

CSIT115/CSIT815 Data Management and Security
Assignment 4
14 May 2018

Scope

This assignment is related to creation of a logical backup and application of the backup to restore the original data in a corrupted database and to implementation of simple auditing in a database system.

Important messages

Please read the messages listed below before implementation of the tasks included in a specification of Assignment 4.

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 4 are due by **Saturday, 2 June, 2018, 10.00pm (sharp)**.

Assignment 4 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 4 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.

Files left in "draft/not submitted" mode 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`, `disaster.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`. If some tables are left you must drop the tables manually with `DROP TABLE` statement. Sometimes, you may have to drop the referential integrity constraints first.

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

To 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 (3 marks)

Perform the following actions.

- (1) Create a logical backup of relational table `APPLICANT`. Save a logical backup in a file with the same name as *a prefix of your University email account.bak*.
- (2) Execute a script `disaster.sql`. The script simulates a disaster that has happened to a number of relational tables included in a sample database created and loaded in the **Prologue** step. 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.

No report is expected from the actions performed so far.

Your task is to use a logical backup of a relational table `APPLICANT` to find the rows that have been deleted, the rows that have been inserted, and the rows that have been updated in a relational table `APPLICANT`. To make this task unambiguous assume that none of the values of primary key in a relational table `APPLICANT` have been changed.

Implement SQL script file `solution1.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.

Hint

Of course, before processing of a script file `solution1.sql` you have to restore from a logical backup the old contents of a relational table `APPLICANT`.

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

Task 2 (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 `csit115` 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 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 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 process a script `solution2.sql` and save a report from processing of the script 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 4**
- (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 4** 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