NAME: RAHUL DHANOLA ***ROLL NUMBER:R177219139*** ***BATCH:4***

EXPERIMENT NO:- 10

OBJECTIVE: **To understand the concepts of function and procedure in PL/SQL.**

1) **Create Write a PL/SQL code to accept the value of A, B & C display which is greater.**

```
SQL> set serveroutpu on
SQL> CREATE OR REPLACE FUNCTION GREATEST(A IN NUMBER,B IN NUMBER,C IN NUMBER)RET
URN NUMBER IS G NUMBERS
  2  BEGIN
  3  IF(A>B) THEN
  4  G:=A;
  5  ELSE
  6  G:=B;
  7  IF(C>G) THEN
  8  G:=C;
  9  END IF;
 10  END IF;
 11  RETURN(G);
 12  END;
 13  /

Warning: Function created with compilation errors.

SQL> DECLARE A NUMBER(3);
  2  B NUMBER(3);
  3  C NUMBER(3);
  4  D NUMBER(3);
  5  BEGIN
  6  A:=&A;
  7  B:=&B;
  8  C:=&C;
  9  D:=GREATEST(A,B,C);
 10  DBMS_OUTPUT.PUT_LINE('GREATEST AMONG THREE IS:');
 11  DBMS_OUTPUT.PUT_LINE(D);
 12  END;
 13  /
Enter value for a: 45
old   6: A:=&A;
new   6: A:=45;
Enter value for b: 35
old   7: B:=&B;
new   7: B:=35;
Enter value for c: 25
old   8: C:=&C;
new   8: C:=25;
GREATEST AMONG THREE IS:
45

PL/SQL procedure successfully completed.

SQL>
```

2) Using PL/SQL Statements create a simple loop that display message "Welcome to PL/SQL Programming" 20 times.

```
SQL> DECLARE
NUM NUMBER;
PROCEDURE PRINTED(NUM IN OUT NUMBER) IS
BEGIN
DBMS_OUTPUT.PUT_LINE('THE GIVEN STATEMENT 20 TIMES IS:');
WHILE NUM<20 LOOP
DBMS_OUTPUT.PUT_LINE('Welcome to PL/SQL Programming');
NUM:=NUM+1;
END LOOP;
END;
BEGIN
NUM:=0;
PRINTED(NUM);
END;
/

PL/SQL procedure successfully completed.
THE GIVEN STATEMENT 20 TIMES IS:
Welcome to PL/SQL Programming
Welcome to PL/SQL Programming
Welcome to PL/SQL Programming
Welcome to PL/SQL Programming
Welcome to PL/SQL Programming
Welcome to PL/SQL Programming
Welcome to PL/SQL Programming
Welcome to PL/SQL Programming
Welcome to PL/SQL Programming
Welcome to PL/SQL Programming
Welcome to PL/SQL Programming
Welcome to PL/SQL Programming
Welcome to PL/SQL Programming
Welcome to PL/SQL Programming
Welcome to PL/SQL Programming
Welcome to PL/SQL Programming
Welcome to PL/SQL Programming
Welcome to PL/SQL Programming
Welcome to PL/SQL Programming
Welcome to PL/SQL Programming

PL/SQL procedure successfully completed.

SQL>
```

3) Write a PL/SQL code block to find the factorial of a number.

```
SQL> set serveroutput on
SQL> CREATE OR REPLACE FUNCTION FACT(NUM IN NUMBER)
RETURN NUMBER
IS
RES_FACT NUMBER:=1;
BEGIN
FOR I IN 1..5 LOOP
RES_FACT:=RES_FACT*I;
 DBMS_OUTPUT.PUT_LINE(RES_FACT);
 -- DBMS_OUTPUT.PUT_LINE('FACTORIAL OF '||NUM||' = '||RES_FACT);
 END LOOP;
 RETURN RES_FACT;
 DBMS_OUTPUT.PUT_LINE(RES_FACT);
 END;
 RES_FACT=RES_FACT*I;
 AS I CALL FUNCTION I USED TO GET THE FACTORIAL OF THAT INPUT NUMBER
 RES_FACT=5*4*3*2*1;
 RES_FACT=120
/

Warning: Function created with compilation errors.
```

```
Enter value for 1: 5
old   2: INP NUMBER:=&1;
new   2: INP NUMBER:=5;
THE FACTORIAL IS:
120

PL/SQL procedure successfully completed.

SQL>
```

4) Write a PL/S QL program to generate Fibonacci series.

```
SQL> CREATE OR REPLACE FUNCTION FIB (N POSITIVE) RETURN INTEGER IS
  2  BEGIN
  3  IF (N = 1) OR (N = 2) THEN -- TERMINATING CONDITION
  4  RETURN 1;
  5  ELSE
  6  RETURN FIB(N - 1) + FIB(N - 2); -- RECURSIVE CALL
  7  END IF;
  8  END FIB;
  9
 10  /

Function created.

SQL> DECLARE
  2  RESULT NUMBER := 1;
  3  PREVIOUS NUMBER := -1;
  4  L_SUM NUMBER;
  5  N NUMBER;
  6  L_IN NUMBER := 10;
  7  BEGIN
  8  FOR N IN 1 .. L_IN
  9  LOOP
 10  L_SUM := RESULT + PREVIOUS;
 11  PREVIOUS := RESULT;
 12  RESULT := L_SUM;
 13  DBMS_OUTPUT.PUT_LINE (L_SUM);
 14  END LOOP;
 15  END;
 16  /
0
1
1
2
3
5
8
13
21
34

PL/SQL procedure successfully completed.

SQL>
```

5) Write a PL/SQL code to find the sum of first N numbers

```
SQL> set serveroutput on
SQL> --DECLARATION SECTION
SQL> DECLARE
  2  X NUMBER;
  3  N NUMBER;
  4
  5  --UTILITY FUNCTION
  6  FUNCTION FINDMAX(N IN NUMBER)
  7     RETURN NUMBER
  8  IS
  9     Z NUMBER;
 10  BEGIN
 11
 12     -- FORMULA FOR FINDING SUM
 13     Z := (N * (N + 1) * (N + 2)) / 6;
 14     RETURN Z;
 15  END;
 16  BEGIN
 17     N := 4;
 18     X := FINDMAX(N);
 19     DBMS_OUTPUT.PUT_LINE(' SUM: '
 20     || X);
 21  END;
 22
 23  --END OF PROGRAM
 24  /
SUM: 20

PL/SQL procedure successfully completed.

SQL>
```