School of Computing and Information Technology
University of Wollongong

Session: **Spring 2018**
Lecturer: **Tianbing Xia**

CSIT115/CSIT815 Data Management and Security
Assignment 3

Scope

This assignment is related to application of data definition and data manipulation statements of SQL.

Important messages

Please read the messages listed below before implementation of the tasks included in a specification of Assignment 3.

More implementation related information can be found in "How to … ?" Cookbook available through Moodle or at:
`http://www.uow.edu.au/~jrg/115/COOKBOOK`.

The outcomes of Assignment 3 are due by **Saturday, 27 October, 2018, 11.55 pm (sharp).**

Assignment 3 contributes to 8% of the total evaluation in the subject. 3 tasks are included in this assignment.

A submission procedure is explained at the end of this document.

Only one submission of the outcomes of Assignment 3 is allowed and only one submission per student is accepted. Please make sure that you submit the correct files.

A submission that contains an incorrect file attached is treated as a correct submission with all consequences coming from the evaluation of the file attached.

Compressed (zipped, rared, tared, etc) files will not be evaluated.

All files left on Moodle in a state `"Draft(not submitted)"` will not be evaluated.

A submission marked by Moodle as "late" is treated as a late submission no matter how many seconds it is late.

A policy regarding late submissions is included in CSIT115/815 Subject Outline.

Prologue

Download the files `dbcreate.sql`, `dbdrop.sql`, `dbload.sql`, template files `solution1.sql`, `solution2.sql` and `solution3.sql`, and `dbschema.pdf`. Copy the files to your USB drive or email the files to yourself such that you can access all of them either through command line interface mysql or graphical user interface MySQL Workbench.

Connect to MySQL database server either through command line interface mysql or graphical user interface MySQL Workbench.

When connected, select a database `csit115` with a command

```
use csit115
```

It is strongly recommended to process SQL script file `dbdrop.sql` to drop all relational tables used for implementation of the previous assessment task.

To re-create the relational tables of a sample database, process SQL script `dbcreate.sql`.

To load data into the relational tables created in the previous step process SQL script `dbload.sql`.

To list the names of relational tables created, use a command `show tables`.

To list a structure of a relational table `<table-name>` use a command: `describe <table-name>`.

Use a pdf viewer to open a file `dbschema.pdf` with a conceptual schema of the sample database.

No report is expected from the implementation of the steps listed above.

Tasks

Task 1 (2 marks)

Process SQL script file `dbdrop.sql` to drop all relational tables used for implementation of the previous task.

To re-create the relational tables of a sample database, process SQL script `dbcreate.sql`.

To load data into the relational tables created in the previous step process SQL script `dbload.sql`.

No report is expected from the implementation of the steps listed above.

At the moment MySQL database server does not enforce domain (also called as `CHECK`) consistency constraint. The domain constraints can be specified together with `CREATE/ALTER TABLE` statements. However the constraints are not enforced during the manipulations on data.

It is possible to enforce a domain constraint in a different way than directly through `CREATE TABLE` statement.

Implement SQL script in a template file `solution1.sql` that enforces the constraints listed below in the sample database.

(1) The total hours for each employee works on the projects must be less than or equals to 30 hours.

List the employee number, name and the message "works on the projects more than 30 hours" for the employees who violate the above constraint.

(2) The manager of a department must work in the same department with the employees that supervised by him / her.

List the employee number, department number that the employee works in, manager number, department number that the manager manages, and the message "are not in the same department" for the managers and employees who violate the above constraint.

When ready execute SQL scrip `solution1.sql` and save a report from execution in a file `solution1.rpt`.

Start mysql command line in a terminal by

```
mysql -u csit115 -p  -v  -c
```

Then input a password `csit115. It` allows MySQL database server to display comments and SQL script for each question in a report.

Use the commands as follows to generate a report for Task 1.

```
tee solution1.rpt;
source solution1.sql;
notee;
```

Hint: You can find similar `SELECT` statement already implemented in the "Cookbook".

Deliverables
A file `solution1.rpt` with a report from processing of SQL script `solution1.sql`. The report MUST have no errors and the report MUST list all SQL statements processed.

A report that contains no listing of executed SQL statements scores no marks and report that contains errors also scores no marks !

Submission of a file with a different name and/or different extension and/or different type scores no marks.

Task 2 (3 marks)

Perform the following actions.

(1) Connect to MySQL as a user csit115. Select a database `csit115.` Process SQL script file `dbdrop.sql` to drop all relational tables used for implementation of the previous task.

To re-create the relational tables of a sample database, process SQL script `dbcreate.sql`.

To load data into the relational tables created in the previous step process SQL script `dbload.sql`.

(2) Create a logical backup of relational table `Department` and `DeptLocation`. Save a logical backup in a file with the same name as a *prefix of your University email account*.`bak`.

(3) Connect to MySQL as a user csit115. Select a database `csit115.` Execute a script `disaster.sql`. The script simulates a disaster that has happened to a relational table `Department` and a table `DeptLocation`. Some of the rows are deleted, some of the rows are incorrectly inserted, and the others are modified. Unfortunately, all deletions, insertions, and modifications have been committed.

(4) Connect to MySQL as a user `root` either through command line interface `mysql` or graphical user interface `MySQL Workbench` and create a new database with a name the same as a *prefix of your University email account*.

