

CSCI235/CSCI835 Database Systems
Laboratory 1
3 August 2020

Scope

This laboratory includes implementation of simple SQL scripts in order to refresh your SQL skills and to make you familiar with the command line and graphical interfaces to Oracle database servers. It also includes a task related to discovering the redundancies in the relational tables.

The outcomes of the laboratory work are due by **Saturday 15 August, 2020, 7.00 pm (sharp)**.

Please read very carefully information listed below.

This laboratory contributes to 1% of the total evaluation in the subject.

A submission procedure is explained at the end of specification.

This laboratory consists of 3 tasks and specification of each task starts from a new page.

It is recommended to solve the problems before attending a laboratory class in order to efficiently use supervised laboratory time.

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

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

A submission of compressed files (zipped, gzipped, rared, tared, 7-zipped, lhzed, ... etc) is not allowed. The compressed files will not be evaluated.

All files left on Moodle in a state "Draft (not submitted) " will not be evaluated.

It is expected that all tasks included within **Laboratory 1** 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 the assessment task.

Prologue

Start and connect to your Windows 10 system and to Moodle. Download a conceptual schema `dbschema.bmp` of a sample database and SQL scripts `dbcreate.sql`, `dbload.sql`, `dbcount.sql`, and `dbdrop.sql` included in a section **SAMPLE DATABASES**.

Tasks

Task 1 (0.4 mark)

- (1) To implement Task 1 read in the Cookbook Recipe 2.2 How to access Oracle database servers from a remote system ? It explains how to connect to Oracle database servers from your computer system at home and how to use virtual machine with the installation of Oracle 19c.
- (2) Start `SQLcl` client (command line interface to Oracle database server) and connect to your Oracle 19c database server account. Note, that your Oracle user name and password have been already emailed to you through SOLS email. If you enrolled the subject late and you got no email with your Oracle user name and password then please contact your lecture over email. Try to connect to your account on `csora` Oracle database server first and later to one database server running `data-pc..` systems in 3.125.
- (3) Open a file `dbschema.bmp` with a conceptual schema of a sample database and familiarize yourself with a domain of a sample database. Next, use a text editor to read and to analyse the contents of SQL script `dbcreate.sql` that can be used to create a sample database. Compare the contents of the scrip with the respective conceptual schema. Process SQL script `dbcreate.sql` to create the relational tables. Next, process SQL script `dbload.sql` to load data into the relational tables. Finally, process a script `dbcount.sql` to count the total number of rows in each relational table. You should get the following results.

LINEITEM	ORDERS	CUSTOMER	PARTSUPP	SUPPLIER	PART	NATION	REGION
-----	-----	-----	-----	-----	-----	-----	-----
309	85	16	40	5	10	30	10

No report is expected from the first three steps listed above.

- (4) Implement SQL script `solution11.sql` that performs the following actions.

First, your script modifies the structure of the sample database such that after the modifications it is possible to store information about the total number of parts supplied by each supplier. Please, remember that some supplier may supply no parts at the moment due to a poor economic situation. It is important to find the best design. Remember to enforce appropriate consistency constraints.

Next, your script saves in the sample database information about the total number of parts supplied by each supplier.

When ready use `SQLcl` to process the script and save a report in a file `solution11.lst`. It is explained in the Cookbook Recipe 2.5 How to use `SQLcl` client ?, Step 9 how to create and how to save a report from processing of SQL script.

Your report must include listing of all SQL statement processed. To achieve that put the following `SQLcl` commands:

```
SPOOL solution11
SET ECHO ON
SET FEEDBACK ON
SET LINESIZE 100
SET PAGESIZE 200
```

at the beginning of SQL script `solution11.sql` and

```
SPOOL OFF
```

at the end of the script.

- (5) Start SQL Developer (GUI interface to Oracle database server) and connect to your Oracle 19c database server account. You can connect to anyone of `data-pc..` database servers located in a lab room or to `csora` server.
- (6) Implement SQL script `solution12.sql` that performs the following actions.

First, the script saves in a separate relational table (a name of the table and the names of columns are up to you) information about the names of relational tables included in a sample database and the total number of rows in each table.

Next, the script lists the contents of the new table in the ascending order of the total number of rows in the relational tables.

Use `SQL Developer` to test your solution. It is explained in the Cookbook Recipe 2.3 How to access Oracle database server from `SQL Developer` ? how to use `SQL Developer`.

- (7) When ready use `SQLcl` to process SQL script file `solution12.sql` and to save a report in a file `solution12.lst`. It is explained in the Cookbook Recipe 2.5 How to use `SQLcl` client ?, Step 9 how to create and how to save a report from processing of SQL script.

Your report must include listing of all SQL statement processed. To achieve that put the following SQL*Plus commands:

```
SPOOL solution12  
SET ECHO ON  
SET FEEDBACK ON  
SET LINESIZE 100  
SET PAGESIZE 200
```

at the beginning of SQL script `solution12.sql` and

```
SPOOL OFF
```

at the end of the script.

Deliverables

The files `solution11.lst` and `solution12.lst` that contain the reports from processing of SQL script `solution11.sql` and `solution12.sql`. The reports 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 any kind of processing errors scores no marks !

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

Task 2 (0.6 mark)

Analyze a collection of incorrectly designed relational schemas listed below.

To find what is wrong with the relational schemas listed below use a method of row insertions explained in a presentation 01 Database Design Quality. Insert into the relational tables with the schemas (headers) listed below from 3 to 5 rows that demonstrate the redundancies.

Include into a file `solution2.pdf` the drawings of relational tables with redundancies and briefly explain the reasons behind each redundancy. The scanned neat hand drawings are acceptable.

`STUDENT(snumber, first-name, last-name, ccode)`

A relational table `STUDENT` contains information about the students and the courses enrolled by the students. A course (`ccode`) is enrolled by more than one students (`snumber`) and each student enrolls several course. Student number (`snumber`) uniquely identifies each students and course code (`ccode`) uniquely identifies each course. The first (`first-name`) and the last (`last-name`) names describe the students.

`HOTEL(name, city, capacity, enumber, salary)`

A relational table `HOTEL` contains information about the hotels and employees working in the hotels. A hotel is identified by a pair of attributes (`name, city`) and it is also described by the total number of rooms available (`capacity`). Each employee is identified by employee number (`enumber`) and it is described by a salary (`salary`).

`TEAM(tname, player, supporter)`

A relational table `TEAM` contains information about football teams, football players who belong to the teams and supporters of the teams. Each football team is described a unique name (`tname`). Players and supporters are described by unique names (`player`) and (`supporter`). A team has many players and many supporters.

Deliverables

A file `solution2.pdf` with the drawings of relational tables with redundancies and the brief explanations of the reasons behind each redundancy

Submission

Submit the files **solution11.lst**, **solution12.lst**, and **solution2.pdf** 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 **CSCI835/CSCI235 (S220) Database Systems**
- (4) Scroll down to a section **SUBMISSIONS**
- (5) Click at a link **In this place you can submit the outcomes of Laboratory 1**
- (6) Click at a button **Add Submission**
- (7) Move a file **solution11.lst** into an area **You can drag and drop files here to add them**. You can also use a link **Add...**
- (8) Repeat a step (7) for the files **solution12.lst**, and **solution2.pdf**.
- (9) Click at a button **Save changes**
- (10) Click at a button **Submit assignment**
- (11) 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.
- (12) Click at a button **Continue**

End of specification