School of Computing and Information Technology
University of Wollongong

Session: **Autumn 2019**
Lecturers: **Tianbing Xia & Janusz Getta**

CSCI235/CSCI835 Database Systems
Assignment 1
Published on 25 March 2019

Scope

This assignment includes the tasks related to implementation in PL/SQL stored procedures, stored functions, statement and row triggers.

More implementation related information can be found in "How to … ?" Cookbook available through Moodle or at:
`http://www.uow.edu.au/~jrg/235/COOKBOOK`.

The outcomes of the assignment work are due by **Saturday, 6 April, 2019, 11.55 pm (sharp).**

This assignment contributes to 10% of the total evaluation in the subject.

A submission procedure is explained at the end of specification.

This assignment consists of 4 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.

Only one submission of the outcomes of Assignment 1 is allowed and only one submission per student is accepted. Please make sure that you submit the correct files.

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.

Compressed (zipped, rared, tared, etc) files will not be evaluated.

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

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

A report from processing of SQL script that contains no listing of processed SQL and/or PL/SQL statements scores no marks.

A report from processing of SQL script 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!

A policy regarding late submissions is included in CSCI235/CSCI835 Subject Outline.

Prologue

Start and connect to your Windows 7 system (if it is not started yet and you are not connected). Download the script files `dbcreate.sql` and `dbdrop.sql` from the links of the `SAMPLE DATABASE`. Execute the script file `dbcreate.sql` to create the sample database. Execute the script file `dbdrop.sql` to drop all tables in the sample database later.

Do not drop the relational tables now.

Tasks

Task 1. Stored procedure (2.5 marks)

Implement a stored PL/SQL procedure `APPLICATIONS` to list the applicants and their applications.

The names of applicants must be listed in the descending order of last names of applicants. The position number and title of a position applied by an applicant must be listed in the ascending order of position number.

Execute the stored PL/SQL procedure `APPLICATIONS`. A fragment of expected sample printout is given below.

```
16 Zhi Chao Zhong:
2  Johnny Walker:
   1 lecturer
   6 professor
14 Ivan TheTerrible:
17 Richard TheLionheart:
…
```

Implement a solution as PL/SQL stored procedure and save it in SQL script file `solution1.sql`. Then, process the script and save a report in a file `solution1.lst`. It is explained in Cookbook, Recipe 2.5 "How to use SQL*Plus 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:

```
SET ECHO ON
SET FEEDBACK ON
SET LINESIZE 100
SET PAGESIZE 100
SET SERVEROUTPUT ON
```

at the beginning of SQL script `solution1.sql`.

Deliverables

Submit a file `solution1.lst` 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 PL/SQL statements and control statements that implement and execute the procedure of Task 1 and NO OTHER statements.

Task 2. Stored function (2.5 marks)
Implement a stored PL/SQL function `APPLICANTSKILLS` that takes an applicant number (`anumber`) as a parameter, and finds all the skills possessed by the applicant.

The function must return a string of characters that contains the first and last name of an applicant, skill name and level that the applicant possessed.

Execute the stored PL/SQL function `APPLICANTSKILLS` for all applicants. A fragment of sample printout is given below:

```
1 Harry Potter: C programming 4 Java programming 9
cooking 9
2 Johnny Walker: Java programming 9 driving 9
3 Mary Poppins: C++ programming 10 Java programming 9
painting 5
4 Michael Collins:
5 Margaret Finch: SQL programming 6
6 Claudia Kowalewski: SQL programming 8
…
```

Save your implementation of Task 2 in SQL script file `solution2.sql`. Then, process the script and save a report in a file `solution2.lst`. It is explained in Cookbook, Recipe 2.5 "How to use SQL*Plus 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:

```
SET ECHO ON
SET FEEDBACK ON
SET LINESIZE 100
SET PAGESIZE 200
SET SERVEROUTPUT ON
```

at the beginning of SQL script `solution2.sql`.

Deliverables
Submit a file `solution2.lst` 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**. The report MUST include ONLY PL/SQL statements and control statements that implement and execute the function of Task 2 and NO OTHER statements.

Task 3 Statement trigger (2.5 marks)
Implement and comprehensively test a **`statement trigger`** that verifies the following consistency constraint.

> *"A position cannot need more than 4 skills".*

When ready save your `CREATE TRIGGER` statement and all SQL statements that comprehensively test a trigger in a script `solution3.sql`. Comprehensive testing means that the trigger must reject SQL statements that violate the consistency constraint and accept SQL statements that do not violate the consistency constraint. It is a part of your task to find what SQL statements should be tested. Whenever SQL statement violates the consistency constraint a trigger must return `ORA-...` error message. Use a procedure `RAISE_APPLICATION_ERROR` to return `ORA-...` error message. If SQL statement does not violate the consistency constraint then a trigger must return no messages.

Process SQL script file `solution3.sql` and save a report from processing in a file `solution3.lst`.

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

```
SET ECHO ON
SET FEEDBACK ON
```

at the beginning of SQL script `solution3.sql`.

Deliverables
Submit a file `solution3.lst` with a report from processing of SQL script `solution3.sql`. The report MUST have no errors other than reported by a trigger and the report MUST list all SQL statements processed. The report MUST include ONLY SQL statements and comprehensively test statements that implement the specifications of Task 3 and NO OTHER statements.

Task 4 Row trigger (2.5 marks)
Implement and comprehensively test a **`row trigger`** that verifies the following consistency constraint.

> *"An applicant cannot apply for a position during the last 30 days from his/her the previous application for the same position".*

Hint:
No need to consider the UPDATE event.

When ready save your `CREATE TRIGGER` statement and all SQL statements that comprehensively test a trigger in a script `solution4.sql`. Comprehensive testing means that the trigger must reject SQL statements that violate the consistency constraint and accept SQL statements that do not violate the consistency constraint. It is a part of your task to find what SQL statements should be tested. Whenever SQL statement violates the consistency constraint a trigger must return `ORA-...` error message. Use a procedure `RAISE_APPLICATION_ERROR` to return `ORA-...` error message. If SQL statement does not violate the consistency constraint then a trigger must return no messages.

Process SQL script file `solution4.sql` and save a report from processing in a file `solution4.lst`.

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

```
SET ECHO ON
SET FEEDBACK ON
```

at the beginning of SQL script `solution4.sql`.

Deliverables
Submit a file `solution4.lst` with a report from processing of SQL script `solution4.sql`. The report MUST have no errors other than reported by a trigger and the report MUST list all SQL statements processed. The report MUST include ONLY SQL statements and comprehensively test statements that implement the specifications of Task 4 and NO OTHER statements.

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.lst**, **solution2.lst**, **solution3.lst** and **solution4.lst** to 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 (S119) Database Systems**
(4) Scroll down to a section **SUBMISSIONS**
(5) Click at a link **In this place you can submit the outcomes of Assignment 1**
(6) Click at a button **Add Submission**
(7) Move a file **solution1.lst** 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.lst**, **solution3.lst** and **solution4.lst.**
(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**

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

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

End of specification