Experiment No 25: Create functions for given database

I. **Practical Significance:** PL/SQL function is a named, self-contained block of code that performs a specific task and returns a value. Functions are designed to encapsulate logical operations, promoting code modularity and reusability. Developers can build a more maintainable and efficient PL/SQL codebase by isolating specific tasks within functions. This practical allows students to implement PL/SQL Functions for the given database.

II. INDUSTRY / EMPLOYER EXPECTED OUTCOME:

To implement PL/SQL function on database.

III. COURSE LEVEL LEARNING OUTCOMES (COS):

CO4 - Implement PL/SQL codes for given application.

IV. LABORATORY LEARNING OUTCOME:

Implement Function for given database.

V. Relevant Affective Domain related outcome(s)

a. Follow precautionary measures.

b. Follow installation steps.

c. Follow ethical practices.

VI. Relevant Theoretical Background

A typical PL/SQL function consists of the following components:

Function Declaration

The function declaration serves as the initial step in defining a PL/SQL function. It includes the function's name, input parameters (if any), and the return type. Parameters can be either input-only or input-output, allowing for dynamic and flexible function behavior.

Local Declarations

Within the function's block, local declarations allocate memory for variables that are used to store intermediate values or perform calculations. These variables are confined to the scope of the function, preventing them from interfering with other parts of the codebase.

Function Body

The function body contains the actual logic of the function. It consists of a series of statements that manipulate the input parameters and local variables to achieve the desired outcome. This is where the magic happens, as developers can employ conditional statements, loops, and other programming constructs to implement complex operations.

Return Statement

The return statement marks the end of the function and specifies the value that will be returned to the caller. It is essential to ensure that the return type matches the one declared in the function header.

SYNTAX:

```
FUNCTION function_name (parameter1 [IN | OUT | IN OUT] datatype, parameter2 [IN | OUT
| IN OUT] datatype, ...)
  RETURN return_datatype
IS
  -- Declaration section (optional)
  variable1 datatype;
  variable2 datatype;
  ...
BEGIN
  -- Function body
  -- SQL and PL/SQL statements
  RETURN return_value;
EXCEPTION
  -- Exception handling (optional)
  WHEN exception_name1 THEN
    -- Handle exception 1
  WHEN exception_name2 THEN
    -- Handle exception 2
  ...
END;
```

VII. **Required Resources/apparatus/equipment with specifications**

Sr.No	Equipment Name with Broad Specifications	Relevant LLO Number
1	Computer system with all necessary components like; motherboard, random access memory (RAM), read-only memory (ROM), internal hard disk drives, Mouse, Keyboard, and RDBMS applications such as Oracle Express Edition,MySql,SQLite,Oracle Apex etc.	All

VIII. **Procedure**

1.Define the PL/SQL block structure

2.Implement the logic for the given problem

IX. **Result(s)**

In this practical we studied to create functions for given database.

X. **Practical related questions (Provide space for answers)**

Note: Below are a few sample questions for reference. Teacher must design more such questions to ensure the achievement of identified CO.

1. What is the difference between Function and Procedure.
2. Write syntax for creating and replacing function.

(Space for answer)

2] →

```
CREATE OR REPLACE FUNCTION <procedure_name>
(
<parameter 1 IN/OUT <datatype>
)
RETURN <datatype> [IS/AS]
   <declaration_part>
BEGIN
      <execution part>
Exception
      <exception handling part>
END;
```

I →

	Procedures	Functions
①	Execute as a PL/SQL Statement	① Invoke as part of an expression
②	Do not Contain RETURN Clause in the header	② Must Contain a RETURN Clause in the header
③	Can return Values (if any) in output parameters	③ Must return a Single Value
④	Can contain a RETURN statement without a Value	④ Must contain at least one RETURN Statement

XI. Exercise

1. Write PL/SQL function which will compute and return the maximum of two values.
2. Write PL/SQL function to calculate the factorial of given no.

XII. References/Suggestions for further reading: include websites/links

1. https://www.youtube.com/watch?v=yGU4YfSSjdM
2. https://www.javatpoint.com/pl-sql-Functions.
3. https://www.tutorialspoint.com/plsql/plsql_functions.html
4. PL/SQL Developer - Allround Automations

* Exercise

1] ⟶

```
CREATE OR REPLACE FUNCTION get_max (
P_Value1  IN NUMBER,
P_Value2  IN NUMBER
) RETURN NUMBER is V_Max NUMBER;
BEGIN
if P_Value1 > P_Value2  THEN
V_max := P_value1;
ELSE
V_max  := P_Value2;
END if;
RETURN V_max;
END get_max;
 /


-- Use Function
DECLARE
 V_result NUMBER;
BEGIN
 V_result := get_max (10 , 20);
DBMS_OUTPUT.PUT_LINE ('The maximum value
                  is:' || v_result);
END;
 /
```

2] →

```
CREATE OR REPLACE FUNCTION Calculate_Factorial
(
 p_number IN NUMBER
) RETURN NUMBER IS
V_result NUMBER :=1;
 BEGIN
For i IN 1..p_number Loop
V_result := V_result * i;
END Loop;
RETURN V_result;
END calculate_Factorial;
/


-- USE Function


DECLARE
V_factorial NUMBER;
BEGIN
V_factorial:= calculate_factorial(5);
DBMS_OUTPUT.PUT_LINE ('The factorial of 5
    is:' || V_factorial);
END;
/
```