Task: 7.1
Date: 8/8/25

PL/SQL Procedure, functions, Loops

Aim:

To implement PL/SQL procedures, Functions and loops on number theory & business scenarios.

Slno	section & Description
1	Declarations This section starts with the keyword DECLARE. It is an optional section & defines all variables, cursors, subprograms and other elements to be used in the progsam
2	Executable Command This section is enclosed between the keywods BEGIN & END it is a mandatory section. It consists of the executabe statements of the program.
3	Exception Handeling This section starts with the keyword EXCERTION

Simple program to print a sentence:

SYNTAX: DECLARE

< section>

BEGIN

<executable section>

Exception

<Exception handling>

END;

Program

```
Declare
    message  varchar2(20) := 'booking closed';
BEGIN
    dbms_output.put_line(message);
END;
```

Static input

```
SQL> DECLARE
  2  x number(5);
  3  y number(5);
  4  z number(9);
  5  begin
  6  x:=10;
  7  y:=12;
  8  z:= x+y;
  9  dbms_output.put_line('sum is' || z);
 10  end;
 11  /
```

Sum is 22.

★

```
Declare
    nid number(3) := 100;
BEGIN
    IF(nid = 10) THEN
        dbms_output.put_line('value of nid is 10');
    ELSEIF (nid = 20) Then
        dbms_output.put_line('value of nid is 20');
```

```
ELSEIF (nid = 30) Then
        dbms_output.put_line ('value of nid is 30');

ELSE
    dbms_output.put_line ('None of the value is
                           matching')

END IF;
        dbms_outputline ('Exact value of nid is:')

END;
/
```

PL/SQL procedure successfully completed.

```
DECLARE
    nid number(1);
    oid number(1);
BEGIN
    << outer loop>>
    FOR oid IN 1..3 LOOP
        dbms_output.put_line ('nid is:' || nid || 'and
                               oid is:' || oid);
    END loop inner_loop;
END;
/
```

nid is: 1 and oid is: 1
nid is: 1 and oid is: 2
nid is: 1 and oid is 3
nid is: 2 and oid is: 1
nid is: 2 andoid is: 2
nid is: 2 and oid is: 3
nid is: 3 and oid is: 1
nid is: 3 and oid is: 2
nid is: 3 and oid is: 3

sample program for only function

```
SQL> create or replace function csinformation
(c_id in number, c_name in varchar2)
Return varchar 2
Is
Begin
If c_id > 200 then
Return ('no booking available');
Else
Return ('booking open')
End if;
End;
```

Function created

```
SQL> declare
2 mesg varchar 2(200);
3 begin
4 mesg =csinformation 2 (206, 'raam');
5 dbms_output.put_line (mesg);
6 end;
7 /
```

No vehicle available.

VEL TECH	
EX No.	
PERFORMANCE (5)	
RESULT AND ANALYSIS (5)	
VIVA VOCE (5)	
RECORD (5)	
TOTAL (20)	
SIGN WITH DATE	

Result: Implementation of PL & SQL procedures functions & loops on number theory and business has been done successfully.

Task: 07(a)

PL/SQL Procedure for loops

Aim:

To write PL/SQL programs using loops for printing prime numbers customers IDS and for demonstrating loop control in different scenarios

Procedure

1) Start a PL/SQL block or procedure
2) Use a cursor (if required) to fetch customers IDs
3) For each ID, check wheather it is a prime number ~~using~~ checking.
4) USE FOR LOOP/ WHILE LOOP to demonstrate prime number checking.
5) Print the result using DBMS_OUTPUT.PUT_LINE
6) End the block

Example 1: Using WHILE LOOP Cursor

Prime check using WHILE loop

```
Create OR REPLACE PROCEDURE print_prime_customers IS
    Cursor cust_cur IS
        SELECT cusmer_id FROM customers;
    v_id Number;
    v_is_prime Boolean;
    v_i NUMBER;
```

```
BEGIN
    open cust_cur;
    LOOP
        FETCH cust_cur INFO v_id;
        EXIT WHEN cust_cur%. NOT FOUND;
        IF v_id < 2 THEN
            V_is_prime := FALSE;
        ELSE
            Vis_prime:= True;
            v_i := 2;
            WHILE v_i <= TRUNC (SQRT (v_id)) Loop
                IF MOD(v_id, v_id) = 0 Then
                    v_is_prime := False;
                    EXIT;
                END IF;
                v_i := v_i + 1 ;
                ENDLOOP;
            ENDIF

        IF v_is_prime THEN
            DBMS_OUTPUT. PUT_LINE ('Prime Customer
                                    ID: '||
                                    v_id);

        END IF;
    ENDLOOP;
    CLOSE cust_cur;
END;
```

This procedure checks all customers IDs in the table and prints the prime one using a WHILE LOOP.

Example 2: Using For Loop for First N prime Numbers

```
Create OR REPLACE PROCEDURE print_first_n
                                    primes
  v_num NUMBER := 2;
  v_count NUMBER := 0;

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;
  END IF
  v_num := v_num + 1;
  END LOOP;

END;
```

output

Prime : 2

Prime : 3

Prime : 5

Prime : 7

Prime : 11

Prime : 13

Prime : 17

Prime : 19

Prime : 23

Prime : 29

VEL TECH	
EX NO.	7
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	5
TOTAL (20)	20
SIGN WITH DATE	23/9/24

Result: thus the implementation of PL/SQL procedures functions & loops on number theory has been successfully completed