CSCI235/MCS9235 Databases
Assignment 3
(Total 8 marks)

Due on Saturday, 14 May 2016 at 11:55 PM

Scope

The tasks of this assignment include implementation and testing of relational views, implementation and testing of advanced data manipulation statements of SQL, granting access rights to database objects, design and implementation of hierarchical data structures in XML.

This assignment consists of 5 tasks.

It is very important that you do what is written in **Prologue** section for Task 1 – Task 4, and follow all recommendations written at the beginning of each task.

The deliverables from each task are listed at the section Deliverables before the submission section.

A submission procedure is described at the end of assignment specifications.

Prologue

Download and unzip a file **all-files.zip**. You should obtain the following files: a3create.sql and a3drop.sql.
Connect to your account on one of the Oracle servers (data-pc01 .. data-pc40) and execute a script a3create.sql. The script creates the relational tables of a sample database used in Assignment 3. Read the script and discover and draw a conceptual schema of the sample database.
You can use a script a3drop.sql to drop all relational tables created by dbcreate.sql. Do not drop the relational tables now.

Tasks

Task 1 Implementation and testing of relational views (1.2 marks)

It is recommended to do Experiment 6.3 included in Homework 6 before implementation of task 1.
Implement SQL script **task1.sql** that creates and tests the following relational views. The names of views are up to you.
Testing of the views must be performed by insertion into the relational tables the rows such, that selection of information from each view returns non empty results. It means that your script must include CREATE VIEW statements, INSERT statements, and SELECT * FROM … statements.

Implementation of the views without testing that returns non empty contents of each view will score no marks!

(1) The first view must contain information about all academics who taught at least one subject worth 12 credit points and such that total enrolment in a subject was higher than 50 students. Include into a view the first and last name of academic, code and name of a subject.

(2) The second view must contain information about the total number of lecturers who taught each one of the subjects CSCI235, CSCI212, and CSCI222 in the past, i.e. before 2016.
For each subject a view must contain subject code, subject name, and the total number of lecturers who taught a subject in the past.

(3) The third view must contain information about the academics who do not teach any subjects in autumn session of the current year. Note that the view must contain correct information no matter in which year it is accessed. Include information about staff umber, first name, and last name of academics.

(4) The fourth a view must that contain information about academics that have taught the subjects CSCI235 and CSCI204 and CSCI222 in the past or are teaching the subjects now. Include only information about staff number, first and last names of academics who taught each one of the subjects listed above.

Task 2 Implementation of advanced data manipulations in SQL (1.2 marks)

It is recommended to do Experiment 6.3 included in Homework 6 before implementation of task 2.
Implement SQL script **task2.sql** that performs and tests the following modification of the sample database.

(1) Delete information about all lecturers who taught no subjects in the last 3 years.

(2) Create a relational table **CSCI235** that consists of the same columns as a relational table ACADEMIC. Copy the data from the table ACADEMIC into the relational table CSCI235 of the information about all academics who have taught a subject CSCI235 at least once. Note that you must enforce appropriate primary and foreign key constraints on a relational table CSCI235 before copying data from a relational table Academics.

(3) Add a column **total_subjects** to a relational table ACADEMIC and use **one** UPDATE statement to fill the column with information about the total number of subjects taught by each academic so far. Remember, that some academics are very busy with their research and they do not teach any subjects.

Testing of the modifications must be performed by insertion into the relational tables the rows that will be affected by data manipulation statements (INSERT, UPDATE, DELETE). You are allowed to use INSERT statements already implemented for task 1.

Implementation of the modification without testing that affects at least one row in the relational tables will score no marks!

Task 3 Granting access rights to database objects (0.6 marks)

It is recommended to do Experiment 6.4 included in Homework 6 before implementation of task 3.
Implement SQL script **task3.sql** that grants the following access rights to a user SCOTT.

(1) A user SCOTT should be able to access a relational table Teaches both in read and write mode.

(2) A user SCOTT should be able to create any relational table that has a foreign key referencing a primary key in a relational table RunningSubject.

(3) A user SCOTT should be able to read information about the academics who taught at least one of the subjects CSCI235, CSCI204, and CSCI124.

Task 4 Design and implementation stored PL/SQL procedure and function (2 marks)

It is recommended to do Experiment 7.3 included in Homework 7 before implementation of task 4.
Implement SQL script **task4.sql** that define and execute PL/SQL stored procedure and function for the following questions.

(1) Define a stored procedure **WHOTAUGHT** that takes a parameter of a subject code and display subject title, lecturers' names who have taught the subject before 2016. Execute the procedure **WHOTAUGHT** by prompt input a subject code and display the results.

