94-981B-F28254913C13}')
-- and
insert into @techpreq values (@number,'12130','{05512423-252F-41cc-B7D9-1D28D644F928}')
-- and
insert into @techpreq values (@number,'12131','{9C28091B-2226-4780-9FE3-B6C50B4EA62A}')

set @number = '12243'
insert into @techpreq values (@number,'12242','{F94CEE9C-E56C-4eb3-9CB6-A84D48904C0A}')
-- and
insert into @techpreq values (@number,'63424','{8A6FE9F0-264B-413f-AAD9-4378F2E624C8}')
insert into @techpreq values (@number,'63425','{8A6FE9F0-264B-413f-AAD9-4378F2E624C8}')
-- and
insert into @techpreq values (@number,'12121','{FA6EF9CD-206E-4966-8C90-EE4AF2A1B4E6}')
-- and
insert into @techpreq values (@number,'63233','{EF0EB32A-E776-44d1-AA8A-017ED681E9B6}')
-- and
insert into @techpreq values (@number,'12231','{1C68EA10-A5CE-4a9e-B11C-04B900EE7FFD}')
-- and
insert into @techpreq values (@number,'63431','{9A1596F2-0D43-4d5b-9E47-2D01CEDE1E4E}')

set @number = '12244'
insert into @techpreq values (@number,'12240','{AA071CA7-108B-431f-8BA9-10DEC3A22649}')

set @number = '12300'
insert into @techpreq values (@number,'01005','{BFBC01E0-95B8-4f47-A461-C9B36DC6B594}')

-- course 12320: technical prerequisites can not be handled.

set @number = '12322'
insert into @techpreq values (@number,'12320','{443EF44A-5A10-4624-A80A-98199D9C21CA}')

set @number = '12325'
insert into @techpreq values (@number,'12320','{D671EB8B-8ADA-4378-A466-F00BD24C2446}')
-- and
insert into @techpreq values (@number,'12321','{EAC1B749-D6C8-41cf-8530-222871395F64}')

set @number = '12330'
insert into @techpreq values (@number,'12100','{26821F12-0594-401e-959F-00C5A89CE1FE}')
-- and
insert into @techpreq values (@number,'12300','{81D10E9A-E79D-48aa-BD2A-B722DAC4BF56}')

set @number = '12331'
insert into @techpreq values (@number,'63370','{B0CD9B57-84DD-43ec-A927-F40B632ECA4F}')
-- and
insert into @techpreq values (@number,'12330','{4F5269C1-7BFA-4180-93C3-F14DD1E41BB2}')

set @number = '12332'
insert into @techpreq values (@number,'12100','{416E15D3-9FC1-41d2-B2EB-174FE71C2D10}')
-- and
insert into @techpreq values (@number,'12400','{7DBFEF6A-A363-44f6-AB68-48E0CEF7FD52}')

set @number = '12333'
insert into @techpreq values (@number,'12320','{4676532A-7384-4e18-B41F-F57AD95B8082}')

set @number = '12341'
insert into @techpreq values (@number,'12320','{2D95ACE0-656E-4f16-94BC-39FB2FDB0A3C}')

set @number = '12401'
insert into @techpreq values (@number,'12400','{74DEA6EA-62E1-4dce-B489-5E014A7D842A}')

set @number = '12440'
insert into @techpreq values (@number,'12411','{2BEA4D36-0F7B-4dc4-83BA-9EE2B7103E64}')

set @number = '12444'
insert into @techpreq values (@number,'12411','{47AC6527-56C1-46aa-8176-636A8D05E454}')

set @number = '13002'
insert into @techpreq values (@number,'13000','{54C046A7-6098-4393-9145-D65FF840724A}')
-- and
insert into @techpreq values (@number,'13110','{DA65FBF5-987B-45f0-AA95-0F1A8D84F3BC}')

set @number = '13110'
insert into @techpreq values (@number,'13000','{E092353D-7987-4206-BC7E-61DB003140E7}')

set @number = '13120'
insert into @techpreq values (@number,'13110','{103B67DE-0F0D-4604-94FA-C0795BAF6609}')
-- and
insert into @techpreq values (@number,'13003','{69B38612-45E2-45b0-B652-03B37AE685CC}')

set @number = '13121'
insert into @techpreq values (@number,'13110','{FA9A15CF-13D9-4c2c-BB7B-78E984C6FE62}')
-- and
insert into @techpreq values (@number,'02401','{90F1E797-98DC-4436-B841-81DECBF7728F}')
-- and
insert into @techpreq values (@number,'02402','{732AE8AF-5FD9-452b-A329-BE9B443BE71E}')

set @number = '13310'
```

```
insert into @techpreq values (@number,'13000','{78318626-96CF-414e-9E61-78A401E1FF6E}')

set @number = '13311'
insert into @techpreq values (@number,'13310','{DADCFD8F-3732-481f-A462-B894331DC25C}')

set @number = '13320'
insert into @techpreq values (@number,'13000','{DA0618F7-DDFB-4abe-9B5C-0B1F64325551}')

set @number = '13430'
insert into @techpreq values (@number,'02713','{F130F1CD-F286-44e8-A6FA-18806FDE2E06}')

-- course 26026: no technical prerequisites are specified.

set @number = '26122'
insert into @techpreq values (@number,'21000','{59983B47-9134-4a09-A3C2-4DB7CDFDD1FA}')
insert into @techpreq values (@number,'26000','{1EB7D4BC-428F-477c-B98A-01607D0A7C17}')

set @number = '26124'
insert into @techpreq values (@number,'21210','{069F7B62-4756-402d-9E74-B2003E7FD348}')
insert into @techpreq values (@number,'26001','{069F7B62-4756-402d-9E74-B2003E7FD348}')

set @number = '26126'
insert into @techpreq values (@number,'26124','{C2D535FD-D529-459b-931D-5A207208D0D8}')
insert into @techpreq values (@number,'21215','{C2D535FD-D529-459b-931D-5A207208D0D8}')
insert into @techpreq values (@number,'25111','{C2D535FD-D529-459b-931D-5A207208D0D8}')

set @number = '26128'
insert into @techpreq values (@number,'21210','{5EB7F6EB-3AE0-49c1-9D43-59619174216A}')
insert into @techpreq values (@number,'21262','{5EB7F6EB-3AE0-49c1-9D43-59619174216A}')
insert into @techpreq values (@number,'26220','{5EB7F6EB-3AE0-49c1-9D43-59619174216A}')

set @number = '26140'
insert into @techpreq values (@number,'98101','{9434FD5A-CABB-40d7-81F1-A70B731B54BD}')
insert into @techpreq values (@number,'98001','{9434FD5A-CABB-40d7-81F1-A70B731B54BD}')
insert into @techpreq values (@number,'98003','{9434FD5A-CABB-40d7-81F1-A70B731B54BD}')
insert into @techpreq values (@number,'98171','{9434FD5A-CABB-40d7-81F1-A70B731B54BD}')
insert into @techpreq values (@number,'98071','{9434FD5A-CABB-40d7-81F1-A70B731B54BD}')
insert into @techpreq values (@number,'98073','{9434FD5A-CABB-40d7-81F1-A70B731B54BD}')

set @number = '26142'
insert into @techpreq values (@number,'98101','{98491B44-6E3F-452c-BA6A-D7B57B5BD148}')
insert into @techpreq values (@number,'98001','{98491B44-6E3F-452c-BA6A-D7B57B5BD148}')
insert into @techpreq values (@number,'98003','{98491B44-6E3F-452c-BA6A-D7B57B5BD148}')
insert into @techpreq values (@number,'98171','{98491B44-6E3F-452c-BA6A-D7B57B5BD148}')
insert into @techpreq values (@number,'98071','{98491B44-6E3F-452c-BA6A-D7B57B5BD148}')
insert into @techpreq values (@number,'98073','{98491B44-6E3F-452c-BA6A-D7B57B5BD148}')

-- course 26170: no technical prerequisites are specified.

set @number = '26172'
insert into @techpreq values (@number,'91315','{874F212F-7DF4-49c5-B4D4-046D5388141E}')
insert into @techpreq values (@number,'26170','{874F212F-7DF4-49c5-B4D4-046D5388141E}')

set @number = '26201'
insert into @techpreq values (@number,'26000','{5EEEAD21-CC5F-495b-ADA1-425257627B1E}')
insert into @techpreq values (@number,'01000','{5EEEAD21-CC5F-495b-ADA1-425257627B1E}')

set @number = '26222'
insert into @techpreq values (@number,'26201','{81F7A58F-735D-4473-948B-B11F16A7E072}')
insert into @techpreq values (@number,'21061','{81F7A58F-735D-4473-948B-B11F16A7E072}')
insert into @techpreq values (@number,'21001','{81F7A58F-735D-4473-948B-B11F16A7E072}')
insert into @techpreq values (@number,'10005','{81F7A58F-735D-4473-948B-B11F16A7E072}')
insert into @techpreq values (@number,'10012','{81F7A58F-735D-4473-948B-B11F16A7E072}')

set @number = '26233'
insert into @techpreq values (@number,'26001','{5E874609-0B91-41b6-BE3B-DA429A2A01C3}')
insert into @techpreq values (@number,'26220','{5E874609-0B91-41b6-BE3B-DA429A2A01C3}')
insert into @techpreq values (@number,'10005','{5E874609-0B91-41b6-BE3B-DA429A2A01C3}')
insert into @techpreq values (@number,'10012','{5E874609-0B91-41b6-BE3B-DA429A2A01C3}')
insert into @techpreq values (@number,'01005','{5E874609-0B91-41b6-BE3B-DA429A2A01C3}')

set @number = '26235'
insert into @techpreq values (@number,'26230','{920A8BC2-C45D-4301-AB48-87D15BAD6BE4}')
insert into @techpreq values (@number,'21263','{920A8BC2-C45D-4301-AB48-87D15BAD6BE4}')
insert into @techpreq values (@number,'26225','{920A8BC2-C45D-4301-AB48-87D15BAD6BE4}')

set @number = '26240'
insert into @techpreq values (@number,'26220','{0B2DAECF-18FB-4147-B7A9-D5B2AAD9DFF9}')

set @number = '26250'
insert into @techpreq values (@number,'01030','{19EC89C4-417A-4e31-8FE9-25A7F6074248}')
insert into @techpreq values (@number,'01031','{19EC89C4-417A-4e31-8FE9-25A7F6074248}')
-- and
insert into @techpreq values (@number,'21262','{86A04720-87EE-47c0-B347-5C0A65BE54B4}')
insert into @techpreq values (@number,'26220','{86A04720-87EE-47c0-B347-5C0A65BE54B4}')
-- and
insert into @techpreq values (@number,'21390','{88B0B2BA-146C-4877-8DB0-C6C530F9D5F3}')
insert into @techpreq values (@number,'26500','{88B0B2BA-146C-4877-8DB0-C6C530F9D5F3}')

set @number = '26260'
insert into @techpreq values (@number,'10911','{5AD3ED9B-3300-43be-AC69-F31568E427C0}')
insert into @techpreq values (@number,'01901','{5AD3ED9B-3300-43be-AC69-F31568E427C0}')
insert into @techpreq values (@number,'01902','{5AD3ED9B-3300-43be-AC69-F31568E427C0}')

set @number = '26270'
insert into @techpreq values (@number,'91315','{A2BADBDB-977D-4085-A16C-89A86F774A57}')
insert into @techpreq values (@number,'26170','{A2BADBDB-977D-4085-A16C-89A86F774A57}')

set @number = '26272'
```

```
insert into @techpreq values (@number,'91315','{1239CEB3-ACCD-40e9-873C-F54141893DE0}')
insert into @techpreq values (@number,'26170','{1239CEB3-ACCD-40e9-873C-F54141893DE0}')
-- and
insert into @techpreq values (@number,'91329','{F470B13B-FC66-455d-9CF6-2CA6E4E01A13}')
insert into @techpreq values (@number,'26172','{F470B13B-FC66-455d-9CF6-2CA6E4E01A13}')
-- and
insert into @techpreq values (@number,'91325','{A1D24CEF-64F3-4ddd-B9C1-3FCFD4845819}')
insert into @techpreq values (@number,'26270','{A1D24CEF-64F3-4ddd-B9C1-3FCFD4845819}')

set @number = '26301'
insert into @techpreq values (@number,'26001','{F67961CD-AADB-439c-8C27-B7D417DDD43B}')

set @number = '26310'
insert into @techpreq values (@number,'2113','{B57016D9-DB05-4fc9-84FE-0AEC66B350BA}')
insert into @techpreq values (@number,'21210','{B57016D9-DB05-4fc9-84FE-0AEC66B350BA}')
insert into @techpreq values (@number,'21001','{B57016D9-DB05-4fc9-84FE-0AEC66B350BA}')
insert into @techpreq values (@number,'21002','{B57016D9-DB05-4fc9-84FE-0AEC66B350BA}')
insert into @techpreq values (@number,'26001','{B57016D9-DB05-4fc9-84FE-0AEC66B350BA}')
insert into @techpreq values (@number,'26002','{B57016D9-DB05-4fc9-84FE-0AEC66B350BA}')
-- and
insert into @techpreq values (@number,'2107','{0CEC9EF7-C4B0-4015-A9F7-3D885C545587}')
insert into @techpreq values (@number,'21245','{0CEC9EF7-C4B0-4015-A9F7-3D885C545587}')
insert into @techpreq values (@number,'26300','{0CEC9EF7-C4B0-4015-A9F7-3D885C545587}')
-- and
insert into @techpreq values (@number,'8859','{A20D4B39-A751-42e2-9034-6BC3A4371911}')
insert into @techpreq values (@number,'56059','{A20D4B39-A751-42e2-9034-6BC3A4371911}')
insert into @techpreq values (@number,'8860','{A20D4B39-A751-42e2-9034-6BC3A4371911}')
insert into @techpreq values (@number,'56060','{A20D4B39-A751-42e2-9034-6BC3A4371911}')

set @number = '26312'
insert into @techpreq values (@number,'2140','{A7DD5D29-B4D4-4c85-A2C7-AFE492265437}')
insert into @techpreq values (@number,'21250','{A7DD5D29-B4D4-4c85-A2C7-AFE492265437}')
insert into @techpreq values (@number,'26310','{A7DD5D29-B4D4-4c85-A2C7-AFE492265437}')

set @number = '26320'
insert into @techpreq values (@number,'21000','{D9075DB2-8274-4e67-B1D6-9D304171DAE4}')
insert into @techpreq values (@number,'26000','{D9075DB2-8274-4e67-B1D6-9D304171DAE4}')
insert into @techpreq values (@number,'21001','{D9075DB2-8274-4e67-B1D6-9D304171DAE4}')
insert into @techpreq values (@number,'26001','{D9075DB2-8274-4e67-B1D6-9D304171DAE4}')
insert into @techpreq values (@number,'21010','{D9075DB2-8274-4e67-B1D6-9D304171DAE4}')
insert into @techpreq values (@number,'26026','{D9075DB2-8274-4e67-B1D6-9D304171DAE4}')
insert into @techpreq values (@number,'10004','{D9075DB2-8274-4e67-B1D6-9D304171DAE4}')
insert into @techpreq values (@number,'10005','{D9075DB2-8274-4e67-B1D6-9D304171DAE4}')

set @number = '26322'
insert into @techpreq values (@number,'21000','{1A73F48A-6BEC-4392-96C3-C0F05DB9DC99}')
insert into @techpreq values (@number,'26000','{1A73F48A-6BEC-4392-96C3-C0F05DB9DC99}')
insert into @techpreq values (@number,'21001','{1A73F48A-6BEC-4392-96C3-C0F05DB9DC99}')
insert into @techpreq values (@number,'26001','{1A73F48A-6BEC-4392-96C3-C0F05DB9DC99}')
insert into @techpreq values (@number,'21010','{1A73F48A-6BEC-4392-96C3-C0F05DB9DC99}')
insert into @techpreq values (@number,'26026','{1A73F48A-6BEC-4392-96C3-C0F05DB9DC99}')
insert into @techpreq values (@number,'10004','{1A73F48A-6BEC-4392-96C3-C0F05DB9DC99}')
insert into @techpreq values (@number,'10005','{1A73F48A-6BEC-4392-96C3-C0F05DB9DC99}')

set @number = '26334'
insert into @techpreq values (@number,'26410','{14ED1E9C-FCF5-46ee-804B-E1AE88D569BD}')
insert into @techpreq values (@number,'23220','{14ED1E9C-FCF5-46ee-804B-E1AE88D569BD}')

set @number = '26370'
insert into @techpreq values (@number,'26170','{FB6D6F93-9888-4701-B46C-C6190281D38E}')

set @number = '26372'
insert into @techpreq values (@number,'91311','{E303C4B5-D623-4956-BA76-EC5A16C88BBA}')
insert into @techpreq values (@number,'91315','{E303C4B5-D623-4956-BA76-EC5A16C88BBA}')
insert into @techpreq values (@number,'26170','{E303C4B5-D623-4956-BA76-EC5A16C88BBA}')

-- course 26376: no technical prerequisites are specified.

set @number = '26378'
insert into @techpreq values (@number,'26170','{02E7F9A5-B32A-45fb-AE45-427CDCA7C3E7}')
insert into @techpreq values (@number,'91315','{02E7F9A5-B32A-45fb-AE45-427CDCA7C3E7}')

set @number = '26400'
insert into @techpreq values (@number,'26001','{93D81FDC-4C72-4773-A650-850E89EE417C}')
insert into @techpreq values (@number,'21001','{93D81FDC-4C72-4773-A650-850E89EE417C}')
insert into @techpreq values (@number,'21210','{93D81FDC-4C72-4773-A650-850E89EE417C}')
-- and
insert into @techpreq values (@number,'21211','{065EB14A-01A9-404f-9350-3F64962349C0}')
insert into @techpreq values (@number,'21000','{065EB14A-01A9-404f-9350-3F64962349C0}')

set @number = '26401'
insert into @techpreq values (@number,'26001','{1BC5490D-7C8E-4112-B90D-62FD3A8F29FF}')
insert into @techpreq values (@number,'21001','{1BC5490D-7C8E-4112-B90D-62FD3A8F29FF}')
insert into @techpreq values (@number,'21210','{1BC5490D-7C8E-4112-B90D-62FD3A8F29FF}')
insert into @techpreq values (@number,'21010','{1BC5490D-7C8E-4112-B90D-62FD3A8F29FF}')
insert into @techpreq values (@number,'10005','{1BC5490D-7C8E-4112-B90D-62FD3A8F29FF}')

set @number = '26406'
insert into @techpreq values (@number,'26300','{96CD50EB-003F-4aa7-A6DD-53E2E905CACC}')
insert into @techpreq values (@number,'21245','{96CD50EB-003F-4aa7-A6DD-53E2E905CACC}')
insert into @techpreq values (@number,'56069','{96CD50EB-003F-4aa7-A6DD-53E2E905CACC}')

set @number = '26410'
insert into @techpreq values (@number,'26001','{45572B17-88E0-40b1-A4ED-6664E8F2644B}')
insert into @techpreq values (@number,'26400','{45572B17-88E0-40b1-A4ED-6664E8F2644B}')
insert into @techpreq values (@number,'26401','{45572B17-88E0-40b1-A4ED-6664E8F2644B}')
insert into @techpreq values (@number,'23110','{45572B17-88E0-40b1-A4ED-6664E8F2644B}')
insert into @techpreq values (@number,'2310','{45572B17-88E0-40b1-A4ED-6664E8F2644B}')
```

```
set @number = '26416'
insert into @techpreq values (@number,'26410','{A11795CE-A438-469d-827A-50520524F399}')
insert into @techpreq values (@number,'23220','{A11795CE-A438-469d-827A-50520524F399}')

set @number = '26418'
insert into @techpreq values (@number,'26410','{2068FFEB-73BD-40f2-87A4-FF79CF0E5634}')
insert into @techpreq values (@number,'23220','{2068FFEB-73BD-40f2-87A4-FF79CF0E5634}')

set @number = '26430'
insert into @techpreq values (@number,'26410','{9A882B38-53F4-47d4-8691-87CDE926E24A}')

set @number = '26470'
insert into @techpreq values (@number,'26370','{EE67F15D-9AA2-4760-9479-E247A45FDAF6}')
-- and
insert into @techpreq values (@number,'26372','{54327009-8A65-4f4c-A4BA-885E8C73B191}')

set @number = '26472'
insert into @techpreq values (@number,'26370','{A4ED1087-11A5-4209-B889-FD7EA7EE5E6D}')

set @number = '26474'
insert into @techpreq values (@number,'91332','{A34EBB52-9478-4da6-ABFA-B1A28CA3D379}')
insert into @techpreq values (@number,'91328','{A34EBB52-9478-4da6-ABFA-B1A28CA3D379}')
insert into @techpreq values (@number,'26470','{A34EBB52-9478-4da6-ABFA-B1A28CA3D379}')

-- course 26478: no technical prerequisites are specified.

-- course 26510: technical prerequisites can not be handled.

-- course 26900: technical prerequisites can not be handled.

set @number = '26926'
insert into @techpreq values (@number,'26430','{ECE33D8F-6D7D-4682-B012-356712BF6032}')

set @number = '26930'
insert into @techpreq values (@number,'26430 ','{81E715A7-B08F-407e-9346-638A82D1C38C}')

-- course 26936: technical prerequisites can not be handled.

set @number = '27000'
insert into @techpreq values (@number,'26001','{224288DD-EABB-44b9-B461-F8931C064574}')
-- and
insert into @techpreq values (@number,'01005','{09F029AA-EE0A-44e5-BACC-F2252933DD7B}')

set @number = '27001'
insert into @techpreq values (@number,'26001','{5A34E4D8-616A-4fca-B5C3-6E18105A5D8A}')
-- and
insert into @techpreq values (@number,'01005','{04D32156-E500-44c7-A6FE-D4365A05530A}')

set @number = '27011'
insert into @techpreq values (@number,'27000','{9A7D0ADB-6892-470c-81DE-AB58738BBF4E}')
insert into @techpreq values (@number,'27001','{CC185485-0344-4ecd-9B5E-151F924F50B1}')
insert into @techpreq values (@number,'27931','{0B7C1E87-6846-4087-A607-A38575F766FE}')

set @number = '27012'
insert into @techpreq values (@number,'27000','{FB41B9C0-FDFD-4b42-8DB8-4A7C6C7304F1}')
insert into @techpreq values (@number,'27001','{FB41B9C0-FDFD-4b42-8DB8-4A7C6C7304F1}')
insert into @techpreq values (@number,'27931','{FB41B9C0-FDFD-4b42-8DB8-4A7C6C7304F1}')
insert into @techpreq values (@number,'01005','{FB41B9C0-FDFD-4b42-8DB8-4A7C6C7304F1}')

set @number = '27014'
insert into @techpreq values (@number,'27231','{F271838C-B2F2-4544-BF3E-F0C6584B3FA8}')

set @number = '27021'
insert into @techpreq values (@number,'27931','{312AEA0D-F162-40d3-B680-6C0A2237F704}')

set @number = '27031'
insert into @techpreq values (@number,'27000','{780716F2-54BE-4a4b-876B-AE58D4E1C61D}')
insert into @techpreq values (@number,'27001','{780716F2-54BE-4a4b-876B-AE58D4E1C61D}')
-- and
insert into @techpreq values (@number,'28120','{603B5E7E-A824-43d2-95E0-EDA7460CB2A5}')
-- and
insert into @techpreq values (@number,'01005','{B3576069-EEDA-4cf0-B4FE-F7C6AD0D2E43}') -- Mega MAT
insert into @techpreq values (@number,'01006','{B3576069-EEDA-4cf0-B4FE-F7C6AD0D2E43}') -- Mega MAT (repeaters' version)

set @number = '27033'
insert into @techpreq values (@number,'27961','{B7787F58-4B82-43b0-B959-E50A208370D7}')

set @number = '27231'
insert into @techpreq values (@number,'27021','{FC567428-25F7-4ec4-A621-4F9CF6F8E075}')

set @number = '27232'
insert into @techpreq values (@number,'27021','{5889E899-99BF-4ab7-8E8A-581624B12FCF}')
-- and
insert into @techpreq values (@number,'27231','{8C1C54A9-0F24-4c0c-955A-CE88354D3D18}')

set @number = '27233'
insert into @techpreq values (@number,'27021','{539FC5AE-E0AA-47a3-82AE-98FE6E21686B}')

set @number = '27253'
insert into @techpreq values (@number,'27021','{05F25534-89FD-4552-9E0D-C497412F07C5}')

set @number = '27254'
insert into @techpreq values (@number,'27253','{B96CA9AE-D3BC-4050-8171-7D01341BB772}')
-- and
insert into @techpreq values (@number,'27221','{5FD9E7D5-73D8-40ec-870A-F00B370D7B5F}')
-- and
insert into @techpreq values (@number,'27251','{25C353B8-845C-4925-B9B8-C7EFC783399E}')

set @number = '27258'
```

```
insert into @techpreq values (@number,'27021','{DB0FDB88-9E6A-491d-932F-238D18B02A0D}')
insert into @techpreq values (@number,'27031','{DB0FDB88-9E6A-491d-932F-238D18B02A0D}')

set @number = '27262'
insert into @techpreq values (@number,'24001','{CE56252D-38E7-49a2-B85B-4491B4DFAFA6}')
insert into @techpreq values (@number,'27000','{CE56252D-38E7-49a2-B85B-4491B4DFAFA6}')
insert into @techpreq values (@number,'27001','{CE56252D-38E7-49a2-B85B-4491B4DFAFA6}')
insert into @techpreq values (@number,'30000','{CE56252D-38E7-49a2-B85B-4491B4DFAFA6}')
insert into @techpreq values (@number,'30001','{CE56252D-38E7-49a2-B85B-4491B4DFAFA6}')
insert into @techpreq values (@number,'27931','{CE56252D-38E7-49a2-B85B-4491B4DFAFA6}')

set @number = '27301'
insert into @techpreq values (@number,'25111','{FEF49F93-81FC-46b7-AE5D-874D599E8688}')
insert into @techpreq values (@number,'27021','{FEF49F93-81FC-46b7-AE5D-874D599E8688}')
insert into @techpreq values (@number,'27731','{FEF49F93-81FC-46b7-AE5D-874D599E8688}')

set @number = '27302'
insert into @techpreq values (@number,'27301','{07A9D386-089F-4bc3-BB14-D0A0CBE67CA1}')

set @number = '27311'
insert into @techpreq values (@number,'24001','{68319C01-6056-49d2-AC9C-691526A80B17}')
insert into @techpreq values (@number,'27001','{68319C01-6056-49d2-AC9C-691526A80B17}')
insert into @techpreq values (@number,'27000','{68319C01-6056-49d2-AC9C-691526A80B17}')
insert into @techpreq values (@number,'27731','{68319C01-6056-49d2-AC9C-691526A80B17}')
insert into @techpreq values (@number,'30000','{68319C01-6056-49d2-AC9C-691526A80B17}')
insert into @techpreq values (@number,'30001','{68319C01-6056-49d2-AC9C-691526A80B17}')

set @number = '27321'
insert into @techpreq values (@number,'27021','{23069C33-EB2F-4412-8523-42BF65FDE95F}')
-- and
insert into @techpreq values (@number,'27311','{F0B38CB9-D999-4bcf-9E92-6278F5F70BE8}')

set @number = '27403'
insert into @techpreq values (@number,'26410','{904B750D-CFEA-4efc-962B-0B8860AD8827}')
-- and
insert into @techpreq values (@number,'27021','{DB02CC5C-D215-43e3-9847-512BB0CAF552}')
-- and
insert into @techpreq values (@number,'27031','{9C0901A3-F116-4c20-BB28-B4C336ED8A3D}')
-- and
insert into @techpreq values (@number,'27311','{50188576-098B-47f5-81D4-7D82A1B09182}')

-- course 27404: technical prerequisites can not be handled.

set @number = '27405'
insert into @techpreq values (@number,'27031','{1EF8BD28-9097-4a37-9AAB-60033D4341D8}')
-- and
insert into @techpreq values (@number,'28120','{140CA8DC-4EBE-4943-8661-74604DEB7BFB}')

-- course 27406: technical prerequisites can not be handled.

set @number = '27407'
insert into @techpreq values (@number,'27031','{2FA7E9A4-B78B-416b-83E9-5FF8577028A5}')

-- course 27442: some parts of the technical prerequisites specified can not be handled.
set @number = '27442'
insert into @techpreq values (@number,'27031','{92E0C802-E8BB-497e-AF10-82E1375EDC73}')
insert into @techpreq values (@number,'27961','{92E0C802-E8BB-497e-AF10-82E1375EDC73}')
insert into @techpreq values (@number,'27403','{92E0C802-E8BB-497e-AF10-82E1375EDC73}')

-- course 27443: technical prerequisites can not be handled.

set @number = '27444'
insert into @techpreq values (@number,'27011','{56649003-B4C9-4bb0-8036-9E33C430ECCB}')
-- and
insert into @techpreq values (@number,'27000','{5BBA59AE-94BC-4095-BC93-787948293039}')

set @number = '27485'
insert into @techpreq values (@number,'27011','{CED7A791-09A8-4818-A232-ABE4367DB283}')
-- and
insert into @techpreq values (@number,'27321','{0107ACD3-4698-4b5b-9C21-A7DFA8421904}')

set @number = '27500'
insert into @techpreq values (@number,'25111','{25B8A5A8-05E9-46b8-8A5D-F1D0350C009C}')
insert into @techpreq values (@number,'25771','{25B8A5A8-05E9-46b8-8A5D-F1D0350C009C}')
insert into @techpreq values (@number,'27731','{25B8A5A8-05E9-46b8-8A5D-F1D0350C009C}')
insert into @techpreq values (@number,'24101','{25B8A5A8-05E9-46b8-8A5D-F1D0350C009C}')
insert into @techpreq values (@number,'27021','{25B8A5A8-05E9-46b8-8A5D-F1D0350C009C}')

set @number = '27501'
insert into @techpreq values (@number,'27721','{2A1BC075-F204-4127-BD3C-99E8FC8B996A}')
-- and
insert into @techpreq values (@number,'27731','{1F3D3380-C120-44a4-BE4F-D24069CC5A86}')

set @number = '27502'
insert into @techpreq values (@number,'27721','{7AD8C57E-A98B-4a08-A5E9-2095C9593E28}')
-- and
insert into @techpreq values (@number,'27731','{F6280A84-9E14-420e-8FA6-E23EB0BFF605}')

-- course 27505: technical prerequisites can not be handled.

set @number = '27520'
insert into @techpreq values (@number,'27500','{B5B598A0-16DA-4af4-918E-348F0C1132FF}')
-- and
insert into @techpreq values (@number,'30221','{2769E4FC-939D-4975-B71E-6D3F60AB8D0D}')
insert into @techpreq values (@number,'27711','{2769E4FC-939D-4975-B71E-6D3F60AB8D0D}')
-- and
insert into @techpreq values (@number,'30773','{6837DDAE-27BE-4749-B3DA-4DD0AB874707}')
insert into @techpreq values (@number,'36100','{6837DDAE-27BE-4749-B3DA-4DD0AB874707}')
-- and
```

```
insert into @techpreq values (@number,'91533','{50416A4A-B7B4-4914-AD09-F73A6CA2874C}')
-- and
insert into @techpreq values (@number,'27710','{012A3709-476A-4d46-836E-47682774BA2A}')

set @number = '27584'
insert into @techpreq values (@number,'27000','{6978EEB4-3179-4153-8CC2-8970A35995E2}')
insert into @techpreq values (@number,'27001','{6978EEB4-3179-4153-8CC2-8970A35995E2}')
insert into @techpreq values (@number,'30000','{6978EEB4-3179-4153-8CC2-8970A35995E2}')
insert into @techpreq values (@number,'30001','{6978EEB4-3179-4153-8CC2-8970A35995E2}')

-- course 27700: technical prerequisites can not be handled.

