# EXERCISE-16

## PROCEDURES AND FUNCTIONS

### PROCEDURES

### DEFINITION

A procedure or function is a logically grouped set of SQL and PL/SQL statements that perform a specific task. They are essentially sub-programs. Procedures and functions are made up of,

- Declarative part
- Executable part
- Optional exception handling part

These procedures and functions do not show the errors.

### KEYWORDS AND THEIR PURPOSES

**REPLACE:** It recreates the procedure if it already exists.

**PROCEDURE:** It is the name of the procedure to be created.

**ARGUMENT:** It is the name of the argument to the procedure. Paranthesis can be omitted if no arguments are present.

**IN:** Specifies that a value for the argument must be specified when calling the procedure ie. used to pass values to a sub-program. This is the default parameter.

**OUT:** Specifies that the procedure passes a value for this argument back to it's calling environment after execution ie. used to return values to a caller of the sub-program.

**INOUT:** Specifies that a value for the argument must be specified when calling the procedure and that procedure passes a value for this argument back to it's calling environment after execution.

**RETURN:** It is the datatype of the function's return value because every function must return a value, this clause is required.

### PROCEDURES – SYNTAX

```
create or replace procedure <procedure name> (argument {in,out,inout} datatype ) {is,as}
variable declaration;
constant declaration;
begin
PL/SQL subprogram body;
exception
exception PL/SQL block;
end;
```

### FUNCTIONS – SYNTAX

```
create or replace function <function name> (argument in datatype ) return datatype {is,as}
```

## Program 1

## FACTORIAL OF A NUMBER USING FUNCTION

```
CREATE OR REPLACE FUNCTION get_factorial ( n IN NUMBER)
RETURN NUMBER
IS
   v_fact NUMBER := 1;
BEGIN
   -- Loop from 1 to n (or n down to 1)
  FOR i IN 1..n LOOP
  v_fact := v_fact * i;
  END. LOOP;
  RETURN v_fact;
END;


SET SERVEROUTPUT ON;
DECLARE
   v_num NUMBER := 5;
   v_result NUMBER;
BEGIN
    v_result := get_factorial (v_num);
    DBMS_OUTPUT.PUT_LINE ('Factorial of' || v_num || 'is' ||
 v_result);
END;
```

## Program 2

**Write a PL/SQL program using Procedures IN,INOUT,OUT parameters to retrieve the corresponding book information in library**

```
CREATE TABLE library-books ( book_id NUMBER PRIMARY KEY,
title VARCHAR2(100), author VARCHAR2(100), checkout_count NUMBER DEFAULT 0
); INSERT INTO library-books (book_id, title, author) VALUES (101, 'The
Hobbit', 'J.R.R.Tolkien'); INSERT INTO library_books (book_id, title, author)
VALUES (102, '1984', 'George Orwell'); COMMIT;

SET SERVEROUTPUT ON;
DECLARE
    v_id NUMBER := 101;
    v_title VARCHAR2(100):
    v_author VARCHAR2(100);
    v_checkouts NUMBER := 0;
BEGIN
    get_book_info(v_id, v_title, v_author, v_checkouts);
    DBMS_OUTPUT.PUT_LINE('Title: ' || v_title);
```

| Evaluation Procedure | Marks awarded |
|---|---|
| PL/SQL Procedure(5) | |
| Program/Execution (5) | |
| Viva(5) | |
| Total (15) | |
| Faculty Signature | |