School of Computing and Information Technology
University of Wollongong

Session: **Autumn 2019**
Lecturers: **Janusz R. Getta**
Tianbing Xia

CSIT115/CSIT815 Data Management and Security
Assignment 3

Published on 20 May 2019

Scope

This assignment is related to implementation of verification of complex consistency constraint, implementation of a simple auditing system, and backup and recovery of a database.

Please read very carefully information listed below.

This assignment contributes to 8% of the total evaluation in a subject CSIT115 and it contributes to 8% of the total evaluation in a subject CSIT815.

The outcomes of the assignment work are due by **Saturday 8 June 2019, 7.00 pm (sharp).**

A submission procedure is explained at the end of specification.

This assignment consists of 3 tasks and specification of each task starts from a new page.

It is recommended to solve the problems before attending the laboratory classes in order to efficiently use supervised laboratory time.

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 the subject outline.

A submission of compressed files (zipped, gzipped, rared, tared, 7-zipped, lhzed, … etc) is not allowed. The compressed files will not be evaluated.

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

An implementation that does not compile due to one or more syntactical errors scores no marks and implementation that has the processing errors scores no marks.

It is expected that all tasks included within **Assignment 3** 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 the assessment task.

Prologue

Connect to Moodle and download the files `dbcreate.sql`, `dbdrop.sql`, `dbload.sql`, `dbcount.sql`, and `dbschema.pdf` from Sample database section on Moodle.

SQL script `dbcreate.sql` can be used to create the relational tables of a sample database. SQL script `dbdrop.sql` can be used to drop the tables of a sample database. SQL script `dbload.sql` can be used to load data into a sample database. SQL script `dbcount.sql` can be used to display the total number of rows in each table included in a sample database. Finally, a file `dbschema.pdf` contains a conceptual schema of a sample database.

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`.

To 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>`.

To list the total number of rows in each relational table process a script `dbcount.sql`.

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 actions listed above.

Tasks

Task 1 (2 marks)

An objective of this task is to implement SQL script that verifies the following logical consistency constraint imposed on the contents of a sample database.

"All orders submitted after 30 April 2019 must not include discontinued products"

Download a file `solution1.sql` and insert into the file the implementations of the following actions.

(1) First, the script inserts into a sample database information about a new order submitted today that includes two products. One of the products is discontinued while the other is not discontinued. You are allowed to examine the contents of a sample database to find out which products are discontinued and which products are not discontinued and later on apply `INSERT` statements to implement this step. All other information related to a new order is up to you.

(2) Next, the script creates a single column relational table `MESSAGE` to store variable size strings no longer than 500 characters.

(3) Next, the script inserts into a relational table `MESSAGE` information about the contents of a sample database that violate the following consistency constraint.

"All orders submitted after 30 April 2019 must not include discontinued products"

The script must list the outcomes of verification of the consistency constraint as a single column table with the following messages as the rows in the table.

`Order` <insert `order_id` here> `submitted on` <insert `order_date` here> `includes a discontinued product` <insert `product_name` here>

For example, if `order_id` of an order submitted on 1 May 2019 is equal to `278` and product `Chai` is discontinued and the product is included into the order then verification of the consistency constraint must return the following message.

```
Order 278 submitted on 2019-05-01 includes a discontinued product
Chai
```

Use a function `CONCAT` to create the messages like the one listed above.

(4) Finally, the script makes the contents of a relational table `MESSAGE` permanent and lists the contents of the table.

When ready process a script file `solution1.sql` and save a report from the processing in a file `solution1.rpt`.

To create a report from processing of a script file `solution1.sql` open a Terminal

window and start the command line interface `mysql` in the following way:

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

Next, process SQL script `solution1.sql` and save a report in a file `solution1.rpt`. Note, that when started with the options **`-v`** and **`-c`** the command line interface includes both listing of `SELECT` statements processed and the comments included in the original version of a file `solution1.sql`.

Deliverables

A file `solution1.rpt` with a report from processing of SQL script `solution1.sql`. The report must be created with the command line interface mysql, the report MUST NOT include any errors, and the report must list all SQL statements processed and all comments included in the original (downloaded) version of `solution1.sql`. Marks will be deducted for the missing comments. Submission of a file with a different name and/or different extension and/or different type scores no marks.

Task 2 (3 marks)
An objective if this task is to use backup and recovery feature of a database system to find corrupted data and to restore the original contents of a sample database.

