

**CSIT115/CSIT815 Data Management and Security**  
**Assignment 3**  
25 April 2017

---

**Scope**

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

The outcomes of the assignment are due by **Saturday, 13 May, 2017, 7.00 pm sharp**.

This assignment contributes to 6% (5% for CSIT815) of the total evaluation in the subject.

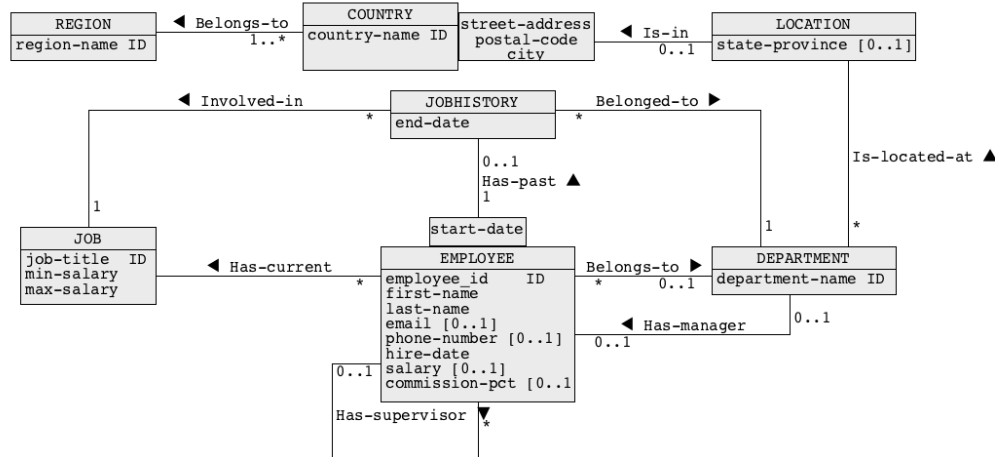
A submission procedure is explained at the end of assignment specification.

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`, `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`.

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**

### **Task1 (1.5 mark)**

Implement the following data manipulations in SQL as the sequences of INSERT, UPDATE, DELETE statements. Note, that each task may need more than one data manipulation statement to be implemented and note, that in some cases you must use advanced DML statements of SQL. An important condition is that you are **NOT ALLOWED** to drop, to alter, and to create any consistency constraints during the modifications.

- (1) A new employee James Clark has been hired at 20/04/2017 for a job of Database analyst. He works in a department Information Technology. The supervisor id is 104. His employee id is 207. His new email address is james.clark@bigmail.com. His salary is 8500. The new employee does not have commission and phone number. A new job Database analyst needs to be added into the database. The salary of the job is between 8000 and 10000.
- (2) A company decides to replace employee\_id 201 with a new employee\_id 208. An email of the employee with employee\_id equal to 201 is MHARTSTE and mobile phone is 515.123.5555. Update related data in the database.
- (3) An employee with employee\_id equal to 102 left the company. Remove all information about the employee from the database. Note that the employee is a supervisor of other employees and he is not a manager of any department.
- (4) A company decided to merge Shipping department with Marketing department. The new department name is called Logistics. It is located in the same place as Shipping department and it has the same manager as Shipping department. Update the database. Note, that after merge all information about both Shipping and Marketing departments must be removed from the database.
- (5) An employee with employee\_id equal to 180 has finished his job at 15/04/2017. Save his old job information in a relational table JOBHISTORY. Now, he is hired as Finance Manager in Purchasing department. Hire date is 20/04/2017. His salary is 4000 and commission percentage is 20. His supervisor is the same as before.

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.

Note, that at a development stage after every processing of a script solution1.sql it is recommended to use the scripts dbdrop.sql, dbcreate.sql, and

dbload.sql to refresh the database with the original data. It allows 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 dbdrop.sql, dbcreate.sql and dbload.sql. A file solution1.rpt must contain only INSERT, DELETE, and UPDATE statements of SQL with optionally embedded SELECT statements.

**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 included in the script solution1.sql.

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 INSERT, DELETE, UPDATE, source, and notee with optionally embedded SELECT statements scores no marks !

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

---

## Task 2 (2 marks)

Connect to MySQL either through command line interface `mysql` or graphical user interface MySQL Workbench. Execute the script files `dbdrop.sql`, `dbcreate.sql`, and `dbload.sql` to refresh a sample database before implementation of Task 2.

Implement the following queries as `SELECT` statements of SQL and save the statements in SQL script file `solution2.sql`. Note, that finding the relevant date processing functions in MySQL Reference Manual

(<https://dev.mysql.com/doc/refman/5.7/en/>)

is a subtask of this assignment.