-- course 27706: technical prerequisites can not be handled.

set @number = '27710'
insert into @techpreq values (@number,'27700','{FCC58043-0E6D-465c-B8CE-FEDB02A70212}')
insert into @techpreq values (@number,'91924','{FCC58043-0E6D-465c-B8CE-FEDB02A70212}')

set @number = '27711'
insert into @techpreq values (@number,'27710','{C7EA5369-9750-4f93-908E-121506ED23D5}')
insert into @techpreq values (@number,'91533','{C7EA5369-9750-4f93-908E-121506ED23D5}')
-- and
insert into @techpreq values (@number,'072511','{044C161F-A0D4-452a-87C5-B7865FC0C429}')

set @number = '27722'
insert into @techpreq values (@number,'24101','{93537197-4F2F-4734-BD13-217A4AD9BB28}')
insert into @techpreq values (@number,'24773','{93537197-4F2F-4734-BD13-217A4AD9BB28}')
insert into @techpreq values (@number,'27721','{93537197-4F2F-4734-BD13-217A4AD9BB28}')
insert into @techpreq values (@number,'27931','{93537197-4F2F-4734-BD13-217A4AD9BB28}')

set @number = '27731'
insert into @techpreq values (@number,'91522','{C7387ADA-D848-4091-AEBD-D0F63A34D252}')
-- and
insert into @techpreq values (@number,'91313','{052A9494-C174-489a-88A7-CF554EE3412B}')
-- and
insert into @techpreq values (@number,'91923','{1083C9AB-726A-4817-A20E-5A0C66EEF102}')

set @number = '27732'
insert into @techpreq values (@number,'27731','{53E52EFE-043B-4308-84FE-507385EE32FB}')
-- and
insert into @techpreq values (@number,'27021','{5FD59268-3D84-4fdf-9437-015BD73352BA}')

-- course 27751: technical prerequisites can not be handled.

-- course 27752: some parts of the technical prerequisites specified can not be handled.
set @number = '27752'
insert into @techpreq values (@number,'27021','{AE76FE31-FCCB-475c-90B2-3DE8E8157AD5}')

-- course 27763: some parts of the technical prerequisites specified can not be handled.
set @number = '27763'
insert into @techpreq values (@number,'27717','{B8A6C72F-5361-47a4-A2BB-1A4CACEF5D7C}')
insert into @techpreq values (@number,'22751','{B8A6C72F-5361-47a4-A2BB-1A4CACEF5D7C}')
insert into @techpreq values (@number,'078020','{B8A6C72F-5361-47a4-A2BB-1A4CACEF5D7C}')

set @number = '27931'
insert into @techpreq values (@number,'26172','{C60E861E-05AA-4d4b-AECE-2EE3D09A833E}')
-- and
insert into @techpreq values (@number,'26470','{8DC8338D-9457-4a0a-ADDA-2604B7CDEE21}')

set @number = '27942'
insert into @techpreq values (@number,'26470','{CE6539B8-D991-4d2b-9AFC-F60B937FFB1C}')

set @number = '27961'
insert into @techpreq values (@number,'28021','{C100F0DE-9801-499f-833F-4482E0C6D484}')
-- and
insert into @techpreq values (@number,'28351','{16BAE575-8532-4873-90EC-475568F663F2}')

-- course 28010: technical prerequisites does not make sense.

set @number = '28021'
insert into @techpreq values (@number,'28011','{5351E561-E8D7-48f0-9FE1-D321B13B504F}')
-- and
insert into @techpreq values (@number,'01901','{4566FF9D-8A71-4b49-869E-2BA1B1F6B543}')
-- and
insert into @techpreq values (@number,'10911','{12B5A58C-8D9B-408a-A5DE-6858C2AC5BBA}')
-- and
insert into @techpreq values (@number,'26270','{2A58191C-9769-48d6-873A-93C8C4AB6F9B}')

set @number = '28120'
insert into @techpreq values (@number,'10010','{9B482D11-CBEA-4c97-B18C-F9EFF2DE6071}')
insert into @techpreq values (@number,'10011','{9B482D11-CBEA-4c97-B18C-F9EFF2DE6071}')

set @number = '28153'
insert into @techpreq values (@number,'28021','{A6BC5877-E975-4e67-8766-60364D526CC9}')

set @number = '28181'
insert into @techpreq values (@number,'28171','{EC4BA34B-7205-4caa-9550-AC9D00154B6F}')

set @number = '28210'
insert into @techpreq values (@number,'26001','{B8443BFE-3E2C-455d-8571-281BC0117ACA}')

set @number = '28212'
insert into @techpreq values (@number,'26410','{4F1A48A1-80A3-4fa3-97FA-640005C09F89}')

set @number = '28213'
insert into @techpreq values (@number,'28220','{7C5C04AC-6A08-4ecb-BB82-2E6A133EA3F6}')
insert into @techpreq values (@number,'01031','{7C5C04AC-6A08-4ecb-BB82-2E6A133EA3F6}')
insert into @techpreq values (@number,'26400','{7C5C04AC-6A08-4ecb-BB82-2E6A133EA3F6}')
```

```
set @number = '28221'
insert into @techpreq values (@number,'26222','{D62A248D-E2AA-4699-914D-0AD576F50CB3}')

set @number = '28240'
insert into @techpreq values (@number,'C3600','{545EA84A-D66D-4550-8470-5A9C729B77D0}')
insert into @techpreq values (@number,'36100','{545EA84A-D66D-4550-8470-5A9C729B77D0}')
-- and
insert into @techpreq values (@number,'36261','{62BB94AA-327E-4256-BE5F-F9C76B3F7F8F}')
insert into @techpreq values (@number,'36260','{62BB94AA-327E-4256-BE5F-F9C76B3F7F8F}')
-- and
insert into @techpreq values (@number,'28120','{10E5E8EB-1138-4734-BE00-A971CE2152FC}')
-- and
insert into @techpreq values (@number,'28260','{A1F75240-6D9C-452d-9D38-5737253F46B2}')
-- and
insert into @techpreq values (@number,'28863','{2B2FE1F8-E3FE-4bc6-89A5-DD556C5E5021}')

set @number = '28241'
insert into @techpreq values (@number,'28220','{CCAD953C-4DFC-4791-9F7D-70D11D28F0E9}')
insert into @techpreq values (@number,'10004','{CCAD953C-4DFC-4791-9F7D-70D11D28F0E9}')
insert into @techpreq values (@number,'10005','{CCAD953C-4DFC-4791-9F7D-70D11D28F0E9}')

set @number = '28244'
insert into @techpreq values (@number,'28120','{ED042175-C896-42e9-92E3-1D6480F83146}')
-- and
insert into @techpreq values (@number,'41401','{60CBCAE1-49C0-4ecc-94AD-FF268CC51123}')

set @number = '28250'
insert into @techpreq values (@number,'28120','{3B0A05A0-C262-41f6-8EA6-8A56042D4343}')
insert into @techpreq values (@number,'28260','{3B0A05A0-C262-41f6-8EA6-8A56042D4343}')
insert into @techpreq values (@number,'36100','{3B0A05A0-C262-41f6-8EA6-8A56042D4343}')
insert into @techpreq values (@number,'36260','{3B0A05A0-C262-41f6-8EA6-8A56042D4343}')

set @number = '28252'
insert into @techpreq values (@number,'28250','{2A85A42C-C5B0-4bd2-9B4B-236361F03282}')

set @number = '28260'
insert into @techpreq values (@number,'01005','{FBA196C4-9F7C-464b-8B5F-B2BE69ADF74D}')
insert into @techpreq values (@number,'01006','{FBA196C4-9F7C-464b-8B5F-B2BE69ADF74D}')
-- and
insert into @techpreq values (@number,'28010','{277EB903-4268-45ab-8C34-AA5AA746074C}') -- Products and Processes
insert into @techpreq values (@number,'28011','{277EB903-4268-45ab-8C34-AA5AA746074C}') -- Introduction to Chemical Process Engineering

set @number = '28316'
insert into @techpreq values (@number,'28315','{51A358DA-37F2-46fe-9BFA-F30A949BA68A}')

set @number = '28320'
insert into @techpreq values (@number,'28120','{EA0AB911-ACB2-49fa-9035-2574AF590C8B}')
insert into @techpreq values (@number,'28863','{EA0AB911-ACB2-49fa-9035-2574AF590C8B}')

set @number = '28321'
insert into @techpreq values (@number,'26272','{6A5686C9-42F6-47b3-BC8C-E992E348A5BE}')

set @number = '28325'
insert into @techpreq values (@number,'36250','{7D80B0DF-EF27-4f3d-88B1-D58090C7381A}')
insert into @techpreq values (@number,'28220','{D406C4F9-5B48-4a07-BB29-39C977684906}')

set @number = '28341'
insert into @techpreq values (@number,'01901','{B30AF34B-F427-4af1-BC3D-206B1A4889C7}')
-- and
insert into @techpreq values (@number,'26272','{C345BEC6-801B-4076-9B63-1A11ADCF1D39}')
-- and
insert into @techpreq values (@number,'28021','{9096DAC0-ABED-47ae-8F71-303D9C2DD330}')

set @number = '28350'
insert into @techpreq values (@number,'28010','{5ACE85A1-4303-4faa-B834-479FA9AB7DAE}')
-- and
insert into @techpreq values (@number,'28120','{C7CE50F2-C816-4bcf-B3C9-8AC47E123E8C}')

set @number = '28351'
insert into @techpreq values (@number,'01902','{EFF3AFF6-2173-45f9-962F-239D85C71400}')
-- and
insert into @techpreq values (@number,'28021','{22947F4F-FBAE-439a-AD58-6550B298A722}')

set @number = '28414'
insert into @techpreq values (@number,'28230','{267F3948-F2BD-4c2f-A0A6-64E47E70AC06}')

-- course 28415: technical prerequisites can not be handled.

set @number = '28416'
insert into @techpreq values (@number,'28120','{D011FA76-2F2A-4a1e-A543-BB0F80A234A2}')

set @number = '28423'
insert into @techpreq values (@number,'28220','{F35BA30A-5E2A-4cf9-8254-FC3501E1B9AF}')

set @number = '28434'
insert into @techpreq values (@number,'28120','{1B1767B3-3738-4c88-AC59-1D5BAF0DF726}')

set @number = '28443'
insert into @techpreq values (@number,'28240','{890B6BBE-824F-4307-8F80-4A4F6FBD2325}')
-- and
insert into @techpreq values (@number,'28341','{B18BB1BD-D63E-4534-B1FC-5A4BC995BF40}')

set @number = '28463'
insert into @techpreq values (@number,'28260','{2EBF5DE3-A14B-4df0-B109-D616297CE64B}')
-- and
insert into @techpreq values (@number,'28860','{A9CC73DD-DFA1-46df-8A9D-29EF2AA5B189}')
-- and
insert into @techpreq values (@number,'28863','{993B1440-252F-4658-816C-A6FE07C369DF}')
```

```
-- course 28515: technical prerequisites can not be handled.

-- course 28811: technical prerequisites can not be handled.

-- course 28835: technical prerequisites can not be handled.

set @number = '28845'
insert into @techpreq values (@number,'28341','{9B926804-0BD2-48d3-9E52-F90E24691B63}')

set @number = '28851'
insert into @techpreq values (@number,'28250','{870C3546-8202-4146-BAA0-8FF02C8A256E}')
-- and
insert into @techpreq values (@number,'28351','{903E9CE5-FE00-4390-AD9B-7EE5BFBA4AA4}')

set @number = '28852'
insert into @techpreq values (@number,'28120','{84BCBCA1-7AA5-4213-A6D2-201FBDA1AB5A}')

set @number = '28861'
insert into @techpreq values (@number,'01030','{54F1E69A-B262-4dad-8F58-1343A164345A}')
insert into @techpreq values (@number,'01902','{54F1E69A-B262-4dad-8F58-1343A164345A}')
insert into @techpreq values (@number,'01912','{54F1E69A-B262-4dad-8F58-1343A164345A}')

set @number = '28863'
insert into @techpreq values (@number,'26500','{DD48FE5D-36CD-45cc-909F-646A23049AF0}')

-- course 28885: technical prerequisites can not be handled.

set @number = '31011'
insert into @techpreq values (@number,'31000','{A3B537C0-18D0-4561-918C-9D88B865D3C9}')

set @number = '31022'
insert into @techpreq values (@number,'01962','{07ADAB06-84FA-47a3-B29A-6457C1F4BE28}')
-- and
insert into @techpreq values (@number,'31020','{9E12F17D-A6F4-4ac7-A398-B37BCBFF63FB}')

set @number = '31023'
insert into @techpreq values (@number,'31022','{B537FF4C-D5A7-44f2-ABBD-397AA87B3A3C}')
-- and
insert into @techpreq values (@number,'01962','{C2642AC5-FA2B-42e0-92D0-A127C34EF255}')

set @number = '31024'
insert into @techpreq values (@number,'31021','{22188C51-ADAE-4d02-80AF-8D2113262F5A}')
-- and
insert into @techpreq values (@number,'32022','{AEE8FD3A-B4BF-478f-8BD4-57C543FCD913}')

set @number = '31025'
insert into @techpreq values (@number,'01961','{9A3E155B-5468-4e05-9B8E-9EAE9AB943E8}')
-- and
insert into @techpreq values (@number,'31021','{1CA8A871-F94F-4e00-8C5E-4C445B43E9FE}')

set @number = '31027'
insert into @techpreq values (@number,'31020','{C238E1AD-962B-442c-A62A-1330D8F3182E}')
insert into @techpreq values (@number,'02311','{C238E1AD-962B-442c-A62A-1330D8F3182E}')
insert into @techpreq values (@number,'31026','{C238E1AD-962B-442c-A62A-1330D8F3182E}')
insert into @techpreq values (@number,'02310','{C238E1AD-962B-442c-A62A-1330D8F3182E}')

set @number = '31028'
insert into @techpreq values (@number,'31027','{B3A28697-E3AC-4c13-BBB4-1C537C0BCD48}')

set @number = '31029'
insert into @techpreq values (@number,'01962','{F8556E65-36D0-4437-9456-97B2064BD6FB}')
-- and
insert into @techpreq values (@number,'31022','{791114B9-4853-4c36-A1A9-8D0533C09B97}')

set @number = '31030'
insert into @techpreq values (@number,'31029','{646510EC-4594-44a8-9E90-234BEABD7B2B}')

set @number = '31050'
insert into @techpreq values (@number,'02199','{67994AFC-7D35-46b4-A14B-176A6D5872D6}')
-- and
insert into @techpreq values (@number,'31000','{998450E6-64C4-4212-AA30-7948455A09F9}')

-- course 31070: technical prerequisites are described in free text specifying
-- required qulifications/skills which can be represented (only course numbers can
-- be specified)

set @number = '31200'
insert into @techpreq values (@number,'01030','{1C5FFFEB-9F38-4b3d-BDE0-9BC4B4ED24A0}')
-- and
insert into @techpreq values (@number,'10010','{2F943E26-B548-4da8-BDB6-F4AD377DF1B6}')

set @number = '31220'
insert into @techpreq values (@number,'31000','{5FCDE6CE-2A57-42b7-9969-29BC8DAA4563}')
-- and
insert into @techpreq values (@number,'31200','{D98FE5E7-ED1D-4293-A976-4FA46C57724B}')

set @number = '31230'
insert into @techpreq values (@number,'31200','{A3F06708-77DD-465f-8CE1-D53B0BE3019E}')
insert into @techpreq values (@number,'51001','{A3F06708-77DD-465f-8CE1-D53B0BE3019E}')

set @number = '31232'
insert into @techpreq values (@number,'31200','{91C37B71-6C3F-4fd8-B222-4EE8F1D44AB9}')
-- and
insert into @techpreq values (@number,'31230','{C3158D37-70CD-4c81-A3AC-F524C2AB38EC}')

set @number = '31236'
insert into @techpreq values (@number,'31230','{647EE499-3072-49ea-B3A9-B5F65EABC7EE}')

set @number = '31240'
```

```
insert into @techpreq values (@number,'51001','{0C3C1E07-5D0A-47e9-9385-D0CA0A0133D4}')
insert into @techpreq values (@number,'31200','{0C3C1E07-5D0A-47e9-9385-D0CA0A0133D4}')
insert into @techpreq values (@number,'64004','{0C3C1E07-5D0A-47e9-9385-D0CA0A0133D4}')
insert into @techpreq values (@number,'11003','{0C3C1E07-5D0A-47e9-9385-D0CA0A0133D4}')

set @number = '31250'
insert into @techpreq values (@number,'51001','{3AC7601B-00D1-4d8f-BD09-CF2D6C556CBF}')
insert into @techpreq values (@number,'31200','{3AC7601B-00D1-4d8f-BD09-CF2D6C556CBF}')
insert into @techpreq values (@number,'31240','{3AC7601B-00D1-4d8f-BD09-CF2D6C556CBF}')

set @number = '31260'
insert into @techpreq values (@number,'31200','{4AA244DD-F632-494f-938D-EB78F46F5FCE}')

set @number = '31270'
insert into @techpreq values (@number,'31200','{CCA8B7B7-430C-4295-86CD-5E92573A877A}')

-- course 31275: technical prerequisites are described in free text specifying
-- required qulifications/skills which can be represented (only course numbers can
-- be specified)

set @number = '31300'
insert into @techpreq values (@number,'01030','{C66CCA07-48E2-4a37-957A-6F2F575AE99F}')
insert into @techpreq values (@number,'01031','{C66CCA07-48E2-4a37-957A-6F2F575AE99F}')
insert into @techpreq values (@number,'01032','{C66CCA07-48E2-4a37-957A-6F2F575AE99F}')
insert into @techpreq values (@number,'01034','{C66CCA07-48E2-4a37-957A-6F2F575AE99F}')
insert into @techpreq values (@number,'36260','{C66CCA07-48E2-4a37-957A-6F2F575AE99F}')
insert into @techpreq values (@number,'36261','{C66CCA07-48E2-4a37-957A-6F2F575AE99F}')
insert into @techpreq values (@number,'28260','{C66CCA07-48E2-4a37-957A-6F2F575AE99F}')
insert into @techpreq values (@number,'92052','{C66CCA07-48E2-4a37-957A-6F2F575AE99F}')
-- and
insert into @techpreq values (@number,'92061','{E95D8EE8-D08E-49e5-9FD3-AB554A2FFF0C}')
insert into @techpreq values (@number,'01961','{E95D8EE8-D08E-49e5-9FD3-AB554A2FFF0C}')
insert into @techpreq values (@number,'01981','{E95D8EE8-D08E-49e5-9FD3-AB554A2FFF0C}')

set @number = '31305'
insert into @techpreq values (@number,'50300','{A6D8BD47-147C-48f1-BE2A-E815938786E6}')
insert into @techpreq values (@number,'31300','{A6D8BD47-147C-48f1-BE2A-E815938786E6}')
insert into @techpreq values (@number,'72231','{A6D8BD47-147C-48f1-BE2A-E815938786E6}')
insert into @techpreq values (@number,'92081','{A6D8BD47-147C-48f1-BE2A-E815938786E6}')

set @number = '31310'
insert into @techpreq values (@number,'50300','{07171B25-2F48-4e8f-8D4B-D3E20ECE9F66}')
insert into @techpreq values (@number,'36230','{07171B25-2F48-4e8f-8D4B-D3E20ECE9F66}')
insert into @techpreq values (@number,'50300','{07171B25-2F48-4e8f-8D4B-D3E20ECE9F66}')
insert into @techpreq values (@number,'31300','{07171B25-2F48-4e8f-8D4B-D3E20ECE9F66}')
insert into @techpreq values (@number,'72131','{07171B25-2F48-4e8f-8D4B-D3E20ECE9F66}')
insert into @techpreq values (@number,'72231','{07171B25-2F48-4e8f-8D4B-D3E20ECE9F66}')

set @number = '31320'
insert into @techpreq values (@number,'31310','{E4738A45-CD85-4669-BABF-6D9B46E21161}')

set @number = '31340'
insert into @techpreq values (@number,'31300','{9EDF96D1-2EF1-4307-BB89-E606AED65FCA}')

set @number = '31350'
insert into @techpreq values (@number,'01032','{729609A2-DA1E-4505-857F-F29B1AD16D5A}')
insert into @techpreq values (@number,'53013','{729609A2-DA1E-4505-857F-F29B1AD16D5A}')
-- and
insert into @techpreq values (@number,'48000','{3D20189C-675B-4e24-B462-3EC1023540B5}')
insert into @techpreq values (@number,'10007','{3D20189C-675B-4e24-B462-3EC1023540B5}')
insert into @techpreq values (@number,'10008','{3D20189C-675B-4e24-B462-3EC1023540B5}')

set @number = '31352'
insert into @techpreq values (@number,'92053','{1B27E058-EA1C-4a1f-B3BA-7C5C0E1CDAA2}')
insert into @techpreq values (@number,'92200','{1B27E058-EA1C-4a1f-B3BA-7C5C0E1CDAA2}')
insert into @techpreq values (@number,'50280','{1B27E058-EA1C-4a1f-B3BA-7C5C0E1CDAA2}')
insert into @techpreq values (@number,'45120','{1B27E058-EA1C-4a1f-B3BA-7C5C0E1CDAA2}')
insert into @techpreq values (@number,'32020','{1B27E058-EA1C-4a1f-B3BA-7C5C0E1CDAA2}')
insert into @techpreq values (@number,'31350','{1B27E058-EA1C-4a1f-B3BA-7C5C0E1CDAA2}')

set @number = '31353'
insert into @techpreq values (@number,'92506','{DDD58583-235B-4eb6-BB5A-E835FCA06F36}')
insert into @techpreq values (@number,'92509','{DDD58583-235B-4eb6-BB5A-E835FCA06F36}')
insert into @techpreq values (@number,'32620','{DDD58583-235B-4eb6-BB5A-E835FCA06F36}')
insert into @techpreq values (@number,'31352','{DDD58583-235B-4eb6-BB5A-E835FCA06F36}')
-- and
insert into @techpreq values (@number,'92081','{918F0EF9-D3C9-4fa0-944D-98C7085BAF1A}')
insert into @techpreq values (@number,'50300','{918F0EF9-D3C9-4fa0-944D-98C7085BAF1A}')
insert into @techpreq values (@number,'31300','{918F0EF9-D3C9-4fa0-944D-98C7085BAF1A}')

set @number = '31356'
insert into @techpreq values (@number,'31350','{77BC96D1-5DBE-4e54-BC97-E4EB0C89220C}')

set @number = '31358'
insert into @techpreq values (@number,'45120','{85CA16A8-8C72-4a37-B15F-9FC5F2FCA5C5}')
insert into @techpreq values (@number,'50280','{85CA16A8-8C72-4a37-B15F-9FC5F2FCA5C5}')
insert into @techpreq values (@number,'92053','{85CA16A8-8C72-4a37-B15F-9FC5F2FCA5C5}')
insert into @techpreq values (@number,'92200','{85CA16A8-8C72-4a37-B15F-9FC5F2FCA5C5}')
insert into @techpreq values (@number,'32020','{85CA16A8-8C72-4a37-B15F-9FC5F2FCA5C5}')
insert into @techpreq values (@number,'31350','{85CA16A8-8C72-4a37-B15F-9FC5F2FCA5C5}')
-- and
insert into @techpreq values (@number,'31352','{1001EC9B-003F-41dd-9C4B-023ABE0A3C9D}')
insert into @techpreq values (@number,'32620','{1001EC9B-003F-41dd-9C4B-023ABE0A3C9D}')
insert into @techpreq values (@number,'92509','{1001EC9B-003F-41dd-9C4B-023ABE0A3C9D}')
insert into @techpreq values (@number,'92506','{1001EC9B-003F-41dd-9C4B-023ABE0A3C9D}')
-- and
insert into @techpreq values (@number,'92081','{93D72C88-8152-478c-B876-4FFC3AE27033}')
insert into @techpreq values (@number,'50300','{93D72C88-8152-478c-B876-4FFC3AE27033}')
insert into @techpreq values (@number,'31300','{93D72C88-8152-478c-B876-4FFC3AE27033}')
```

```
set @number = '31360'
insert into @techpreq values (@number,'36230','{02D890FE-CF4B-4da9-BE64-405AED0C266D}')
insert into @techpreq values (@number,'50300','{02D890FE-CF4B-4da9-BE64-405AED0C266D}')
insert into @techpreq values (@number,'72131','{02D890FE-CF4B-4da9-BE64-405AED0C266D}')
insert into @techpreq values (@number,'72231','{02D890FE-CF4B-4da9-BE64-405AED0C266D}')
insert into @techpreq values (@number,'31300','{02D890FE-CF4B-4da9-BE64-405AED0C266D}')

set @number = '31361'
insert into @techpreq values (@number,'36230','{9539EB79-D369-4974-9BA7-2EAD199B4E50}')
insert into @techpreq values (@number,'50300','{9539EB79-D369-4974-9BA7-2EAD199B4E50}')
insert into @techpreq values (@number,'72131','{9539EB79-D369-4974-9BA7-2EAD199B4E50}')
insert into @techpreq values (@number,'72231','{9539EB79-D369-4974-9BA7-2EAD199B4E50}')
insert into @techpreq values (@number,'31300','{9539EB79-D369-4974-9BA7-2EAD199B4E50}')

set @number = '31365'
insert into @techpreq values (@number,'50300','{912B621D-29B2-406c-8F29-2A367DBC7601}')
insert into @techpreq values (@number,'36230','{912B621D-29B2-406c-8F29-2A367DBC7601}')
insert into @techpreq values (@number,'72231','{912B621D-29B2-406c-8F29-2A367DBC7601}')

-- course 31368: technical prerequisites can not be handled.