(2) Define a stored function **FINDSUBJECTS** that takes an academic number and returns the academic's full name and a list of all subjects' code that the academic taught. Execute the function **FINDSUBJECTS** to find subject lists of all academics like the following:

```
ACADEMIC                SUBJECTS

Bill Gates              CSCI124 CSCI321

Steven Jobs             CSCI124 CSCI204


......
```

Implementation of the PL/SQL scripts without execution of the procedure and function will score no marks!

Task 5 XML

(1) **Design and implementation of hierarchical data structures in XML. (2 marks)**
You **must** do Experiment 9.1 included in Homework 9 before implementation of task 5.
Read the following specification of a sample database domain.

```
The multinational companies consist of the national
divisions like for example Oracle Australia, SAP New Zealand,
Microsoft Vanuatu, etc. A national division is described by
a name of country it is located in, and an address of its
headquarters.
An address of headquarters consists of city, street and
building number and it optionally includes phone and fax
number.
Each company has a number of branches located in different
cities. There is at most one branch of a given company in a
city. A branch is described by an address and a full name
(first name, optional initials, last name) of its director.
Each branch consists of a number of departments.
A department is described by its name, a full name of its
manager (first name, optional initials, last name) and
budget allocated in the current year. A department is
located in one building over several floors.
Employees work at departments. An employee is described by a
full name (first name, optional initials, last name) and
position he/she is assigned to. An employee works for only
one department.
```

Design a conceptual schema for a sample database domain given above. Use a notation of simplified UML class diagrams explained to you in this subject. Save your diagram in a file **task5.pdf**.

Create **at least two** sample instances of objects for each class included in your conceptual schema and implement all these instances in one XML document. Save your document in a file **task5.xml**. Start Command Prompt on XP system and use a program oraxml available on XP system to make sure that your document is well formed. A program oraxml is not available on Unix system!

(2) **Creation of DTD and validation of XML documents against DTD. (1 mark)**
You **must** do Experiment 9.2 and Experiment 9.3 included in Homework 9 and you **must** implement task 5.(1) of this assignment before implementation of task 5.(2).

Implement an external DTD that validates XML documents contents of the database designed in task 5.(1). Save your DTD in a file **task5.dtd**. Use a program oraxml to make sure that document in a file **task5.xml** validates against DTD in a file task5.dtd.

Deliverables

Task 1

Submit a file **task1.lst** which contains the execution results of the script file **task1.sql**. **Remember to put SQL*Plus command SET ECHO ON in the front of the script file.**

Task 2

Submit a file **task2.lst** which contains the execution results of the script file **task2.sql**. **Remember to put SQL*Plus command SET ECHO ON in the front of the script file**.

Task 3

Submit a file **task3.lst** which contains the execution results of the script file **task3.sql**. **Remember to put SQL*Plus command SET ECHO ON in the front of the script file**.

Task 4

Submit a file **task4.lst** which contains the execution results of the script file **task4.sql**. **Remember to put SQL*Plus command SET ECHO ON and SET SERVEROUTPUT ON in the front of the script file**.

Task 5

A file **task5.pdf** with a conceptual schema of the sample database, a file **task5.xml** with well formed XML document that contains sample contents of the database. A file **task5.dtd** that can be used to comprehensively test DTD for a well formed XML file task5.xml.

Submission procedure

Zip the files task1.lst, task2.lst, task3.lst, task4.lst, task5.pdf, task5.xml and task5.dtd into a file assignment3.zip.

(1) Connect to Moodle.
(2) Navigate to a folder ASSIGNMENT SUBMISSIONS
(3) Click at Assignment 3, Submit your solution here link.
(4) Click at Add Attachments button.
(5) Navigate to a location where a file **assignment3.zip** has been saved.
(6) Select the file and click at Open button.
(7) Click at Submit button.
(8) Click at OK button to return to Home Page.

A policy regarding late submissions is included in the course outline.

The assignment must be submitted as **soft copy** only.

The assignment is an **individual assignment** and it is expected that all its tasks will be solved **individually without any cooperation** with the other students. If you have any doubts, questions, etc. please consult your lecturer or tutor during lab classes or office hours. Plagiarism will result in a **FAIL** grade being recorded for that assessment task.

Late submissions do not have to be requested. Late submissions will be allowed for a few days after close of scheduled submission (up to 3 days). Late submissions attract a mark penalty; this penalty may be waived if an appropriate request for academic consideration (for medical or similar problem) is made via the university SOLS system *before* the Due date. No work can be submitted after the late submission time.

Marks and comments on the assignments will be returned to the students in two weeks after the submission on Moodle.

End of specification