

Assignment 2 (15% of total marks)

Due date: Sunday, 10 May 2020

Scope:

The tasks of this assignment cover **stored PL/SQL procedure, function, and trigger**. The assignment covers the topics discussed in lecture 7, 8, and 9.

Assessment criteria:

Marks will be awarded for:

- Correct,
- Comprehensive, and
- Appropriate

application of the materials covered in this subject.

Assignment Specification:

Preliminary actions

Download SQL script dbCreateTruck-2020SP20.sql from Moodle. Execute the script to create and to load a sample database for trips made by drivers. The database contains information about the employees working for a transportation company, drivers employed by the company, trucks owned by the company, trips made by the drivers and trucks, and all legs of each trip. It is strongly recommended to discover a conceptual schema (UML diagram) of the database. However, there will be no mark awarded for producing the conceptual schema.

Task 1 (7.0 marks)

Stored PL/SQL Procedure

Implement a stored PL/SQL procedure that insert into a database full information about an employee i.e., the values of the following attributes E#, NAME, DOB, ADDRESS, HIREDATE, L#, STATUS, EXPERIENCE (only for mechanics).



Your procedure should enforce a logical consistency constraint such that "the sets of drivers and mechanics must be disjoint". It means that it is not allowed to have in the tables MECHANIC and DRIVER the rows with the same employee number (E#) and/or the same driving license number (L#).

Execute your procedure twice. The first execution should insert full information about a new employee. The second execution should fail due to the violation of logical consistency constraint given above.

Deliverables

Hand in your solution1.lst in pdf format. Your report **MUST** have no errors and the report **MUST** list all SQL statements processed. All SQL statements of the script must be executed with SET ECHO ON, SET FEEDBACK ON and SET SERVEROUTPUT ON options of SQL*Plus. It is a good idea to set the options at the beginning of the script. The script must be implemented with SQL*Plus. Oracle database server used for the testing and the final execution of the script is up to you. **The printouts that do not satisfy the requirements listed above will score NO MARKS!**

Task 2 (5.0 marks) Stored PL/SQL Trigger



Implement **a row trigger** that enforces the following consistency constraint.



A column **totalTripMade** in the relational table **DRIVER** is currently does not contain any value. Create a row trigger that automatically updates the values in the column (totalTripMade) when a new trip made by a driver is **inserted into the relational table TRIP**. Your trigger, once activated, will compute the total number of trips made by the driver and update the totalTripMade column in the relational table DRIVER. NOTE: You do not need to consider any other cases that may change the value in the column totalTripMade; that is, **NO NEED to consider delete and update cases**.

When ready, process the SQL script solution2.sql and record the results of processing in a file solution2.lst.

Deliverables

Hand in your solution2.lst in pdf format. Your report **MUST** have no errors and the report **MUST** list all SQL statements processed. All SQL statements of the script must be executed with SET ECHO ON, SET FEEDBACK ON and SET SERVEROUTPUT ON options of SQL*Plus. It is a good idea to set the options at the beginning of the script. The script must be implemented with SQL*Plus. Oracle database server used for the testing and the final execution of the script is up to you. **The printouts that do not satisfy the requirements listed above will score NO MARKS!**

Task 3 (3.0 marks)

Stored PL/SQL function

Implement a stored **PL/SQL function** LONGTRIP(DLNUM) that finds the length (the total number of legs) of the longest trip performed by a driver identified by a driving license number (L# attribute in table DRIVER and parameter DLNUM parameter in the function). Remember to include or consider the drivers **that performed no trips**; that is to say, do not ignore drivers that do not perform any trip. **If a driver has not performed any trip, then output the longest trip a 0.**

Use a stored function LONGTRIP in SELECT statement to list the names of all drivers together with the length of the longest trip performed by each driver.

Deliverables

Hand in your solution3.lst in pdf format. Your report **MUST** have no errors and the report **MUST** list all SQL statements processed. All SQL statements of the script must be executed with SET ECHO ON, SET FEEDBACK ON and SET SERVEROUTPUT ON options of SQL*Plus. It is a good idea to set the options at the beginning of the script. The script must be implemented with SQL*Plus. Oracle database server used for the testing and the final execution of the script is up to you. **The printouts that do not satisfy the requirements listed above will score NO MARKS!**

Submissions

This assignment is due by 9:00 pm (2100 hours) Sunday, 10 May 2020, Singapore time.

Submit the files **solution1.pdf, solution2.pdf, solution3.pdf** through Moodle in the following way:

- 1) Zip all the files (Solution1.pdf, solution2.pdf and solution3.pdf) into one zipped folder. Name your zipped file as YourName-A3)
- 2) Access Moodle at **<http://moodle.uowplatform.edu.au/>**
- 3) 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
- 4) When successfully logged in, select a site CSCI235 (SP220) Database Systems
- 5) Scroll down to a section Submissions of Assignments
- 6) Click at Submit your Assignment 1 here link.
- 7) Click at a button Add Submission
- 8) Move the zipped file created in Step 1 above into an area provided in Moodle. You can drag and drop files here to add them. You can also use a link *Add...*
- 9) Click at a button Save changes,
- 10) Click at check box to confirm authorship of a submission,
- 11) When you are satisfied, remember to click at a button Submit assignment.

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

Only one submission per student is accepted.

Assignment 2 is an individual assignment and it is expected that all its tasks will be solved individually without any cooperation with the other students. Plagiarism is treated seriously. Students involved will likely receive zero. If you have any doubts, questions, etc. please consult your lecturer or tutor during lab classes or over e-mail.

End of specification