

DPIT115 Data Management and Security

Assignment 3

Published on 8 September 2020

Scope

This assignment is related to discretionary access control in relational database systems, verification of a complex consistency constraint, implementation of a simple auditing system, and database backup and recovery techniques.

You will also complete an incident report on an incident reported in the media and discuss the incident with your tutorial class.

Please read the information listed below very carefully.

This assignment contributes to **15%** of the total evaluation in a subject DPIT115.

The outcomes of the assignment work are due by **Tuesday 22 September 2020, 11.55 pm.**

This assignment consists of **5 tasks**, and specification of each task starts from a new page.

A submission procedure is explained at the end of the specification.

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

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 in **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.

If it is suspected that you have received assistance from another person to complete the tasks, the matter will be investigated as an alleged breach of the UOW College Academic Integrity and Student Conduct Policy, in accordance with the Procedure for Managing Alleged Student Misconduct. As part of this investigation, you may be required to undergo an oral examination to verify your understanding of the assessment content.

Configuring the Virtual Server

Connect to Moodle and download the **Sample database** (Sample_database.zip) on Moodle. Extract the files dbcreate.sql, dbdrop.sql, dbload.sql, dbcount.sql, and dbschema.bmp.

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 number of rows in each database table.

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.

1. When connected, select a database csit115 with a command use csit115.
2. To create the relational tables of a sample database, process SQL script dbcreate.sql.
3. To load data into the relational tables created in the previous step process SQL script dbload.sql.
4. To list the names of relational tables created, use a command show tables.
5. To list a structure of a relational table <table-name> use a command describe <table-name>.
6. To list the total number of rows in each relational table process a script dbcount.sql.
7. Use a pdf viewer to open a file dbschema.pdf with a conceptual schema of the sample database.

No report is expected to be submitted from the implementation of the actions listed above.

Tasks

Task 1 (3 marks)

Your task is to implement and to process SQL script `solution1.sql` that creates a new database, creates the new user accounts, creates the new roles, grants roles and privileges to the new user accounts, sets resource limits and locks the accounts. Insert into a file `solution1.sql` implementation of the steps listed below. Note, that a user `csit115` does not have the privileges required to process these steps. You must connect as a user `root` with a password `csit115`. The steps to be implemented are the following.

- (1) Create a database with the same name as a prefix of your University email account. For example, if your University email account is `xyz007@uow.edu.au` then a name of a database should be `xyz007`. (0.1 mark)
- (2) Create two new user accounts. The names of user accounts and the passwords are up to you. (0.1 mark)
- (3) While connected as a user `root`, process the scripts `dbcreate.sql` and `dbload.sql` to create and to load data into the relational tables later on used in this laboratory class. All relational tables must be located in a database created in step (1). A listing of SQL statements processed by the scripts must NOT be included in a report from processing of a script `solution1.sql`. It means that before processing of the script you must process `notee` statement to turn the spooling off and after processing of the scripts you must process a statement `tee solution1.rpt` to turn the spooling on into a report file. (0.2 mark)
- (4) Next, create two new roles: `driver` and `admin` and grant to a role `admin` the read privileges on the entire database. The privileges must be granted such that any owner of a role `admin` cannot grant the same privileges to another role or user. (0.2 mark)
- (5) Next, grant to a role `driver` a read privilege on a relational table `DRIVER` located in the database. A privilege must be granted such that any owner of a role `driver` can grant the same privilege to another role or user. (0.2 mark)
- (6) Next, grant to a role `driver` the read and write privileges on the relational table `TRIPLEG` located in the database. The privileges must be granted such that any owner of a role `driver` cannot grant the same privilege to another role or user. (0.2 mark)
- (7) Next, grant to a role `driver` a read privilege on the columns `FNAME`, `INITIALS`, `LNNAME` in a relational table `EMPLOYEE`. A privilege must be granted such that any owner of a role `driver` cannot grant the same privilege to another role or user. (0.3 mark)
- (8) Next, grant to a role `admin` an insert privilege on a relational table `EMPLOYEE`. A privilege must be granted such that any owner of a role `admin` can propagate the same privileges to another role or user (0.2 mark)

- (9) Next, grant to a role `admin` a privilege to create relational tables located in the database. The privileges must be granted such that any owner of a role `admin` cannot grant the same privileges to another role or user. (0.2 mark)
- (10) Next, grant to a role `admin` a privilege to create relational views located in the database. The privileges must be granted such that any owner of a role `admin` cannot grant the same privileges to another role or user. (0.2 mark)
- (12) Next, grant to a role `admin` a read privilege on information about the trips completed in 2018. A hint is on create a relational view and grant a read privilege on the view. A privilege must be granted such that any owner of a role `admin` cannot grant the same privilege to another role or user. (0.5 mark)
- (13) Next, grant a role `driver` to a role `admin` and then grant a role `admin` to one of the users created in step (2) and a role `driver` to another user. (0.2 mark)
- (14) Next, set a resource limit on maximum total number of concurrent connections available to both users created in step (2). The maximum number of concurrent connections is up to you. (0.3 mark)
- (15) Finally, lock the accounts of the new users. (0.1 mark)

To implement and to test SQL script file `solution1.sql` you can either use graphical user interface MySQL Workbench or command line interface `mysql`.

