Task 7: Procedure function and loops: Program using PL/SQL procedures, functions & loops.

Aim:- To implement PL/SQL produres, functions and loops on number theory and business sceanarios.

1. Simple PL/SQL program (Static input)

```
DECLARE
  message VARCHAR2(20)!='Booking closed:
BEGIN
  dbms-output.put-line (message):
END,
```

Output:

```
Booking closed
```

2. Conditional statement (Dynamic input):

```
DECLARE
   hid NUMBER (3): =100;
BEGIN
  If (hid = 10) THEN
      dbms-output.put-line ('value of hid is 10'):
  ELSIf (hid=20) THEN.
      dbms-output.put-line ('value of hid is 20');
  ELSIF (hid=30) THEN.
      dbms-~~opput~~.put-line ('value of hid is 30'):
  ELSE
      dbms-output.put-line ('None of the value's matching'):
  END If;
      dbms-output.put-line ('Exact value of hid is (hid)):
END;
```

Output.

```
None of the value is matching
Exact value of hid in : 100.
```

3. Nested Loops Example:

```
DECLARE
    hid NUMBER(1);
    Oid NUMBER(1);
BEGIN
   << outer-loop>>
   for hid IN 1..3 loop
       <<inner-loops>>
       for oid IN 1..3 Loop
           dbms_output.put_line ('hid is:|| hid ||
                                  and oid is:||out)
       END Loop inner-loop;
   END Loop outer_loop;
END;
```

Output:-

```
hid is :1 and oid is:1
hid is :1 and oid is:2
hid is :1 and oid is:3
hid is :2 and oid is:1
hid is: 2 and oid is:2
hid is : 2 and oid is: 3
hid is :3 and oid is:1
hid is :3 and oid is: 2
hid is :3 and oid is: 3
```

4. Procedure Example

```
CREATE OR REPLACE PROCEDURE booking_status
                                        (Cid IN NUMBER)
IS
BEGIN.
```

```
    If c-id > 200 THEN
        dbms_output.put_line('No booking available');
    ELSE
        dbms_output.put_line('Booking open');
    END If;
END;
/
```

Execution-

```
BEGIN
    booking_status (150);
    booking_status (250);
END;
/
```

Output:

```
Booking open
No Booking available
```

PL/SQL Procedure for Loops.

Example 1: Using WHILE Loop with Cursor.

Prime check using while loop

```
CREATE OR REPLACE PROCEDURE Print_prime_customers IS
    CURSOR cust_cur IS
        SELECT customer_id FROM customers;
    v_id NUMBER;
    v_is_prime BOOLEAN;
    v_i NUMBER;
BEGIN
    open cust_cur;
    Loop
        FETCH cust_cur INTO v_id;
```

```
EXIT THEN Cust-Cur%NOTFOUND;

If V-id<2 THEN
    V-is-prime := FALSE;
ELSE
    V-is-prime := TRUE;
    V-i:=2;
    WHILE V-i <= TRUNC (SQRT(V-id) Loop
        If MOD(V-id, V-i) = 0 THEN
            V-is-prime:= FALSE;
            EXIT;
        END If;
        V-i:=V-i+1;
    END LOOP;
END If;

If V-is-prime THEN
    DBMS-OUTPUT.PUT-LINE('Prime customer
                         ID: '||V-id);
END If;
END LOOP;
CLOSE Cust-Cur;
END;
```

The procedure checks all customer IDs in the table and prints the prime ones using a WHILE LOOP.

Example2: Using FOR LOOP for first N prime Numbers.

```
CREATE OR REPLACE PROCEDURE Print_first_n_
                               primes (n number) IS
   V_num NUMBER:=2;
   V_Count NUMBER:=0;
   V_is_prime BOOLEAN;
BEGIN
   WHILE V_Count <n loop
      V_is_prime : = True;

      for i IN 2.. TRUNC (SQRT(V_num)) Loop
        If MOD (V_num,i) = 0 THEN
           V_is_prime : = fALSE;
           EXIT;
        END If;
      END LOOP;

      If V_is_prime THEN
         DBMS_OUTPUT_PUT_LINE ('Prime: '||V_num);
         V_Count : = V_Count +1;
      END If;

      V_num: = V_num+1;
   END LOOP;
END;
```

This procedure prints the first N prime numbers using a FOR LOOP.

```
BEGIN
   Print_first_n_primes(10);
END;
```

EX No.	7
PERFORMAN[illegible]	5
RESULT AND [illegible]	5
VIVA VOCE ( )	
RECORD (3)	
TOTAL (20)	15
SIGN WITH DATE	[illegible] 22/[illegible]

<u>Result</u>:- Thus, the procedure function and loops program using PL/SQL procedurs, functions & loops are executed succesfully.