# Assignment 2 (15% of total marks)

**Due date:** Sunday, 11 May 2025

**Scope:**

The tasks of this assignment cover topics on **PLSQL**.

## Assessment criteria:

Marks will be awarded for:

- Correct,
- Comprehensive, and
- Appropriate

application of the materials covered in this subject.

**Please read carefully information listed below.**

This assignment contributes to 15% of the total assessment mark for the subject CSCI235.

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.

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

SQL Script Requirements:

- Include the following settings at the **beginning of your script**:
  - `\set ECHO all` – This will ensure that all SQL commands are echoed to the terminal.
  - `\set ECHO none` – Use this to stop echoing the commands when needed.

## Assignment Specification:

## Task 1 (5.0 marks)

### Stored PL/SQL procedure

Implement a **stored** PL/SQL procedure SUPPLIERACCBAL that lists information about the account balances of suppliers in nations within a specified region. The procedure first computes the average account balance of a nation. The procedure then extracts the supplier's information within a nation and determine if the supplier's account balance is above or below the nation's average. The information to be displayed include the supplier name, supplier phone number, the account balance, and a comment indicating whether the account balance is above or below the nation's average. An example of a segment of the output for suppliers from various nation within the region of ASIA is as follow:

Nation name: CHINA

| Supplier name | Supplier phone | Account balance | Comment |
|---|---|---|---|
| Supplier#000001221 | 28-332-756-9313 | $4,533.75 | The account balance is above the nation average of $4,369.32. |
| Supplier#000002325 | 28-167-932-2440 | $4,875.37 | The account balance is above the nation average of $4,369.32. |
| Supplier#000000255 | 28-629-327-4139 | $4,663.08 | The account balance is above the nation average of $4,369.32. |
| Supplier#000000302 | 28-734-845-8630 | $4,422.77 | The account balance is above the nation average of $4,369.32. |

Nation name: INDIA

| Supplier name | Supplier phone | Account balance | Comment |
|---|---|---|---|
| Supplier#000001037 | 18-415-126-3978 | $4,176.78 | The account balance is below the nation average of $4,554.48. |
| Supplier#000000136 | 18-175-739-8397 | $4,623.48 | The account balance is above the nation average of $4,554.48. |
| Supplier#000000060 | 18-550-360-2464 | $4,515.80 | The account balance is below the nation average of $4,554.48. |
| Supplier#000001516 | 18-431-532-9957 | $4,755.71 | The account balance is above the nation average of $4,554.48. |
| Supplier#000001906 | 18-738-147-3630 | $4,978.10 | The account balance is above the nation average of $4,554.48. |

…

It is up to you to decide if you want to handle exception in your procedure.

### Deliverables

Hand in an SQL script and the report from execution of the script. The report must have no errors related to the implementation of your task and it must list all PL/SQL and SQL statements processed.

## Task 2 (5.0 marks)

### Stored PL/SQL Function

Implement a **stored** PL/SQL function that finds a supplier who supplies the cheapest (lowest supply cost) or the dearest (highest supply cost) for a part specified by a user. The function should obtain the part key as its formal parameter, and it should return a string of values consisting of the supplier key, supplier name, and the cost (cheapest and dearest). Example of the string output are as follow:

```
Supplier with cheapest cost: 1344, Supplier#000001344, $172.09.
Supplier with dearest cost: 2882, Supplier#000002882, $791.07.

Supplier with cheapest cost: 770, Supplier#000000770, $74.55.
Supplier with dearest cost: 2308, Supplier#000002308, $826.35.
```

Next use the function to list the string output as specified above for part numbers 3753, 43064, 57574 and 60000 using a select statement.

### Deliverables

Hand in an SQL script and the report from execution of the script. The report must have no errors related to the implementation of your task and it must list all PL/SQL and SQL statements processed.

## Task 3 (5.0 marks)

**Stored trigger**

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

The column c_comment in the relational table CUSTOMER of the TPCHR benchmark database is defined as 'NOT NULL'. Create a row trigger that automatically updates the values in the column (c_comment) to 'New customer was created on <the system date>' if the comment of the newly inserted record is left as NULL when a new customer is inserted into the relational table CUSTOMER. Your trigger, once activated, will enforce the consistency constraint described.
When ready, process the SQL script solution2.sql and record the results of processing in a file solution2.lst.

**Deliverables**

Hand in the SQL script and the report from execution of scripts.

**Submissions**

This assignment is due by 9:00 pm (21:00 hours) 11 May 2025, **Singapore time**.

Submit the files **solution1.pdf**, **solution2.pdf**, and **solutions3.pdf** through Moodle in the following way:

1) Zip all the files (Solution1.pdf, solution2.pdf, and solution3.pdf into one zipped folder.)
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 (SP225) Database Systems
5) Scroll down to a section Submissions of Assignments
6) Click at Submit your Assignment 2 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*