Assignment 3

Aim

The objectives of this assignment include:

- tasks related to implementation of discretionary access control, granting system resources, verification of complex consistency constraints,
- tasks related to using backup and restore features of DBMS to find the differences between two states of a relational table, and implementation of a simple auditing system.

Prologue

Download and unzip a file A3-all-files.zip. You should get the files CSIT115_Assn3_2023S4.pdf, A3create.sql, A3load.sql, A3drop.sql, A3Task2.sql and A3Task4.sql.

The script files create and load data into a database that contains information about customers, members, products, and purchases. The database also contains information about the types of members.

Task 1 (3 Marks)

(1) Connect to MySQL as a user root and create a new database with a name the same as <*prefix of your UOW email account*>.

(2) While connected as a user root use SQL script A3create.sql to create the relational tables in a database created in the previous step. A script A3create.sql creates relational tables that can be used to store information about customers, members, products, and purchases. Execute a script A3load.sql to load data into the relational tables that were created by A3create.sql. You can use a script A3drop.sql to drop the relational tables. Do not drop the relational tables now!

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

Implement SQL script that performs the following actions as a user root.

(1) Create three users with the following usernames:
<*prefix of your UOW email account*>*_1*
<*prefix of your UOW email account*>*_2*
<*prefix of your UOW email account*>*_3*
For example, if your UOW email prefix is abc123 then the names of users are abc123_1, abc123_2 and abc123_3. Set all passwords to be the same as the usernames. For example, if the username is abc123_1, the password should be set to abc123_1.

(2) Next, the script grants the user <*prefix of your UOW email account*>*_1* the privileges to alter relational tables, to drop relational tables, and to read and write relational tables in a database with the same name as <*prefix of your UOW email account*>*.* The privilege must be granted such that the user <*prefix of your UOW email account*>*_1* is able to grant all privileges listed above to the other users.

(3) Next, the script grants to the user <*prefix of your UOW email account*>*_2* the privileges to create views and to read data from the relational tables CUSTOMER and MEMBER in a database with the same name as <*prefix of your UOW email account*>. The privilege must be granted such that the user <*prefix of your UOW email account*>*_2* is NOT able to grant all privileges listed above to the other users.

(4) Next, the script sets the following values of resource limits to a user <*prefix of your UOW email account*>*_3*: total number of queries an account owner can issue per hour must be set to 100, and total number of updates an account owner can issue per hour must be set to 10.

(5) Next, the script expires the password of the account <*prefix of your UOW email account*>*_2*

(6) Finally, the script lists the privileges granted to all new users, <*prefix of your UOW email account*>*_1*, <*prefix of your UOW email account*>*_2*, <*prefix of your UOW email account*>*_3,* the values of resource limits set in a step (4) and a status of <*prefix of your UOW email account*>*_2* set in a step (5). To do so your script must access appropriate relational tables in a database mysql. Do not list information NOT related to the actions performed above!

Task 2 (2 Marks)

(1) Connect to MySQL as user csit115 and execute a script file A3drop.sql and immediately after that the scripts A3create.sql and A3load.sql to refresh the contents of a database csit115. Exit MySQL.

(2) Create a logical backup of a relational table CUSTOMER and save it in a file with the same name as *<prefix of your UOW email account>.bak*.

(3) Connect as a user csit115 to MySQL and execute a script file A3Task2.sql.

(4) Use a text editor and modify a backup file obtained in a step (2) such that a backup of a relational table CUSTOMER can be restored into a relational table with the same name as *a <prefix of your UOW email account>_DOC*.

(5) Use an updated backup file *<prefix of your UOW email account>*.bak to load the contents of the backup into a relational table *<prefix of your UOW email account>_DOC*. DO NOT delete the backup file!

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

Implement SQL script that finds the differences between the contents of a relational table CUSTOMER and a relational table with the same name as *<prefix of your UOW email account>_DOC.*

The script must first list the rows added to the relational table CUSTOMER after the backup file was created, then the rows deleted from a relational table CUSTOMER after the backup file was created, and finally list the rows changed in relational table CUSTOMER after the backup file was created.

In brief, the script must first list all added rows, then all deleted rows, and finally all changed rows in a relational table CUSTOMER. It is allowed to use more than one SELECT statement to implement this task.

Task 3 (2 Marks)

Connect to MySQL as user csit115 and execute a script file A3drop.sql and immediately after that the scripts A3create.sql and A3load.sql to refresh the contents of a database csit115.

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

Implement SQL script that performs the following actions.

(1) The script finds all cases that violate in a database csit115 the following consistency constraint.
"A member which is an elite member should NOT be a regular member"

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

A member with the customer number of <insert CNum here> and having the name <insert CName here> is both an elite member and a regular member

Use the function CONCAT to create the messages above and only display unique rows. It is NOT allowed to use more than one SELECT statement to implement this task.

Note that it is NOT your task to eliminate the violations of consistency constraint listed above.

Task 4 (3 Marks)

In this task you will implement your own simple method of auditing the database activities.

Connect to MySQL as a user root and execute a script file A3drop.sql and immediately after that execute script A3create.sql and A3load.sql to refresh a database csit115.

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

Implement SQL script that performs the following actions.

(1) First, the script sets the appropriate values of the variables that allow create a general log, to save a general log in a relational table, and to start recording a general log from now.

(2) Next, the script makes a relational table that contains a general log empty.

(3) Next, the script executes a script file A3Task4.sql. (Do NOT put results of execution of script A3Task4.sql into a report.)

(4) Next, the script sets the appropriate values of all variables that stop recording a general log from now.

(5) Next, the script lists the DDL statements (CREATE, ALTER, DROP) processed in a period of time when a general log was recorded.

(6) Next, the script lists the total number of times each one of DML statements (SELECT, INSERT, DELETE, UPDATE) processed in a period of time when a general log was recorded. Sort the results in a descending order of the total number of times a DML statement has been processed.

Submission

A) Please submit the following:

- Four text files that include the SQL statements and execution result for each task (one text file for each task)
- A .bakfile for task 2

B) The SQL statements must be arranged in the correct sequence of execution.

C) Please name your file as: <Your group>_<UOW ID>_<full name>_<Task number>.txt. For instance, T1_1234567_WangXinTien_Task2.txt.

D) Submission has to be uploaded to Moodle before the due date (set on Moodle). NO email submission is allowed.

E) Late submission penalty (25% per day) will be applied regardless of the reasons: network delay, disconnection, etc. Therefore, you are advised to avoid peak submission period near the submission deadline.

F) Re-submission (because of incorrect file included in the original submission) will be treated as late submission if it is submitted after the deadline.