

CSIT115/CSIT815 Data Management and Security
Assignment 3
23 April 2018

Scope

This assignment is related to application of advanced data manipulation statements of SQL and relational views.

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, 12 May, 2018, 10.00pm (sharp)**.

Assignment 3 contributes to 6% (5% for CSIT815 students) of the total evaluation in the subject.

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.

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

The reports from processing of SQL scripts must return NO ERRORS ! A solution with the errors is worth no marks !

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

Prologue

Download the files `dbcreate.sql`, `dbdrop.sql`, `dbload.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`.

You must drop all tables earlier created in `csit115` database. A script `dbdrop.sql` can be used to drop the relational tables. To make sure that all tables have been dropped use a command `show tables`.

Next, re-create all tables of `csit115` database and load data into the database before implementation of both tasks included in the assignment.

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

To load data into the relational tables process SQL script `dbload.sql`.

Your SQL statements must operate on the sample database loaded with data.

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. The green blobs represent the relational tables that implement the classes of objects and associations.

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

Tasks

Task 1 (4 marks)

Implement the following advanced manipulations on data in SQL.

- (1) Create an empty relational table to store information about the names of employers and the total number of positions offered by each employer. A name of relational table and the names of columns are up to you. Enforce the appropriate primary key and referential integrity constraints (if any).

Next, load into the table information about the names of employers together with the total number of positions offered by each employer. If an employer offers no position then its name must be recorded with a number zero (0).

- (2) Create a relational table that contains information about the names of all skills and the largest skill level possessed by an applicant and a number of applicant who possesses a skill at the highest level. Ignore the skills not possessed by any applicant.

All data must be loaded into the table by the same SQL statement that creates the table.

Enforce the appropriate primary key and referential integrity constraints (if any) after data is loaded.

- (3) Add a column to a relational table `POSITIONS` to store information about the total number of skills needed by each advertised position. A name of the column is up to you. Assume that no more than 9 skills are needed for each position.

Next, use a single `UPDATE` statement to set the values in the new column consistent with the present contents of the sample database.

- (4) Use a single `DELETE` statement to remove from a relational table `APPLIES` all applications submitted by the applicants who do not have any skills. `DELETE` statement must be correct for any contents of the sample database.

When ready save your implementations in SQL script file `solution1.sql` and process a script `solution1.sql`. Note, that your script must contain only SQL statements implementing the actions (1), (2), (3), (4). Save a report from processing of a file `solution1.sql` in a file `solution1.rpt`.

A file `solution1.rpt` must NOT contain the reports from processing of the scripts `dbdrop.sql`, `dbcreate.sql` and `dbload.sql`.

You can also find more information about creating reports from processing of SQL scripts in Cookbook, Recipe 3.1 How to use “mysql? Command based interface to MySQL database server? Step 4 How to save the results of SQL processing in a file?”

Your report must contain a listing of all SQL statements processed. You can find more information on how to display SQL statements while a script is processed in Cookbook, Recipe 3.1 How to use “mysql? Command based interface to MySQL database server? Step 3 How to process SQL script ?”

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

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.

Task 2 (2 marks)

It is recommended to drop all relational tables of a sample database and to create and to load data into a sample database before implementation of Task 2. It may happen that you have to drop some referential integrity constraints first.

Create SQL script `solution2.sql` that implements of the following query in SQL.

Find full names (first name, last name) and an average skill level of all applicants whose average skill level is below an average skill level of all skills possessed.

For example, if applicant X has three skills at the levels 5, 5, and 8 and applicant Y has one skill at level 2 and applicant Z has no skills then an average skill level of all skills possessed is $(5+5+8+2)/4 = 5$. An average skill level of applicant X is $(5+5+8)/3 = 6$. An average skill level of applicant Y is $2/1 = 2$. An average skill level of applicant Z is 0. Therefore, the full names of applicants Y and Z should be listed together with average skill level for each one of them.

Y	2
Z	0

The query must be implemented in the following way.

- (1) First, create a relational view that contains information about an average skill level of all skills possessed by at least one applicant. A name of relational view is up to you.
- (2) Next, create a relational view that for each applicant who possesses at least one skill contains information about an applicant number and an average level of all his/her skills. Remember, that the view must contain information about the applicants who possess at least one skill. A name of relational view is up to you.
- (3) Next, create a relational view that for each applicant who possesses no skills contains information about an applicant number and 0 as his/her an average level of all skills. A name of relational view is up to you.
- (4) Next, create a relational view that is union of the views created in step (2) and step (3). A name of relational view is up to you.
- (5) Next, create a relational view that contains information about applicant numbers whose average skill level is below an average skill level computed in step (1). The new relational view created in this step must use information included in the views created in the steps (1) and (4). A name of relational view is up to you.
- (6) Finally, implement SELECT statement that uses a relational view created in a step (5) and a relational table APPLICANT to *find full names (first name, last name) and*

an average skill level of all applicants whose average skill level is below an average skill level of all skills possessed by at least one applicant.

Include `CREATE VIEW` statements implementing the steps from (1) to (5) and the final `SELECT` statement into SQL script `solution2.sql`. When ready process the script and save a report from the processing in a file `solution2.rpt`.

Deliverables

Submit a report 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. The report **MUST** include **ONLY** SQL statements and control statements that implement a specification of Task 2 and **NO OTHER** statements.

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** and **solution2.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 (S118)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 a file **solution2.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