-- course 31370: technical prerequisites are described in free text specifying
-- required qulifications/skills which can be represented (only course numbers can
-- be specified)

-- course 31380: technical prerequisites are described in free text specifying
-- required qulifications/skills which can be represented (only course numbers can
-- be specified)

-- course 31385: technical prerequisites are described in free text specifying
-- required qulifications/skills which can be represented (only course numbers can
-- be specified)

set @number = '31390'
insert into @techpreq values (@number,'28250','{3FBFB717-2DFA-4177-9020-25BDA468EAA3}')
insert into @techpreq values (@number,'31300','{3FBFB717-2DFA-4177-9020-25BDA468EAA3}')
insert into @techpreq values (@number,'36230','{3FBFB717-2DFA-4177-9020-25BDA468EAA3}')
insert into @techpreq values (@number,'50300','{3FBFB717-2DFA-4177-9020-25BDA468EAA3}')
insert into @techpreq values (@number,'72131','{3FBFB717-2DFA-4177-9020-25BDA468EAA3}')
insert into @techpreq values (@number,'72231','{3FBFB717-2DFA-4177-9020-25BDA468EAA3}')

set @number = '31400'
insert into @techpreq values (@number,'31000','{8ED8E6B2-813C-4f26-A5AD-448FF1DBF494}')
-- and
insert into @techpreq values (@number,'01005','{F578DDDE-FEB2-4f01-B67D-FC153A071171}')

set @number = '31405'
insert into @techpreq values (@number,'31400','{FD7E6A85-6BBA-4430-AFCE-0BF3C199CC23}')
insert into @techpreq values (@number,'10014','{FD7E6A85-6BBA-4430-AFCE-0BF3C199CC23}')
insert into @techpreq values (@number,'31480','{FD7E6A85-6BBA-4430-AFCE-0BF3C199CC23}')

set @number = '31411'
insert into @techpreq values (@number,'31405','{5AFFC9B2-59BE-48db-A294-4CC5FC3A6BA3}')

set @number = '31415'
insert into @techpreq values (@number,'31405','{EB39D5E3-244D-486b-9601-DDA6C137874C}')
insert into @techpreq values (@number,'48110','{EB39D5E3-244D-486b-9601-DDA6C137874C}')

set @number = '31420'
insert into @techpreq values (@number,'31405','{85F0C77D-4A1A-47cd-9345-A7FC98B61BF0}')

set @number = '31425'
insert into @techpreq values (@number,'31405','{C150FAD5-2F6D-4770-B95F-423B16D9BC3F}')
insert into @techpreq values (@number,'48110','{C150FAD5-2F6D-4770-B95F-423B16D9BC3F}')

set @number = '31430'
insert into @techpreq values (@number,'31405','{FCA4633C-5999-4dc9-A57C-23660D1F3CA5}')

set @number = '31435'
insert into @techpreq values (@number,'31430','{A414C00C-1289-492b-AE94-03E51568CC3D}')

set @number = '31440'
insert into @techpreq values (@number,'31405','{B48DA7FF-49AA-4277-8270-9A0F9A81D899}')

set @number = '31480'
insert into @techpreq values (@number,'31025','{903299E6-90FA-4b8c-B5D3-0C1B97084722}')

set @number = '31490'
insert into @techpreq values (@number,'31025','{21B4242F-44BE-40ce-A79C-E11F1393DC3C}')

set @number = '31606'
insert into @techpreq values (@number,'01032','{5FB17666-FD83-4208-AE20-6D473E272FC5}')

set @number = '31610'
insert into @techpreq values (@number,'01030','{731B4E4F-E616-4f20-91BF-2046252F1965}')
insert into @techpreq values (@number,'01032','{731B4E4F-E616-4f20-91BF-2046252F1965}')
-- and
insert into @techpreq values (@number,'02401','{1201CF8D-F7D3-4ef1-887A-8BE3628E3A34}')
insert into @techpreq values (@number,'04040','{1201CF8D-F7D3-4ef1-887A-8BE3628E3A34}')
insert into @techpreq values (@number,'04041','{1201CF8D-F7D3-4ef1-887A-8BE3628E3A34}')

set @number = '31625'
insert into @techpreq values (@number,'49101','{61A43210-1AAE-406e-8218-AAA2CA7DE028}')
insert into @techpreq values (@number,'49108','{61A43210-1AAE-406e-8218-AAA2CA7DE028}')

set @number = '31630'
insert into @techpreq values (@number,'49108','{67BBC928-D9C4-4f5a-9B5D-9579FF532E7F}')
insert into @techpreq values (@number,'31000','{67BBC928-D9C4-4f5a-9B5D-9579FF532E7F}')
```

```
set @number = '31640'
insert into @techpreq values (@number,'31000','{2F8F8309-8E30-4611-87AD-75E3779435CA}')
-- and
insert into @techpreq values (@number,'49116','{3D2D2C7A-C439-436b-8711-2428AF777922}')

set @number = '31641'
insert into @techpreq values (@number,'31640','{24A333F7-1E1D-4fc2-8586-E0924CE2F147}')
insert into @techpreq values (@number,'49325','{24A333F7-1E1D-4fc2-8586-E0924CE2F147}')

set @number = '31650'
insert into @techpreq values (@number,'41621','{E1D20E11-ACC1-41d5-99F4-612AF1B32526}')
insert into @techpreq values (@number,'27311','{E1D20E11-ACC1-41d5-99F4-612AF1B32526}')

set @number = '31651'
insert into @techpreq values (@number,'01032','{B0DCB86E-5F63-4fbb-8D47-12D61FE8ED62}')
-- and
insert into @techpreq values (@number,'31650','{3BE9CFE3-BFBC-4541-8180-5B45B95440D0}')
-- and
insert into @techpreq values (@number,'41621','{380B206B-E112-4d7c-8B5B-A709024A89BA}')

-- course 31652: some parts of the technical prerequisites specified can not be handled.
set @number = '31652'
insert into @techpreq values (@number,'31029','{81B1410A-5158-46d3-B296-1F6E6A77E92E}')
-- and
insert into @techpreq values (@number,'31657','{8D89B330-8FAA-47ed-BA23-E96325544BD6}')

-- course 31653: some parts of the technical prerequisites specified can not be handled.
set @number = '31653'
insert into @techpreq values (@number,'31029','{3426D059-5410-4ab4-9605-59508AFC8F8B}')
-- and
insert into @techpreq values (@number,'31657','{87A41949-0CDF-4678-9AF9-134A6FEFC6BB}')

set @number = '31655'
insert into @techpreq values (@number,'49215','{C85F3D99-2B4C-494b-9B88-98B068466C62}')
-- and
insert into @techpreq values (@number,'49210','{44D5FD4B-747A-4825-9C55-143D63BF4B69}')
insert into @techpreq values (@number,'52201','{44D5FD4B-747A-4825-9C55-143D63BF4B69}')
insert into @techpreq values (@number,'49415','{44D5FD4B-747A-4825-9C55-143D63BF4B69}')
insert into @techpreq values (@number,'31610','{44D5FD4B-747A-4825-9C55-143D63BF4B69}')
insert into @techpreq values (@number,'04361','{44D5FD4B-747A-4825-9C55-143D63BF4B69}')
insert into @techpreq values (@number,'02451','{44D5FD4B-747A-4825-9C55-143D63BF4B69}')
insert into @techpreq values (@number,'31030','{44D5FD4B-747A-4825-9C55-143D63BF4B69}')

set @number = '31656'
insert into @techpreq values (@number,'31650','{72D56483-67CE-4b1a-9F18-CF4FF2577885}')
insert into @techpreq values (@number,'31657','{72D56483-67CE-4b1a-9F18-CF4FF2577885}')
insert into @techpreq values (@number,'31658','{72D56483-67CE-4b1a-9F18-CF4FF2577885}')
insert into @techpreq values (@number,'92604','{72D56483-67CE-4b1a-9F18-CF4FF2577885}')
insert into @techpreq values (@number,'92605','{72D56483-67CE-4b1a-9F18-CF4FF2577885}')

set @number = '31671'
insert into @techpreq values (@number,'31030','{E7BCD71B-CF9D-4994-9A31-5FCB000E9063}')
insert into @techpreq values (@number,'01032','{E7BCD71B-CF9D-4994-9A31-5FCB000E9063}')

-- course 31730: technical prerequisites can not be handled.

set @number = '31740'
insert into @techpreq values (@number,'31730','{3753C122-974E-4cbb-AD35-1C0AB6504B2F}')

set @number = '31750'
insert into @techpreq values (@number,'31730','{86A0208B-C2E0-4c73-A223-253D8B0D8CBE}')

set @number = '31775'
insert into @techpreq values (@number,'32020','{D6931A65-CB76-485c-B430-58B6BFE33557}')
insert into @techpreq values (@number,'32060','{D6931A65-CB76-485c-B430-58B6BFE33557}')

set @number = '31777'
insert into @techpreq values (@number,'10007','{557B570A-16C0-4cc2-B6AA-E86AC4BE3FBF}')
insert into @techpreq values (@number,'10013','{557B570A-16C0-4cc2-B6AA-E86AC4BE3FBF}')
insert into @techpreq values (@number,'48000','{557B570A-16C0-4cc2-B6AA-E86AC4BE3FBF}')
-- and
insert into @techpreq values (@number,'10004','{7B5C8780-9142-47d9-AF8D-0A71106164AA}')
insert into @techpreq values (@number,'10012','{7B5C8780-9142-47d9-AF8D-0A71106164AA}')

-- course 31781: technical prerequisites can not be handled.

-- course 31782: some parts of the technical prerequisites specified can not be handled.
set @number = '31782'
insert into @techpreq values (@number,'31781','{546587A3-1E8C-4e57-A02F-60E1D21A88F2}')

-- course 31790: technical prerequisites can not be handled.

set @number = '31792'
insert into @techpreq values (@number,'31791','{1F9F594B-80A8-495c-8A7B-94BBF1CFD40F}')
insert into @techpreq values (@number,'32810','{1F9F594B-80A8-495c-8A7B-94BBF1CFD40F}')
insert into @techpreq values (@number,'53185','{1F9F594B-80A8-495c-8A7B-94BBF1CFD40F}')

set @number = '31793'
insert into @techpreq values (@number,'53185','{928A9240-A6B1-49c1-82B2-6EB000191B46}')
insert into @techpreq values (@number,'32810','{928A9240-A6B1-49c1-82B2-6EB000191B46}')
insert into @techpreq values (@number,'31791','{928A9240-A6B1-49c1-82B2-6EB000191B46}')

set @number = '31800'
insert into @techpreq values (@number,'10472','{172557CE-52A6-4661-8AD6-42EFE68C7F46}')
insert into @techpreq values (@number,'31650','{172557CE-52A6-4661-8AD6-42EFE68C7F46}')
insert into @techpreq values (@number,'41621','{172557CE-52A6-4661-8AD6-42EFE68C7F46}')
insert into @techpreq values (@number,'27311','{172557CE-52A6-4661-8AD6-42EFE68C7F46}')
-- and
```

```
insert into @techpreq values (@number,'01030','{AD53E75A-43B0-4e72-8BC9-C57A66286391}')
insert into @techpreq values (@number,'01031','{AD53E75A-43B0-4e72-8BC9-C57A66286391}')
insert into @techpreq values (@number,'01032','{AD53E75A-43B0-4e72-8BC9-C57A66286391}')
insert into @techpreq values (@number,'01902','{AD53E75A-43B0-4e72-8BC9-C57A66286391}')
insert into @techpreq values (@number,'01962','{AD53E75A-43B0-4e72-8BC9-C57A66286391}')

set @number = '31820'
insert into @techpreq values (@number,'31000','{108A0218-941B-4ee4-BD8C-BB936555DD94}')
insert into @techpreq values (@number,'31400','{108A0218-941B-4ee4-BD8C-BB936555DD94}')

set @number = '31825'
insert into @techpreq values (@number,'50200','{569DD1F4-5F45-48a8-8D08-6C63854D8C7E}')
insert into @techpreq values (@number,'31820','{569DD1F4-5F45-48a8-8D08-6C63854D8C7E}')

set @number = '31830'
insert into @techpreq values (@number,'31820','{66D324A4-C625-4d43-8F90-9EB46F3342DA}')
insert into @techpreq values (@number,'72141','{66D324A4-C625-4d43-8F90-9EB46F3342DA}')
insert into @techpreq values (@number,'49215','{66D324A4-C625-4d43-8F90-9EB46F3342DA}')

set @number = '31840'
insert into @techpreq values (@number,'01141','{1FE40303-5588-42e6-BB0A-820CCFDAE597}')
insert into @techpreq values (@number,'49105','{1FE40303-5588-42e6-BB0A-820CCFDAE597}')
insert into @techpreq values (@number,'49135','{1FE40303-5588-42e6-BB0A-820CCFDAE597}')
-- and
insert into @techpreq values (@number,'50250','{43214B81-B83C-4827-8C10-DE8729DF19D3}')
insert into @techpreq values (@number,'31050','{43214B81-B83C-4827-8C10-DE8729DF19D3}')

set @number = '31860'
insert into @techpreq values (@number,'50250','{46784914-8EA8-45ff-BF90-B30281E24F32}')
insert into @techpreq values (@number,'31050','{46784914-8EA8-45ff-BF90-B30281E24F32}')

set @number = '31875'
insert into @techpreq values (@number,'31024','{9C37F6FD-1FD1-4d53-94EF-A62FDCAC5EAD}')

-- course 33205: some parts of the technical prerequisites specified can not be handled.
set @number = '33205'
insert into @techpreq values (@number,'10300','{2FD89AFD-94DA-4708-9013-1E8C867E08D1}')
insert into @techpreq values (@number,'10306','{2FD89AFD-94DA-4708-9013-1E8C867E08D1}')
-- and
insert into @techpreq values (@number,'10112','{4921403A-F3C1-4379-833B-95ED726937DE}')

set @number = '33250'
insert into @techpreq values (@number,'33253','{8F75777B-977A-474f-AE65-604DBDE2B1E5}')
-- and
insert into @techpreq values (@number,'26026','{F0F0D7A8-AACA-41cb-AA09-D7376DD895A3}')
insert into @techpreq values (@number,'10004','{F0F0D7A8-AACA-41cb-AA09-D7376DD895A3}')

set @number = '33251'
insert into @techpreq values (@number,'33253','{93DC1F8E-C16D-4dc6-A0C0-81EB7C270950}')

-- course 33253: technical prerequisites are described in free text with no course
-- numbers specified. Based on the text a set of course numbers have been specified
-- below.
set @number = '33253'
insert into @techpreq values (@number,'10009','{BFE36F03-086C-4519-ABAB-490B2C8F1241}')
insert into @techpreq values (@number,'10013','{BFE36F03-086C-4519-ABAB-490B2C8F1241}')
insert into @techpreq values (@number,'10405','{BFE36F03-086C-4519-ABAB-490B2C8F1241}')
insert into @techpreq values (@number,'31000','{BFE36F03-086C-4519-ABAB-490B2C8F1241}')
insert into @techpreq values (@number,'31025','{BFE36F03-086C-4519-ABAB-490B2C8F1241}')
insert into @techpreq values (@number,'31400','{BFE36F03-086C-4519-ABAB-490B2C8F1241}')
insert into @techpreq values (@number,'31600','{BFE36F03-086C-4519-ABAB-490B2C8F1241}')
-- and
insert into @techpreq values (@number,'10010','{46FF9883-4B93-4f15-8273-A86AEC1C34B3}')
insert into @techpreq values (@number,'10011','{46FF9883-4B93-4f15-8273-A86AEC1C34B3}')
insert into @techpreq values (@number,'10911','{46FF9883-4B93-4f15-8273-A86AEC1C34B3}')
insert into @techpreq values (@number,'10912','{46FF9883-4B93-4f15-8273-A86AEC1C34B3}')
insert into @techpreq values (@number,'41015','{46FF9883-4B93-4f15-8273-A86AEC1C34B3}')

-- course 33320: technical prerequisites can not be handled.

set @number = '33321'
insert into @techpreq values (@number,'33320','{19668F76-1B4B-4e21-AF53-B0C32B42F97F}')
insert into @techpreq values (@number,'10009','{19668F76-1B4B-4e21-AF53-B0C32B42F97F}')
-- and
insert into @techpreq values (@number,'31400','{ADD63F10-0D14-437e-9750-4E7F80CA87FB}')
insert into @techpreq values (@number,'10111','{ADD63F10-0D14-437e-9750-4E7F80CA87FB}')
insert into @techpreq values (@number,'26260','{ADD63F10-0D14-437e-9750-4E7F80CA87FB}')
insert into @techpreq values (@number,'10100','{ADD63F10-0D14-437e-9750-4E7F80CA87FB}')

set @number = '33441'
insert into @techpreq values (@number,'10201','{1346F470-4CD2-4f8b-8B6B-34B3EA7FBA1F}')
insert into @techpreq values (@number,'21270','{1346F470-4CD2-4f8b-8B6B-34B3EA7FBA1F}')
-- and
insert into @techpreq values (@number,'10203','{FB97C49C-AEAC-46d2-A332-C8C56134851D}')
insert into @techpreq values (@number,'10221','{FB97C49C-AEAC-46d2-A332-C8C56134851D}')
insert into @techpreq values (@number,'21475','{FB97C49C-AEAC-46d2-A332-C8C56134851D}')

-- course 33443: technical prerequisites does not make sense.

set @number = '33470'
insert into @techpreq values (@number,'33253','{1EB895DE-8168-4d8d-85FF-C97723903081}')
-- and
insert into @techpreq values (@number,'26020','{71D7F9F9-D8AF-48a5-9A22-57ECC73EC076}')
insert into @techpreq values (@number,'10004','{71D7F9F9-D8AF-48a5-9A22-57ECC73EC076}')

set @number = '33471'
insert into @techpreq values (@number,'33253','{C9BD1290-FD2A-4900-83BE-DB3A8C35307D}')
insert into @techpreq values (@number,'33320','{C9BD1290-FD2A-4900-83BE-DB3A8C35307D}')
insert into @techpreq values (@number,'33355','{C9BD1290-FD2A-4900-83BE-DB3A8C35307D}')
```

```
insert into @techpreq values (@number,'33250','{C9BD1290-FD2A-4900-83BE-DB3A8C35307D}')
insert into @techpreq values (@number,'33251','{C9BD1290-FD2A-4900-83BE-DB3A8C35307D}')
insert into @techpreq values (@number,'33232','{C9BD1290-FD2A-4900-83BE-DB3A8C35307D}')

set @number = '33522'
insert into @techpreq values (@number,'33321','{8EB7499D-B18B-40d0-8654-A8B3F34D3FB2}')
insert into @techpreq values (@number,'33320','{8EB7499D-B18B-40d0-8654-A8B3F34D3FB2}')

set @number = '33534'
insert into @techpreq values (@number,'33232','{19FAFF93-510F-4b4c-8D0F-8E933CBFF8B3}')

set @number = '33556'
insert into @techpreq values (@number,'33355','{543F85BA-1846-446b-8B66-EB53DA6D6F7D}')

set @number = '34030'
insert into @techpreq values (@number,'10013','{BEF16955-92E2-407d-B171-EC244061637A}')
insert into @techpreq values (@number,'31400','{BEF16955-92E2-407d-B171-EC244061637A}')
-- and
insert into @techpreq values (@number,'10300','{CC594485-DE55-4b3f-BCDB-AB75EA2B90BE}')
insert into @techpreq values (@number,'33253','{CC594485-DE55-4b3f-BCDB-AB75EA2B90BE}')
-- and
insert into @techpreq values (@number,'10100','{8DE99C0C-C9FA-4e87-893A-669E22DC86FC}')

set @number = '34040'
insert into @techpreq values (@number,'44207','{C1F5F77A-778A-46fc-AE17-6F83E1E427CD}')
-- and
insert into @techpreq values (@number,'48110','{1BEE66DB-C7E5-46e6-B4B4-76F2A5F72A7C}')

set @number = '34049'
insert into @techpreq values (@number,'34030','{B832A06D-FAE0-42ee-B7C8-7F0CC0A2DA57}')
insert into @techpreq values (@number,'34040','{B832A06D-FAE0-42ee-B7C8-7F0CC0A2DA57}')
insert into @techpreq values (@number,'34140','{B832A06D-FAE0-42ee-B7C8-7F0CC0A2DA57}')
insert into @techpreq values (@number,'46210','{B832A06D-FAE0-42ee-B7C8-7F0CC0A2DA57}')
insert into @techpreq values (@number,'44207','{B832A06D-FAE0-42ee-B7C8-7F0CC0A2DA57}')
insert into @techpreq values (@number,'46220','{B832A06D-FAE0-42ee-B7C8-7F0CC0A2DA57}')
insert into @techpreq values (@number,'44201','{B832A06D-FAE0-42ee-B7C8-7F0CC0A2DA57}')
insert into @techpreq values (@number,'46240','{B832A06D-FAE0-42ee-B7C8-7F0CC0A2DA57}')
insert into @techpreq values (@number,'48240','{B832A06D-FAE0-42ee-B7C8-7F0CC0A2DA57}')

-- course 34051: some parts of the technical prerequisites specified can not be handled.
set @number = '34051'
insert into @techpreq values (@number,'34040','{DC9CC938-AA72-4c5f-8BE3-523FF6B38D6D}')
-- and
insert into @techpreq values (@number,'10111','{7B628949-E62D-4e70-BBB8-7C54C97E02FA}')

set @number = '34140'
insert into @techpreq values (@number,'31400','{08EC4DB8-996C-4e5d-971C-2010F543635E}')
insert into @techpreq values (@number,'48000','{08EC4DB8-996C-4e5d-971C-2010F543635E}')
insert into @techpreq values (@number,'10015','{08EC4DB8-996C-4e5d-971C-2010F543635E}')

set @number = '34150'
insert into @techpreq values (@number,'34140','{E7ECA8F4-2AEF-4e05-994B-F4BD3E02DAFA}')
insert into @techpreq values (@number,'46240','{E7ECA8F4-2AEF-4e05-994B-F4BD3E02DAFA}')
insert into @techpreq values (@number,'48240','{E7ECA8F4-2AEF-4e05-994B-F4BD3E02DAFA}')

set @number = '34230'
insert into @techpreq values (@number,'01032','{F7D23B85-D1F7-4c73-8080-EA7ED05831EB}')
insert into @techpreq values (@number,'02451','{F7D23B85-D1F7-4c73-8080-EA7ED05831EB}')
insert into @techpreq values (@number,'04361','{F7D23B85-D1F7-4c73-8080-EA7ED05831EB}')
-- and
insert into @techpreq values (@number,'01142','{E0ED8363-8F1F-4edf-8633-59C8492FD656}')
insert into @techpreq values (@number,'04040','{E0ED8363-8F1F-4edf-8633-59C8492FD656}')
insert into @techpreq values (@number,'04041','{E0ED8363-8F1F-4edf-8633-59C8492FD656}')

set @number = '34240'
insert into @techpreq values (@number,'34230','{51866352-2581-4d9e-9A26-FDA0DB257DE3}')
insert into @techpreq values (@number,'52200','{51866352-2581-4d9e-9A26-FDA0DB257DE3}')
insert into @techpreq values (@number,'52201','{51866352-2581-4d9e-9A26-FDA0DB257DE3}')
insert into @techpreq values (@number,'02501','{51866352-2581-4d9e-9A26-FDA0DB257DE3}')
insert into @techpreq values (@number,'04250','{51866352-2581-4d9e-9A26-FDA0DB257DE3}')

set @number = '34241'
insert into @techpreq values (@number,'02501','{EAF8E2F2-0BEC-4468-B931-EA0A2ED49583}')
insert into @techpreq values (@number,'04250','{EAF8E2F2-0BEC-4468-B931-EA0A2ED49583}')
-- and
insert into @techpreq values (@number,'34230','{65CF5C40-C194-403f-8028-E3AFB13A114C}')
insert into @techpreq values (@number,'34320','{65CF5C40-C194-403f-8028-E3AFB13A114C}')
insert into @techpreq values (@number,'52200','{65CF5C40-C194-403f-8028-E3AFB13A114C}')
insert into @techpreq values (@number,'52201','{65CF5C40-C194-403f-8028-E3AFB13A114C}')

set @number = '34300'
insert into @techpreq values (@number,'31000','{A5EF3FEB-E8BC-4db2-AC3A-D3B9B8E0F936}')

set @number = '34340'
insert into @techpreq values (@number,'01142','{DB2B6348-AF2B-44c3-A4CA-D00E022E809E}')
insert into @techpreq values (@number,'01963','{DB2B6348-AF2B-44c3-A4CA-D00E022E809E}')
insert into @techpreq values (@number,'04041','{DB2B6348-AF2B-44c3-A4CA-D00E022E809E}')
insert into @techpreq values (@number,'02401','{DB2B6348-AF2B-44c3-A4CA-D00E022E809E}')
insert into @techpreq values (@number,'04040','{DB2B6348-AF2B-44c3-A4CA-D00E022E809E}')

set @number = '34341'
insert into @techpreq values (@number,'34320','{0F4A0E93-AE71-49a0-B14B-428E77494DBD}')
insert into @techpreq values (@number,'34321','{0F4A0E93-AE71-49a0-B14B-428E77494DBD}')
insert into @techpreq values (@number,'52235','{0F4A0E93-AE71-49a0-B14B-428E77494DBD}')
insert into @techpreq values (@number,'52210','{0F4A0E93-AE71-49a0-B14B-428E77494DBD}')
-- and
insert into @techpreq values (@number,'01142','{10A07265-D8A5-4c3d-AB6E-3C27B6544024}')

set @number = '34349'
```

```
insert into @techpreq values (@number,'02200','{E363ACD9-D57E-4c0f-B397-CBBB05C17FA7}')
insert into @techpreq values (@number,'02206','{E363ACD9-D57E-4c0f-B397-CBBB05C17FA7}')

set @number = '34350'
insert into @techpreq values (@number,'34320','{C71E4C95-EBE0-4c5e-B938-05DC2AA8AC83}')
insert into @techpreq values (@number,'34321','{C71E4C95-EBE0-4c5e-B938-05DC2AA8AC83}')

set @number = '34351'
insert into @techpreq values (@number,'34320','{904D9B87-CDA8-487f-9512-A39955B371C4}')
insert into @techpreq values (@number,'34321','{904D9B87-CDA8-487f-9512-A39955B371C4}')

set @number = '34352'
insert into @techpreq values (@number,'34350','{BB8A8409-AF07-4322-83E4-E4E5AE2F2FC9}')

set @number = '34353'
insert into @techpreq values (@number,'34150','{2C1F3289-7FCD-4bf6-87C4-0F881D918C0B}')
insert into @techpreq values (@number,'46310','{2C1F3289-7FCD-4bf6-87C4-0F881D918C0B}')
insert into @techpreq values (@number,'48342','{2C1F3289-7FCD-4bf6-87C4-0F881D918C0B}')

set @number = '34354'
insert into @techpreq values (@number,'34350','{DFE4F510-48FE-4037-B379-A96025447E2D}')

set @number = '34355'
insert into @techpreq values (@number,'34320','{76A3F9E4-0093-4454-843E-4FD888EC190E}')
insert into @techpreq values (@number,'34321','{76A3F9E4-0093-4454-843E-4FD888EC190E}')

set @number = '34356'
insert into @techpreq values (@number,'34350','{3D3C32FB-1076-4864-8A89-96516AA42BDC}')

-- course 34390: technical prerequisites can not be handled.

set @number = '34641'
insert into @techpreq values (@number,'02100','{2DD0591E-404B-4e12-A3F1-AD43A8DD686D}')
insert into @techpreq values (@number,'02197','{2DD0591E-404B-4e12-A3F1-AD43A8DD686D}')
insert into @techpreq values (@number,'02199','{2DD0591E-404B-4e12-A3F1-AD43A8DD686D}')
insert into @techpreq values (@number,'21390','{2DD0591E-404B-4e12-A3F1-AD43A8DD686D}')
insert into @techpreq values (@number,'49135','{2DD0591E-404B-4e12-A3F1-AD43A8DD686D}')
insert into @techpreq values (@number,'49161','{2DD0591E-404B-4e12-A3F1-AD43A8DD686D}')
insert into @techpreq values (@number,'41861','{2DD0591E-404B-4e12-A3F1-AD43A8DD686D}')

set @number = '34642'
insert into @techpreq values (@number,'02264','{D53D0D00-D4D7-4777-8ED8-921EC96563ED}')

set @number = '34643'
insert into @techpreq values (@number,'34320','{5F6BF173-0EFC-4999-8F9E-81B4A725F3B4}')
insert into @techpreq values (@number,'34351','{5F6BF173-0EFC-4999-8F9E-81B4A725F3B4}')
insert into @techpreq values (@number,'34531','{5F6BF173-0EFC-4999-8F9E-81B4A725F3B4}')

set @number = '34821'
insert into @techpreq values (@number,'85221','{B7F19BC5-204E-4837-9F2F-4ECA929E7225}')

set @number = '34841'
insert into @techpreq values (@number,' 85221','{54238A29-0359-4c83-9409-D255B9828BDA}')

set @number = '34842'
insert into @techpreq values (@number,'34531','{69824281-0C66-4555-804D-15280917195B}')

set @number = '41031'
insert into @techpreq values (@number,'41012','{27DF7B7C-675C-4710-8A4A-BF56B1DE4247}')
-- and
insert into @techpreq values (@number,'42020','{EF95DBA8-BF05-4c20-844C-56AA6C317E80}')

set @number = '41035'
insert into @techpreq values (@number,'41015','{E0A7DF0E-6022-4d26-ACBE-02DD10100076}')
-- and
insert into @techpreq values (@number,'01007','{AFA79383-CDCB-46d4-A630-0A19679678E0}')

set @number = '41045'
insert into @techpreq values (@number,'01007','{11032DBD-3D38-4af6-A78C-5276D6829BDB}')

set @number = '41112'
insert into @techpreq values (@number,'41110','{F02464C2-312F-421a-B7EF-626BAE625240}')

set @number = '41120'
insert into @techpreq values (@number,'41110','{7BF6E526-7A64-4a93-B5B9-19F0FCFEF33E}')

set @number = '41121'
insert into @techpreq values (@number,'41110','{E1AACCCF-40C1-4888-B175-4CB241E828E4}')

set @number = '41126'
insert into @techpreq values (@number,'57105','{0F22DD51-2863-4372-8037-A02A73622DB6}')
insert into @techpreq values (@number,'12520','{0F22DD51-2863-4372-8037-A02A73622DB6}')
insert into @techpreq values (@number,'41110','{0F22DD51-2863-4372-8037-A02A73622DB6}')

set @number = '41123'
insert into @techpreq values (@number,'41110','{80378ED9-B5AD-45a6-80A9-C4D3AF7A5F90}')
-- and
insert into @techpreq values (@number,'41120','{504D01A0-7BED-4280-9971-78D792415996}')
-- and
insert into @techpreq values (@number,'41121','{CECC902F-0EA5-43be-95AE-99B74BF09993}')

set @number = '41124'
insert into @techpreq values (@number,'57105','{58CA55DA-F38C-4e85-8019-2AFE1D349543}')
insert into @techpreq values (@number,'12520','{58CA55DA-F38C-4e85-8019-2AFE1D349543}')
insert into @techpreq values (@number,'41110','{58CA55DA-F38C-4e85-8019-2AFE1D349543}')

-- course 41201: technical prerequisites can not be handled.

