# Lab 5 – Week 7

# (Stored Procedures/Conditional Statements)

## Submission

***Your submission will be a single text-based SQL file with appropriate header and commenting. Please ensure your file runs when the entire file is executed in SQL Developer.***

Create a new Worksheet in SQL Developer. Save the file as L05\_ID#\_LASTNAME.sql

## Your submission needs to be commented and include the question, the solutions.

In this Lab, you create PL/SQL stored procedures to perform the following tasks. As you know, a stored procedure does not return any value. To send values back to the caller, you can use OUT parameters. A parameter can be

* IN patameter
* OUT parameter
* IN OUT parameter

See the following template: (Make sure you separate the stored procedures by “/” so you can run all statements in your .sql using “Run Script” botton).

|  |
| --- |
| **CREATE** **OR REPLACE** *procedure1\_name*(**arg1** **IN**/**OUT/IN OUT** data\_type, ...) AS  **BEGIN**  ....  **EXCEPTION**  **WHEN OTHERS**  **THEN**  DBMS\_OUTPUT.PUT\_LINE ('Error!');  **END** procedure1\_name;  /  **CREATE** **OR REPLACE** *procedure2\_name*(**arg1** **IN**/**OUT/IN OUT** data\_type, ...) AS  **BEGIN**  ....  DBMS\_OUTPUT.PUT\_LINE (Error!');  **END** procedure2\_name; |

For all the stored procedures make sure you handle all exceptions such as

* TOO\_MANY\_ROWS
* NO\_DATA\_FOUND
* OTHERS
* . . .

Besides checking all required exceptions, have the OTHER exception checked just in case any error occurs that has not been anticipated at the time you write the code.

## Tasks

1.Write a store procedure that gets a string and prints the string (with all letters capital) and the number of characters in that string.See the following sample outputs:

Input: flower  
FLOWER has 6 chracters.

Input: Te$t  
TE$T has 4 characters.

The procedure displays a proper error message if any error occurs.

|  |
| --- |
| 2.The company needs a report that shows three categories of customers based on their credit limits: low, average, and high. To determine the category of customers, the minimum credit limit, maximum credit limit, and the average credit limit of all customers must be calculated.   * If the credit limit is less than (average credit limit – minimum credit limit) / 2, then: The customer’s credit limit is low. * If the credit limit is greater than (maximum credit limit – average credit limit) / 2, then: The customer’s credit limit is high. * If the credit limit is between (average credit limit - minimum credit limit) / 2 and (maximum credit limit - average credit limit) / 2, then:  The customer’s credit limit is average.   Write a procedure named *credit\_report* to show the number of customers in each category: See the following sample output: The number of customers with average credit limit: 23 The number of customers with high credit limit: 55 The number of customers with low credit limit: 17    The values in the above examples are just random values and may not match the real numbers in your result. The procedure has no parameter. First, you need to find the average, minimum, and maximum credit limit in your database and store them into variables avg\_credit, min\_credit, and max\_credit. You need more three variables to store the number of customers in each category: avg\_count high\_count low\_count Make sure you choose a proper type for each variable. You may need to define more variables based on your solution.  The procedure displays a proper error message if any error occurs. |
| |  |  | | --- | --- | |  |  | |  |  | |  |  | |  |  | |

3.Write a store procedure that gets an integer number as an employee ID and prints the number of years the employee has been working in the company. See the following sample output:  
The employee with ID 1004 has worked 5 years.  
The values in the sample output are random numbers and may not match the real numbers in the database.

The procedure displays a proper error message if any error occurs.

|  |
| --- |
|  |
| 4. NOTE: For this question, create a table exactly the same as the products table.  CREATE TABLE new\_products AS SELECT \* FROM products;  Do not include the create statement in your answer.  Use the new\_products in this question to update list prices. Every year, the company increases the price of all products in some categories by 2%. Write a stored procedure that gets an integer number as category ID named categoryID and a number named amount of type NUMBER(9,2) and increases the list price of all products in that category by 2% if the average list price of that category is less than the given amount. Name the stored procedure update\_price. The procedure gets two parameters: •    category\_id IN NUMBER •    amount NUMBER(9,2)  The procedure displays a proper error message if any error occurs. |
|  |
| |  | | --- | |  | |  | |

Example Submission

-- Question 1 – write a brief note about what the question is asking  
-- Q1 SOLUTION –

CREATE OR REPLACE procedure\_name(arg1 data\_type, ...) AS

BEGIN

....

EXCEPTION

WHEN OTHERS

THEN

DBMS\_OUTPUT.PUT\_LINE (Error!');

END procedure\_name;  
  
-- Question 2 –

-- Q2 Solution –