(5) Restore the logical backup of relational tables `Department` and `DeptLocation` in the database *prefix of your University email account.*

No report is expected from the actions performed so far.

Your task is to use restored logical backup of relational tables `Department` and `DeptLocation` to find the rows that have been `deleted`, the rows that have been `inserted`, and the rows that have been `updated` in the relational tables `Department` and `DeptLocation`. To make this task unambiguous assume that none of the values of primary keys in the relational tables `Department` and `DeptLocation` have been changed.

Implement SQL script in a template file `solution2.sql` that lists `all rows that have been deleted`, `all rows that have been inserted` and `all rows that have been updated`. You may use the contents of SQL file `disaster.sql` to check if your script lists the correct results.

When ready execute SQL scrip `solution2.sql` and save a report from the processing of the script in a file `solution2.rpt`.

Start mysql command line in a terminal by

```
mysql -u root -p  -v  -c
```

Then input a password `csit115. It` allows MySQL database server to display comments and SQL script for each question in a report.

Use the commands as follows to generate a report for Task 2.

```
tee solution2.rpt;
source solution2.sql;
notee;
```

Hint: You can find a lot of applications of database definitions and manipulation statements in the "COOKBOOK".

Deliverables
A file `solution2.rpt` with a report from processing of SQL script `solution2.sql`. The report MUST have no errors and the report MUST list all SQL statements processed.

A report that contains no listing of executed SQL statements scores no marks and report that contains errors also scores no marks!

Submission of a file with a different name and/or different extension and/or different type scores no marks.

Task 3 (3 marks)
Some of simpler Database Management Systems, like for example MySQL 5.7 Community Edition, do not have the features that allow for automated auditing the database activities. In this task you will implement your own simple method of auditing the database activities.

Connect as `root` user and drop all relational tables in a database `csit115`. Next, process the scripts `dbcreate.sql` and `dbload.sql` to create and to load data into a sample database.

Next, perform the following actions.

(1) Make a relational table that supposed to contain a general query log empty.

(2) Set the appropriate values of the variables that allow the database to create a general query log, to save the log in a relational table, and to start recording the log from now.

(3) Process SQL script file `disaster.sql`.

(4) Set the appropriate values of all variables that stop recording a general query log from now.

(5) Use `SELECT` statement to list the contents of general query log and compare it with the contents of SQL script file `disaster.sql`.

No report is expected from the actions performed so far.

Your task is to use a general query log to find which relational tables from the sample database have been accessed during the processing of SQL script file `disaster.sql` and how many times each table have been accessed. "Accessed" means any DML or query operation on a relational table. To make this task simpler assume that a relational table can be used only one time per single SQL statement and that SQL script `disaster.sql` does not create any new relational tables.

(1) First, the SQL script creates a relational table to store information about the names of tables included in the sample database and the counters how many times each table has been accessed during the processing of SQL script file `disaster.sql`. A name of a new relational table and the names of its columns are up to you.

(2) Next, the script uses a general query log to fill the relational table created in the previous step with information about the names of tables included in the sample database and the counters how many times each table has been accessed during the processing of SQL script file `disaster.sql`.

(3) Finally, the script lists the contents of the relational table with the names of relational tables and the counters how many times each table has been accessed during the processing of SQL script file `disaster.sql`.

When ready execute SQL scrip `solution3.sql` and save a report from execution in a file `solution3.rpt`.

Start mysql command line in a terminal by

```
mysql -u root -p  -v  -c
```

Then input a password `csit115. It` allows MySQL database server to display comments and SQL script for each question in a report.

Use the commands as follows to generate a report for Task 2.

```
tee solution3.rpt;
source solution3.sql;
notee;
```

Hint: You can find similar `SELECT` statement already implemented in the "Cookbook".

Deliverables
A file `solution3.rpt` with a report from processing of SQL script `solution3.sql`. The report MUST have no errors and the report MUST list all SQL statements processed.

A report that contains no listing of executed SQL statements scores no marks and report that contains errors also scores no marks !

Submission of a file with a different name and/or different extension and/or different type scores no marks.

Submission
Note, that you have only one submission. So, make it absolutely sure that you submit the correct files with the correct contents and correct types. No other submission is possible !

Submit the files **`solution1.rpt`**, **`solution2.rpt`**, and **`solution3.rpt`** through Moodle in the following way:

(1) Access Moodle at **`http://moodle.uowplatform.edu.au/`**
(2) To login use a **`Login`** link located in the right upper corner the Web page or in the middle of the bottom of the Web page
(3) When logged select a site **`CSIT115/DPIT115/CSIT815 (S218)Data Management and Security`**
(4) Scroll down to a section **`Submissions`**
(5) Click at a link **`In this place you can submit the outcomes of Assignment 3`**
(6) Click at a button **`Add Submission`**
(7) Move a file **`solution1.rpt`** into an area **`You can drag and drop files here to add them`**. You can also use a link **`Add…`**
(8) Repeat step (7) for the files **`solution2.rpt`** and **`solution3.rpt`**.
(9) Click at a button **`Save changes`**
(10) Click at a button **`Submit assignment`**
(11) Click at the checkbox with a text attached: **`By checking this box, I confirm that this submission is my own work, …`** in order to confirm the authorship of your submission
(12) Click at a button **`Continue`**

It is expected that all its tasks included within **Assignment 3** will be implemented **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.

End of specification