EX.NO:12 DATE:	**PROCEDURES AND FUNCTIONS**

AIM:

To develop and execute PL/SQL procedures and functions to perform specific tasks, understand parameter passing, and demonstrate modular programming in Oracle PL/SQL.

CREATING TABLE

```
SQL> CREATE TABLE Triangle (
  2      Base NUMBER(5),
  3      Height NUMBER(5),
  4      Area NUMBER(10, 2)
  5  );

Table created.

SQL> INSERT INTO Triangle (Base, Height, Area) VALUES (4, 3, 6);

1 row created.

SQL> INSERT INTO Triangle (Base, Height, Area) VALUES (6, 2, 6);

1 row created.

SQL> INSERT INTO Triangle (Base, Height, Area) VALUES (5, 4, 10);

1 row created.
```

A SIMPLE PL/SQL PROCEDURE

```
SQL> DECLARE
2    length NUMBER := 8;
3    breadth NUMBER := 5;
4    area NUMBER(10,2);
5    BEGIN
```

```
6    area := length * breadth;
7
8    INSERT INTO Rectangle (Length, Breadth, Area)
9    VALUES (length, breadth, area);
10
11     DBMS_OUTPUT.PUT_LINE('Rectangle inserted: Area = ' || area);
12     END;
13     /
PL/SQL procedure successfully completed.
SQL> select * from Triangle;

      BASE     HEIGHT       AREA
---------- ---------- ----------
         4          3          6
         6          2          6
         5          4         10
         8          5         20
         1          4          2
         2          4          4
         3          4          6
         4          4          8
         5          4         10
         1          3        1.5
         2          3          3
         3          3        4.5
         4          3          6

         5          3        7.5

         8          5         20
```

PL/SQL PROCEDURE WITH SIMPLE LOOP

```
SQL> DECLARE
  2    base NUMBER := 1;
  3    height NUMBER := 4;
  4    area NUMBER(10,2);
  5  BEGIN
  6    FOR i IN 1..5 LOOP
  7      area := 0.5 * base * height;
  8
  9      INSERT INTO Triangle (Base, Height, Area)
 10      VALUES (base, height, area);
 11
 12      base := base + 1;
 13    END LOOP;
 14
 15    DBMS_OUTPUT.PUT_LINE('5 triangles inserted using FOR loop.');
 16  END;
 17  /

PL/SQL procedure successfully completed.

SQL> select * from Triangle;


      BASE     HEIGHT       AREA

---------- ---------- ----------

         4          3          6

         6          2          6

         5          4         10

         8          5         20

         1          4          2

         2          4          4
         3          4          6
         4          4          8
         5          4         10
         1          3        1.5
         2          3          3
         3          3        4.5
         4          3          6
         5          3        7.5
         8          5         20
```

PL/SQL PROCEDURE WITH FOR LOOP

```
SQL> DECLARE
  2    base NUMBER := 1;
  3    height NUMBER := 4;
  4    area NUMBER(10,2);
  5  BEGIN
  6    FOR i IN 1..5 LOOP
  7      area := 0.5 * base * height;
  8
  9      INSERT INTO Triangle (Base, Height, Area)
 10      VALUES (base, height, area);
 11
 12      base := base + 1;
 13    END LOOP;
 14
 15    DBMS_OUTPUT.PUT_LINE('5 triangles inserted using FOR loop.');
 16  END;
 17  /

PL/SQL procedure successfully completed.
SQL> select * from Triangle;

      BASE     HEIGHT       AREA
---------- ---------- ----------
         4          3          6
         6          2          6
         5          4         10
         8          5         20
         1          4          2
         2          4          4
         3          4          6
         4          4          8
         5          4         10
         1          3        1.5
         2          3          3
         3          3        4.5
         4          3          6
         5          3        7.5
         8          5         20
         1          4          2
         2          4          4
         3          4          6
         4          4          8
         5          4         10

20 rows selected.
```

PL/SQL PROCEDURE WITH WHILE LOOP

```
SQL> DECLARE
  2    base NUMBER := 1;
  3    height NUMBER := 3;
  4    area NUMBER(10,2);
  5  BEGIN
  6    WHILE base <= 5 LOOP
  7      area := 0.5 * base * height;
  8
  9      INSERT INTO Triangle (Base, Height, Area)
 10      VALUES (base, height, area);
 11
 12      base := base + 1;
 13    END LOOP;
 14
 15    DBMS_OUTPUT.PUT_LINE('5 triangles inserted using WHILE loop.');
 16  END;
 17  /
```

```
PL/SQL procedure successfully completed.

      BASE     HEIGHT       AREA
---------- ---------- ----------
         4          3          6
         6          2          6
         5          4         10
         8          5         20
         1          4          2
         2          4          4
         3          4          6
         4          4          8
         5          4         10
         1          3        1.5
         2          3          3
         3          3        4.5
         4          3          6
         5          3        7.5
         8          5         20
         1          4          2
         2          4          4
         3          4          6
         4          4          8
         5          4         10
         1          3        1.5
         2          3          3

        3          3        4.5

         4          3          6

         5          3        7.5
25 rows selected.
```

PL/SQL PROCEDURE WITH EXCEPTION

```
SQL> DECLARE
   base NUMBER := 10;
   height NUMBER := 0; -- Deliberate zero to cause error
   area NUMBER(10,2);
 BEGIN
   area := 0.5 * base / height; -- Will cause division by zero error

   DBMS_OUTPUT.PUT_LINE('Area: ' || area);
 EXCEPTION
   WHEN ZERO_DIVIDE THEN
     DBMS_OUTPUT.PUT_LINE('Error: Division by zero is not allowed.');
   WHEN OTHERS THEN
     DBMS_OUTPUT.PUT_LINE('An unexpected error occurred.');
 END;
 /

PL/SQL procedure successfully completed.


SQL> select * from Triangle;
```

```
    BASE    HEIGHT      AREA
---------- ---------- ----------
       4         3         6
       6         2         6
       5         4        10
       8         5        20
       1         4         2
       2         4         4
       3         4         6
       4         4         8
       5         4        10
       1         3       1.5
       2         3         3
       3         3       4.5
       4         3         6
       5         3       7.5
       8         5        20
       1         4         2
       2         4         4
       3         4         6
       4         4         8
       5         4        10
       1         3       1.5
       2         3         3
       3         3       4.5
```

```
         4          3          6

         5          3        7.5


25 rows selected.
```

CONTENTS	MARKS ALLOTED	MARKS OBTAINED
Aim,algorithm,SQL,PL/SQL	30	
Execution and Result	20	
Viva	10	
Total	60	

RESULT:

The PL/SQL procedure and function executed successfully, performing tasks such as data insertion and result calculation using parameters. This experiment showcased the use of modular coding and reusability in PL/SQL.