-- course 41211: some parts of the technical prerequisites specified can not be handled.
```

```
set @number = '41211'
insert into @techpreq values (@number,'41201','{6E2925A6-EFC4-456d-85CB-398295B2DA02}')
-- and
insert into @techpreq values (@number,'41501','{64713610-C7D8-4280-B608-A67B42B7E1CC}')
-- and
insert into @techpreq values (@number,'01142','{783CAD00-1D75-409f-8AC1-0B20D8CD2B0B}')
insert into @techpreq values (@number,'02323','{783CAD00-1D75-409f-8AC1-0B20D8CD2B0B}')
insert into @techpreq values (@number,'02402','{783CAD00-1D75-409f-8AC1-0B20D8CD2B0B}')
insert into @techpreq values (@number,'02591','{783CAD00-1D75-409f-8AC1-0B20D8CD2B0B}')

-- course 41212: technical prerequisites can not be handled.

set @number = '41213'
insert into @techpreq values (@number,'11511','{ACAF9C94-3025-49c0-B580-23A650B2A22E}')
insert into @techpreq values (@number,'41501','{ACAF9C94-3025-49c0-B580-23A650B2A22E}')

set @number = '41221'
insert into @techpreq values (@number,'41201','{6AF1E9C9-2903-4603-8A8D-B2E771709C3C}')

set @number = '41222'
insert into @techpreq values (@number,'41502','{FDA032CB-BFAE-4014-BA5C-5C791B4F852B}')
-- and
insert into @techpreq values (@number,'41812','{EBFCCD3C-EE3E-4267-BA17-CF6E1C503633}')

-- course 41223: some parts of the technical prerequisites specified can not be handled.
set @number = '41223'
insert into @techpreq values (@number,'12521','{AB5AD7BA-1D91-44ef-B696-5BB2DB2E4C7E}')
insert into @techpreq values (@number,'41311','{AB5AD7BA-1D91-44ef-B696-5BB2DB2E4C7E}')
insert into @techpreq values (@number,'10346','{AB5AD7BA-1D91-44ef-B696-5BB2DB2E4C7E}')
-- and
insert into @techpreq values (@number,'01030','{8F312042-37CE-4da4-8184-5036565B613D}')
insert into @techpreq values (@number,'01032','{8F312042-37CE-4da4-8184-5036565B613D}')
insert into @techpreq values (@number,'01034','{8F312042-37CE-4da4-8184-5036565B613D}')

set @number = '41224'
insert into @techpreq values (@number,'41213','{3C35AE01-ECC8-4fd0-B2F9-0D0184BA108C}')
insert into @techpreq values (@number,'11551','{3C35AE01-ECC8-4fd0-B2F9-0D0184BA108C}')
insert into @techpreq values (@number,'59318','{3C35AE01-ECC8-4fd0-B2F9-0D0184BA108C}')

set @number = '41225'
insert into @techpreq values (@number,'01005','{53E9FAE6-5BEA-43a5-8E74-6C92C027217F}')
insert into @techpreq values (@number,'01007','{53E9FAE6-5BEA-43a5-8E74-6C92C027217F}')
-- and
insert into @techpreq values (@number,'01142','{5D2824CE-55A7-4a69-BFA3-E7123486737E}')
insert into @techpreq values (@number,'02401','{5D2824CE-55A7-4a69-BFA3-E7123486737E}')
-- and
insert into @techpreq values (@number,'01034','{31AD4A88-F900-47bb-9B00-50AC3764B879}')
insert into @techpreq values (@number,'11511','{31AD4A88-F900-47bb-9B00-50AC3764B879}')
insert into @techpreq values (@number,'41501','{31AD4A88-F900-47bb-9B00-50AC3764B879}')

set @number = '41245'
insert into @techpreq values (@number,'41230','{554391F3-F50F-414f-9687-D93A802D0271}')
insert into @techpreq values (@number,'76700','{554391F3-F50F-414f-9687-D93A802D0271}')

-- course 41260: technical prerequisites can not be handled.

-- course 41262: technical prerequisites can not be handled.

set @number = '41271'
insert into @techpreq values (@number,'41201','{2A741D30-3708-437d-AC0E-90C8BD616A94}')
insert into @techpreq values (@number,'41262','{2A741D30-3708-437d-AC0E-90C8BD616A94}')
insert into @techpreq values (@number,'76161','{2A741D30-3708-437d-AC0E-90C8BD616A94}')

set @number = '41272'
insert into @techpreq values (@number,'41201 ','{4B000502-68CA-4aa0-B638-0A207A07B1C2}')

set @number = '41312'
insert into @techpreq values (@number,'01005','{8079B774-ACB7-4b59-B614-F37CEF3FC655}')

-- course 41313: technical prerequisites can not be handled.

set @number = '41320'
insert into @techpreq values (@number,'02601','{FF63B8B4-F12A-4c86-BD8B-685F47E08738}')
-- and
insert into @techpreq values (@number,'41312','{67AC672D-CB82-4e11-BD53-540E6DF892C7}')
insert into @techpreq values (@number,'41110','{67AC672D-CB82-4e11-BD53-540E6DF892C7}')
insert into @techpreq values (@number,'41221','{67AC672D-CB82-4e11-BD53-540E6DF892C7}')

-- course 41322: technical prerequisites can not be handled.

set @number = '41323'
insert into @techpreq values (@number,'41312','{C9850C61-2768-4da4-B6A2-8E8A78C0E6C7}')
insert into @techpreq values (@number,'41110','{C9850C61-2768-4da4-B6A2-8E8A78C0E6C7}')

set @number = '41324'
insert into @techpreq values (@number,'41312','{F84FDF89-E5D1-4541-A4D4-AA4FB5D63377}')
insert into @techpreq values (@number,'41110','{F84FDF89-E5D1-4541-A4D4-AA4FB5D63377}')
-- and
insert into @techpreq values (@number,'41861','{CBA60D12-CC3A-458a-8C20-FDA5F05F1321}')

set @number = '41401'
insert into @techpreq values (@number,'10012','{CC90F109-B93A-4485-8398-2011FE674C44}')
insert into @techpreq values (@number,'10004','{CC90F109-B93A-4485-8398-2011FE674C44}')
insert into @techpreq values (@number,'10005','{CC90F109-B93A-4485-8398-2011FE674C44}')
-- and
insert into @techpreq values (@number,'26201','{C2C23538-860B-445c-AFD1-A8E1E497B867}')

set @number = '41402'
insert into @techpreq values (@number,'41401','{87FE522C-43E3-4a02-A7AC-B72F97838CB8}')
```

```
set @number = '41410'
insert into @techpreq values (@number,'41403','{18EC120B-77C1-4ee8-8EAD-BA18438C2FAE}')
-- and
insert into @techpreq values (@number,'26026','{4EECC185-019C-473d-9E74-2B6660D37607}')

set @number = '41414'
insert into @techpreq values (@number,'77141','{2612F0BE-D0F8-4676-94A2-823805199E6E}')
insert into @techpreq values (@number,'41401','{2612F0BE-D0F8-4676-94A2-823805199E6E}')

set @number = '41415'
insert into @techpreq values (@number,'41401','{BC2EECC2-183E-4db3-B032-E6BA8D8F76D9}')

set @number = '41416'
insert into @techpreq values (@number,'41401','{5905899E-53BE-4123-A700-AAAC5F89AFFC}')
insert into @techpreq values (@number,'41311','{5905899E-53BE-4123-A700-AAAC5F89AFFC}')
insert into @techpreq values (@number,'41312','{5905899E-53BE-4123-A700-AAAC5F89AFFC}')

set @number = '41418'
insert into @techpreq values (@number,'41401','{6A3551EA-5ECE-476d-B5A7-EB70987AD969}')
insert into @techpreq values (@number,'26201','{6A3551EA-5ECE-476d-B5A7-EB70987AD969}')

set @number = '41430'
insert into @techpreq values (@number,'41531','{C5706031-82EC-462a-BD47-5DA48D935647}')

set @number = '41501'
insert into @techpreq values (@number,'01020','{DF6A65CF-0E1A-4023-BC5C-1062622BB397}')
insert into @techpreq values (@number,'01021','{DF6A65CF-0E1A-4023-BC5C-1062622BB397}')
insert into @techpreq values (@number,'01002','{DF6A65CF-0E1A-4023-BC5C-1062622BB397}')
insert into @techpreq values (@number,'01003','{DF6A65CF-0E1A-4023-BC5C-1062622BB397}')
insert into @techpreq values (@number,'01005','{DF6A65CF-0E1A-4023-BC5C-1062622BB397}')
-- and
insert into @techpreq values (@number,'10002','{F35597FF-4AD4-4191-9901-9057E9518DC7}')
insert into @techpreq values (@number,'10010','{F35597FF-4AD4-4191-9901-9057E9518DC7}')
insert into @techpreq values (@number,'10011','{F35597FF-4AD4-4191-9901-9057E9518DC7}')
insert into @techpreq values (@number,'10001','{F35597FF-4AD4-4191-9901-9057E9518DC7}')

set @number = '41502'
insert into @techpreq values (@number,'70101','{EB5335B7-74D0-4d41-B603-A6D5F6FD21A1}')
insert into @techpreq values (@number,'41501','{EB5335B7-74D0-4d41-B603-A6D5F6FD21A1}')

set @number = '41511'
insert into @techpreq values (@number,'70102','{70FA8F03-2A8F-4079-99C8-3CE5B49EAA03}')
insert into @techpreq values (@number,'41502','{70FA8F03-2A8F-4079-99C8-3CE5B49EAA03}')

-- course 41515: some parts of the technical prerequisites specified can not be handled.
set @number = '41515'
insert into @techpreq values (@number,'41861','{A071784D-D5FB-4bb5-AEBD-AC48691FF7BE}')
insert into @techpreq values (@number,'26500','{A071784D-D5FB-4bb5-AEBD-AC48691FF7BE}')
insert into @techpreq values (@number,'02197','{A071784D-D5FB-4bb5-AEBD-AC48691FF7BE}')

set @number = '41521'
insert into @techpreq values (@number,'41560','{F9E0D99E-33D7-4d93-A364-1C913AF006E6}')
insert into @techpreq values (@number,'41512','{F9E0D99E-33D7-4d93-A364-1C913AF006E6}')
insert into @techpreq values (@number,'70205','{F9E0D99E-33D7-4d93-A364-1C913AF006E6}')

set @number = '41522'
insert into @techpreq values (@number,'70102','{730A60E7-3B3C-4955-85AA-AF0FADB8BFA6}')
insert into @techpreq values (@number,'41502','{730A60E7-3B3C-4955-85AA-AF0FADB8BFA6}')

set @number = '41525'
insert into @techpreq values (@number,'70102','{EA2E3547-8892-4600-9F77-438B398FD1A5}')
insert into @techpreq values (@number,'41502','{EA2E3547-8892-4600-9F77-438B398FD1A5}')
-- and
insert into @techpreq values (@number,'70212','{3069BBE6-2A8F-4d10-92E8-7BD46ABEB0C6}')
insert into @techpreq values (@number,'41515','{3069BBE6-2A8F-4d10-92E8-7BD46ABEB0C6}')

set @number = '41531'
insert into @techpreq values (@number,'01921','{3B4312DC-BDB5-4e3d-B33B-CBA55E9460B2}')
insert into @techpreq values (@number,'94101','{3B4312DC-BDB5-4e3d-B33B-CBA55E9460B2}')
-- and
insert into @techpreq values (@number,'41530','{DE9135AD-A6A4-4709-8AA9-34E453A9A228}')
insert into @techpreq values (@number,'94301','{DE9135AD-A6A4-4709-8AA9-34E453A9A228}')

set @number = '41533'
insert into @techpreq values (@number,'41532','{D3390449-9BCA-47b1-83E8-60D19EC815B1}')

set @number = '41534'
insert into @techpreq values (@number,'41532','{8A2F1F5C-49ED-4fb7-800E-BCD102A12510}')
-- and
insert into @techpreq values (@number,'41533','{A92F71A2-E9AE-4c23-BC5F-BEA2059458CA}')

set @number = '41535'
insert into @techpreq values (@number,'01905','{1AA11A9B-A442-480f-A783-1C7F340F2A97}')

set @number = '41560'
insert into @techpreq values (@number,'01030','{5850E3D7-C732-416a-9BF2-A153218FDA23}')
insert into @techpreq values (@number,'01032','{5850E3D7-C732-416a-9BF2-A153218FDA23}')
insert into @techpreq values (@number,'01034','{5850E3D7-C732-416a-9BF2-A153218FDA23}')
insert into @techpreq values (@number,'28260','{5850E3D7-C732-416a-9BF2-A153218FDA23}')
-- and
insert into @techpreq values (@number,'41501','{89341562-C90C-4e55-922F-8628AB37CF68}')

set @number = '41602'
insert into @techpreq values (@number,'41601','{27269986-94DC-4364-BC37-8A5CEA5609E7}')

set @number = '41611'
insert into @techpreq values (@number,'10011','{64249305-B1C6-454c-AB68-85D260431E8D}')
-- and
```

```
insert into @techpreq values (@number,'41501','{867390A6-46D0-4220-9408-B4AA0FCCE307}')
-- and
insert into @techpreq values (@number,'41502','{211D61B2-4CA2-4043-A752-89486A699328}')

set @number = '41612'
insert into @techpreq values (@number,'41601','{F0A23B65-D4A8-4130-8C1C-12D720558B68}')
-- and
insert into @techpreq values (@number,'42301','{EC79BE00-FCB2-4ef0-9833-3B6959288D8D}')
insert into @techpreq values (@number,'42302','{EC79BE00-FCB2-4ef0-9833-3B6959288D8D}')
insert into @techpreq values (@number,'41305','{EC79BE00-FCB2-4ef0-9833-3B6959288D8D}')
insert into @techpreq values (@number,'41309','{EC79BE00-FCB2-4ef0-9833-3B6959288D8D}')

set @number = '41614'
insert into @techpreq values (@number,'10010','{4F5C30A7-3BD0-4b09-B80F-63484FCCD74E}')
-- and
insert into @techpreq values (@number,'41560','{4538404A-07F5-4808-B157-B8D534BE6F4C}')
insert into @techpreq values (@number,'41512','{4538404A-07F5-4808-B157-B8D534BE6F4C}')
insert into @techpreq values (@number,'70205','{4538404A-07F5-4808-B157-B8D534BE6F4C}')

set @number = '41622'
insert into @techpreq values (@number,'41611','{440C6862-9AB3-4ce6-BA1A-C16A523A2290}')
-- and
insert into @techpreq values (@number,'01030','{59A36456-F482-4201-BD61-EBF4051C4346}')

set @number = '41625'
insert into @techpreq values (@number,'31300','{710288D3-EF5E-40c0-8ECF-EEEDFBEA5BBD}')
insert into @techpreq values (@number,'41659','{710288D3-EF5E-40c0-8ECF-EEEDFBEA5BBD}')

set @number = '41627'
insert into @techpreq values (@number,'41612','{356DEA17-FCEF-4399-9D02-91C614AF4B43}')

set @number = '41628'
insert into @techpreq values (@number,'41602','{B8C061A9-255F-401f-98D7-B373D98B1112}')
-- and
insert into @techpreq values (@number,'41612','{19951349-9A03-47b3-A0FD-50F43C5CB4F1}')

set @number = '41629'
insert into @techpreq values (@number,'41612','{733CF413-F2EF-474b-A76B-E2EA0F6DDF44}')

set @number = '41632'
insert into @techpreq values (@number,'41530','{D185F673-2DBE-43f6-A3B9-C96B96A7782B}')
insert into @techpreq values (@number,'94301','{D185F673-2DBE-43f6-A3B9-C96B96A7782B}')
-- and
insert into @techpreq values (@number,'41630','{10FAC287-EA5E-4901-9C7A-00AB286F2164}')
insert into @techpreq values (@number,'94346','{10FAC287-EA5E-4901-9C7A-00AB286F2164}')

set @number = '41647'
insert into @techpreq values (@number,'41530','{97AE2284-73CC-4233-9B2A-714C28DDAD3B}')
-- and
insert into @techpreq values (@number,'41632','{DE210273-ED30-4b06-B1BC-8DB6D302F56B}')

set @number = '41660'
insert into @techpreq values (@number,'41646','{AB796B57-AEF8-44c2-87F9-E1ADE5F67349}')
insert into @techpreq values (@number,'94215','{AB796B57-AEF8-44c2-87F9-E1ADE5F67349}')
-- and
insert into @techpreq values (@number,'41632','{F0E683DF-B869-477e-9CCC-DF320FFA7F4C}')
insert into @techpreq values (@number,'94315','{F0E683DF-B869-477e-9CCC-DF320FFA7F4C}')
-- and
insert into @techpreq values (@number,'41612','{63806971-CE2D-475f-AF0D-D0C01C12C326}')
insert into @techpreq values (@number,'72221','{63806971-CE2D-475f-AF0D-D0C01C12C326}')
-- and
insert into @techpreq values (@number,'49163','{CB265BBD-F05D-4cd7-B07C-7A7743DFED23}')
insert into @techpreq values (@number,'31600','{CB265BBD-F05D-4cd7-B07C-7A7743DFED23}')

set @number = '41661'
insert into @techpreq values (@number,'28250','{6C4E26B3-3E2B-483c-89E7-A39B8CC5D07F}')
insert into @techpreq values (@number,'31300','{6C4E26B3-3E2B-483c-89E7-A39B8CC5D07F}')
insert into @techpreq values (@number,'41637','{6C4E26B3-3E2B-483c-89E7-A39B8CC5D07F}')
insert into @techpreq values (@number,'41659','{6C4E26B3-3E2B-483c-89E7-A39B8CC5D07F}')
insert into @techpreq values (@number,'41626','{6C4E26B3-3E2B-483c-89E7-A39B8CC5D07F}')

-- course 41670: technical prerequisites can not be handled.

set @number = '41721'
insert into @techpreq values (@number,'41311','{C99BA587-7054-428f-B2D8-1E34D7FFFB4B}')
-- and
insert into @techpreq values (@number,'41814','{E3236C50-2CC2-489c-849B-9EAFA175185A}')
-- and
insert into @techpreq values (@number,'41815','{BC91274C-BF44-48ef-93FF-334079278D0C}')

set @number = '41811'
insert into @techpreq values (@number,'70102','{37047DB0-7880-4919-8CB4-E76E73D5A06F}')
insert into @techpreq values (@number,'41502','{37047DB0-7880-4919-8CB4-E76E73D5A06F}')

-- course 41812: some parts of the technical prerequisites specified can not be handled.
set @number = '41812'
insert into @techpreq values (@number,'41501','{0668263F-51F5-4599-85E4-2CE3C51B79A5}')
insert into @techpreq values (@number,'41502','{0668263F-51F5-4599-85E4-2CE3C51B79A5}')

set @number = '41813'
insert into @techpreq values (@number,'41401','{262BCCBD-40BA-457d-BF65-DEA82A676AC6}')
insert into @techpreq values (@number,'77141','{262BCCBD-40BA-457d-BF65-DEA82A676AC6}')
insert into @techpreq values (@number,'41430','{262BCCBD-40BA-457d-BF65-DEA82A676AC6}')
insert into @techpreq values (@number,'94200','{262BCCBD-40BA-457d-BF65-DEA82A676AC6}')

set @number = '41814'
insert into @techpreq values (@number,'01030','{0DC6E238-9726-49bf-86C0-99DD040D352C}')
-- and
insert into @techpreq values (@number,'41401','{EA6BD72E-CADE-4fed-B2F5-B28C105CF223}')
```

```
set @number = '41815'
insert into @techpreq values (@number,'41401','{179DB008-3590-4b20-8A81-B982F94A2EDF}')

-- course 41816: technical prerequisites can not be handled.

set @number = '41822'
insert into @techpreq values (@number,'41312','{D94D6D88-26A0-4ed9-956D-E887BB069B52}')
insert into @techpreq values (@number,'41110','{D94D6D88-26A0-4ed9-956D-E887BB069B52}')

set @number = '41841'
insert into @techpreq values (@number,'01905','{30C3DF11-F956-4209-9C01-CE13DCA4B745}')

set @number = '41861'
insert into @techpreq values (@number,'01005','{F6593800-B042-425c-B4B9-E733DA3AA960}')
-- and
insert into @techpreq values (@number,'10010','{36D3A5D0-7E45-40b4-BC18-BB93FF25BD5F}')
insert into @techpreq values (@number,'10011','{36D3A5D0-7E45-40b4-BC18-BB93FF25BD5F}')
-- and
insert into @techpreq values (@number,'10012','{43D4909F-BB51-4bc6-BA46-D8BEECB4976A}')

set @number = '42020'
insert into @techpreq values (@number,'41010','{58620EAE-4BFA-4514-9A71-8CD782033C09}')

set @number = '42021'
insert into @techpreq values (@number,'42011','{46E39A02-6F76-4a09-91C1-6DF6A6258737}')

set @number = '42041'
insert into @techpreq values (@number,'42021','{BD04A090-87CD-4e06-AE1F-453E202795A9}')

set @number = '42120'
insert into @techpreq values (@number,'42110','{A30C95AE-F561-453e-B3B8-6D0AE9151084}')
insert into @techpreq values (@number,'80251','{A30C95AE-F561-453e-B3B8-6D0AE9151084}')

set @number = '42135'
insert into @techpreq values (@number,'42130','{4FD81995-ABFC-4d98-ABB1-9BDFB479A4BE}')
insert into @techpreq values (@number,'80252','{4FD81995-ABFC-4d98-ABB1-9BDFB479A4BE}')
-- and
insert into @techpreq values (@number,'42120','{D6E4AC43-D1AC-4457-8E8A-297E151CE386}')
insert into @techpreq values (@number,'80253','{D6E4AC43-D1AC-4457-8E8A-297E151CE386}')

set @number = '42155'
insert into @techpreq values (@number,'42936','{F6A8F840-ACAC-4ce0-B23C-AEB838EBA6A3}')

set @number = '42221'
insert into @techpreq values (@number,'42110','{5A0B31A3-E25E-4ea4-B215-29E68EA923D5}')

set @number = '42222'
insert into @techpreq values (@number,'42301','{D43B37B6-440A-46a4-8D2E-2947B7DF46C4}')
insert into @techpreq values (@number,'80001','{D43B37B6-440A-46a4-8D2E-2947B7DF46C4}')

set @number = '42224'
insert into @techpreq values (@number,'42301','{D2F49BCD-821D-49a1-B3CB-592E1F377B2D}')
insert into @techpreq values (@number,'80001','{D2F49BCD-821D-49a1-B3CB-592E1F377B2D}')
-- and
insert into @techpreq values (@number,'10012','{744CEA33-347A-4640-8E72-BA4E62CB7A4D}')
-- and
insert into @techpreq values (@number,'41861','{B8FEB27C-3254-4f0a-946F-7C284C0154E1}')

set @number = '42226'
insert into @techpreq values (@number,'42301','{6E057F0A-B2F7-4bcf-9235-11D1CBC79FC8}')
insert into @techpreq values (@number,'80001','{6E057F0A-B2F7-4bcf-9235-11D1CBC79FC8}')

set @number = '42250'
insert into @techpreq values (@number,'42301','{6D9FBF18-640F-4d20-8D67-A7AB17CF31ED}')
insert into @techpreq values (@number,'80001','{6D9FBF18-640F-4d20-8D67-A7AB17CF31ED}')

set @number = '42260'
insert into @techpreq values (@number,'42301','{5C7B38AB-41FF-4619-8FA4-5E3A40520839}')
insert into @techpreq values (@number,'80001','{5C7B38AB-41FF-4619-8FA4-5E3A40520839}')

-- course 42305: technical prerequisites can not be handled.

set @number = '42341'
insert into @techpreq values (@number,'42301','{B9D5C6BE-AB5C-4332-AEE7-B457B777FB37}')
insert into @techpreq values (@number,'80001','{B9D5C6BE-AB5C-4332-AEE7-B457B777FB37}')
insert into @techpreq values (@number,'42302','{B9D5C6BE-AB5C-4332-AEE7-B457B777FB37}')
insert into @techpreq values (@number,'80002','{B9D5C6BE-AB5C-4332-AEE7-B457B777FB37}')
insert into @techpreq values (@number,'42305','{B9D5C6BE-AB5C-4332-AEE7-B457B777FB37}')
insert into @techpreq values (@number,'80005','{B9D5C6BE-AB5C-4332-AEE7-B457B777FB37}')
insert into @techpreq values (@number,'42309','{B9D5C6BE-AB5C-4332-AEE7-B457B777FB37}')
insert into @techpreq values (@number,'80009','{B9D5C6BE-AB5C-4332-AEE7-B457B777FB37}')

set @number = '42371'
insert into @techpreq values (@number,'42405','{0A7835C5-C02F-41ea-B3BA-A37E05282279}')
insert into @techpreq values (@number,'80171','{0A7835C5-C02F-41ea-B3BA-A37E05282279}')
insert into @techpreq values (@number,'83171','{0A7835C5-C02F-41ea-B3BA-A37E05282279}')
-- and
insert into @techpreq values (@number,'42415','{ABA0FFE1-1277-4efd-90F8-0F077BAE5D40}')
insert into @techpreq values (@number,'80175','{ABA0FFE1-1277-4efd-90F8-0F077BAE5D40}')
insert into @techpreq values (@number,'83175','{ABA0FFE1-1277-4efd-90F8-0F077BAE5D40}')
-- and
insert into @techpreq values (@number,'94510','{DC5B7D82-B199-4600-A885-1D949249C177}')

-- course 42412: no technical prerequisites specified.

set @number = '42415'
insert into @techpreq values (@number,'42412','{114B94E4-FA88-4148-8EE4-13082292FC16}')
```

```
set @number = '42440'
insert into @techpreq values (@number,'42412','{E0F66E50-E9C9-43f5-A986-4553F15C2F75}')
insert into @techpreq values (@number,'85221','{E0F66E50-E9C9-43f5-A986-4553F15C2F75}')

set @number = '42450'
insert into @techpreq values (@number,'42405','{5504A3DD-2ABB-4366-AB61-B27D00CF0A25}')
insert into @techpreq values (@number,'80171','{5504A3DD-2ABB-4366-AB61-B27D00CF0A25}')
insert into @techpreq values (@number,'83171','{5504A3DD-2ABB-4366-AB61-B27D00CF0A25}')
insert into @techpreq values (@number,'42415','{5504A3DD-2ABB-4366-AB61-B27D00CF0A25}')
insert into @techpreq values (@number,'80175','{5504A3DD-2ABB-4366-AB61-B27D00CF0A25}')
insert into @techpreq values (@number,'83175','{5504A3DD-2ABB-4366-AB61-B27D00CF0A25}')

set @number = '42455'
insert into @techpreq values (@number,'42405','{14667B23-2AFD-4ac4-807B-B25C0CE21C76}')
insert into @techpreq values (@number,'80171','{14667B23-2AFD-4ac4-807B-B25C0CE21C76}')
insert into @techpreq values (@number,'83171','{14667B23-2AFD-4ac4-807B-B25C0CE21C76}')
-- and
insert into @techpreq values (@number,'42415','{3EC4BED7-595E-47b1-B444-5CB1F70D4BD0}')
insert into @techpreq values (@number,'80175','{3EC4BED7-595E-47b1-B444-5CB1F70D4BD0}')
insert into @techpreq values (@number,'83175','{3EC4BED7-595E-47b1-B444-5CB1F70D4BD0}')

set @number = '42460'
insert into @techpreq values (@number,'42405','{3D785E5C-5EEF-41a0-B439-FFEE39A36BB9}')
insert into @techpreq values (@number,'80171','{3D785E5C-5EEF-41a0-B439-FFEE39A36BB9}')
insert into @techpreq values (@number,'83171','{3D785E5C-5EEF-41a0-B439-FFEE39A36BB9}')

set @number = '42465'
insert into @techpreq values (@number,'42405','{678C3D7D-591A-440a-AD53-B9E47E58C386}')
insert into @techpreq values (@number,'83171','{678C3D7D-591A-440a-AD53-B9E47E58C386}')
insert into @techpreq values (@number,'80171','{678C3D7D-591A-440a-AD53-B9E47E58C386}')
insert into @techpreq values (@number,'42420','{678C3D7D-591A-440a-AD53-B9E47E58C386}')
insert into @techpreq values (@number,'83174','{678C3D7D-591A-440a-AD53-B9E47E58C386}')
insert into @techpreq values (@number,'86174','{678C3D7D-591A-440a-AD53-B9E47E58C386}')
insert into @techpreq values (@number,'42415','{678C3D7D-591A-440a-AD53-B9E47E58C386}')
insert into @techpreq values (@number,'83175','{678C3D7D-591A-440a-AD53-B9E47E58C386}')
insert into @techpreq values (@number,'80175','{678C3D7D-591A-440a-AD53-B9E47E58C386}')

-- course 42510: no technical prerequisites specified.