It is strongly recommended to connect to MySQL either through command line interface mysql or graphical user interface MySQL Workbench and process a script file `dbdrop.sql` and immediately after that the scripts `dbcreate.sql` and `dbload.sql` to refresh a sample database.

(1) Create a backup of a relational table `ORDER_DETAIL` and save it in a file with the same name as *a prefix of your University email account*`.bak`.

(2) Use a text editor `gedit` to modify a backup file obtained in a step (1) such that a backup of a relational table `ORDER_DETAIL` can be restored into a relational table with the same name as *a prefix of your University email account*. Do not restore the table yet.

(3) Download SQL script file `lazy-hacker.sql`. Connect as a user `csit115` and process the script file. Processing of the script simulates an activity of a lazy hacker who performs pretty random deletions, insertions, updates on a relational table `ORDER_DETAIL`.

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

Due to the modifications performed by a hacker a relational table `ORDER_DETAIL` has been corrupted. Your task is to find all insertions, updates, and deletions performed on the original contents of a relational table `ORDER_DETAIL` and then to restore the original contents from a backup taken earlier.

Download a file `solution2.sql` and insert into the file the implementations of the following actions.

(4) Create a relational table with a name *a prefix of your University email account* and with the same structure as already created relational table `ORDER_DETAIL`. Remember, to enforce the appropriate consistency constraints for the new table.

(5) Use a backup modified in step (2) to load the original contents of a relational table `ORDER_DETAIL` into a relational table with a name *a prefix of your University email account*.

(6) Use `SELECT` statements to list the rows that have been deleted, modified, and inserted into a relational table `ORDER_DETAIL` by a lazy hacker.

(7) Restore the original contents of a relational table `ORDER_DETAIL` and make the restored contents permanent.

(8) Drop a relational table *a prefix of your University email account*.

When ready process a script file `solution2.sql` and save a report from processing in a file `solution2.rpt`.

To create a report from processing of a file `solution2.sql` open a Terminal window and start the command line interface `mysql` in the following way:

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

Next, process SQL script `solution2.sql` and save a report in a file `solution2.rpt`. Note, that when started with the options **`-v`** and **`-c`** the command line interface includes both listing of SQL statements processed and the comments included in the original version of a file `solution2.sql`.

Deliverables

A file `solution2.rpt` with a report from processing of SQL script `solution2.sql`. The report must be created with the command line interface mysql, the report MUST NOT include any errors, and the report must list all SQL statements processed and all comments included in the original (downloaded) version of `solution2.sql`. Marks will be deducted for the missing comments. Submission of a file with a different name and/or different extension and/or different type scores no marks.

Task 3 (3 marks)
An objective of this task is to implement your own simple method of auditing the database activities.

It is strongly recommended to connect to MySQL either through command line interface mysql or graphical user interface MySQL Workbench and process a script file `dbdrop.sql` and immediately after that the scripts `dbcreate.sql` and `dbload.sql` to refresh a sample database.

Download a file `solution3.sql` and insert into the file the implementations of the following actions.

(1) First, the script makes a relational table that contains a general log empty.

(2) Next, the script sets the appropriate values of the variables that allow to create a general log, to save a general log in a relational table, and to start recording a general log from now. Then, the script executes a script file `lazy-hacker.sql`. Next, the script sets the appropriate values of all variables that stop recording a general log from now.

(3) Next, the script lists DML statements `INSERT`, `DELETE`, and `UPDATE` processed in a period of time when a general log was recorded.

(4) Next, the script lists total number of times a relational table `ORDER_DETAIL` have been accessed by DML statements `INSERT`, `DELETE`, and `UPDATE`.

When ready process a script file `solution3.sql` and save a report from processing in a file `solution3.rpt`.

To create a report from processing of a file `solution3.sql` open a Terminal window and start the command line interface `mysql` in the following way:

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

Next, process SQL script `solution3.sql` and save a report in a file `solution3.rpt`. Note, that when started with the options **`-v`** and **`-c`** the command line interface includes both listing of `SELECT` statements processed and the comments included in the original version of a file `solution3.sql`.

Deliverables
A file `solution3.rpt` with a report from processing of SQL script `solution3.sql`. The report must be created with the command line interface mysql, the report MUST NOT include any errors, and the report must list all SQL statements processed and all comments included in the original (downloaded) version of

`solution3.sql`. Marks will be deducted for the missing comments. 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/CSIT815 (S119)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`**

End of specification