**PL/SQL – Procedural Language/Structured Query Language**

1. **To find the Greater Number**

SQL> set serveroutput on

SQL> edit sqltest

//\*This will open a notepad;

declare

a int;

b int;

begin

a:=&a;

b:=&b;

if(a>b) then

dbms\_output.put\_line('a is greater'||a);

else

dbms\_output.put\_line('b is greater '||b);

end if;

end;

//\*save & close the note pad

SQL>@sqltest

Enter value for a: 6

old 5: a:=&a;

new 5: a:=6;

Enter value for b: 5

old 6: b:=&b;

new 6: b:=5;

a is greater6

PL/SQL procedure successfully completed.

1. **Trigger** – to compute the Salary difference between old salary and new salary

SQL> CREATE OR REPLACE TRIGGER display\_salary\_changes

2 BEFORE DELETE OR INSERT OR UPDATE ON customers

3 FOR EACH ROW

4 WHEN (NEW.ID > 0)

5 DECLARE

6 sal\_diff number;

7 BEGIN

8 sal\_diff := :NEW.salary - :OLD.salary;

9 dbms\_output.put\_line('Old salary: ' || :OLD.salary);

10 dbms\_output.put\_line('New salary: ' || :NEW.salary);

11 dbms\_output.put\_line('Salary difference: ' || sal\_diff);

12 END;

13 /

Trigger created.

SQL> INSERT INTO CUSTOMERS (ID,NAME,AGE,ADDRESS,SALARY)

2 VALUES (7, 'Kriti', 22, 'HP', 7500.00 );

Old salary:

New salary: 7500

Salary difference:

1 row created.

SQL> UPDATE customers

2 SET salary = salary + 500

3 WHERE id = 2;

Old salary: 3000

New salary: 3500

Salary difference: 500

1 row updated.

* **Functions** − These subprograms return a single value; mainly used to compute and return a value.
* **Procedures** − These subprograms do not return a value directly; mainly used to perform an action.

**3.Procedure –** To find the minimum number from the given number

SQL> DECLARE

2 a number;

3 b number;

4 c number;

5 PROCEDURE findMin(x IN number, y IN number, z OUT number) IS

6 BEGIN

7 IF x < y THEN

8 z:= x;

9 ELSE

10 z:= y;

11 END IF;

12 END;

13 BEGIN

14 a:= 23;

15 b:= 45;

16 findMin(a, b, c);

17 dbms\_output.put\_line(' Minimum of (23, 45) : ' || c);

18 END;

19 /

Minimum of (23, 45) : 23

PL/SQL procedure successfully completed.

**To display Hello world as a procedure**

SQL> CREATE OR REPLACE PROCEDURE greetings

2 AS

3 BEGIN

4 dbms\_output.put\_line('Hello World!');

5 END;

6 /

Procedure created.

SQL> execute greetings;

Hello World!

PL/SQL procedure successfully completed.

**4. Function –** To find the count of the total customers count from the table

A function is same as a procedure except that it returns a value.

SQL> CREATE OR REPLACE FUNCTION totalCustomers

2 RETURN number IS

3 total number(2) := 0;

4 BEGIN

5 SELECT count(\*) into total

6 FROM customers;

7

8 RETURN total;

9 END;

10 /

Function created.

SQL> DECLARE

2 c number(2);

3 BEGIN

4 c := totalCustomers();

5 dbms\_output.put\_line('Total no. of Customers: ' || c);

6 END;

7 /

Total no. of Customers: 6

PL/SQL procedure successfully completed.

SQL> insert into customers values(10, 'Vel',40, 'KRpuram', 4000);

Old salary:

New salary: 4000

Salary difference:

1 row created.

SQL> DECLARE

2 c number(2);

3 BEGIN

4 c := totalCustomers();

5 dbms\_output.put\_line('Total no. of Customers: ' || c);

6 END;

7 /

Total no. of Customers: 7

PL/SQL procedure successfully completed.

**5. Cursors**

SQL> DECLARE

2 total\_rows number(2);

3 BEGIN

4 UPDATE customers

5 SET salary = salary + 500;

6 IF sql%notfound THEN

7 dbms\_output.put\_line('no customers selected');

8 ELSIF sql%found THEN

9 total\_rows := sql%rowcount;

10 dbms\_output.put\_line( total\_rows || ' customers selected ');

11 END IF;

12 END;

13 /

Old salary: 5000

New salary: 5500

Salary difference: 500

Old salary: 3500

New salary: 4000

Salary difference: 500

Old salary: 3000

New salary: 3500

Salary difference: 500

Old salary: 50000

New salary: 50500

Salary difference: 500

Old salary: 7500

New salary: 8000

Salary difference: 500

Old salary: 7500

New salary: 8000

Salary difference: 500

Old salary: 4000

New salary: 4500

Salary difference: 500

7 customers selected

PL/SQL procedure successfully completed.

SQL> /

Old salary: 5500

New salary: 6000

Salary difference: 500

Old salary: 4000

New salary: 4500

Salary difference: 500

Old salary: 3500

New salary: 4000

Salary difference: 500

Old salary: 50500

New salary: 51000

Salary difference: 500

Old salary: 8000

New salary: 8500

Salary difference: 500

Old salary: 8000

New salary: 8500

Salary difference: 500

Old salary: 4500

New salary: 5000

Salary difference: 500

7 customers selected

PL/SQL procedure successfully completed.