set @number = '42540'
insert into @techpreq values (@number,'42342','{FAFFD6EB-60FF-4068-8651-52F109D11069}')
insert into @techpreq values (@number,'42470','{FAFFD6EB-60FF-4068-8651-52F109D11069}')
insert into @techpreq values (@number,'42521','{FAFFD6EB-60FF-4068-8651-52F109D11069}')
insert into @techpreq values (@number,'42531','{FAFFD6EB-60FF-4068-8651-52F109D11069}')
insert into @techpreq values (@number,'42630','{FAFFD6EB-60FF-4068-8651-52F109D11069}')

set @number = '42541'
insert into @techpreq values (@number,'42531','{8A6F7C36-47CD-4d51-A698-4FD2F9E9828B}')
insert into @techpreq values (@number,'42420','{8A6F7C36-47CD-4d51-A698-4FD2F9E9828B}')

-- course 42640: some parts of the technical prerequisites specified can not be handled.
set @number = '42640'
insert into @techpreq values (@number,'42531','{C41B4CEA-4B80-47b8-B115-0B869B2CBE1A}') -- Organization, Work and Behaviour
insert into @techpreq values (@number,'02725','{C41B4CEA-4B80-47b8-B115-0B869B2CBE1A}') -- Planning and Participation

set @number = '42642'
insert into @techpreq values (@number,'42412','{A85E8D53-234D-4ef4-8EF2-B55982891BA2}') -- Introduction to Economics
-- and
insert into @techpreq values (@number,'42422','{9B44ACEA-9039-42bc-92E0-70C294A3F0EE}') -- Management and Organization
-- and
insert into @techpreq values (@number,'42531','{FAF39539-A972-4a63-96CF-A5F79DBD9815}') -- Organization, Work and Behaviour
insert into @techpreq values (@number,'42643','{FAF39539-A972-4a63-96CF-A5F79DBD9815}') -- Philosophy of Technology
insert into @techpreq values (@number,'11254','{FAF39539-A972-4a63-96CF-A5F79DBD9815}') -- Industrial Production Concepts in Construction

set @number = '42930'
insert into @techpreq values (@number,'41631','{6C1247B4-B2A0-4499-8BF9-483D4B574143}')
insert into @techpreq values (@number,'94400','{6C1247B4-B2A0-4499-8BF9-483D4B574143}')

set @number = '42931'
insert into @techpreq values (@number,'42930','{F21C022F-3B29-48c6-80FA-15C283C5064B}')
insert into @techpreq values (@number,'94405','{F21C022F-3B29-48c6-80FA-15C283C5064B}')

set @number = '42932'
insert into @techpreq values (@number,'41631','{B092BCE4-6B2E-42db-B370-D1251AC3971C}')
insert into @techpreq values (@number,'94400','{B092BCE4-6B2E-42db-B370-D1251AC3971C}')

set @number = '42964'
insert into @techpreq values (@number,'41830','{81CBCC1C-63E1-43f8-8E88-EA8B9844415F}')

-- course 42967: technical prerequisites can not be handled.

set @number = '42968'
insert into @techpreq values (@number,'02323','{10114183-B6E4-4753-8F12-1F343E076C5C}') -- Probability and Statistics
insert into @techpreq values (@number,'02402','{10114183-B6E4-4753-8F12-1F343E076C5C}') -- Introduction to Statistics
insert into @techpreq values (@number,'02591','{10114183-B6E4-4753-8F12-1F343E076C5C}') -- Statistics
-- and
insert into @techpreq values (@number,'42455','{56C39090-773D-470b-80F1-ADB9FCA05A44}') -- Production and Quality Management
insert into @techpreq values (@number,'42952','{56C39090-773D-470b-80F1-ADB9FCA05A44}') -- Production Control, Logistics and Statistics
insert into @techpreq values (@number,'42954','{56C39090-773D-470b-80F1-ADB9FCA05A44}') -- Production Control and Statistics

set @number = '42981'
insert into @techpreq values (@number,'28021','{2DC36BB4-66CF-44e7-8F43-1FD68CE435B6}')

set @number = '42983'
insert into @techpreq values (@number,'42981','{CD06744D-1E50-45ec-90EB-327C2E484DB0}')
insert into @techpreq values (@number,'42110','{CD06744D-1E50-45ec-90EB-327C2E484DB0}')

-- course 88383: technical prerequisites can not be handled.
```

```
set @number = '88891'
insert into @techpreq values (@number,'88890','{94FD321D-4F06-40bb-8B5B-C3B5CE183C58}')

set @number = '88893'
insert into @techpreq values (@number,'88892','{6B2F6CCD-D123-41dd-9E19-C7DDC414CA4E}')

------------------------------------------------------------------------------------------------------------------------

-- ********************************************************************************** --
------------------------------------------------------------------------------------------------------------------------


declare @userid uniqueidenti↓er
set @userid = '{40AA0D66-B5A3-4E42-9B13-BCDA0905EB9D}'

declare @curdate datetime
set @curdate = getdate()

declare @RelationCourseType_ID int
set @RelationCourseType_ID = 2

declare @courseversion_id uniqueidenti↓er

declare preq_cursor cursor for
select distinct relationcourse_id, number
from @techpreq
order by number
for read only
open preq_cursor

declare @coursenumber varchar(20)
declare @relationcourse_id uniqueidenti↓er

fetch next from preq_cursor into @relationcourse_id, @coursenumber
while @@fetch_status = 0
begin
  if exists (select courseversion_id from courseversion where number=@coursenumber)
  begin
    select @courseversion_id=courseversion_id from courseversion where number=@coursenumber

    insert into Course_RelationCourse
    (Course_RelationCourse_ID,
     CourseVersion_ID,
     Course_RelationCourseType_ID,
     Created,
     CreatedBy,
     Updated,
     UpdatedBy)
    values
    (@relationcourse_id,
     @courseversion_id,
     @RelationCourseType_ID,
     @curdate,
     @userid,
     @curdate,
     @userid)
  IF @@ROWCOUNT = 0
    RAISERROR(@coursenumber, 14, 1);
  end
  fetch next from preq_cursor into @relationcourse_id, @coursenumber
end

close preq_cursor
deallocate preq_cursor

------------------------------------------------------------

declare items_cursor cursor for
select number, techpreqcourse, relationcourse_id
from @techpreq
order by number, relationcourse_id
for read only
open items_cursor

declare @techpreqcourse varchar(20)
declare @tpc_relationcourse_id uniqueidenti↓er

fetch next from items_cursor into @coursenumber, @techpreqcourse, @tpc_relationcourse_id
while @@fetch_status = 0
begin
  if exists (select courseversion_id from courseversion where number=@coursenumber)
  begin
    insert into Course_RelationCourseItem
    (Course_RelationCourseItem_ID,
     Course_RelationCourse_ID,
     Course_ID,
     CourseNumber,
     Created,
     CreatedBy,
     Updated,
     UpdatedBy)
    values
    (newid(),
     @tpc_relationcourse_id,
     null,
     @techpreqcourse,
     @curdate,
     @userid,
     @curdate,
```

```
      @userid)
  end
  fetch next from items_cursor into @coursenumber, @techpreqcourse, @tpc_relationcourse_id
end

close items_cursor
deallocate items_cursor

-------------------------------------------------------------

declare items_cursor cursor for
select course_relationcourseitem_id, coursenumber
from course_relationcourseitem
for read only
open items_cursor

declare @course_relationcourseitem_id uniqueidenti↓er
declare @course_id uniqueidenti↓er

fetch next from items_cursor into @course_relationcourseitem_id, @coursenumber
while @@fetch_status = 0
begin
  if exists (select course_id from courseversion where number=@coursenumber)
  begin
    select @course_id=course_id from courseversion where number=@coursenumber

    update Course_RelationCourseItem
    set course_id=@course_id
    where course_relationcourseitem_id=@course_relationcourseitem_id
  end
  fetch next from items_cursor into @course_relationcourseitem_id, @coursenumber
end

close items_cursor
deallocate items_cursor
```

4.2.4 Point Blocking

```
declare @number varchar(20)

declare @pointblock table (
  number varchar(20),
  pointblockingcourse varchar(20),
  relationcourse_id uniqueidenti↓er
)

-----------------------------------------------------------------------------------------------------------------------

-----------------------------------------------------------------------------------------------------------------------


set @number = '01005'
insert into @pointblock values (@number,'01000','{A4917B1A-0754-4edc-B914-039F3942F131}')
insert into @pointblock values (@number,'01001','{A4917B1A-0754-4edc-B914-039F3942F131}')
insert into @pointblock values (@number,'01002','{A4917B1A-0754-4edc-B914-039F3942F131}')
insert into @pointblock values (@number,'01003','{A4917B1A-0754-4edc-B914-039F3942F131}')
insert into @pointblock values (@number,'01010','{A4917B1A-0754-4edc-B914-039F3942F131}')
insert into @pointblock values (@number,'01011','{A4917B1A-0754-4edc-B914-039F3942F131}')
insert into @pointblock values (@number,'01012','{A4917B1A-0754-4edc-B914-039F3942F131}')
insert into @pointblock values (@number,'01013','{A4917B1A-0754-4edc-B914-039F3942F131}')
insert into @pointblock values (@number,'01014','{A4917B1A-0754-4edc-B914-039F3942F131}')
insert into @pointblock values (@number,'01020','{A4917B1A-0754-4edc-B914-039F3942F131}')
insert into @pointblock values (@number,'01021','{A4917B1A-0754-4edc-B914-039F3942F131}')
insert into @pointblock values (@number,'01007','{A4917B1A-0754-4edc-B914-039F3942F131}')

set @number = '01006'
insert into @pointblock values (@number,'01000','{25273583-968A-4af5-9DDB-6BC68F04ECCF}')
insert into @pointblock values (@number,'01001','{25273583-968A-4af5-9DDB-6BC68F04ECCF}')
insert into @pointblock values (@number,'01002','{25273583-968A-4af5-9DDB-6BC68F04ECCF}')
insert into @pointblock values (@number,'01003','{25273583-968A-4af5-9DDB-6BC68F04ECCF}')
insert into @pointblock values (@number,'01010','{25273583-968A-4af5-9DDB-6BC68F04ECCF}')
insert into @pointblock values (@number,'01011','{25273583-968A-4af5-9DDB-6BC68F04ECCF}')
insert into @pointblock values (@number,'01012','{25273583-968A-4af5-9DDB-6BC68F04ECCF}')
insert into @pointblock values (@number,'01013','{25273583-968A-4af5-9DDB-6BC68F04ECCF}')
insert into @pointblock values (@number,'01014','{25273583-968A-4af5-9DDB-6BC68F04ECCF}')
insert into @pointblock values (@number,'01020','{25273583-968A-4af5-9DDB-6BC68F04ECCF}')
insert into @pointblock values (@number,'01021','{25273583-968A-4af5-9DDB-6BC68F04ECCF}')
insert into @pointblock values (@number,'01007','{25273583-968A-4af5-9DDB-6BC68F04ECCF}')

set @number = '01007'
insert into @pointblock values (@number,'01005','{410FAC6D-47DA-489b-BD0C-8CA5C3FF3D23}')
insert into @pointblock values (@number,'01006','{410FAC6D-47DA-489b-BD0C-8CA5C3FF3D23}')

set @number = '01030'
insert into @pointblock values (@number,'01031','{C6CEC8FA-FC82-437d-B5F9-E81ED3637476}')
insert into @pointblock values (@number,'01032','{C6CEC8FA-FC82-437d-B5F9-E81ED3637476}')
insert into @pointblock values (@number,'45020','{C6CEC8FA-FC82-437d-B5F9-E81ED3637476}')
insert into @pointblock values (@number,'01034','{C6CEC8FA-FC82-437d-B5F9-E81ED3637476}')
insert into @pointblock values (@number,'36260','{C6CEC8FA-FC82-437d-B5F9-E81ED3637476}')

set @number = '01031'
insert into @pointblock values (@number,'01030','{3602F0FA-A45C-441e-9A2D-217D98795B56}')
insert into @pointblock values (@number,'01032','{3602F0FA-A45C-441e-9A2D-217D98795B56}')
insert into @pointblock values (@number,'45020','{3602F0FA-A45C-441e-9A2D-217D98795B56}')
insert into @pointblock values (@number,'01034','{3602F0FA-A45C-441e-9A2D-217D98795B56}')
```

```
insert into @pointblock values (@number,'36260','{3602F0FA-A45C-441e-9A2D-217D98795B56}')

set @number = '01032'
insert into @pointblock values (@number,'01030','{D892B65C-2C22-4e8a-B875-338355BAB266}')
insert into @pointblock values (@number,'01031','{D892B65C-2C22-4e8a-B875-338355BAB266}')
insert into @pointblock values (@number,'01034','{D892B65C-2C22-4e8a-B875-338355BAB266}')
insert into @pointblock values (@number,'36260','{D892B65C-2C22-4e8a-B875-338355BAB266}')
insert into @pointblock values (@number,'45020','{D892B65C-2C22-4e8a-B875-338355BAB266}')

set @number = '01034'
insert into @pointblock values (@number,'01030','{2949508F-DCAA-49fe-A9EB-4DBDC22AF6C9}')
insert into @pointblock values (@number,'01031','{2949508F-DCAA-49fe-A9EB-4DBDC22AF6C9}')
insert into @pointblock values (@number,'01032','{2949508F-DCAA-49fe-A9EB-4DBDC22AF6C9}')
insert into @pointblock values (@number,'45020','{2949508F-DCAA-49fe-A9EB-4DBDC22AF6C9}')
insert into @pointblock values (@number,'36260','{2949508F-DCAA-49fe-A9EB-4DBDC22AF6C9}')

set @number = '01141'
insert into @pointblock values (@number,'02643','{91941E07-55CB-44b2-94B7-0C5F5C3E2284}')
insert into @pointblock values (@number,'04124','{91941E07-55CB-44b2-94B7-0C5F5C3E2284}')

set @number = '01902'
insert into @pointblock values (@number,'91128','{7D5116D2-4C4D-4512-9EC1-CC48DD24B357}')

set @number = '01911'
insert into @pointblock values (@number,'91118','{1A4C0968-4750-47ac-A262-0A09352D1808}')

set @number = '01912'
insert into @pointblock values (@number,'01902','{D2607294-31C9-46ac-B999-9A643D3BE299}')

set @number = '01922'
insert into @pointblock values (@number,'94110','{25F43CF7-5C25-4650-8C5E-01C899B0A847}')
insert into @pointblock values (@number,'94111','{25F43CF7-5C25-4650-8C5E-01C899B0A847}')

set @number = '01931'
insert into @pointblock values (@number,'94100','{596A464B-BC4B-4800-B833-30ACE14D0E92}')
insert into @pointblock values (@number,'94101','{596A464B-BC4B-4800-B833-30ACE14D0E92}')

set @number = '01932'
insert into @pointblock values (@number,'94110','{E1001C6A-F96F-4c0c-B7D7-BCFFF07264FA}')
insert into @pointblock values (@number,'94111','{E1001C6A-F96F-4c0c-B7D7-BCFFF07264FA}')

set @number = '01962'
insert into @pointblock values (@number,'01982','{076BEBD2-F177-4360-97A7-C7230147DB2E}')

set @number = '02100'
insert into @pointblock values (@number,'02199','{AB98B0D0-ED07-47e1-BADE-593306060B21}')
insert into @pointblock values (@number,'02115','{AB98B0D0-ED07-47e1-BADE-593306060B21}')
insert into @pointblock values (@number,'02312','{AB98B0D0-ED07-47e1-BADE-593306060B21}')

set @number = '02115'
insert into @pointblock values (@number,'02100','{0FF85531-FE68-40c7-A0E5-86E6905F2141}')
insert into @pointblock values (@number,'02199','{0FF85531-FE68-40c7-A0E5-86E6905F2141}')

set @number = '02160'
insert into @pointblock values (@number,'02344','{F76A3297-AA63-48c4-8808-C4EBE3F0D0D3}')
insert into @pointblock values (@number,'02264','{F76A3297-AA63-48c4-8808-C4EBE3F0D0D3}')

set @number = '02170'
insert into @pointblock values (@number,'02344','{7B1726BA-7355-4779-A6B8-D49CA106B9A3}')

set @number = '02197'
insert into @pointblock values (@number,'49105','{894C01DF-30E5-41ee-9AE2-D850B6ACDD28}')
insert into @pointblock values (@number,'02199','{894C01DF-30E5-41ee-9AE2-D850B6ACDD28}')

set @number = '02199'
insert into @pointblock values (@number,'02100','{A50AB677-6928-4b2c-A4A8-DCDC886DF8A6}')
insert into @pointblock values (@number,'02199','{A50AB677-6928-4b2c-A4A8-DCDC886DF8A6}')
insert into @pointblock values (@number,'02312','{A50AB677-6928-4b2c-A4A8-DCDC886DF8A6}')

set @number = '02200'
insert into @pointblock values (@number,'49280','{7EA88FD9-A1B5-4921-B161-64003444F6AC}')
insert into @pointblock values (@number,'02340','{7EA88FD9-A1B5-4921-B161-64003444F6AC}')

set @number = '02201'
insert into @pointblock values (@number,'49281','{1E85ECBE-C1E1-4ec3-B883-0E8BEFC53AC4}')

set @number = '02261'
insert into @pointblock values (@number,'49234','{0DA8332B-66F8-40cb-9D4F-C94F89245FCC}')

set @number = '02312'
insert into @pointblock values (@number,'02100','{FFF67214-3CFC-4508-B7D8-03EA836B31AA}')
insert into @pointblock values (@number,'02115','{FFF67214-3CFC-4508-B7D8-03EA836B31AA}')
insert into @pointblock values (@number,'02199','{FFF67214-3CFC-4508-B7D8-03EA836B31AA}')
insert into @pointblock values (@number,'02318','{FFF67214-3CFC-4508-B7D8-03EA836B31AA}')

set @number = '02323'
insert into @pointblock values (@number,'01963','{EFE69672-6BB6-46e5-87C6-E5C6FB6F109E}')
insert into @pointblock values (@number,'02401','{EFE69672-6BB6-46e5-87C6-E5C6FB6F109E}')
insert into @pointblock values (@number,'02402','{EFE69672-6BB6-46e5-87C6-E5C6FB6F109E}')

set @number = '02333'
insert into @pointblock values (@number,'02220','{D1D0C40D-5573-4cce-8089-A0221B5D0A76}')

set @number = '02335'
insert into @pointblock values (@number,'02391','{AE3F6A57-D5A0-4f94-BCA3-3DF27E745D38}')
insert into @pointblock values (@number,'34531','{AE3F6A57-D5A0-4f94-BCA3-3DF27E745D38}')
insert into @pointblock values (@number,'42967','{AE3F6A57-D5A0-4f94-BCA3-3DF27E745D38}')
insert into @pointblock values (@number,'91173','{AE3F6A57-D5A0-4f94-BCA3-3DF27E745D38}')
```

```
set @number = '02340'
insert into @pointblock values (@number,'02200','{9F17AA05-69E7-4acb-8E01-677AA46B021F}')

set @number = '02348'
insert into @pointblock values (@number,'92403','{34AC7A6B-7D3A-493a-AB6C-47962BCA29FF}')

set @number = '02393'
insert into @pointblock values (@number,'91174','{25908A28-1A2F-4e4c-8365-9D8A1829036C}')
insert into @pointblock values (@number,'49423','{25908A28-1A2F-4e4c-8365-9D8A1829036C}')
insert into @pointblock values (@number,'49420','{25908A28-1A2F-4e4c-8365-9D8A1829036C}')
insert into @pointblock values (@number,'02198','{25908A28-1A2F-4e4c-8365-9D8A1829036C}')

set @number = '02402'
insert into @pointblock values (@number,'01963','{50A9AA27-1F49-47ed-AAE0-EDC5BDEA74BC}')
insert into @pointblock values (@number,'02323','{50A9AA27-1F49-47ed-AAE0-EDC5BDEA74BC}')
insert into @pointblock values (@number,'02401','{50A9AA27-1F49-47ed-AAE0-EDC5BDEA74BC}')

set @number = '02405'
insert into @pointblock values (@number,'01142','{CDB1E3CD-BB09-4255-A1FF-7778988BC3E7}')

set @number = '02407'
insert into @pointblock values (@number,'04141','{35CEB430-377F-455c-B4D1-DF380188BCD2}')

set @number = '02409'
insert into @pointblock values (@number,'04241','{FBC3211A-3008-48dd-BA5A-A1D7476E7B76}')

set @number = '02411'
insert into @pointblock values (@number,'02593','{6DE8A312-83F5-47e8-8E8E-BD116B64D879}')
insert into @pointblock values (@number,'04242','{6DE8A312-83F5-47e8-8E8E-BD116B64D879}')

set @number = '02413'
insert into @pointblock values (@number,'04243','{778FC6F3-FB93-4469-9985-C432A53DB270}')

set @number = '02417'
insert into @pointblock values (@number,'04244','{6BF91970-2113-4db1-80D9-494ABB2569B6}')

set @number = '02451'
insert into @pointblock values (@number,'04362','{60ADC58C-9E33-4b0a-8BB3-8236568D6825}')

set @number = '02532'
insert into @pointblock values (@number,'02533','{8C7E7F98-76C3-441b-A6B6-C8145159F161}')
insert into @pointblock values (@number,'02535','{8C7E7F98-76C3-441b-A6B6-C8145159F161}')

set @number = '02591'
insert into @pointblock values (@number,'02401','{8979CCB3-8028-4391-A327-06CB20D00B6A}')

set @number = '02643'
insert into @pointblock values (@number,'01141','{20AD236F-D7E2-4e0c-9D47-CBFA6CAE1984}')
insert into @pointblock values (@number,'04124','{20AD236F-D7E2-4e0c-9D47-CBFA6CAE1984}')

set @number = '02709'
insert into @pointblock values (@number,'04230','{0A2E0550-AAB9-4f41-892D-063E4626AB12}')

set @number = '02731'
insert into @pointblock values (@number,'04434','{2CA9FD7E-38A7-4ade-AED2-3E2606F7B371}')

set @number = '02735'
insert into @pointblock values (@number,'04430','{14442BEB-C2D0-424c-B7CD-88CFC5DE55F9}')

set @number = '10001'
insert into @pointblock values (@number,'10010','{8C65344A-BA68-45ec-BD64-53CDDA46E75B}')
insert into @pointblock values (@number,'10011','{8C65344A-BA68-45ec-BD64-53CDDA46E75B}')

set @number = '10004'
insert into @pointblock values (@number,'10012','{0D398531-4F20-4cac-A053-2B498D00C13B}')
insert into @pointblock values (@number,'21061','{0D398531-4F20-4cac-A053-2B498D00C13B}')
insert into @pointblock values (@number,'21262','{0D398531-4F20-4cac-A053-2B498D00C13B}')
insert into @pointblock values (@number,'21010','{0D398531-4F20-4cac-A053-2B498D00C13B}')
insert into @pointblock values (@number,'26201','{0D398531-4F20-4cac-A053-2B498D00C13B}')

set @number = '10009'
insert into @pointblock values (@number,'10013','{C3D70D15-3E75-4051-B020-6F02BAC65C02}')
insert into @pointblock values (@number,'31400','{C3D70D15-3E75-4051-B020-6F02BAC65C02}')
insert into @pointblock values (@number,'48000','{C3D70D15-3E75-4051-B020-6F02BAC65C02}')

set @number = '10010'
insert into @pointblock values (@number,'10001','{D0C213EB-EF5F-48ae-8951-0D881F472366}')
insert into @pointblock values (@number,'10002','{D0C213EB-EF5F-48ae-8951-0D881F472366}')
insert into @pointblock values (@number,'10011','{D0C213EB-EF5F-48ae-8951-0D881F472366}')
insert into @pointblock values (@number,'10032','{D0C213EB-EF5F-48ae-8951-0D881F472366}')

set @number = '10011'
insert into @pointblock values (@number,'10001','{33A92E3C-F5FA-42f5-BFFF-6677A1C8AEB1}')
insert into @pointblock values (@number,'10002','{33A92E3C-F5FA-42f5-BFFF-6677A1C8AEB1}')
insert into @pointblock values (@number,'10010','{33A92E3C-F5FA-42f5-BFFF-6677A1C8AEB1}')
insert into @pointblock values (@number,'10032','{33A92E3C-F5FA-42f5-BFFF-6677A1C8AEB1}')

set @number = '10012'
insert into @pointblock values (@number,'10004','{3B736860-D6BB-4785-AB47-04720E1E16CF}')
insert into @pointblock values (@number,'10005','{3B736860-D6BB-4785-AB47-04720E1E16CF}')
insert into @pointblock values (@number,'26201','{3B736860-D6BB-4785-AB47-04720E1E16CF}')

set @number = '10013'
insert into @pointblock values (@number,'31400','{06B8A44B-7A0B-45c1-9857-488577CE5648}')
insert into @pointblock values (@number,'48000','{06B8A44B-7A0B-45c1-9857-488577CE5648}')
insert into @pointblock values (@number,'10007','{06B8A44B-7A0B-45c1-9857-488577CE5648}')
-- and
insert into @pointblock values (@number,'10008','{C6917CA4-E3AC-452e-9604-6747F0230135}')
insert into @pointblock values (@number,'10009','{C6917CA4-E3AC-452e-9604-6747F0230135}')
```

```
insert into @pointblock values (@number,'10014','{C6917CA4-E3AC-452e-9604-6747F0230135}')
-- and
insert into @pointblock values (@number,'10015','{EB604C09-FB01-411e-B7F5-2229A6943DCE}')

set @number = '10061'
insert into @pointblock values (@number,'10215','{A8D6BF31-C171-4fdf-BF89-F8AB87701EA2}')

set @number = '10100'
insert into @pointblock values (@number,'1201','{D1452C9F-920A-4501-BF3F-73D523CAB4D4}')
insert into @pointblock values (@number,'1202','{D1452C9F-920A-4501-BF3F-73D523CAB4D4}')

set @number = '10111'
insert into @pointblock values (@number,'1555','{6831CDAD-8BB3-4a7c-ADC0-46233043ED74}')
insert into @pointblock values (@number,'21270','{6831CDAD-8BB3-4a7c-ADC0-46233043ED74}')

set @number = '10112'
insert into @pointblock values (@number,'1557','{C6DDC746-EF50-48f9-A740-40FEDFD39885}')
insert into @pointblock values (@number,'4046','{C6DDC746-EF50-48f9-A740-40FEDFD39885}')

set @number = '10120'
insert into @pointblock values (@number,'10110','{E54A7F52-D7FD-4e4b-A764-7B627BD07198}')

set @number = '10300'
insert into @pointblock values (@number,'1556','{9F5518B2-2600-49d7-8B0E-A488A15F6FA2}')
insert into @pointblock values (@number,'10203','{9F5518B2-2600-49d7-8B0E-A488A15F6FA2}')

set @number = '10302'
insert into @pointblock values (@number,'10223','{BAF6A933-5B15-49e9-9696-4704CA1E0FF7}')

set @number = '10322'
insert into @pointblock values (@number,'10465','{83D776F2-881B-489a-8669-8D3F4FF88D7B}')

set @number = '10343'
insert into @pointblock values (@number,'10233','{FAC7E757-0866-41fc-867A-DC57D431169D}')
insert into @pointblock values (@number,'10433','{FAC7E757-0866-41fc-867A-DC57D431169D}')

set @number = '10344'
insert into @pointblock values (@number,'10237','{B3E6B906-7E53-4808-B037-97F58489C4EF}')

set @number = '10345'
insert into @pointblock values (@number,'02653','{646684CE-0000-429f-B598-7EC8788D944B}')

set @number = '10374'
insert into @pointblock values (@number,'10243','{12334242-B7FF-4f7c-B970-1BFA46DB0313}')

set @number = '10376'
insert into @pointblock values (@number,'10247','{D77A7D65-A383-4dfa-9355-DFE4D5B101EC}')

set @number = '10911'
insert into @pointblock values (@number,'1510','{EEC08C0C-18EA-4a12-957B-6C14E2A8594C}')
insert into @pointblock values (@number,'10010','{EEC08C0C-18EA-4a12-957B-6C14E2A8594C}')
insert into @pointblock values (@number,'1521','{EEC08C0C-18EA-4a12-957B-6C14E2A8594C}')
insert into @pointblock values (@number,'1532','{EEC08C0C-18EA-4a12-957B-6C14E2A8594C}')
insert into @pointblock values (@number,'10032','{EEC08C0C-18EA-4a12-957B-6C14E2A8594C}')

set @number = '11001'
insert into @pointblock values (@number,'64000','{8A07B38E-E2A0-4b92-87CB-20B1A4401375}')

set @number = '11002'
insert into @pointblock values (@number,'67000','{C14F1851-C4DA-4738-826D-0F34153181F0}')

set @number = '11003'
insert into @pointblock values (@number,'64116','{B8B2D57C-A09B-4326-9C4D-CE58A232FCAF}')
insert into @pointblock values (@number,'64004','{B8B2D57C-A09B-4326-9C4D-CE58A232FCAF}')

set @number = '11101'
insert into @pointblock values (@number,'64040','{3BA0B0EB-EAC0-4149-90D1-B04B7A16F6B9}')
insert into @pointblock values (@number,'11732','{3BA0B0EB-EAC0-4149-90D1-B04B7A16F6B9}')

set @number = '11102'
insert into @pointblock values (@number,'64044','{F2D9F234-EA75-436e-BD04-9449FFAF7578}')
insert into @pointblock values (@number,'64350','{F2D9F234-EA75-436e-BD04-9449FFAF7578}')

set @number = '11103'
insert into @pointblock values (@number,'64350','{7CEC3D55-910B-4dd3-8EF1-3FE96489D569}')
-- and
insert into @pointblock values (@number,'64420','{A1FA5DFC-8D89-47b3-8965-3A7FA415AF28}')

set @number = '11104'
insert into @pointblock values (@number,'64360','{581E2067-0EB1-4286-AC30-7CD44FBF5E2C}')

set @number = '11105'
insert into @pointblock values (@number,'64492','{4AF79574-7699-4808-9ADF-0C3BE0C1D584}')

set @number = '11200'
insert into @pointblock values (@number,'67161','{190A281F-CD79-4fcc-B12F-9CB32B036FFB}')

set @number = '11201'
insert into @pointblock values (@number,'67162','{98341449-25E1-4b57-B633-25F622B7CEB3}')

set @number = '11202'
insert into @pointblock values (@number,'67271','{3505CB56-A544-4c39-BAC0-1A09927BEC1B}')
insert into @pointblock values (@number,'64006','{3505CB56-A544-4c39-BAC0-1A09927BEC1B}')

set @number = '11204'
insert into @pointblock values (@number,'67261','{93E4AFF3-8BF3-4898-A4CB-A0144596EEEE}')
insert into @pointblock values (@number,'67262','{93E4AFF3-8BF3-4898-A4CB-A0144596EEEE}')
insert into @pointblock values (@number,'67462','{93E4AFF3-8BF3-4898-A4CB-A0144596EEEE}')
```

