

CSIT115/CSIT815 Database Management and Security
Laboratory 7
9 May 2017

Scope

This laboratory includes the tasks related to the applications of CREATE DATABASE, CREATE USER, and GRANT statements of SQL.

The outcomes of the laboratory work are due by **Saturday, 20 May, 2017, 7.00 pm.**

This laboratory contributes to 2% of the total evaluation in the subject.

A submission procedure is explained at the end of specification.

This laboratory consists of 2 tasks and specification of each task starts from a new page.

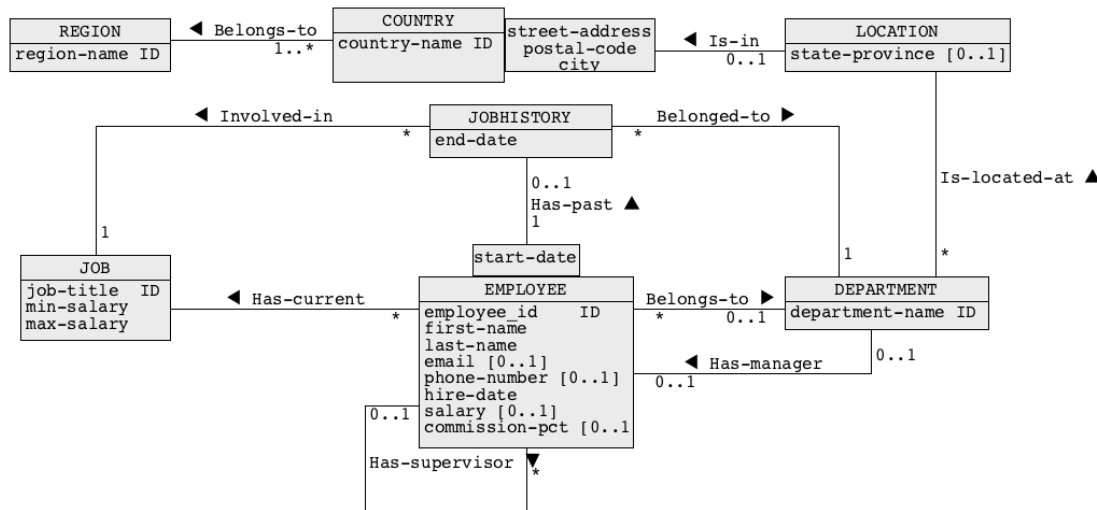
It is strongly recommended to solve the problems included in this specification **before coming to a laboratory class** and bring the preliminary solutions to a laboratory class such that any doubts, question, problems, etc can be discussed with a tutor in a laboratory class. Such procedure allows for more effective use of time spent in a supervised laboratory class.

Prologue

Perform the following actions.

Download and unzip a file `laboratory7-all-files.zip`. You should get the files `Laboratory7.pdf`, `dbcreate.sql`, `dbload.sql` and `dbdrop.sql`. Copy the files to your USB drive such that you can access both files either through command line interface `mysql` or graphical user interface `MySQL Workbench`. Copy the files to your USB drive such that you can access both files either through command line interface `mysql` or graphical user interface `MySQL Workbench`. You can also email a file `laboratory7-all-files.zip` to yourself such that you can access it on different systems. Finally, the simplest solution is to download the file directly to Ubuntu Linux from <http://www.uow.edu.au/~jrg/115/LABORATORIES/LABORATORY7>.

Connect to MySQL either through command line interface `mysql` or graphical user interface `MySQL Workbench` and execute script files `dbcreate.sql` and `dbload.sql`. The script files create and load data into a database that contain information about a company and its employees. The company consists of several departments located in the cities all over the world. The database also contains information about the present and past jobs of its employees and about the present managerial structure. A conceptual schema of the database is given below.



Tasks

Task 1 (1 mark)

Connect to a database server as a user `root` with a password `csit115`.

Implement SQL script `solution1.sql` that performs the following actions.

- (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 must be `xyz007`. **Note, that if you create a user `xyz007` then you can be absolutely sure that you will get no marks for this laboratory work.**
- (2) Create three users with the following user names: *prefix of your University email account_1*, *prefix of your University email account_2*, and *prefix of your University email account_3*. For example, if a prefix of your University email account is `xyz007` then the names of users are `xyz007_1`, `xyz007_2`, and `xyz007_3`. All passwords are up to you.
- (3) While connected as a user `root`, execute 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). SQL statements processed by the scripts must NOT be included in a report `solution1.rpt`. It means that before processing of the scripts you must execute `notee` statement to turn off spooling and after processing of the scripts `tee solution1.rpt` to turn on spooling into a report file.
- (4) The script grants a read privilege on entire database *prefix of your University email account* to a user *prefix of your University email account_1*. The privilege must be granted such that a user *prefix of your University email account_1* is not allowed to grant the same privilege to another user.
- (5) Next, the script grants write privileges on a relational table `EMPLOYEE` located in a database *prefix of your University email account* to a user *prefix of your University email account_2*. The privileges must be granted such that a user *prefix of your University email account_2* is able to grant the same privileges to the other users.
- (6) Next, the script grants a privilege to create relational tables located in a database *prefix of your University email account* to a user *prefix of your University email account_3*. The privilege must be granted such that a user *prefix of your University email account_3* is not allowed to grant the same privilege to another user.
- (7) Next, the script grants a privilege to read the columns (`department_name`, `street_address`, `city`, `country_name`) in a relational table `DEPARTMENT` located in a database *prefix of your University email account* to a user *prefix of your University email account_3*. The privilege must be granted such that a user *prefix of your University email account_3* is not allowed to grant the same privilege to another user.