- (1) Find the full names of employees who have been hired or who had their jobs in 1995.
- (2) Find the names of departments together with the total number of employees working at each department. Include the departments that have no employees.
- (3) Find the full names of supervisors together with the total number of employees directly supervised by each one of them.
- (4) Find the full names of employees who had completed their jobs and worked for more than 1000 days.

When ready execute SQL script `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

Submit 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 file `solution2.rpt` must contain only `SELECT` statements of SQL.

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

A report that contains any 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 `SELECT`, `source`, and `notee` statements scores no marks !

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

---

### Task 3 (1.5 mark)

Connect to MySQL either through command line interface `mysql` or graphical user interface MySQL Workbench. Execute the script files `dbdrop.sql`, `dbcreate.sql`, and `dbload.sql` to refresh a sample database before implementation of Task 3.

Implement the following queries as `SELECT` statements of SQL and save the statements in SQL script file `solution3.sql`. Please pay attention to the statements type in a bold font below !

- (1) Find the full names of all employees who currently working in Canada. **The query must be implemented as a nested query.**
- (2) Find the full names of all employees who have no information about themselves recorded in `JOBHISTORY` table. **The query must be implemented as a correlated nested query with a negated existential quantifier.**
- (3) Find the full names of all employees who have completed two jobs. **The query must be implemented as a nested query.**
- (4) Find the names of regions together with the names of departments located in each region. **The query must be implemented as a join query.**
- (5) Find the names of departments located in Europe. **The query must be implemented as nested query with an 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

Submit 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 file `solution2.rpt` must contain only `SELECT` statements of SQL.

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

A report that contains any 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 `SELECT`, `source`, and `notee` statement scores no marks !

SELECT statement implemented in a different technique from the one required in the specification scores no marks !

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

---

**Task 4 (2 marks)**

Connect to MySQL either through command line interface `mysql` or graphical user interface MySQL Workbench. Execute the script files `dbdrop.sql`, `dbcreate.sql`, and `dbload.sql` to refresh a sample database before implementation of Task 4.

Implement the following subtasks in SQL. Note, that each subtask may need more than one SQL statement to be implemented.

- (1) Use a single SQL statement to create a relational table `EMPCONTACT` that consists of the columns `EID`, `FNAME`, `LNAME`, `EMAIL`, and `PHONENUM` and such that it contains all appropriate data copied from a relational table `EMPLOYEE`. When ready, enforce appropriate consistency constraints on a relational table `EMPCONTACT`.
- (2) Create an empty relational table `SALESEMPLOYEE`, which has the following columns: `EID`, `FNAME`, `LNAME`, and `JOBTITLE` with the same types as the columns with the respective names in a relational table `EMPLOYEE`. Enforce primary key and referential integrity constraints on a relational table `SALESEMPLOYEE`. Copy information about all employees who have job related to Sales into `SALESEMPLOYEE`. Note that if an employee has a job related with Sales then it means that job title contains a word `Sales`.
- (3) Use a single `UPDATE` statement to increase salary by 500 for all employees that have been hired in 1998 and earlier.
- (4) Add a column `TOTSTAFF` to a relational table `DEPARTMENT`. A type of the column must be `DECIMAL (3)` with no other constraints imposed on the column. Next, insert into a column `TOTSTAFF` the total number of employees in each department.
- (5) Use a single `DELETE` statement to remove all data from `JOBHISTORY` related to the employees who completed at least one jobs before 1996.

When ready save your implementations in SQL script file `solution4.sql` and execute a script `solution4.sql`. Note, that your script must contain only SQL statements implementing the subtasks listed above and NO other statements. Save a report from the processing of SQL script file `solution4.sql` in a file `solution4.rpt`.

Note, that at a development stage after every processing of a script `solution4.sql` it is recommended to use the scripts `dbdrop.sql`, `dbcreate.sql`, and `dbload.sql` to refresh the database with the original data and to drop the relational tables `EMPCONTACT` and `SALESEMPLOYEE`. It allows 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 `dbdrop.sql`, `dbcreate.sql` and `dbload.sql`.

**Deliverables**

Submit 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 !

A report that contains any processing errors scores no marks !

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

Processing of the script that contains SQL statements that do not implement the subtasks listed above 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 the correct files with the correct contents. No other submission is possible!**

Submit the files **solution1.rpt**, **solution2.rpt**, **solution3.rpt**, and **solution4.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 (S116) Data Management and Security**
- (4) Scroll down to a section assignment **Submissions**
- (5) Click at a link **in this place you can submit the outcomes of Assignment 3**
- (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 the files **solution2.rpt**, **solution3.rpt**, and **solution4.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 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 third assignment 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 its 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 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 7.00 pm on Saturday, 3 June, 2017.**

---

*End of specification*