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 root -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 (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.

"A driver is not allowed to perform more than 1 trip per day"

Download a file `solution2.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 trip that consist of two legs. A trip must be performed by a driver who has already performed a trip on the same day. You are allowed to examine the contents of a sample database to find out which driver performed at least one trip and later on apply `INSERT` statements to insert information about the same driver who performed another trip on the same day. Next, insert information about two legs that belong to the already inserted trip. All other information about a new trip and its legs 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 consistency constraint.

"A driver is not allowed to perform more than 1 trip per day"

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.

A driver <insert driving licence number here> performed more than one trip on <insert trip date here>

For example, if a driving licence number of a driver who performed more than one trip on 1 May 2019 is equal to 7 then verification of the consistency constraint must return the following message.

A driver 7 performed more than one trip on 1 May 2019

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 connect as `csit115` user, process a script file `solution2.sql`, and save a report from the processing in a file `solution2.rpt`.

To create a report from processing of a script 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 `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)

An objective of this task is to use a backup file and recovery feature of a database system to find historical information from a sample database. On 31 May 2015 a database administrator created a backup of a relational table `DRIVER` and saved a backup in a file `old_driver.bak`. Then to make your task easier, a database administrator updated the file such that it can be restored into a relational table `OLD_DRIVER`.

Implement a script file `solution3.sql` that performs the following actions.

- (1) Create a relational table `OLD_DRIVER` that has the same columns as a relational table `DRIVER`. Enforce appropriate consistency constraints for a relational table `OLD_DRIVER`.
- (2) Use a backup file `old_driver.bak` to load the old pre 31 May 2015 contents of a relational table `DRIVER` into a relational table `OLD_DRIVER`.
- (3) Use `SELECT` statements to list the employee numbers and driving license numbers of drivers and who left the transportation company after 31 May 2015.
- (4) Use `SELECT` statements to list the employee numbers, driving license numbers, first name and last name of drivers and joined the transportation company after 31 May 2015.

Before processing of a script file `solution3.sql` 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.

When ready connect as `csit115` user, 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 SQL 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.

Task 4 (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 `solution4.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 to save a general log in a relational table and to start recording a general log from now.
- (3) Next, the script makes a database `csit115` a default database, it stops recording a report, it executes a script file `workload.sql`, and it resumes recording a report into a file `solution4.rpt`.
- (4) Next, the script sets the appropriate values of all variables to stop recording a general log from now.
- (5) Finally, the script finds and lists how many times each one of the relational tables included in a sample database have been used by the successfully processed SQL statements included in SQL script `workload.sql`. You have to consider the relational tables with the following names `EMPLOYEE`, `DRIVER`, `ADMIN`, `TRUCK`, `TRIP`, and `TRIPLEG` and no other relational tables. The script must list the names of relational tables together with the total number of times each table has been used. Please, find a fragment of a sample output listed below.

```
+-----+-----+
| TABLE_NAME | TOTAL |
+-----+-----+
| EMPLOYEE    |      5 |
| TRIP        |      3 |
| ...         | ...   |
+-----+-----+
6 rows in set (0.01 sec)
```

To simplify this task, assume that a relational table is used no more than one time in SQL statement.

The results must be listed in the descending order of the total number of times each one of the relational tables has been used by the successfully processed SQL statements included in a script `workload.sql`. Note, that some of SQL statements included in a script `workload.sql` cannot be successfully processed and because of that counting

the total number of names of relational tables in the script does not provide the correct results. To find the correct results you must access an earlier recorded general log.

When ready connect as root user, process a script file `solution4.sql`, and save a report from processing in a file `solution4.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 root -p -v -c
```

Next, process SQL script `solution4.sql` and save a report in a file `solution4.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 `solution4.sql`.

Deliverables

A file `solution4.rpt` with a report from processing of SQL script `solution4.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 `solution4.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 5 (4 marks)

Find a recent news story regarding a data security incident and prepare a one to two-page report and discuss with the tutorial class.

Introduction

What happened and in what company did it happen? What were the circumstances, and what might be the implications?

Background

Discuss the threats, existing countermeasures in place. How did the countermeasures they fail to prevent the security incident?

Response

What was the response by the company to manage the problem. How effective was the response in resolving the incident? How long did it take for the company to respond?

Results

What was the situation at the end of the incident? How was the state of the organisation's reputation at the end of the incident?

Lessons

What can be learned from the incident to assist with future similar events?

Conclusion

Like the introduction, summarise your main points and emphasise the lessons learned.

References

Provide a reference list to your key resources and news articles.

Deliverables

Use the template **solution5.docx** provided to create your report and submit the file as **solution5.pdf**, submission of a file with a different name and/or different extension and/or different type scores no marks.

During your tutorial in week 12, you will be asked to discuss the report with the class and your tutor. This discussion will contribute 50% to your mark for this task.

Submission

Note that you have only one submission. So, make it sure that you submit correct files with the correct contents.

Submit the files `solution1.rpt`, `solution2.rpt`, `solution3.rpt`, `solution4.rpt` and `solution5.pdf` to the Assignment 3 submission area on the Moodle site.

End of specification