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}
variable declaration;
```

101	2500	500	201
102	3000	1600	202
103	4000	600	202

PROCEDURE FOR 'IN' PARAMETER – CREATION, EXECUTION

```
SQL> set serveroutput on;

SQL> create procedure yyy (a IN number) is price number;
 2  begin
 3  select actualprice into price from ititems where itemid=a;
 4  dbms_output.put_line('Actual price is ' || price);
 5  if price is null then
 6  dbms_output.put_line('price is null');
 7  end if;
 8  end;
 9  /
Procedure created.

SQL> exec yyy(103);
Actual price is 4000
PL/SQL procedure successfully completed.
```

PROCEDURE FOR 'OUT' PARAMETER – CREATION, EXECUTION

```
SQL> set serveroutput on;

SQL> create procedure zzz (a in number, b out number) is identity number;
 2  begin
 3  select ordid into identity from ititems where itemid=a;
 4  if identity<1000 then
 5  b:=100;
 6  end if;
 7  end;
 8  /
Procedure created.

SQL> declare
 2  a number;
 3  b number;
 4  begin
 5  zzz(101,b);
 6  dbms_output.put_line('The value of b is '|| b);
 7  end;
 8  /
The value of b is 100
PL/SQL procedure successfully completed.
```

PROCEDURE FOR 'INOUT' PARAMETER – CREATION, EXECUTION

```
SQL> create procedure itit ( a in out number) is
 2  begin
 3  a:=a+1;
```

Program 1

FACTORIAL OF A NUMBER USING FUNCTION

```
SET SERVEROUTPUT ON;
CREATE OR REPLACE FUNCTION factorial (n Number)
RETURN NUMBER.
IS
fact NUMBER := 1;
BEGIN
 FOR i IN 1; n. LOOP
  fact := fact * i;
 END LOOP;
 RETURN fact;
 END;
/
DECLARE
num NUMBER := 5;
result NUMBER;
BEGIN
result := factorial (num);
DBMS_OUTPut. PuT_LINE ('factorial of ' || num || ' is ' || result);
 END;
/
```

Program 2

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

```
SET SERVER OUTPUT ON;
Create OR REPLACE Procedure get_book info (P_book_id IN number, P_
  -name OUT VARCHAR2, P_author OUT
VARCHAR2, P_Price IN OUT NUMBER) is
BEGIN
SELECT book_name, author price INTO P_book name, P_author,
P_price from library where book_id = P_book_id;
DBMS_OUTPUT_LINE ("Book Name: "||P_book_name);
DBMS_ OUTput_ Line ('Author: '||P_Author);
DBMS_output_ line (Price: '||P_price);
Exception.
  when NO_Data_found then
  DBMS_ output.PUT_ line ("No book found with ID'|| P_book_id
  END;
  DECLARE.
  V_ book_name VARCHAR 2(50);
  V_ author VARCHAR2(50);
V_price number := 0;
BEGIN
get_book_info(101,V, book_na
  V_author, V_pri
DBMS_output.Put_line ("Book in
  Retrieved Successfully.")
END;
```

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