School of Computing and Information Technology
University of Wollongong

Session: **Autumn 2020**
Lecturer: **Tianbing Xia**

CSIT115/CSIT815 Data Management and Security
Assignment 2

Published on 1 May 2020

Scope

This assignment is related to implementation of simple `SELECT` statements, `SELECT` statement with `GROUP BY` and `HAVING` clauses, `SELECT` statements that join and antijoin relational tables, nested `SELECT` statements with set membership operation, and nested `SELECT` statements with `EXISTS/NOT EXISTS` clauses.

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 16 May 2020, 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 2** 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 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.bmp` 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)

Download a file solution1.sql and insert into the file the implementations of the following queries as SELECT statements of SQL.
Your implementation must directly follow a comment with a specification of a subtask.

(1) Find the pub names which located at KING ST. or OXFORD ST..

(2) Find the pub names that serve WHISKY, or VODKA, or COGNAC. Display each pub name only once.

(3) Find the drinkers who have ordered drinks in January 2020. Display each drinker only once.

(4) Find the drinkers who have ordered the drink WHITE WINE at LONG JOHN. Display each drinker only once.

(5) Find the drink and the rating of drinks that liked. The results must be displayed in the descending order of the ratings, and for all drinks that have the same rating the results must be displayed in the ascending order of drinks. Display each pair only once.

(6) Find the drink and the lowest price of each drink served in pubs.

(7) Find the drinker and the total number of drinks ordered by each drinker in the first three months of 2020.

(8) Find the drink and the total amount of ratings of drinks for each drink. Do not display a drink if it hasn't been rated.

(9) Find the pub and the total number of drinks served in each pub that has more than three types of drinks.

(10) Find the pub, the drink, and the price of pubs that drinks contain a letter E.

When ready process a script file solution1.sql with SELECT statements.

To create a report from processing of SELECT statements 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)
Download a file `solution2.sql` and insert into the file the implementations of the following queries as `SELECT` statements of SQL.

Your implementation must directly follow a comment with a specification of a subtask.

The queries listed below must be implemented as `SELECT` statements with `JOIN` or `LEFT / RIGHT OUTER JOIN` operation.

(1) Find the distinct drinkers who have ordered drinks at a pub on `VICTORIA AVE.` in `March 2020`.

(2) Find the drinker and total number of times that the drinker rated drinks for `all drinkers`. Include drinkers who haven't rated a drink.

(3) Find the drinker and total amount of money spent on drinks for `all drinkers` in `February 2020`. Include drinkers who haven't ordered a drink in this period.

(4) Find the drinkers who haven't ordered any drinks so far.

(5) Find the drinkers who haven't order any drinks in `April 2020`.

(6) Find the drinker and the drink liked by the drinker but the drinker hasn't order the drink so far. Sort the drinker and the drink pairs in the ascending order of drinkers and drinks.

When ready process a script file `solution2.sql` with `SELECT` statements.

To create a report from processing of `SELECT` statements 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 `SELECT` 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)
Download a file solution3.sql and insert into the file the implementations of the following queries as SELECT statements of SQL.

Your implementation must directly follow a comment with a specification of a subtask.

The queries listed below must be implemented as nested SELECT statements with IN/NOT IN set membership operation.

(1) Find the distinct drinkers who have ordered drinks at a pub on VICTORIA AVE. in March 2020.

(2) Find the drinker and the drink liked by the drinker but the drinker hasn't order the drink so far. Sort the drinker and the drink pairs in the ascending order of drinkers and drinks.

The queries listed below must be implemented as nested queries with EXISTS/NOT EXISTS clauses.

(3) Find the distinct drinkers who have ordered drinks at a pub on VICTORIA AVE. in March 2020.

(4) Find the drinker and the drink liked by the drinker but the drinker hasn't order the drink so far. Sort the drinker and the drink pairs in the ascending order of drinkers and drinks.

A query listed below must be implemented with a set algebra operation.

(5) Find the distinct drinkers who like either BEER or RED WINE.

A query listed below must be implemented as a nested query.

(6) Find the distinct drinkers that ordered both VODKA and WHISKY.

When ready process a script file solution3.sql with SELECT statements.

To create a report from processing of SELECT statements 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 (S120)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 2`**

(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