Date: 22/09/25

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

AIM: To implement PL/SQL produced function and loop on number theory and business scenarios.

1. Simple PL/SQL program (Static input)

```
DECLARE
    msage VARCHAR 2(20) = 'Booking closed';
BEGIN
    dbms_output.put_line (msage);
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');
ELSE IF (hid = 20) THEN
    dbms_output.put_line ('value of hid is 20');
ELSEIF (hid = 30) THEN
    dbms_output.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 is : 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 : ||oid)

    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
                                (c_id 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

Date: 22/09/25

PL/SQL Procedure for loops.

Example 1: Using WHILE loop with Cursor.

Prime check using while loop.

```
CREATE OR REPLACE PROCEDURE Print_prime_customers I

  CURSOR   cost_cur IS

     SELECT customer_id from customers;

  v_id  NUMBER;
  v_is_prime BOOLEAN;
  v_i  NUMBER;

BEGIN
     Open cost_cur;
     Loop
       FETCH cost_cur INTO v_id;
EXIT THEN cost_cur %. NOT FOUND;
If v_id < 2 THEN
   v_is_prime := FALSE;
ELSE
   v_is_prime := TRUE;
   v_i := 2;
   WHILE v_i <= TRUNC (SORT(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.

Example 2: 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;
```

VELTECH	
EX No.	7
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	
RECORD (5)	5
TOTAL (20)	
SIGN WITH DATE	24/9/25

Result: Thus, the procedure function and loops program using PL/SQL procedures, function & loops are executed successfully.