```
set @number = '11208'
insert into @pointblock values (@number,'88318','{9271A410-EF25-42bb-9EC7-8BBC54D36168}')

set @number = '11301'
insert into @pointblock values (@number,'11712','{0406B0C9-F153-44be-80B9-24BCD4BB4E5B}')

set @number = '11302'
insert into @pointblock values (@number,'67241','{646E2CD8-C576-4e4f-8485-A950BE2D79E2}')

set @number = '11305'
insert into @pointblock values (@number,'88310','{C5D1E883-62E4-4ea0-9ADA-38B2819FBE69}')

set @number = '11306'
insert into @pointblock values (@number,'67445','{8DC91C13-6C6C-4c23-965C-443BBC4C5E40}')

set @number = '11310'
insert into @pointblock values (@number,'64285','{1CCED0BE-70CF-48b0-A391-4F9A3C3B0C80}')
insert into @pointblock values (@number,'64480','{1CCED0BE-70CF-48b0-A391-4F9A3C3B0C80}')

set @number = '11411'
insert into @pointblock values (@number,'11743','{50898AE0-C513-4944-9FE7-50F5FFCB88B5}')

set @number = '11421'
insert into @pointblock values (@number,'56432','{D9EA441E-3E68-404c-BB1C-D49DD6862190}')

set @number = '11423'
insert into @pointblock values (@number,'56204','{AAF0339B-C1AE-4d36-A021-336EAABD5E9E}')

set @number = '11503'
insert into @pointblock values (@number,'59204','{713B33DE-538D-4c88-AC71-6F17466B83C1}')

set @number = '11521'
insert into @pointblock values (@number,'59331','{8F4CF138-8DC3-44ee-9103-FC1814841629}')

set @number = '11532'
insert into @pointblock values (@number,'59435','{F34C4F88-6576-4571-9CDC-E2376363A523}')
insert into @pointblock values (@number,'59335','{F34C4F88-6576-4571-9CDC-E2376363A523}')

set @number = '11601'
insert into @pointblock values (@number,'59341','{2D6B59D3-B0F7-476e-97C2-BD0CA9ED717C}')
insert into @pointblock values (@number,'64300','{2D6B59D3-B0F7-476e-97C2-BD0CA9ED717C}')
insert into @pointblock values (@number,'64302','{2D6B59D3-B0F7-476e-97C2-BD0CA9ED717C}')

set @number = '11690'
insert into @pointblock values (@number,'56800','{8CF008E2-C6D3-448d-97D2-C2DCBDC51917}')
insert into @pointblock values (@number,'57800','{8CF008E2-C6D3-448d-97D2-C2DCBDC51917}')
insert into @pointblock values (@number,'59800','{8CF008E2-C6D3-448d-97D2-C2DCBDC51917}')
insert into @pointblock values (@number,'63800','{8CF008E2-C6D3-448d-97D2-C2DCBDC51917}')
insert into @pointblock values (@number,'64800','{8CF008E2-C6D3-448d-97D2-C2DCBDC51917}')
insert into @pointblock values (@number,'67800','{8CF008E2-C6D3-448d-97D2-C2DCBDC51917}')

set @number = '11713'
insert into @pointblock values (@number,'11301','{97A9CDC3-2962-45b1-940E-5255E3463C56}')

set @number = '11714'
insert into @pointblock values (@number,'93251','{05B8BFAE-9ABA-4e77-B55A-A4078D655582}')
insert into @pointblock values (@number,'93161','{05B8BFAE-9ABA-4e77-B55A-A4078D655582}')
insert into @pointblock values (@number,'11711','{05B8BFAE-9ABA-4e77-B55A-A4078D655582}')

set @number = '11715'
insert into @pointblock values (@number,'93151','{B0F0BEC7-4F8B-4e1b-BB73-EAF454E8FDE3}')
insert into @pointblock values (@number,'11711','{B0F0BEC7-4F8B-4e1b-BB73-EAF454E8FDE3}')

set @number = '11722'
insert into @pointblock values (@number,'93152','{6F08D252-5F94-4086-8B5F-72B2A89E2B0E}')
insert into @pointblock values (@number,'11711','{6F08D252-5F94-4086-8B5F-72B2A89E2B0E}')
insert into @pointblock values (@number,'11721','{6F08D252-5F94-4086-8B5F-72B2A89E2B0E}')

set @number = '11743'
insert into @pointblock values (@number,'93603','{9FB2FD01-023C-4baf-923E-1D0B3A265F41}')
insert into @pointblock values (@number,'11411','{9FB2FD01-023C-4baf-923E-1D0B3A265F41}')

set @number = '11802'
insert into @pointblock values (@number,'11711','{7E343D0B-53E4-419d-8083-61C6EE60DE84}')

set @number = '12102'
insert into @pointblock values (@number,'63130','{B425334E-2B8A-4fa0-B2B2-E3DCB977BA2B}')
-- and
insert into @pointblock values (@number,'12100','{BDBE2DDB-74B9-45c5-83F0-2DDC78955E5B}')

set @number = '12103'
insert into @pointblock values (@number,'12100','{85A0BDC9-420B-43a5-A8BE-445136979A46}')
-- and
insert into @pointblock values (@number,'12102','{E21A87D6-5443-4505-A401-D26C6F2E2E24}')

set @number = '12122'
insert into @pointblock values (@number,'12120','{55D9B129-F727-4cb5-8EA6-B7C8177813D5}')

set @number = '12131'
insert into @pointblock values (@number,'63340','{FDCDFB2F-7F40-4434-BDA3-03ABF22D5FC1}')

set @number = '12133'
insert into @pointblock values (@number,'91779','{58E40308-7BC5-4524-8396-2A3FECCC53A2}')
insert into @pointblock values (@number,'30261','{58E40308-7BC5-4524-8396-2A3FECCC53A2}')

set @number = '12140'
insert into @pointblock values (@number,'63421','{87BB5126-49DF-4c8c-805C-796FF4641E0E}')

set @number = '12200'
```

```
insert into @pointblock values (@number,'63001','{B5CA4B45-6860-48f8-855A-B8E48E6E95A9}')
-- and
insert into @pointblock values (@number,'56059','{A3F50D55-F66A-4bc1-B84F-47EB66E6349D}')

set @number = '12201'
insert into @pointblock values (@number,'64090','{B8ABCC08-2F26-4486-B182-B4C85ED6005E}')

set @number = '12210'
insert into @pointblock values (@number,'63107','{BD9E13B2-09D0-44b7-AFC5-13982E906F1B}')

set @number = '12225'
insert into @pointblock values (@number,'63280','{72982819-7C8C-4c65-901E-EDCC1247C4EC}')
-- and
insert into @pointblock values (@number,'12220','{53DACB56-AF09-4d70-A619-BCA09407FB6F}')

set @number = '12230'
insert into @pointblock values (@number,'63335','{97D53B10-FEA9-4ecf-8E22-44B790B0AE78}')
-- and
insert into @pointblock values (@number,'63235','{4615A7AC-1277-4054-BE8F-E5C6199E9A13}')
-- and
insert into @pointblock values (@number,'63336','{B76DEBCB-6AEE-43c1-8BBF-6431B19164FE}')

set @number = '12233'
insert into @pointblock values (@number,'12232','{B43C1AB4-1928-415e-85CD-091BB4DD1EB0}')
-- and
insert into @pointblock values (@number,'63350','{3AB08823-6E5A-4180-8A1A-4DB8C2A6DE89}')

set @number = '12240'
insert into @pointblock values (@number,'63480','{AC10CF43-99DE-43c2-9085-EB8E80B3ECD9}')

set @number = '12243'
insert into @pointblock values (@number,'63409','{E7D9E6CB-9280-44d7-84D8-601090EE6381}')

set @number = '12300'
insert into @pointblock values (@number,'57002','{78BF7B34-D3F3-4411-B2A8-B57CF0052A9D}')

set @number = '12320'
insert into @pointblock values (@number,'57131','{5F385863-703F-407b-AE2D-98D14F2C3B39}')

set @number = '12330'
insert into @pointblock values (@number,'63370','{7EAE4066-9CB5-4b74-B9C9-737EDBBCAE8B}')
-- and
insert into @pointblock values (@number,'63365','{A965395D-BA52-44d3-81A4-4556A7B12E12}')

set @number = '12331'
insert into @pointblock values (@number,'63372','{19377E01-24B8-43bf-B35B-DADB8F0EB3A6}')

set @number = '12332'
insert into @pointblock values (@number,'56230','{ECFD8DBF-FF50-43e0-B0FD-9D8881EB22BF}')

set @number = '12400'
insert into @pointblock values (@number,'5602','{E040417B-E7C6-40f7-9559-C0FAC212320E}')
-- and
insert into @pointblock values (@number,'11401','{A337F357-9D42-4073-B5AF-0308A01D755B}')

set @number = '12401'
insert into @pointblock values (@number,'C5652','{CB82F312-6527-438d-8D7B-27B910F0A746}')
-- and
insert into @pointblock values (@number,'56452','{AE8C3FF9-4B17-4c6e-9699-FAA36E78EE7B}')

set @number = '12423'
insert into @pointblock values (@number,'12422','{D6756E6F-05AA-4edf-B420-939BC4169C3F}')

set @number = '12700'
insert into @pointblock values (@number,'93352','{86D94FB1-4BC8-405b-9158-3B95CDFFBEC5}')
insert into @pointblock values (@number,'93552','{86D94FB1-4BC8-405b-9158-3B95CDFFBEC5}')

set @number = '13000'
insert into @pointblock values (@number,'67122','{FFB06B29-A6C3-4192-BE77-61E548461837}')

set @number = '13110'
insert into @pointblock values (@number,'6721','{77565E74-C640-4990-BDB7-0E61ABAB547A}')
insert into @pointblock values (@number,'67231','{77565E74-C640-4990-BDB7-0E61ABAB547A}')

set @number = '13120'
insert into @pointblock values (@number,'6723','{EBA3EF96-4386-4ae4-B16C-3091B55C6271}')

set @number = '13210'
insert into @pointblock values (@number,'67232','{87063DA0-A6FF-4188-80AF-A4F7872966E7}')

set @number = '13320'
insert into @pointblock values (@number,'67223','{C06BFDF3-107F-497e-8B62-5EAC311E8FC7}')

set @number = '26000'
insert into @pointblock values (@number,'10004','{2E9F05E9-5E27-4122-8377-24FBDE2E6B9B}')
insert into @pointblock values (@number,'10005','{2E9F05E9-5E27-4122-8377-24FBDE2E6B9B}')
insert into @pointblock values (@number,'21010','{2E9F05E9-5E27-4122-8377-24FBDE2E6B9B}')
insert into @pointblock values (@number,'21001','{2E9F05E9-5E27-4122-8377-24FBDE2E6B9B}')

set @number = '26001'
insert into @pointblock values (@number,'21000','{F472FFD7-09B0-4a5a-8749-9217FFC87641}')
insert into @pointblock values (@number,'21210','{F472FFD7-09B0-4a5a-8749-9217FFC87641}')
insert into @pointblock values (@number,'21211','{F472FFD7-09B0-4a5a-8749-9217FFC87641}')
insert into @pointblock values (@number,'21061','{F472FFD7-09B0-4a5a-8749-9217FFC87641}')
insert into @pointblock values (@number,'23110','{F472FFD7-09B0-4a5a-8749-9217FFC87641}')
insert into @pointblock values (@number,'23111','{F472FFD7-09B0-4a5a-8749-9217FFC87641}')

set @number = '26004'
insert into @pointblock values (@number,'10004','{FF6CB7BC-0ECF-4037-8C90-A15A8887E87A}')
```

```
insert into @pointblock values (@number,'10005','{FF6CB7BC-0ECF-4037-8C90-A15A8887E87A}')
insert into @pointblock values (@number,'21001','{FF6CB7BC-0ECF-4037-8C90-A15A8887E87A}')
insert into @pointblock values (@number,'21010','{FF6CB7BC-0ECF-4037-8C90-A15A8887E87A}')
insert into @pointblock values (@number,'21210','{FF6CB7BC-0ECF-4037-8C90-A15A8887E87A}')
insert into @pointblock values (@number,'21211','{FF6CB7BC-0ECF-4037-8C90-A15A8887E87A}')
insert into @pointblock values (@number,'26000','{FF6CB7BC-0ECF-4037-8C90-A15A8887E87A}')
insert into @pointblock values (@number,'26001','{FF6CB7BC-0ECF-4037-8C90-A15A8887E87A}')

set @number = '26010'
insert into @pointblock values (@number,'21103','{5F53D18C-8939-4f4d-B2D1-E9BB34AC5ABC}')

set @number = '26025'
insert into @pointblock values (@number,'21000','{D7BAFF6D-C2F4-45cc-84A1-2876144630B6}')
insert into @pointblock values (@number,'21001','{D7BAFF6D-C2F4-45cc-84A1-2876144630B6}')
insert into @pointblock values (@number,'26000','{D7BAFF6D-C2F4-45cc-84A1-2876144630B6}')
insert into @pointblock values (@number,'26001','{D7BAFF6D-C2F4-45cc-84A1-2876144630B6}')
insert into @pointblock values (@number,'21010','{D7BAFF6D-C2F4-45cc-84A1-2876144630B6}')
insert into @pointblock values (@number,'26026','{D7BAFF6D-C2F4-45cc-84A1-2876144630B6}')
insert into @pointblock values (@number,'10004','{D7BAFF6D-C2F4-45cc-84A1-2876144630B6}')
insert into @pointblock values (@number,'10005','{D7BAFF6D-C2F4-45cc-84A1-2876144630B6}')

set @number = '26026'
insert into @pointblock values (@number,'10004','{38491478-67B6-4472-80A3-9DCAB9DA6FCD}')
insert into @pointblock values (@number,'10005','{38491478-67B6-4472-80A3-9DCAB9DA6FCD}')
insert into @pointblock values (@number,'26000','{38491478-67B6-4472-80A3-9DCAB9DA6FCD}')
insert into @pointblock values (@number,'21000','{38491478-67B6-4472-80A3-9DCAB9DA6FCD}')
insert into @pointblock values (@number,'21001','{38491478-67B6-4472-80A3-9DCAB9DA6FCD}')

set @number = '26120'
insert into @pointblock values (@number,'21220','{1C8070F5-D04A-4132-9511-9E2F06BA8607}')

set @number = '26122'
insert into @pointblock values (@number,'21230','{38C4C621-C9E3-4055-A293-B8815D14269E}')

set @number = '26124'
insert into @pointblock values (@number,'21215','{31069C68-C922-4824-A758-0573928C3A05}')

set @number = '26126'
insert into @pointblock values (@number,'21235','{09B46866-E5FE-4aff-8281-8081CB3BFCCD}')

set @number = '26128'
insert into @pointblock values (@number,'21370','{24918100-097F-4351-BD95-2A31EF26DF29}')

set @number = '26140'
insert into @pointblock values (@number,'21380','{ACDA6311-6840-485d-87D4-1FD9CAB43A83}')

set @number = '26142'
insert into @pointblock values (@number,'21382','{CFDFCC74-4266-4247-90D7-02185AB4AB85}')

set @number = '26170'
insert into @pointblock values (@number,'91315','{48926B77-D217-43e7-8164-15BC9440F032}')
-- and
insert into @pointblock values (@number,'91311','{F5B18681-9EDA-4bc2-91CF-EC399A49ED2B}')

set @number = '26172'
insert into @pointblock values (@number,'91321','{189968D0-E091-4db9-BAA6-50DEC50E58D5}')
-- and
insert into @pointblock values (@number,'91329','{D97D5769-292D-4397-AB12-6EC1B640E12F}')

set @number = '26201'
insert into @pointblock values (@number,'10005','{722A7CEA-8D15-461f-AD16-929B60F2F5F2}')
insert into @pointblock values (@number,'10012','{722A7CEA-8D15-461f-AD16-929B60F2F5F2}')
insert into @pointblock values (@number,'21061','{722A7CEA-8D15-461f-AD16-929B60F2F5F2}')

set @number = '26222'
insert into @pointblock values (@number,'26220','{3E1C5C53-6528-406e-8C4C-3FE7CD0E18D3}')
insert into @pointblock values (@number,'28220','{3E1C5C53-6528-406e-8C4C-3FE7CD0E18D3}')
insert into @pointblock values (@number,'26225','{3E1C5C53-6528-406e-8C4C-3FE7CD0E18D3}')

set @number = '26233'
insert into @pointblock values (@number,'21262','{58FF0B34-9477-4601-A72F-5B8A34E74E7D}')
insert into @pointblock values (@number,'21263','{58FF0B34-9477-4601-A72F-5B8A34E74E7D}')
insert into @pointblock values (@number,'26230','{58FF0B34-9477-4601-A72F-5B8A34E74E7D}')
insert into @pointblock values (@number,'26225','{58FF0B34-9477-4601-A72F-5B8A34E74E7D}')

set @number = '26240'
insert into @pointblock values (@number,'21490','{B48F31C2-E24E-444b-9729-81950C5DC7E9}')

set @number = '26250'
insert into @pointblock values (@number,'21550','{1AD92E38-7823-428e-91CA-7399E846B024}')

set @number = '26270'
insert into @pointblock values (@number,'91331','{C4635855-94C2-45fd-8449-089B4A9C8D6F}')
-- and
insert into @pointblock values (@number,'91325','{0FC25D3B-E451-4f8c-B206-AEBD438B382C}')

set @number = '26272'
insert into @pointblock values (@number,'91341','{1C6C9B47-2F19-4e51-9695-4293DFCFB47C}')
-- and
insert into @pointblock values (@number,'91333','{0119F42D-CA68-49f4-9196-0680E2E218BB}')

set @number = '26300'
insert into @pointblock values (@number,'8859','{A1ADB0A9-2138-4ac9-8AE5-3B2812686247}')
insert into @pointblock values (@number,'21245','{A1ADB0A9-2138-4ac9-8AE5-3B2812686247}')
insert into @pointblock values (@number,'26300','{A1ADB0A9-2138-4ac9-8AE5-3B2812686247}')

set @number = '26301'
insert into @pointblock values (@number,'21245','{B0CCDF0A-84E1-456c-A835-5C425510F0D9}')
insert into @pointblock values (@number,'26300','{B0CCDF0A-84E1-456c-A835-5C425510F0D9}')
```

```
insert into @pointblock values (@number,'26300','{B0CCDF0A-84E1-456c-A835-5C425510F0D9}')

set @number = '26310'
insert into @pointblock values (@number,'2140','{5F618F6D-4294-460b-BACD-522919BDA798}')
insert into @pointblock values (@number,'21250','{5F618F6D-4294-460b-BACD-522919BDA798}')

set @number = '26312'
insert into @pointblock values (@number,'2146','{4A625C61-6360-4ff2-9BC2-68F00B31963F}')
insert into @pointblock values (@number,'21455','{4A625C61-6360-4ff2-9BC2-68F00B31963F}')

set @number = '26322'
insert into @pointblock values (@number,'21330','{39A3A62D-711C-4a1b-8F09-DEF51428DD8E}')

set @number = '26330'
insert into @pointblock values (@number,'21335','{C9CA8DC1-6B6B-471b-9B83-2692F3942895}')

set @number = '26334'
insert into @pointblock values (@number,'23352','{82DEF42F-C011-40fc-9F89-754E7BA9A11A}')
insert into @pointblock values (@number,'23456','{82DEF42F-C011-40fc-9F89-754E7BA9A11A}')

set @number = '26370'
insert into @pointblock values (@number,'91312','{7BCF838C-447E-42f0-B34B-60A25638C848}')

set @number = '26372'
insert into @pointblock values (@number,'91326','{8C1B79F0-5D7B-4984-92F2-0930FAB4A99F}')
-- and
insert into @pointblock values (@number,'91316','{938169D0-45D2-4ee6-9031-C980461B3B82}')

set @number = '26376'
insert into @pointblock values (@number,'91316','{1F5B0D14-C7F5-45d9-9CCC-D8DD565159B8}')
-- and
insert into @pointblock values (@number,'91326','{3DE0F4CF-7A39-4571-8F0C-4D80C7A6C178}')
-- and
insert into @pointblock values (@number,'26372','{60184576-B7C8-4adb-A35D-81E421BA5193}')

set @number = '26378'
insert into @pointblock values (@number,'91327','{FDED7311-9EDC-4326-8D38-F1977949436F}')

set @number = '26400'
insert into @pointblock values (@number,'26001','{ACDE0D4A-F42F-4158-9AA4-369A3B681FDF}')
insert into @pointblock values (@number,'26401','{ACDE0D4A-F42F-4158-9AA4-369A3B681FDF}')
insert into @pointblock values (@number,'26470','{ACDE0D4A-F42F-4158-9AA4-369A3B681FDF}')
insert into @pointblock values (@number,'23110','{ACDE0D4A-F42F-4158-9AA4-369A3B681FDF}')
insert into @pointblock values (@number,'23111','{ACDE0D4A-F42F-4158-9AA4-369A3B681FDF}')

set @number = '26401'
insert into @pointblock values (@number,'26001','{55CE7604-C5FD-42d8-88F0-FFAA83EDCF4B}')
insert into @pointblock values (@number,'26400','{55CE7604-C5FD-42d8-88F0-FFAA83EDCF4B}')
insert into @pointblock values (@number,'26470','{55CE7604-C5FD-42d8-88F0-FFAA83EDCF4B}')
insert into @pointblock values (@number,'23111','{55CE7604-C5FD-42d8-88F0-FFAA83EDCF4B}')
insert into @pointblock values (@number,'23110','{55CE7604-C5FD-42d8-88F0-FFAA83EDCF4B}')

set @number = '26406'
insert into @pointblock values (@number,'26472','{0E9107E3-2622-4dae-A8DF-C83097A8AAD0}')
insert into @pointblock values (@number,'23117','{0E9107E3-2622-4dae-A8DF-C83097A8AAD0}')

set @number = '26416'
insert into @pointblock values (@number,'23225','{50937A0A-1D8A-4552-97C4-53A01DF78CE5}')

set @number = '26418'
insert into @pointblock values (@number,'23227','{E8670604-7A84-44f7-9419-CC84247041BE}')

set @number = '26470'
insert into @pointblock values (@number,'26400','{AEE6A7CE-4EB1-4f44-BDE5-E7614D1DF349}')
-- and
insert into @pointblock values (@number,'26401','{6C3724BC-73F5-4379-9349-C9AEC6819E03}')

set @number = '26472'
insert into @pointblock values (@number,'26406','{BB1D2ADB-039C-4f8f-A8CA-209D753A4A37}')

set @number = '26500'
insert into @pointblock values (@number,'77142','{334A162A-D433-4085-ACAB-DF04BB6C8A2C}')
insert into @pointblock values (@number,'41861','{334A162A-D433-4085-ACAB-DF04BB6C8A2C}')
insert into @pointblock values (@number,'02197','{334A162A-D433-4085-ACAB-DF04BB6C8A2C}')
insert into @pointblock values (@number,'02199','{334A162A-D433-4085-ACAB-DF04BB6C8A2C}')

set @number = '27000'
insert into @pointblock values (@number,'24001','{8AB13D58-4D26-43f9-8E34-3F7C85C42920}')
insert into @pointblock values (@number,'25002','{8AB13D58-4D26-43f9-8E34-3F7C85C42920}')
insert into @pointblock values (@number,'27001','{8AB13D58-4D26-43f9-8E34-3F7C85C42920}')
insert into @pointblock values (@number,'30000','{8AB13D58-4D26-43f9-8E34-3F7C85C42920}')
insert into @pointblock values (@number,'30001','{8AB13D58-4D26-43f9-8E34-3F7C85C42920}')

set @number = '27001'
insert into @pointblock values (@number,'24001','{C2938490-49E9-4900-8B04-D214F8B0EC19}')
insert into @pointblock values (@number,'25002','{C2938490-49E9-4900-8B04-D214F8B0EC19}')
insert into @pointblock values (@number,'27000','{C2938490-49E9-4900-8B04-D214F8B0EC19}')
insert into @pointblock values (@number,'30000','{C2938490-49E9-4900-8B04-D214F8B0EC19}')
insert into @pointblock values (@number,'30001','{C2938490-49E9-4900-8B04-D214F8B0EC19}')

set @number = '27011'
insert into @pointblock values (@number,'30431','{4CF30BD9-EBC4-4dbc-83D9-1915AAA3154C}')

set @number = '27014'
insert into @pointblock values (@number,'27803','{4001C6C1-37B2-4904-BF7B-9EB3478E3FB2}')

set @number = '27021'
insert into @pointblock values (@number,'24101','{0C077383-92F1-47f7-BD8A-39AF6C69D1A8}')
insert into @pointblock values (@number,'24102','{0C077383-92F1-47f7-BD8A-39AF6C69D1A8}')
```

```
insert into @pointblock values (@number,'25111','{0C077383-92F1-47f7-BD8A-39AF6C69D1A8}')
insert into @pointblock values (@number,'25151','{0C077383-92F1-47f7-BD8A-39AF6C69D1A8}')

set @number = '27031'
insert into @pointblock values (@number,'27961','{BEE47EAA-BCF3-4df1-B7D2-8CA7803DE1C0}')

set @number = '27033'
insert into @pointblock values (@number,'27031','{65612A52-0EA6-404f-9760-8B5928792095}')

set @number = '27231'
insert into @pointblock values (@number,'27263','{9B7FE617-0AE2-4fa5-8CC4-78D973D6E027}')
insert into @pointblock values (@number,'24210','{9B7FE617-0AE2-4fa5-8CC4-78D973D6E027}')

set @number = '27232'
insert into @pointblock values (@number,'27264','{02234005-BCDC-4a29-B402-ABCFF4067B01}')
insert into @pointblock values (@number,'27233','{02234005-BCDC-4a29-B402-ABCFF4067B01}')
insert into @pointblock values (@number,'27264','{02234005-BCDC-4a29-B402-ABCFF4067B01}')

set @number = '27233'
insert into @pointblock values (@number,'27242','{62759108-3E74-4149-AD62-DD502171BA67}')
insert into @pointblock values (@number,'27232','{62759108-3E74-4149-AD62-DD502171BA67}')

set @number = '27253'
insert into @pointblock values (@number,'27221','{FCF45250-0389-418c-97E6-FBBAAF992889}')
insert into @pointblock values (@number,'27251','{FCF45250-0389-418c-97E6-FBBAAF992889}')

set @number = '27254'
insert into @pointblock values (@number,'24252','{47E311E9-9021-4652-8425-CCC78EC2F064}')

set @number = '27262'
insert into @pointblock values (@number,'30462','{BA9A0C2A-D1DA-405b-A06E-49845D99CD52}')

set @number = '27301'
insert into @pointblock values (@number,'25221','{E1CBB92E-EB00-41d3-8AFA-91BBAD8E6E05}')
insert into @pointblock values (@number,'27381','{E1CBB92E-EB00-41d3-8AFA-91BBAD8E6E05}')

set @number = '27302'
insert into @pointblock values (@number,'25261','{18A1042F-F5ED-4e4e-B17B-04FBF4D51505}')

set @number = '27311'
insert into @pointblock values (@number,'25241','{EFC27357-96E8-4a66-8CFB-154B98228348}')

set @number = '27321'
insert into @pointblock values (@number,'25242','{E3254AE8-68B8-47bf-8DC8-ADE19211A643}')

set @number = '27403'
insert into @pointblock values (@number,'30203','{205C0F89-47DB-4dca-8E68-FCD3C13E9F54}')

set @number = '27404'
insert into @pointblock values (@number,'30204','{B8F2DA86-EAE0-4f16-97A1-3BB5C2D448F9}')

set @number = '27405'
insert into @pointblock values (@number,'30205','{E0261482-1448-4413-9588-DA6DCF8EE6B5}')

set @number = '27406'
insert into @pointblock values (@number,'30206','{08EDFDA1-A641-48c5-957F-CFF1ED0642A3}')

set @number = '27407'
insert into @pointblock values (@number,'27550','{AC2BA189-1B82-4479-876D-BD15FA206169}')

set @number = '27442'
insert into @pointblock values (@number,'30242','{222DFEFA-CCC1-47fe-AD3A-77E87F71C1F2}')

set @number = '27443'
insert into @pointblock values (@number,'30416','{213C757D-D85B-434d-8FB0-15378F93E88F}')

set @number = '27500'
insert into @pointblock values (@number,'30221','{584D0ADE-C80B-4a4b-A17D-EEB9DC6B197E}')
insert into @pointblock values (@number,'25232','{584D0ADE-C80B-4a4b-A17D-EEB9DC6B197E}')

set @number = '27501'
insert into @pointblock values (@number,'30251','{8BDFC03F-FE84-4eaf-B9DB-EDE720202EF0}')

set @number = '27502'
insert into @pointblock values (@number,'30252','{FB21A6A0-352B-4fc1-BCC2-115824B5B5B8}')

set @number = '27505'
insert into @pointblock values (@number,'30202','{C0192915-50BB-4646-BCF4-8DD2AF73E29A}')

set @number = '27511'
insert into @pointblock values (@number,'30441','{4B6110C1-47FF-4335-A4E5-3A07FA8B6EE4}')

set @number = '27520'
insert into @pointblock values (@number,'30226','{C3B9E114-9A61-4477-89FA-6DD9CE688810}')
insert into @pointblock values (@number,'30227','{85C8736D-17F3-4f27-B8E8-270D9E958A31}')

set @number = '27521'
insert into @pointblock values (@number,'27580','{F64887F2-A446-4ecf-92EF-83F908F81804}')

set @number = '27552'
insert into @pointblock values (@number,'30420','{AD864D84-5352-4889-8793-668D1AEE869F}')

set @number = '27700'
insert into @pointblock values (@number,'91923','{1F4068A1-02CA-49fb-A9B4-E10837A9FB3A}')
insert into @pointblock values (@number,'91924','{1F4068A1-02CA-49fb-A9B4-E10837A9FB3A}')

set @number = '27710'
insert into @pointblock values (@number,'91533','{46CB392B-45E8-4dcd-BDDD-F064E2CB18EC}')
```

