Task 7 : Procedure function and loops : Program using PL/SQL Procedures, Function & loops

Aim: To implement PL/SQL procedures, function and loops on number theory and business scenario.

1. simple PL/SQL program (static Input)

```
DECLARE
    Message VARCHAR 2(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_output.put_line (value of hid is 30');
    ELSE
        dbms_output.put_line (None of the values matching);
    END IF;
        dbms_output.put_line ('Exact value of hid is [hid]');
END;
```

Output :

None of the value is matching

Extract value of wid in = 100

3. Nested loops Example :

```
DECLARE
    wid   NUMBER (1);
    did   NUMBER (1);
BEGIN
    << outer - - loop >>
    for wid IN 1...3 loop
    << inner -- loop >>
    for oid IN 1.. 3 loop
        dbms - output . put - line ('wid is : || wid || and oid
                                    is : || oid)

    END Loop inner -loop ;
    END loop outer -loop ;

END ;
```

output :

wid is : 1 and oid is : 1
wid is : 1 and oid is : 2
wid is : 1 and oid is : 3
wid is : 2 and oid is : 1
wid is : 2 and oid is : 2
wid is : 2 and oid is : 3
wid is : 3 and oid is : 1
wid is : 3 and oid is : 2
wid is : 4 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.

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 %. NOT FOUND;
        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.

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) = 0 THEN
          v_is_prime := FALSE;
          EXIT;
      END IF;
    END LOOP;
    IF v_is_prime THEN
    END IF;
    v_num := v
    END LOOP;
    END;
```

VEL TECH-CSE	
EX NO.	04
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	—
TOTAL (20)	15
SIGN WITH DATE	

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

```
BEGIN
    print_first_n_prime s (10);
END;
```

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