School of Computing and Information Technology
University of Wollongong

Session: **Autumn 2016**
Lecturer: **Janusz R. Getta**

CSIT115/CSIT815 Database Management and Security
Laboratory 8

Scope

This laboratory includes the tasks related to granting access right to the subsets of relational tables and verifying consistency constraints in the relational tables.

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.

Tasks

Task1 (1 mark)

Download and unzip a file `laboratory8-all-files.zip`. You should get the files `Laboratory8.pdf`, `dbcreate8.sql`, and `dbdrop8.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.

Connect as a user that has all privileges on a database `csit115` and execute a script `dbdrop8.sql` to drop the older versions of the relational tables. Next, execute a script `dbcreate8.sql` to create and to load data into a sample database.

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

(1) First, the script creates two database users with the names the same as *a prefix of your University account* concatenated with `_1` in a case of the first user and concatenated with `_2` in a case of the second user.

(2) Next, the script creates in a database `csit115` a relational view `DRVADM` that allows for access to information about drivers and administration people aggregated in the following way.

```
  ENUM  |  LNUM  |  STATUS  | POSITION  |    DOB     | CATEGORY
-------------------------------------------------------------
  1     | 10001  |AVAILABLE | NULL      | 1978-01-01| driver
  17    |  NULL  | NULL     | ceo       |   NULL    | admin
 ...    |  ...   |  ...     |   ...     |    ...    |  ...
```

A relational view `DRVADM` must be organized such that all rows that contain information about drivers must have no value (`NULL`) in a column `POSITION` and all rows that contain information about admin people must have no value (`NULL`) in the columns `LNUM` and `STATUS`.

The following hints provide more information on how to create a view:

- it is possible to use `UNION` or `UNION ALL` operation to "vertically concatenate" two relational tables,
- it is possible to use a symbol `NULL` in `SELECT` clause, for example `SELECT 1+1, NULL FROM DUAL;`
- it is also possible to use a string constant in `SELECT` clause, for example `SELECT 1+1, ' = two' FROM DUAL;`

(3) Next, the script grants a read privilege to all information included a view `DRVADM` to a user with the same name as *a prefix of your University email account*`_1`.

(4) Next, the script grants a read privilege to all information included in a view `DRVADM` except the rows where a date of birth is not empty to a user with the same name as *a prefix of your University email account*`_2`.

(5) Next, the script displays all information available to both users.

(6) Finally, the script displays the read privileges granted to both users.

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.

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

The names of users created in Task 1 must be the same as *a prefix of your University account* concatenated with `_1` for the first user and `_2` for the second user. The different names indicate that your work has been done by another student with all consequences implied by such fact.

Task 2 (1 mark)
Refresh the contents of `csit115` database with SQL scripts `dbdrop8.sql` and `dbcreate8.sql`.

Implement SQL script `solution2.sql` that performs the following actions.

(1) The script uses a database csit115.

(2) Next, the script changes a value of a system variable `AUTOCOMMIT` to `'OFF'`.

(3) The script corrupts then contents of a relational table `TRIPLEG` by changing a name of a departure city in a leg `2` of a trip `25` from `Sydney` into `Hobart` and through changing a name of destination city in a leg `3` of a trip `35` from `Perth` into `Adelaide`.

(4) Next, The script verifies the following consistency constraint.

A destination city of a trip leg and departure city of the next trip leg within the same trip must be the same.

For example, consider a trip number `100`. If a destination city of a trip leg number `3` within a trip number `100` is `Sydney` then if the trip has the next leg, i.e. leg number `4`, then a departure city of such leg must be `Sydney` as well. It means that a trip cannot "jump" from city to city without recording it as a trip leg.

The script must display the violations of the consistency constraint defined above as a relational table with a single column that contains the messages consistent with a pattern listed below.

`A destination city city-name-1 of leg leg-number-1 within a trip trip-number is different from a departure city city-name-2 of leg leg-number-2 within the same trip.`

Obviously, the symbols `city-name-1`, `leg-number-1`, `trip-number`, `city-name-2`, `leg-number-2` used in the pattern above must be replaced with the appropriate values.

To form the messages use a function `CONCAT` to concatenate the values selected from the relational tables with the string constants.

(5) Next, the script reverses the modifications done in a step (3) in the simplest possible way.

(6) Finally, the script repeats verification of the same consistency constraint as in a step (4).

Deliverables

A 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.

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

End of specification