```
set @number = '27711'
insert into @pointblock values (@number,'30771','{3350BB31-54F5-410c-A5C1-FFBBA58AA6F9}')
insert into @pointblock values (@number,'30772','{3350BB31-54F5-410c-A5C1-FFBBA58AA6F9}')
insert into @pointblock values (@number,'30773','{3350BB31-54F5-410c-A5C1-FFBBA58AA6F9}')

set @number = '27721'
insert into @pointblock values (@number,'24771','{437BEF3F-FCB1-41e9-8E8C-2B03F7B04A33}')
insert into @pointblock values (@number,'24772','{437BEF3F-FCB1-41e9-8E8C-2B03F7B04A33}')
insert into @pointblock values (@number,'24773','{437BEF3F-FCB1-41e9-8E8C-2B03F7B04A33}')
insert into @pointblock values (@number,'24101','{437BEF3F-FCB1-41e9-8E8C-2B03F7B04A33}')
insert into @pointblock values (@number,'24001','{437BEF3F-FCB1-41e9-8E8C-2B03F7B04A33}')

set @number = '27722'
insert into @pointblock values (@number,'24783','{4F0798F1-B293-43ce-80D7-3D0C41CB34DD}')
insert into @pointblock values (@number,'24210','{4F0798F1-B293-43ce-80D7-3D0C41CB34DD}')
insert into @pointblock values (@number,'27231','{4F0798F1-B293-43ce-80D7-3D0C41CB34DD}')

set @number = '27731'
insert into @pointblock values (@number,'25771','{1406F951-3424-44de-A648-694BC9A26CEE}')

set @number = '27732'
insert into @pointblock values (@number,'25773','{019C68BF-7085-47e2-8062-EB3FA351E3F0}')

set @number = '27733'
insert into @pointblock values (@number,'25780','{74A8C1C7-7247-4ded-AD4D-F1403AAA0858}')

set @number = '27751'
insert into @pointblock values (@number,'30787','{72750FEF-0A40-4e9b-A779-159DE655BBDD}')

set @number = '27752'
insert into @pointblock values (@number,'30215','{81A69336-95FF-46a8-9AFE-DC072FA597E6}')

set @number = '28010'
insert into @pointblock values (@number,'36001','{9E15AB74-3ED3-4b33-8D96-09186DA24903}')

set @number = '28120'
insert into @pointblock values (@number,'36100','{8D97DE9D-81EF-4a8a-A587-276A433A3976}')

set @number = '28121'
insert into @pointblock values (@number,'36102','{AE1C8A13-CD2D-4250-9827-78CCF5471279}')
insert into @pointblock values (@number,'28153','{AE1C8A13-CD2D-4250-9827-78CCF5471279}')
insert into @pointblock values (@number,'28155','{AE1C8A13-CD2D-4250-9827-78CCF5471279}')

set @number = '28210'
insert into @pointblock values (@number,'26220','{183236C3-1F00-4c30-A84B-9FAA4BC4E932}')
insert into @pointblock values (@number,'26225','{183236C3-1F00-4c30-A84B-9FAA4BC4E932}')

set @number = '28213'
insert into @pointblock values (@number,'C3691','{9E6C1830-C65A-4cd1-B1AB-FF3C07D6346F}')
insert into @pointblock values (@number,'91775','{9E6C1830-C65A-4cd1-B1AB-FF3C07D6346F}')
insert into @pointblock values (@number,'36270','{9E6C1830-C65A-4cd1-B1AB-FF3C07D6346F}')

set @number = '28221'
insert into @pointblock values (@number,'28220','{89281127-AF11-4b45-9F57-907B43D19C4C}')
insert into @pointblock values (@number,'26220','{89281127-AF11-4b45-9F57-907B43D19C4C}')
insert into @pointblock values (@number,'28321','{89281127-AF11-4b45-9F57-907B43D19C4C}')

set @number = '28240'
insert into @pointblock values (@number,'36221','{CD4F0A65-E484-46a4-B0B5-3B1ED0BEA263}')
insert into @pointblock values (@number,'28341','{CD4F0A65-E484-46a4-B0B5-3B1ED0BEA263}')

set @number = '28252'
insert into @pointblock values (@number,'C3633','{E46C30B6-430A-4d0e-9D41-5A7CFED6FFD1}')
insert into @pointblock values (@number,'C3635','{E46C30B6-430A-4d0e-9D41-5A7CFED6FFD1}')
insert into @pointblock values (@number,'36231','{E46C30B6-430A-4d0e-9D41-5A7CFED6FFD1}')
insert into @pointblock values (@number,'36435','{E46C30B6-430A-4d0e-9D41-5A7CFED6FFD1}')

set @number = '28260'
insert into @pointblock values (@number,'28261','{CC07BC1F-783D-4664-AA6A-AC6CC943A91A}')
insert into @pointblock values (@number,'01030','{CC07BC1F-783D-4664-AA6A-AC6CC943A91A}')

set @number = '28315'
insert into @pointblock values (@number,'91765','{B5111C16-6658-48f3-8B7A-BF62F4C5C20D}')

set @number = '28316'
insert into @pointblock values (@number,'91766','{4560E1FD-F41D-4147-B25D-DE2D4574DB00}')

set @number = '28320'
insert into @pointblock values (@number,'36211','{5BDE5C0D-A5BC-4e39-982C-CFF163819B23}')

set @number = '28321'
insert into @pointblock values (@number,'28220','{85772C8E-CF45-4171-A1FD-7D5EFA79AE38}')

set @number = '28325'
insert into @pointblock values (@number,'36451','{A214E215-43B1-4e5b-8C28-9BB1E2A998E0}')
insert into @pointblock values (@number,'28423','{A214E215-43B1-4e5b-8C28-9BB1E2A998E0}')

set @number = '28341'
insert into @pointblock values (@number,'28240','{0A1B7952-5BBE-4339-AB13-C96C3EA3BB84}')

set @number = '28351'
insert into @pointblock values (@number,'36230','{50D5A1D3-4B94-4bb8-8583-C76960EA6BAC}')
insert into @pointblock values (@number,'50300','{50D5A1D3-4B94-4bb8-8583-C76960EA6BAC}')
insert into @pointblock values (@number,'72231','{50D5A1D3-4B94-4bb8-8583-C76960EA6BAC}')

set @number = '28414'
insert into @pointblock values (@number,'36414','{249F39D2-D413-45eb-9B69-4AE6F268845E}')
insert into @pointblock values (@number,'36214','{249F39D2-D413-45eb-9B69-4AE6F268845E}')
```

```
set @number = '28434'
insert into @pointblock values (@number,'36415','{FD281298-28BC-4407-A840-D8A575AD88E9}')

set @number = '28463'
insert into @pointblock values (@number,'36462','{C72F7A12-FA57-4225-A75C-27017F877C85}')

set @number = '28835'
insert into @pointblock values (@number,'91721','{256A9B18-2DB1-46d5-8D47-05C439377E87}')
insert into @pointblock values (@number,'28021','{256A9B18-2DB1-46d5-8D47-05C439377E87}')

set @number = '28845'
insert into @pointblock values (@number,'28240','{D784D9C9-C230-445b-BD7E-B636FD2FB941}')

set @number = '28852'
insert into @pointblock values (@number,'36441','{D2C22BD8-9E72-49ea-BED9-381E24767B5C}')

set @number = '28861'
insert into @pointblock values (@number,'28260','{CA4A0AE3-C66A-46f7-B50E-29069B545FB6}')
insert into @pointblock values (@number,'36260','{CA4A0AE3-C66A-46f7-B50E-29069B545FB6}')
insert into @pointblock values (@number,'36261','{CA4A0AE3-C66A-46f7-B50E-29069B545FB6}')

set @number = '28863'
insert into @pointblock values (@number,'36263','{4DF4C697-BE5C-493a-982D-A5304FEE805A}')

set @number = '28875'
insert into @pointblock values (@number,'91974','{3787BCAA-9394-4b76-8315-78359831D7A9}')

set @number = '31000'
insert into @pointblock values (@number,'31600','{765EE412-A0AC-4f51-84F9-0C84C750493B}')

set @number = '31011'
insert into @pointblock values (@number,'34300','{7C14F53A-9AB1-404e-80B5-6501EDC345B5}')

set @number = '31021'
insert into @pointblock values (@number,'92021','{2D14CC6E-3DC1-48a7-B866-DC3996BE617C}')

set @number = '31024'
insert into @pointblock values (@number,'92013','{FD11D979-1036-4cf5-B9EC-98C3ACF8507A}')

set @number = '31027'
insert into @pointblock values (@number,'02320','{B473BC3D-219B-46fb-8DE3-F10FFBB9ECBA}')

set @number = '31028'
insert into @pointblock values (@number,'92503','{4BC24C1C-0485-42e1-994F-24CF3DEC6E62}')

set @number = '31029'
insert into @pointblock values (@number,'92061','{A80A979A-518A-4e63-AB94-60F66D480103}')
-- and
insert into @pointblock values (@number,'92062','{0DC567D7-1D84-40a5-87D3-A4FB2BFCF13F}')

set @number = '31030'
insert into @pointblock values (@number,'31606','{3FA6AB5F-C6D4-4f5e-AADB-4981CD2577E2}')

set @number = '31050'
insert into @pointblock values (@number,'50250','{B4CBDC91-303D-4f26-8F10-B5851088DDBC}')

set @number = '31200'
insert into @pointblock values (@number,'51001','{8B4D4227-F292-4de6-BDFE-64ECEDD0E782}')

set @number = '31220'
insert into @pointblock values (@number,'51221','{548A651C-33D4-433e-858C-FDA810F8DB08}')
-- and
insert into @pointblock values (@number,'51226','{DC102887-AA95-4b07-ADFE-57F39C886F4F}')
-- and
insert into @pointblock values (@number,'51420','{F074A351-CB7F-402f-9F94-297DF461B726}')

set @number = '31230'
insert into @pointblock values (@number,'51231','{AC950C3E-7886-4909-B9E3-8CEA31D28C23}')

set @number = '31240'
insert into @pointblock values (@number,'51243','{40E8C59E-53F3-4f32-A41F-BB58B37FA2B1}')
-- and
insert into @pointblock values (@number,'51447','{231A9550-4FC6-4193-8B3D-2FA9C630FA28}')

set @number = '31250'
insert into @pointblock values (@number,'51041','{2C3E78CD-199E-4998-97AE-8F092DA0FBD6}')

set @number = '31260'
insert into @pointblock values (@number,'51222','{4FD631DA-68A9-4e5d-8562-8A575FD5B6CA}')

set @number = '31270'
insert into @pointblock values (@number,'51270','{0371522D-BC29-42f4-825B-733D23137F93}')

set @number = '31300'
insert into @pointblock values (@number,'3630','{E76DA772-C474-4ee2-B8FC-11584A015D45}')
insert into @pointblock values (@number,'36230','{E76DA772-C474-4ee2-B8FC-11584A015D45}')
insert into @pointblock values (@number,'3631','{E76DA772-C474-4ee2-B8FC-11584A015D45}')
insert into @pointblock values (@number,'8201','{E76DA772-C474-4ee2-B8FC-11584A015D45}')
insert into @pointblock values (@number,'72131','{E76DA772-C474-4ee2-B8FC-11584A015D45}')
insert into @pointblock values (@number,'72231','{E76DA772-C474-4ee2-B8FC-11584A015D45}')
insert into @pointblock values (@number,'5352','{E76DA772-C474-4ee2-B8FC-11584A015D45}')
insert into @pointblock values (@number,'5001','{E76DA772-C474-4ee2-B8FC-11584A015D45}')
insert into @pointblock values (@number,'50300','{E76DA772-C474-4ee2-B8FC-11584A015D45}')
insert into @pointblock values (@number,'4851','{E76DA772-C474-4ee2-B8FC-11584A015D45}')
insert into @pointblock values (@number,'92081','{E76DA772-C474-4ee2-B8FC-11584A015D45}')

set @number = '31305'
insert into @pointblock values (@number,'50305','{3E6D0EE0-8128-4247-BA10-56900E8B8225}')
```

```
set @number = '31310'
insert into @pointblock values (@number,'5011','{E752B2FA-64CC-4871-8938-20F3EF329701}')
insert into @pointblock values (@number,'50310','{E752B2FA-64CC-4871-8938-20F3EF329701}')

set @number = '31340'
insert into @pointblock values (@number,'50350','{4853D30E-32FC-47e9-B232-210541F69865}')
-- and
insert into @pointblock values (@number,'50355','{3658107C-8582-4348-B0C0-37AAE1E9FA33}')

set @number = '31350'
insert into @pointblock values (@number,'32020','{062F96B6-66F1-4e2c-88C0-CA1B0C030625}')
insert into @pointblock values (@number,'92200','{062F96B6-66F1-4e2c-88C0-CA1B0C030625}')
insert into @pointblock values (@number,'50280','{062F96B6-66F1-4e2c-88C0-CA1B0C030625}')
insert into @pointblock values (@number,'45120','{062F96B6-66F1-4e2c-88C0-CA1B0C030625}')
-- and
insert into @pointblock values (@number,'53114','{010FBF54-F90D-464d-BF3D-87B37594F00F}')
insert into @pointblock values (@number,'32070','{010FBF54-F90D-464d-BF3D-87B37594F00F}')

set @number = '31352'
insert into @pointblock values (@number,'92506','{8D0FACBE-AB36-487e-A0B5-167A14E52C1E}')
insert into @pointblock values (@number,'92509','{8D0FACBE-AB36-487e-A0B5-167A14E52C1E}')
insert into @pointblock values (@number,'32620','{8D0FACBE-AB36-487e-A0B5-167A14E52C1E}')

set @number = '31353'
insert into @pointblock values (@number,'92907','{1901BCFF-59D5-40bb-950F-41891BD93A70}')
insert into @pointblock values (@number,'92908','{1901BCFF-59D5-40bb-950F-41891BD93A70}')
insert into @pointblock values (@number,'32630','{1901BCFF-59D5-40bb-950F-41891BD93A70}')

set @number = '31356'
insert into @pointblock values (@number,'32640','{B6CBA8A2-7242-4afe-BB91-CBA7CB66F54F}')
-- and
insert into @pointblock values (@number,'53256','{6BCA0E0B-B1DD-41eb-A7B7-43FE3DD9E542}')

set @number = '31358'
insert into @pointblock values (@number,'50282','{801BA45F-2802-458d-B8D9-04375BBAF858}')
insert into @pointblock values (@number,'92909','{801BA45F-2802-458d-B8D9-04375BBAF858}')
insert into @pointblock values (@number,'32670','{801BA45F-2802-458d-B8D9-04375BBAF858}')

set @number = '31361'
insert into @pointblock values (@number,'50370','{0447EF88-A60E-41a0-9BF3-8A82A471EFAB}')

set @number = '31370'
insert into @pointblock values (@number,'4705','{EFD1A67B-5B28-4522-8727-105564C23A3B}')
insert into @pointblock values (@number,'5050','{EFD1A67B-5B28-4522-8727-105564C23A3B}')

set @number = '31390'
insert into @pointblock values (@number,'50390','{AE850ACB-E784-4914-962A-461EF6FF3997}')

set @number = '31400'
insert into @pointblock values (@number,'48000','{A9919B94-D29D-47a9-AFC4-C20EB47F63C0}')
insert into @pointblock values (@number,'10009','{A9919B94-D29D-47a9-AFC4-C20EB47F63C0}')
insert into @pointblock values (@number,'10013','{A9919B94-D29D-47a9-AFC4-C20EB47F63C0}')
insert into @pointblock values (@number,'10007','{A9919B94-D29D-47a9-AFC4-C20EB47F63C0}')
insert into @pointblock values (@number,'10008','{A9919B94-D29D-47a9-AFC4-C20EB47F63C0}')
insert into @pointblock values (@number,'10014','{A9919B94-D29D-47a9-AFC4-C20EB47F63C0}')
insert into @pointblock values (@number,'10015','{A9919B94-D29D-47a9-AFC4-C20EB47F63C0}')

set @number = '31411'
insert into @pointblock values (@number,'48412','{73D69077-EA77-4534-91B1-C998B5BBC60E}')
-- and
insert into @pointblock values (@number,'31410','{A673F9A6-48E1-4a4f-96EA-481938CDFABD}')

set @number = '31415'
insert into @pointblock values (@number,'48260','{F353176E-BDE8-487f-A235-900170D44807}')

set @number = '31420'
insert into @pointblock values (@number,'48262','{1A75CEC4-4584-422f-8FB6-5A871AAB059E}')

set @number = '31425'
insert into @pointblock values (@number,'48370','{1B0E27F9-2504-4a29-933F-11E7843ABAC4}')

set @number = '31430'
insert into @pointblock values (@number,'48316','{5FF0C761-C088-4bdc-8B1E-9920C04C28EC}')

set @number = '31435'
insert into @pointblock values (@number,'48420','{18EDF527-AA74-4cde-B71A-43C372E5E06D}')

set @number = '31480'
insert into @pointblock values (@number,'92508','{F4F7FC15-AA03-4b5d-A0EA-280F3FD140AA}')
-- and
insert into @pointblock values (@number,'92602','{E0418C34-104C-4e4f-9AFB-3A56F0C6DDF3}')

set @number = '31490'
insert into @pointblock values (@number,'92603','{26808475-D22F-49ea-B9B5-273A8219C5A8}')

set @number = '31600'
insert into @pointblock values (@number,'31000','{0FC2909A-9827-4266-836A-918F66C4F53C}')
insert into @pointblock values (@number,'49102','{0FC2909A-9827-4266-836A-918F66C4F53C}')

set @number = '31606'
insert into @pointblock values (@number,'31030','{09F3A5AD-37DB-49d4-8070-3D959C88CD82}')

set @number = '31610'
insert into @pointblock values (@number,'49415','{149E4303-515B-4074-9DEB-7F8851BBB725}')
insert into @pointblock values (@number,'49211','{149E4303-515B-4074-9DEB-7F8851BBB725}')
insert into @pointblock values (@number,'02451','{149E4303-515B-4074-9DEB-7F8851BBB725}')
insert into @pointblock values (@number,'04361','{149E4303-515B-4074-9DEB-7F8851BBB725}')

set @number = '31620'
```

```
insert into @pointblock values (@number,'49205','{CF85DB05-BEBC-410a-92FD-21F3915EA80E}')

set @number = '31625'
insert into @pointblock values (@number,'4905','{84BBA208-F0C7-41cf-976E-CA8FD9226043}')
-- and
insert into @pointblock values (@number,'49403','{119411F4-F77F-4804-AEF6-E9056E44F0F1}')

set @number = '31630'
insert into @pointblock values (@number,'49401','{56206B92-38E9-4031-B4FD-A07AD199A185}')

set @number = '31640'
insert into @pointblock values (@number,'49325','{EA50D7D9-BC5F-45a8-B774-19C1181B1128}')

set @number = '31641'
insert into @pointblock values (@number,'49402','{5C8FE1A3-1B3B-4ccf-9595-99F8AE54B6AA}')

set @number = '31655'
insert into @pointblock values (@number,'49414','{F9399AC6-64C7-48aa-AFD8-46E359EC24A9}')

set @number = '31656'
insert into @pointblock values (@number,'49416','{38CA2F87-FCC6-4534-9A9D-4C7D7FCA6702}')

set @number = '31700'
insert into @pointblock values (@number,'53400','{31E23E3B-2AFD-412c-B9D0-A2DEF9B63306}')
-- and
insert into @pointblock values (@number,'32200','{11E6418B-6A4D-43ad-8CE7-6B70C3EF234B}')

set @number = '31730'
insert into @pointblock values (@number,'32020','{ABFF5D0F-6E5D-4b48-8A4D-6C8B7F08F8C8}')

set @number = '31740'
insert into @pointblock values (@number,'32230','{B7E96B06-CC07-454b-9EF9-40751A361070}')

set @number = '31750'
insert into @pointblock values (@number,'32240','{056482A8-E435-47f6-B956-805F30C7E2AB}')

set @number = '31760'
insert into @pointblock values (@number,'53013','{E00F6E27-B54C-4593-BE1B-7D31AB5B5B86}')
-- and
insert into @pointblock values (@number,'32060','{ADC8AF67-BD8D-4c1c-B3E0-3AD348E4F7CB}')

set @number = '31765'
insert into @pointblock values (@number,'64090','{DC07BE90-196E-4ad6-9A36-0ADC144F7D8D}')
-- and
insert into @pointblock values (@number,'32065','{F62E389F-C74D-42fa-8420-FC7F79C5B6C5}')

set @number = '31770'
insert into @pointblock values (@number,'53327','{930F182D-B4EB-4657-AF67-7BAE136BC9F0}')
-- and
insert into @pointblock values (@number,'32410','{B80AFEAD-4D61-44c4-96E9-1FF6132D9B57}')

set @number = '31775'
insert into @pointblock values (@number,'32430','{A4C3069F-4289-4f09-8E3D-BDD0B3C00629}')
-- and
insert into @pointblock values (@number,'53338','{5979E6EB-BD47-46ae-9E8B-7F406D30BB89}')

set @number = '31781'
insert into @pointblock values (@number,'31000','{267CB1F1-9F56-4976-B2E3-6CFC1A227BAD}')
-- and
insert into @pointblock values (@number,'31760','{EB2B79F2-5516-41b4-A422-1E7A846D6C60}')
-- and
insert into @pointblock values (@number,'32060','{19A7C604-CB7C-41d1-BB69-953EC443B3B2}')
-- and
insert into @pointblock values (@number,'32070','{BE6E7F51-B9EC-4956-968C-0AC64B583577}')
-- and
insert into @pointblock values (@number,'32075','{3F8396CB-2A5E-4d20-8724-C26598096B2A}')

set @number = '31782'
insert into @pointblock values (@number,'32020','{903B93BF-F4E8-4b6c-BCF4-CC837F454E65}')
-- and
insert into @pointblock values (@number,'32070','{A822BECA-9F4D-418d-B261-5F5502781994}')
-- and
insert into @pointblock values (@number,'32076','{6A2A3750-DD56-4739-886C-51BE14F7402A}')

set @number = '31790'
insert into @pointblock values (@number,'32490','{A7E5E0EF-52F6-4218-A12C-E8C7F62262F6}')

set @number = '31791'
insert into @pointblock values (@number,'53185','{8726A287-E656-4418-A7F3-051EFFCD8AB9}')
-- and
insert into @pointblock values (@number,'32810','{0649524A-7209-4399-AFAE-45DF99634BC1}')

set @number = '31792'
insert into @pointblock values (@number,'53286','{FABEF915-16DC-4208-803B-8A21266E2B38}')
-- and
insert into @pointblock values (@number,'32820','{E3D535F9-6B9B-4fce-9263-7F7C30F34B9A}')

set @number = '31793'
insert into @pointblock values (@number,'53296','{F2234830-186E-465b-9EC6-179A1BEDAA21}')
-- and
insert into @pointblock values (@number,'32830','{99514127-976C-4994-BDCE-F25E25BB0351}')

set @number = '31800'
insert into @pointblock values (@number,'50230','{BE9B1B77-C151-42ff-AD49-4D805B25B7A9}')

set @number = '31820'
insert into @pointblock values (@number,'50200','{C7A9DEFF-8185-4875-98F3-FB1F2190B0AA}')

set @number = '31825'
```

```
insert into @pointblock values (@number,'50205','{FEE793BA-0C3D-443d-A11F-ECE03848A439}')

set @number = '31830'
insert into @pointblock values (@number,'50520','{12B3F832-37F7-453f-9D69-3A42C5971EBE}')
-- and
insert into @pointblock values (@number,'50220','{63C542ED-B68F-4d08-B4C5-1CB029088580}')

set @number = '31840'
insert into @pointblock values (@number,'50240','{2F20940D-AE95-4289-979D-C8A338D757A9}')

set @number = '31860'
insert into @pointblock values (@number,'50260','{7DC089B4-4909-475e-A03F-8D87B61F2C6F}')

set @number = '31875'
insert into @pointblock values (@number,'31675','{62503B89-34D4-4b3a-B667-38426CF752EE}')

set @number = '33250'
insert into @pointblock values (@number,'44250','{73825287-F7F8-4092-87BC-1182227F9E68}')

set @number = '33251'
insert into @pointblock values (@number,'44251','{2494C61C-138C-48d6-AEFF-CA0632BFF7FB}')

set @number = '33470'
insert into @pointblock values (@number,'44470','{CD693454-FCF5-4b28-AE5C-4379DDB8D2AF}')

set @number = '33471'
insert into @pointblock values (@number,'44471','{50F1FF75-729E-41f2-9D1A-DF37BEDD345C}')

set @number = '34300'
insert into @pointblock values (@number,'31011','{28BC8065-4641-4ead-BBAA-8C631C7A63A9}')

set @number = '34320'
insert into @pointblock values (@number,'52235','{CD471CD1-DBE9-45df-81FF-96928FDDA428}')
insert into @pointblock values (@number,'52281','{CD471CD1-DBE9-45df-81FF-96928FDDA428}')
insert into @pointblock values (@number,'52210','{CD471CD1-DBE9-45df-81FF-96928FDDA428}')
insert into @pointblock values (@number,'34321','{CD471CD1-DBE9-45df-81FF-96928FDDA428}')

set @number = '34321'
insert into @pointblock values (@number,'52235','{2603D3A5-322C-445c-918D-02F23AA7780D}')
insert into @pointblock values (@number,'52281','{2603D3A5-322C-445c-918D-02F23AA7780D}')
insert into @pointblock values (@number,'34320','{2603D3A5-322C-445c-918D-02F23AA7780D}')

set @number = '41012'
insert into @pointblock values (@number,'41601','{47DE4FB1-AA81-405b-9DA2-54F92D784855}')

