School of Computing and Information Technology
University of Wollongong

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

CSIT115/CSIT815 Data Management and Security
Assignment 3

Scope

This assignment consists of the tasks related to implementation of data manipulations and queries in SQL.

This assignment consists of 4 tasks and specification of each task starts from a new page.

Prologue

Download and unzip a file `assignment3-all-files.zip`. You should get the files `Assignment3.pdf`, `dbcreate3.sql`, and `dbdrop3.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 to MySQL either through command line interface mysql or graphical user interface MySQL Workbench and execute a script file `dbcreate3.sql`. A script creates and loads data into the relational tables that contain information about the employees, drivers, administration people, trucks, trips, and trip legs.

Tasks

Task1 (1 mark)

Implement the following data manipulations in SQL as the sequences of simple `INSERT`, `UPDATE`, `DELETE` statements. Note, that each task may need more than one data manipulation statement to be implemented. An important condition is that you **are allowed** to drop, to alter, and to create consistency constraints during the modifications and that at the very end of all modification all original consistency constraints must be enforced.

(1) The transportation company employed Alice B who is a twin sister of a driver with driver license number 10002 and who lives in the same place as a driver with driver license number 10001. Alice B supposed to be employed as an administration staff member at a position "senior analyst". Modify the contents of the database.

(2) The transportation company decided to change the employee numbers (attribute `ENUM`) such that after the modifications all administration people will have the employee numbers starting from 1 and all drivers will have the employee numbers starting from 101. Modify the contents of the database.

(3) The transportation company discovered that a trip number 35 (attribute `TNUM`) has been incorrectly recorded. Correct information about the trip is such that it started at Sydney, then it passed through Melbourne, Hobart, Adelaide, and it ended in Perth. Modify the contents of the database.

(4) The transportation company decided to merge the trips with the numbers 26 and 27 (attribute `TNUM`) into one trip with a number 26 (attribute `TNUM`). A trip 27 (attribute `TNUM`) must be removed from the database. Modify the contents of the database.

(5) The transportation company decided to sell a truck with a registration KKK007 (attribute `REGNUM`). Information about the truck must be removed from the database. Modify the database such that information about all trips and the legs of the trips performed by the truck is still included in the database.

When ready save your implementations in SQL script file `solution1.sql`. When ready execute a script `solution1.sql` and save a report from the processing in a file `solution1.rpt`.

After each execution of a script `solution1.sql` you must use the scripts `dbdrop3.sql` and `dbcreate3.sql` to drop the modified relational tables and to re-create and re-load the tables with the original data. It will allow you to operate on exactly the same database each time you execute the script `solution1.sql`.

A file `solution1.rpt` must NOT contain the reports from processing of `dbdrop3.sql` and `dbcreate3.sql`.

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 !

Task 2 (1.5 mark)

Connect to MySQL either through command line interface mysql or graphical user interface MySQL Workbench and execute a script file `dbdrop3.sql` and immediately after that a script `dbcreate3.sql`. It allows you to refresh a sample database just in case if it has been accidentally modified in the previous task.

Implement the following queries as `SELECT` statements of SQL and save the statements in SQL script file `solution2.sql`.

(1) List full information about the trucks whose registration number starts from a letter P and ends with a digit 8.
(2) Find the total number of times the trips either started or ended or passed through Perth.
(3) Find the largest capacity and the smallest weight of all trucks and list the results in one line.
(4) Find full names of all drivers whose date of birth is unknown.
(5) Find the registration numbers of all truck used for at least one trip in 2012. Do not list duplicated registration numbers.

When ready execute SQL scrip `solution2.sql` and save a report from execution in a file `solution2.rpt`.

Hint: You can find similar `SELECT` statement already implemented in the "Cookbook".

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 !

Task 3 (1.5 mark)

Connect to MySQL either through command line interface mysql or graphical user interface MySQL Workbench and execute a script file `dbdrop3.sql` and immediately after that a script `dbcreate3.sql`. It allows you to refresh a sample database just in case if it has been accidentally modified earlier.

Implement the following queries as `SELECT` statements of SQL and save the statements in SQL script file `solution3.sql`.

(1) Find full names of all drivers whose present status is "on leave". The query must be implemented as a nested query.
(2) Find full names of all drivers whose present status is "on leave". The query must be implemented as a correlated nested query (query with an existential quantifier).
(3) Find full names of all drivers who used a truck with a registration PKR856 at least one time. The query must be implemented as a nested query.
(4) Find the driver license numbers (attribute `LNUM`) of all drivers who never used a truck with a registration PKR856. The query must be implemented as a nested query.
(5) Find the driver license numbers (attribute `LNUM`) of all drivers who never used a truck with a registration PKR856. The query must be implemented as a correlated nested query (a query with a negated existential quantifier).

When ready execute SQL scrip `solution3.sql` and save a report from execution in a file `solution3.rpt`.

Hint: You can find similar `SELECT` statement already implemented in the "Cookbook".

Deliverables

A file `solution3.rpt` with a report from processing of SQL script `solution3.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 !

Task 4 (2 marks)
Connect to MySQL either through command line interface mysql or graphical user interface MySQL Workbench and execute a script file `dbdrop3.sql` and immediately after that a script `dbcreate3.sql`. It allows you to refresh a sample database just in case if it has been accidentally modified earlier.

Implement the following data manipulations in SQL. Note, that each task may need more than one data manipulation statement to be implemented.

(1) Create a relational table `EMPNAME` that consists of the columns `ENUM`, `FNAME`, `INITIALS`, and `LNAME` copied from a relational table `EMPLOYEE`. Enforce appropriate consistency constraints on a relational table `EMPNAME`.
(2) Create an empty relational table `TRIP2015`, which has the following columns: `TNUM`, `LNUM`, `REGNUM` with the same types as the columns with the respective names in a relational table `TRIP`. You can re-use slightly updated `CREATE TABLE` statement included in a script `dbcreate3.sql`. Enforce primary key and referential integrity constraints on a relational table `TRIP2015`. Next, copy information about all trips performed in 2015 from `TRIP` to `TRIP2015`.
(3) Use a single `UPDATE` statement to change a status of truck (attribute `STATUS`) to "maintained" for all trucks that have been used at least one time in 2005 and earlier.
(4) Add a column `TOTALTRIPS` to a relational table `DRIVER`. A type of the column must be `DECIMAL(3)`. Next, insert into a column `TOTALTRIPS` the total number of trips performed by each driver.
(5) Use a single `DELETE` statement to remove all legs of all trips performed in 2015.

When ready save your implementations in SQL script file `solution4.sql` and execute a script `solution4.sql`. Save a report from the processing of SQL script file `solution4.sql` in a file `solution4.rpt`.

After each execution of a script `solution4.sql` you must use the scripts `dbdrop3.sql` and `dbcreate3.sql` to drop the modified relational tables and to re-create and re-load the tables with the original data. It will allow you to operate on exactly the same database each time you execute the script `solution4.sql`.

A file `solution4.rpt` must NOT contain the reports from processing of `dbdrop3.sql` and `dbcreate3.sql`.

Deliverables
A file `solution4.rpt` with a report from processing of SQL script `solution4.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