- (8) Finally, the script lists all privileges granted to the users *prefix of your University email account_1*, *prefix of your University email account_2*, and *prefix of your University email account_3*. The script must use data dictionary views included in mysql database to list the privileges.

Deliverables

Submit 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. The report **MUST** include **ONLY** SQL statements and control statements that implement a specification of Task 1 and **NO OTHER** statements.

A report that contains no listing of executed SQL statements scores no marks !

A report that contains processing errors scores no marks !

Processing of the script on an empty database scores no marks !

Processing of the script that contains statements other from `CREATE DATABASE`, `CREATE USER`, `GRANT`, `SELECT`, `source`, `note`, `tee`, scores no marks !

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

Task 2 (1 mark)

Process a script `solution1.sql` implemented in the previous step and do not drop a database, users, and relational tables created by the script. Do not change the access rights granted by the script. **No report from processing of the script `solution1.sql` is expected in this task.**

Use mysql command line interface to perform the following actions.

- (1) Start mysql command line interface and connect as a user *prefix of your University email account_1*.
- (2) Execute a command `tee solution2.rpt`.
- (3) Execute two SQL statements that show the validity of two different privileges granted to a user *prefix of your University email account_1* in the previous task. Note, that you have to use a database *prefix of your University email account*. Each statement must retrieve precisely two rows.
- (4) Execute any SQL statement that shows a lack of privilege to access a database *prefix of your University email account* in write mode by a user *prefix of your University email account_1*.
- (5) Exit mysql command line interface.
- (6) Start mysql command line interface and connect as a user *prefix of your University email account_2*.
- (7) Execute a command `tee solution2.rpt`.
- (8) Execute two SQL statement that shows the validity of two different privileges on a relational table `EMPLOYEE` located in a database *prefix of your University email account* and granted to a user *prefix of your University email account_2* in the previous task.
- (9) Execute three SQL statement that shows a lack of write privilege on a relational table `DEPARTMENT` located in a database *prefix of your University email account* by a user *prefix of your University email account_2* in the previous task.
- (11) Exit mysql command line interface.
- (12) Start mysql command line interface and connect as a user *prefix of your University email account_3*.
- (13) Execute a command `tee solution2.rpt`.

- (14) Execute any SQL statement that shows the validity of a privilege to create a relational table located in a database *prefix of your University email account* and granted to a user *prefix of your University email account_3* in the previous task.
- (15) Execute any SQL statement that shows a lack of privilege to create a relational table in a database `csit115` by a user *prefix of your University email account_3*.
- (16) Execute any SQL statement that shows the validity of privilege to read the columns `(department_name, street_address, city, country_name)` from a relational table `DEPARTMENT` located in a database *prefix of your University email account* and granted to a user *prefix of your University email account_3* in the previous task. The statement must retrieve precisely one row.
- (18) Execute any SQL statement that shows a lack of privileges to read a column other than `(department_name, street_address, city, country_name)` from a relational table `DEPARTMENT` located in a database *prefix of your University email account* and granted to a user *prefix of your University email account_3* in the previous task.
- (19) Execute a command `notee`.
- (20) Exit mysql command line interface.

Deliverables

Submit a report file `solution2.rpt` that contains results from processing of SQL statements implementing the actions listed above. The report **MUST** list all SQL statements processed.

A report that contains no listing of executed SQL statements scores no marks !

Note, that it is allowed that processing of the appropriate SQL statements returns "... command denied ..." error messages. A report that contains any other error messages scores no marks !

Processing of the script on an empty database scores no marks !

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 correct files with the correct contents. 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 **CSIT815/CSIT115 (S117) 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 Laboratory 7**
- (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**.
- (8) Click at a button **Save changes**
- (9) Click at a button **Submit assignment**
- (10) 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.
- (11) Click at a button **Continue**

A policy regarding late submissions is included in the subject outline.

Only one submission of the outcomes of Laboratory 7 is allowed and only one submission per student is accepted.

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

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.

It is expected that all tasks included within **Laboratory 7** 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 that assessment task.

The evaluated outcomes of will be electronically returned to the students before 11.55pm on Saturday, 4 June, 2017.

End of specification