-- notice, content is interpreted as OR.
set @number = '41015'
insert into @pointblock values (@number,'10001','{42684A87-87B1-469e-B456-C76A268D7F85}')
insert into @pointblock values (@number,'10010','{42684A87-87B1-469e-B456-C76A268D7F85}')
insert into @pointblock values (@number,'10011','{42684A87-87B1-469e-B456-C76A268D7F85}')

set @number = '41025'
insert into @pointblock values (@number,'02100','{D5039E98-E39F-4c3b-9AFF-89998E4BD4D7}')
-- and
insert into @pointblock values (@number,'02199','{31F54A5D-DF73-4ab9-AACC-D3DAF631CB15}')
-- and
insert into @pointblock values (@number,'02115','{AB2473A1-9CE3-418a-BB6C-20A30CDCCC84}')

set @number = '41030'
insert into @pointblock values (@number,'41612','{A36BD880-8F37-4130-9631-29D77343F751}')

set @number = '41045'
insert into @pointblock values (@number,'41401','{2B5073E2-C424-465e-8AF8-D16315D3E42F}')
insert into @pointblock values (@number,'10012','{2B5073E2-C424-465e-8AF8-D16315D3E42F}')
insert into @pointblock values (@number,'10004','{2B5073E2-C424-465e-8AF8-D16315D3E42F}')
insert into @pointblock values (@number,'26201','{2B5073E2-C424-465e-8AF8-D16315D3E42F}')

set @number = '41120'
insert into @pointblock values (@number,'12540','{E8339A70-5513-42b2-A732-AA7464B45312}')

set @number = '41126'
insert into @pointblock values (@number,'57357','{F63681C1-0596-4865-B48A-96CF91CB932D}')

set @number = '41124'
insert into @pointblock values (@number,'57221','{336C4A07-8ABA-4b1b-8BDE-9A132B7D9396}')

set @number = '41201'
insert into @pointblock values (@number,'41262','{C7F49A99-6131-473f-A34E-825151DB4A71}')
-- and
insert into @pointblock values (@number,'76161','{B1266CB0-1BB8-4a60-ACD4-D2D87AB30C4F}')
-- and
insert into @pointblock values (@number,'76170','{3607B49C-C833-41f8-9FA6-0198C1FDD6FD}')
-- and
insert into @pointblock values (@number,'76263','{FEB445E7-1649-4a4c-962D-4972C0A271E6}')
-- and
insert into @pointblock values (@number,'76720','{D608584B-3EA3-4ebc-AAAA-0F3B948B68E0}')

set @number = '41211'
insert into @pointblock values (@number,'76277','{6188AFA2-A2DB-4276-9C77-68B71517B4DD}')

set @number = '41212'
insert into @pointblock values (@number,'76510','{818CFF47-D1A8-45fc-A098-9C12F9E1CFEA}')

set @number = '41213'
insert into @pointblock values (@number,'11551','{2D5199ED-92D0-44af-9F12-388D086211EA}')
insert into @pointblock values (@number,'59318','{2D5199ED-92D0-44af-9F12-388D086211EA}')
insert into @pointblock values (@number,'59118','{2D5199ED-92D0-44af-9F12-388D086211EA}')

set @number = '41223'
```

```
insert into @pointblock values (@number,'04416','{BD52D9E3-F7F3-4b4b-8858-62E33298AFA2}')

set @number = '41224'
insert into @pointblock values (@number,'11552','{B9A9E3AA-2F3A-45d9-9E03-2981947F4BF4}')
insert into @pointblock values (@number,'59419','{B9A9E3AA-2F3A-45d9-9E03-2981947F4BF4}')

set @number = '41262'
insert into @pointblock values (@number,'41201','{FCC8EA7B-2582-48a9-A9FE-1B5F33249A69}')
-- and
insert into @pointblock values (@number,'76161','{45FAACFD-542A-4820-BB66-F255E56DA973}')
-- and
insert into @pointblock values (@number,'76170','{920D2A05-1862-4fab-8E35-D8E23A9687FC}')
-- and
insert into @pointblock values (@number,'76263','{51D9278F-0AB5-43b7-9633-685D0100165C}')
-- and
insert into @pointblock values (@number,'76720','{E9E93E45-B867-48e8-A4D5-F53D861E76B0}')

set @number = '41312'
insert into @pointblock values (@number,'41311','{6235E95F-ECBE-4079-8158-D1E287BCD301}')

set @number = '41313'
insert into @pointblock values (@number,'77376','{D56A05B6-77A6-4a65-B89D-B2515E6A1BD7}')

set @number = '41401'
insert into @pointblock values (@number,'77141','{5388E781-9643-4904-BC1F-65BD6CB130D8}')

set @number = '41414'
insert into @pointblock values (@number,'77382','{1DAEA24C-6622-4734-BDB6-EB266C487580}')
-- and
insert into @pointblock values (@number,'77574','{3F872F63-7BEB-450e-A935-1799F7F81403}')

set @number = '41416'
insert into @pointblock values (@number,'77311','{2331B8A8-101F-4d02-A8B3-26B6FE9480F6}')
-- and
insert into @pointblock values (@number,'77583','{403C3EA1-11B7-412c-B121-3B91B67E37E9}')

set @number = '41418'
insert into @pointblock values (@number,'77252','{F9FC0BF3-7D23-42c0-8C86-68D1AE8FE877}')
-- and
insert into @pointblock values (@number,'77489','{6AEC5654-0834-44e7-9697-B2A0017A1607}')
-- and
insert into @pointblock values (@number,'77576','{7C458B9D-1ACC-434e-A41A-3DBA3F9D957C}')

set @number = '41320'
insert into @pointblock values (@number,'77371','{AD5D781F-7B22-4d38-9607-78703AC3030C}')

set @number = '41420'
insert into @pointblock values (@number,'77387','{6A4F7FFF-A301-4fe8-BA61-DD7645840C0B}')

set @number = '41430'
insert into @pointblock values (@number,'94200','{A92A5AE7-CBDB-44f5-BFA4-873B19CB2B19}')
-- and
insert into @pointblock values (@number,'94210','{25556057-0A38-474a-B00F-78D2E87A2A83}')

set @number = '41431'
insert into @pointblock values (@number,'41430','{B36E564D-F7AF-47a6-8AA9-53EF39FCD54F}')

set @number = '41501'
insert into @pointblock values (@number,'70101','{64D0AC0F-6057-4663-B186-35C5FEEC64E6}')

set @number = '41502'
insert into @pointblock values (@number,'70102','{E4AA8160-3C25-4aa1-888D-77017B1D14A1}')

set @number = '41503'
insert into @pointblock values (@number,'70150','{E94FD2D8-8932-47ac-B01B-CC41E7D438B2}')

set @number = '41511'
insert into @pointblock values (@number,'70203','{0CBBAA34-45CC-43e9-B7A2-BEE240B0C493}')

set @number = '41515'
insert into @pointblock values (@number,'70212','{E3B9A7FB-1122-4048-BE36-DF3F1E62ACE8}')

set @number = '41521'
insert into @pointblock values (@number,'70306','{EA2685ED-4825-4ba7-8277-39151D2529CB}')
-- and
insert into @pointblock values (@number,'70309','{22C028AA-BEB3-49b4-9E76-7E5D7FA551D5}')

-- Notice, content is interpreted as OR.
set @number = '41522'
insert into @pointblock values (@number,'70311','{D64E96F1-3888-45ad-8630-2005DFC2B5FB}')
insert into @pointblock values (@number,'70312','{D64E96F1-3888-45ad-8630-2005DFC2B5FB}')

set @number = '41531'
insert into @pointblock values (@number,'94115','{34F35D53-5476-4d65-A289-3D8DC98FB1E6}')

set @number = '41560'
insert into @pointblock values (@number,'70205','{04D581B1-5A9C-4aa1-864B-9E3C5E1DC4C8}')
insert into @pointblock values (@number,'41512','{04D581B1-5A9C-4aa1-864B-9E3C5E1DC4C8}')

set @number = '41601'
insert into @pointblock values (@number,'67163','{8D8829A8-5590-4b58-B32F-A435F8721043}')
-- and
insert into @pointblock values (@number,'72001','{9763549F-1496-418e-9693-211E85B4409A}')
-- and
insert into @pointblock values (@number,'72101','{99DF9394-1FD3-4dbe-81FB-098C6946F9E2}')
-- and
insert into @pointblock values (@number,'72103','{397D3C15-F8F7-47ad-A80F-BA9EFBBECEAE}')
-- and
insert into @pointblock values (@number,'85151','{6D94043D-86EB-4685-8C66-DFF06E129B0F}')
```

```
set @number = '41614'
insert into @pointblock values (@number,'72113','{E3207D70-F994-459f-A2F6-C224B331BAFA}')
-- and
insert into @pointblock values (@number,'72213','{54DE38D2-4DC7-4459-B46A-6D9DCBC0E2C8}')

set @number = '41625'
insert into @pointblock values (@number,'72238','{F6956F35-F952-42a1-81C3-2547B0BED7C5}')

set @number = '41630'
insert into @pointblock values (@number,'94346','{EF6E9D76-1D83-455f-9003-CDEB4D44F7A6}')
-- and
insert into @pointblock values (@number,'41630','{46D64516-5828-487f-B892-EFD187E33B9F}')

set @number = '41632'
insert into @pointblock values (@number,'94306','{B05576DF-5931-4162-9AE7-F36D489EB6EC}')

set @number = '41636'
insert into @pointblock values (@number,'94205','{14465B83-3BEF-4a8b-97E5-546C6FCCFAA4}')

set @number = '41637'
insert into @pointblock values (@number,'94415','{650176ED-D542-4e6b-9E5F-CDC801DAB106}')

set @number = '41638'
insert into @pointblock values (@number,'41658','{3C9398DD-7975-490e-B4F7-BDB0E0E21C38}')
insert into @pointblock values (@number,'94936','{3C9398DD-7975-490e-B4F7-BDB0E0E21C38}')
insert into @pointblock values (@number,'41634','{3C9398DD-7975-490e-B4F7-BDB0E0E21C38}')

set @number = '41639'
insert into @pointblock values (@number,'41658','{F338DC52-6D7C-4c07-AD90-F4B9A55312FE}')
insert into @pointblock values (@number,'94936','{F338DC52-6D7C-4c07-AD90-F4B9A55312FE}')
insert into @pointblock values (@number,'41634','{F338DC52-6D7C-4c07-AD90-F4B9A55312FE}')

set @number = '41640'
insert into @pointblock values (@number,'94325','{C41EC26F-7E75-4c7a-8FC6-76390051E270}')

set @number = '41647'
insert into @pointblock values (@number,'94940','{73DB602B-08BF-4d94-AE22-FBC09C72E918}')

set @number = '41660'
insert into @pointblock values (@number,'72402','{4ADCAB6C-1690-4229-BD2C-931A82DB5110}')

set @number = '41661'
insert into @pointblock values (@number,'72134','{ABE2FD26-5168-43da-862F-E84CECEB09F8}')
-- and
insert into @pointblock values (@number,'72234','{447968B3-C0C8-442b-A4C5-A0391387F9AA}')

set @number = '41722'
insert into @pointblock values (@number,'77384','{7A0785DD-1D18-4a56-8819-8841C22AD944}')

set @number = '41811'
insert into @pointblock values (@number,'70210','{9093A333-CE89-43a9-B291-ADDBB6D12F71}')

set @number = '41812'
insert into @pointblock values (@number,'41545','{DE843668-BA7D-46da-AC39-A4224CEFA7E4}')

set @number = '41814'
insert into @pointblock values (@number,'77253','{85088260-5AF5-46f4-807A-AD612618AFFE}')

set @number = '41815'
insert into @pointblock values (@number,'77254','{C7E9A3B1-53E2-4046-83D4-06F5F9E34FCA}')

set @number = '41816'
insert into @pointblock values (@number,'70215','{C931E3F7-B80B-4ffc-84EB-2079E4EBDB08}')
insert into @pointblock values (@number,'41648','{C931E3F7-B80B-4ffc-84EB-2079E4EBDB08}')

-- course 41841: some parts of the point blocking courses specification can not be handled.
set @number = '41841'
insert into @pointblock values (@number,'41637','{063B17B5-ABFE-47c7-9D30-9C8201E30101}')

set @number = '41845'
insert into @pointblock values (@number,'94930','{D92A0127-0526-49ad-B18E-C71382E5A1BE}')

set @number = '41859'
insert into @pointblock values (@number,'94946','{AD17158E-A1A1-4c8b-B293-B1190EEACEF5}')

set @number = '41861'
insert into @pointblock values (@number,'26500','{BD65EA4A-25B3-4f3b-A76A-CDA37C6ECEAB}')
insert into @pointblock values (@number,'21390','{BD65EA4A-25B3-4f3b-A76A-CDA37C6ECEAB}')
-- and
insert into @pointblock values (@number,'49104','{3FFFD4DA-9DE1-435d-9CA3-9C4BB9A49A39}')
-- and
insert into @pointblock values (@number,'49105','{740AB3D9-6DA2-4321-93FE-DAFAA015D48A}')

set @number = '42041'
insert into @pointblock values (@number,'42540','{97460B41-DEC7-403b-8443-7B56022A478D}')

set @number = '42110'
insert into @pointblock values (@number,'80251','{4C207BBA-937A-44b2-BE87-18DBF19ED735}')

set @number = '42150'
insert into @pointblock values (@number,'80454','{40081CDC-6CDD-453b-8FDB-B3F712897FA4}')

set @number = '42213'
insert into @pointblock values (@number,'80213','{A3D36039-B6E3-47a7-AAF9-0BD4EE010BCF}')
insert into @pointblock values (@number,'80414','{A3D36039-B6E3-47a7-AAF9-0BD4EE010BCF}')

set @number = '42215'
insert into @pointblock values (@number,'80215','{2BD17551-14AD-4a78-A9E5-E0ED5D05A1D6}')
```

```
set @number = '42221'
insert into @pointblock values (@number,'42220','{18F9273A-6C3D-4226-B223-07B165E59CF4}')

set @number = '42224'
insert into @pointblock values (@number,'80223','{472C7576-E2AF-494f-ACD1-455D3E79829A}')

set @number = '42234'
insert into @pointblock values (@number,'80236','{65354699-3BE3-4721-90BD-FDDB93A8FC93}')

set @number = '42301'
insert into @pointblock values (@number,'80001','{C670B0BC-6777-4562-9E6D-9539396FC368}')

set @number = '42302'
insert into @pointblock values (@number,'42305','{99554DB3-8120-4fb1-BB2C-B59DCF485B2F}')
insert into @pointblock values (@number,'80005','{99554DB3-8120-4fb1-BB2C-B59DCF485B2F}')
insert into @pointblock values (@number,'42309','{99554DB3-8120-4fb1-BB2C-B59DCF485B2F}')
insert into @pointblock values (@number,'80009','{99554DB3-8120-4fb1-BB2C-B59DCF485B2F}')
insert into @pointblock values (@number,'42900','{99554DB3-8120-4fb1-BB2C-B59DCF485B2F}')

set @number = '42305'
insert into @pointblock values (@number,'42302','{ACE94FC5-06DF-48d0-9A94-4811EBD2A619}')
insert into @pointblock values (@number,'80002','{ACE94FC5-06DF-48d0-9A94-4811EBD2A619}')
insert into @pointblock values (@number,'80005','{ACE94FC5-06DF-48d0-9A94-4811EBD2A619}')
insert into @pointblock values (@number,'42309','{ACE94FC5-06DF-48d0-9A94-4811EBD2A619}')
insert into @pointblock values (@number,'80009','{ACE94FC5-06DF-48d0-9A94-4811EBD2A619}')
insert into @pointblock values (@number,'42900','{ACE94FC5-06DF-48d0-9A94-4811EBD2A619}')

set @number = '42309'
insert into @pointblock values (@number,'42302','{6F258C98-17A4-43c0-AF90-FB1DBEEC12CD}')
insert into @pointblock values (@number,'80002','{6F258C98-17A4-43c0-AF90-FB1DBEEC12CD}')
insert into @pointblock values (@number,'42305','{6F258C98-17A4-43c0-AF90-FB1DBEEC12CD}')
insert into @pointblock values (@number,'80005','{6F258C98-17A4-43c0-AF90-FB1DBEEC12CD}')
insert into @pointblock values (@number,'80009','{6F258C98-17A4-43c0-AF90-FB1DBEEC12CD}')
insert into @pointblock values (@number,'42900','{6F258C98-17A4-43c0-AF90-FB1DBEEC12CD}')

set @number = '42341'
insert into @pointblock values (@number,'80173','{F265C631-83B0-4624-A5B9-554FE032E84C}')
insert into @pointblock values (@number,'83173','{F265C631-83B0-4624-A5B9-554FE032E84C}')

set @number = '42371'
insert into @pointblock values (@number,'83489','{259AAA38-FBE5-4617-994F-3FE8AEC7AAE2}')
insert into @pointblock values (@number,'94489','{259AAA38-FBE5-4617-994F-3FE8AEC7AAE2}')

set @number = '42405'
insert into @pointblock values (@number,'80171','{E3D7C2B6-9521-4e1a-A695-2220873B2809}')
insert into @pointblock values (@number,'83171','{E3D7C2B6-9521-4e1a-A695-2220873B2809}')

set @number = '42410'
insert into @pointblock values (@number,'83472','{5F210DBC-4C40-4112-901C-B2A1BC6A6A05}')
-- and
insert into @pointblock values (@number,'83172','{F31F0A3A-2FE3-48b5-93A3-0E7BCA19D311}')
-- and
insert into @pointblock values (@number,'86172','{F2A5EE3F-FB9A-4962-B831-7AB11320E2B5}')
-- and
insert into @pointblock values (@number,'94550','{C47932EF-45CC-469a-A36B-15085AF0E5AC}')

set @number = '42415'
insert into @pointblock values (@number,'83175','{717FAC2E-9212-4ea1-8361-35F20BF5FC27}')
insert into @pointblock values (@number,'80175','{717FAC2E-9212-4ea1-8361-35F20BF5FC27}')

set @number = '42422'
insert into @pointblock values (@number,'83174','{C1727B67-A5E1-49ef-8EB2-0BC09CBF8E31}')
insert into @pointblock values (@number,'86174','{C1727B67-A5E1-49ef-8EB2-0BC09CBF8E31}')

set @number = '42425'
insert into @pointblock values (@number,'86301','{A8FCC80A-5E14-493c-8D6B-C7DE89D48E6D}')

set @number = '42435'
insert into @pointblock values (@number,'86496','{8125C447-0AFC-4085-97AE-142645C2DCEB}')
-- and
insert into @pointblock values (@number,'83496','{0B439F30-CFD0-4050-BE80-0AADB10A501A}')

set @number = '42460'
insert into @pointblock values (@number,'80290','{D5AE992B-EE21-4e0d-A3E4-6F60FAD26631}')
insert into @pointblock values (@number,'83290','{D5AE992B-EE21-4e0d-A3E4-6F60FAD26631}')

set @number = '42465'
insert into @pointblock values (@number,'86280','{C12C88AF-2AAF-43ab-B575-8D4210F0D0B0}')
-- and
insert into @pointblock values (@number,'83280','{CD8AA72E-828F-4580-9ECA-925095909D27}')

set @number = '42470'
insert into @pointblock values (@number,'80493','{5372608F-65DA-4a61-9D41-DEBDDE9424EC}')
insert into @pointblock values (@number,'83150','{5372608F-65DA-4a61-9D41-DEBDDE9424EC}')
insert into @pointblock values (@number,'83493','{5372608F-65DA-4a61-9D41-DEBDDE9424EC}')

set @number = '42643'
insert into @pointblock values (@number,'85211','{85FCAC4D-AE7B-4b8f-9314-2C337C439C4C}')

set @number = '42845'
insert into @pointblock values (@number,'94930','{6448DE38-0C20-4744-8D5C-7E7DFAF43320}')

set @number = '42859'
insert into @pointblock values (@number,'94946','{CA9F46E0-60C0-4bae-A7D8-A74BF92AB2DD}')

set @number = '42930'
insert into @pointblock values (@number,'94405','{8C7C5639-59EF-4fd7-B06A-9965E898EA8D}')
```

```
set @number = '42932'
insert into @pointblock values (@number,'94405','{5CBC48E3-6D0E-4a2b-BC2E-2884A6928F0A}')

set @number = '42953'
insert into @pointblock values (@number,'42951','{CED17FD4-0AE9-429d-8100-B435787ADA38}')
-- and
insert into @pointblock values (@number,'94505','{44D4A351-DE8F-4bb9-9D82-D70DC8AE3A32}')

set @number = '42954'
insert into @pointblock values (@number,'42952','{BFCE782E-0972-43f5-8599-B249CD423802}')

set @number = '42966'
insert into @pointblock values (@number,'94560','{A73494C6-C869-494e-A8E9-493C420F5FD1}')

set @number = '42968'
insert into @pointblock values (@number,'94540','{E4ECDFC1-C18D-43ee-ADA6-2EF460E62D80}')
-- and
insert into @pointblock values (@number,'42962','{4354BDAE-A6C5-4591-90B8-8E3369ADB5B6}')

set @number = '42981'
insert into @pointblock values (@number,'42110','{89C566C1-CF3F-487f-A951-F28BEB55A0E3}')

set @number = '42981'
insert into @pointblock values (@number,'42110','{21E4575C-D217-47d8-8E79-45312D96D6A0}')

set @number = '42983'
insert into @pointblock values (@number,'42135','{9346D670-60A9-4726-9F7D-46137BD7D124}')

set @number = '88383'
insert into @pointblock values (@number,'88384','{8F79C202-09ED-47d7-BC5C-822AD1460392}')
insert into @pointblock values (@number,'88385','{8F79C202-09ED-47d7-BC5C-822AD1460392}')
insert into @pointblock values (@number,'88386','{8F79C202-09ED-47d7-BC5C-822AD1460392}')
insert into @pointblock values (@number,'26010','{8F79C202-09ED-47d7-BC5C-822AD1460392}')

set @number = '88892'
insert into @pointblock values (@number,'C8893','{AB6FDF75-6634-4675-934A-BD0DEABD8432}')

set @number = '88893'
insert into @pointblock values (@number,'C8827','{A89DFCE4-33E8-4196-85C2-084699291121}')

------------------------------------------------------------------------------------------------------------------------

-- ********************************************************************************** --
------------------------------------------------------------------------------------------------------------------------


declare @userid uniqueidenti↓er
set @userid = '{40AA0D66-B5A3-4E42-9B13-BCDA0905EB9D}'

declare @curdate datetime
set @curdate = getdate()

declare @RelationCourseType_ID int
set @RelationCourseType_ID = 0

declare @courseversion_id uniqueidenti↓er

declare preq_cursor cursor for
select distinct relationcourse_id, number
from @pointblock
order by number
for read only
open preq_cursor

declare @coursenumber varchar(20)
declare @relationcourse_id uniqueidenti↓er

fetch next from preq_cursor into @relationcourse_id, @coursenumber
while @@fetch_status = 0
begin
  if exists (select courseversion_id from courseversion where number=@coursenumber)
  begin
    select @courseversion_id=courseversion_id from courseversion where number=@coursenumber
    insert into Course_RelationCourse
    (Course_RelationCourse_ID,
     CourseVersion_ID,
     Course_RelationCourseType_ID,
     Created,
     CreatedBy,
     Updated,
     UpdatedBy)
    values
    (@relationcourse_id,
     @courseversion_id,
     @RelationCourseType_ID,
     @curdate,
     @userid,
     @curdate,
     @userid)
  IF @@ROWCOUNT = 0
    RAISERROR(@coursenumber, 14, 1);
  end
  fetch next from preq_cursor into @relationcourse_id, @coursenumber
end

close preq_cursor
deallocate preq_cursor

------------------------------------------------------------
```

```
declare items_cursor cursor for
select number, pointblockingcourse, relationcourse_id
from @pointblock
order by number, relationcourse_id
for read only
open items_cursor

declare @pointblockingcourse varchar(20)
declare @pbc_relationcourse_id uniqueidenti↓er

fetch next from items_cursor into @coursenumber, @pointblockingcourse, @pbc_relationcourse_id
while @@fetch_status = 0
begin
  if exists (select courseversion_id from courseversion where number=@coursenumber)
  begin
    insert into Course_RelationCourseItem
    (Course_RelationCourseItem_ID,
     Course_RelationCourse_ID,
     Course_ID,
     CourseNumber,
     Created,
     CreatedBy,
     Updated,
     UpdatedBy)
    values
    (newid(),
     @pbc_relationcourse_id,
     null,
     @pointblockingcourse,
     @curdate,
     @userid,
     @curdate,
     @userid)
  end
  fetch next from items_cursor into @coursenumber, @pointblockingcourse, @pbc_relationcourse_id
end

close items_cursor
deallocate items_cursor

_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

declare items_cursor cursor for
select course_relationcourseitem_id, coursenumber
from course_relationcourseitem
for read only
open items_cursor

declare @course_relationcourseitem_id uniqueidenti↓er
declare @course_id uniqueidenti↓er

fetch next from items_cursor into @course_relationcourseitem_id, @coursenumber
while @@fetch_status = 0
begin
  if exists (select course_id from courseversion where number=@coursenumber)
  begin
    select @course_id=course_id from courseversion where number=@coursenumber

    update Course_RelationCourseItem
    set course_id=@course_id
    where course_relationcourseitem_id=@course_relationcourseitem_id
  end
  fetch next from items_cursor into @course_relationcourseitem_id, @coursenumber
end

close items_cursor
deallocate items_cursor
```

4.3 Miscellaneous

4.3.1 Course Points

```
declare @courseversion_id uniqueidenti↓er
declare @course_point_id uniqueidenti↓er
declare @point_id_tot uniqueidenti↓er
declare @point_id uniqueidenti↓er
declare @workload_tot float
declare @workload_rem float
declare @workload_3 float
declare @workload_13 float
declare @workload float

declare @userid uniqueidenti↓er
declare @curdate datetime
declare @periodtype_id int
declare @periods_3 int
declare @parts int
declare @part int

set @userid = '{40AA0D66-B5A3-4E42-9B13-BCDA0905EB9D}'
set @curdate = getdate()
```

```
declare cv_cursor cursor for
select courseversion_id, parts
from courseversion
where parts > 1
--and number = ''
order by number
for read only
open cv_cursor

fetch next from cv_cursor into @courseversion_id, @parts
while @@fetch_status = 0
begin
  set @point_id_tot = (select point_id from course_point where courseversion_id = @courseversion_id)
  set @workload_tot = (select pointmin from point where point_id = @point_id_tot)

  set @workload_rem = @workload_tot

  set @part = @parts
  set @periods_3 = 0;
  while @part > 0
  begin
    set @periodtype_id = (select top 1 periodtype_id
                          from course_period a
                          inner join period b
                          on a.period_id = b.period_id
                          and courseversion_id = @courseversion_id
                          and part = @part)
    if @periodtype_id = 2
    begin
      set @periods_3 = @periods_3 + 1
    end

    set @part = @part - 1
  end

  if ((@periods_3 * 5) >= @workload_tot)
    set @workload_3 = @workload_tot / @parts
  else
    set @workload_3 = 5

  set @workload_13 = (@workload_tot - @periods_3 * @workload_3) / (@parts - @periods_3)

  set @part = @parts
  while @part > 0
  begin
    set @periodtype_id = (select top 1 periodtype_id
                          from course_period a
                          inner join period b
                          on a.period_id = b.period_id
                          and courseversion_id = @courseversion_id
                          and part = @part)
    if @periodtype_id = 2
      set @workload = @workload_3
    else
      set @workload = @workload_13

    update course_point set part = 15 where courseversion_id = @courseversion_id and createdby = '25B22D49-8DDD-406F-A6AA-09
      E0DE40E3DC'

    set @point_id = newid()
    insert into point values(@point_id, @workload, @workload, 0, @curdate, @userid, @curdate, @userid)

    set @course_point_id = newid()
    insert into course_point values(@course_point_id, @courseversion_id, @part, @point_id, @curdate, @userid, @curdate, @userid)

    set @part = @part - 1
  end

  fetch next from cv_cursor into @courseversion_id, @parts
end

close cv_cursor
deallocate cv_cursor
```

4.3.2 Course Parts

```
declare @courseversion_id uniqueidenti↓er
declare @parts int

declare parts_cursor cursor for
select courseversion_id, max(part)
from course_period
group by courseversion_id
for read only
open parts_cursor

fetch next from parts_cursor into @courseversion_id, @parts
while @@fetch_status = 0
begin
  update courseversion
  set parts = @parts
  where courseversion_id = @courseversion_id
  fetch next from parts_cursor into @courseversion_id, @parts
end
```

```
close parts_cursor
deallocate parts_cursor
```

4.3.3 AMS Point Giving Courses

```
update point
set amsgiving = 1
where point_id in
  (select point_id from course_point where courseversion_id in
    (select courseversion_id from courseversion where number in
      (31790, 11251, 11252, 42422, 42521, 02261, 12230, 28310, 42630, 02431, 11302, 11303, 02559, 42643, 02725,
      28350, 34821, 42415, 42405, 42410, 42412, 42440)
    )
  )
```

4.3.4 Update Point for 01005 and 88001

```
update point
set pointmin = 10,
    pointmax = 10
where point_id in (select point_id from courseversion a
                   inner join course_point b
                   on a.courseversion_id = b.courseversion_id
                   where number = '01005'
                   and part < 15)
```

4.3.5 Update Course_ID on Course_RelationCourseItem

```
update course_relationcourseitem
set course_id = (select course_id from courseversion where number = course_relationcourseitem.coursenumber)
```

4.3.6 Update Course_ID on TechnicalLine, TechnicalField and Specialization

```
update technicalline_prerequisitecourse
set course_id = null

update technicalline_prerequisitetechnicalpackagecourse
set course_id = null

update technical↓eld_course
set course_id = null

update specialization_course
set course_id = null

update technicalline_prerequisitecourse
set course_id = (select course_id from courseversion where number = technicalline_prerequisitecourse.number)
where course_id is null

update technicalline_prerequisitetechnicalpackagecourse
set course_id = (select course_id from courseversion where number = technicalline_prerequisitetechnicalpackagecourse.number)
where course_id is null

update technical↓eld_course
set course_id = (select course_id from courseversion where number = technical↓eld_course.number)
where course_id is null

update specialization_course
set course_id = (select course_id from courseversion where number = specialization_course.number)
where course_id is null
```

4.3.7 Update Course_ID on TechnicalPackage

```
update technicalpackage_periodoptionalcourse
set course_id = null

update technicalpackage_periodcourseitem
set course_id = null

update technicalpackage_fundamentalcourseitem
set course_id = null

update technicalpackage_periodoptionalcourse
set course_id = (select course_id from courseversion where number = technicalpackage_periodoptionalcourse.number)
where course_id is null

update technicalpackage_periodcourseitem
set course_id = (select course_id from courseversion where number = technicalpackage_periodcourseitem.number)
```

```
where course_id is null

update technicalpackage_fundamentalcourseitem
set course_id = (select course_id from courseversion where number = technicalpackage_fundamentalcourseitem.number)
where course_id is null
```