Date 23/9/25
Task 7.1

Pl/SQL Procedure functions for ~~for~~ Loops

AIM :- To write PL/SQL programs using loops for printing prime number customer 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 customer IDs from a table.
3. For each ID, check whether it is a prime number using a loop.
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 with Cursor

Prime check using WHILE loop

```
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.. FLOOR (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 checks all customer IDs in the table and prints the prime ones using a WHILE LOOP.

```
SQL >
Prime : 2
Prime : 3
Prime : 5
Prime : 7
Prime : 11
Prime : 13
PL/SQL Procedure successfully
```

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;
```

This procedure prints the First N prime Numbers.

RESULT : Thus, Implementation of PL/SQL Procedures functions and Loops on number theory has been Successfully executed.

PL/SQL procedure successfully completed

VEL TECH	
EX NO.	7
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	[illegible]
TOTAL (20)	18
SIGN WITH DATE	20/9/25

RESULT :- Thus, the implementation of PL/SQL procedure, functions and loops on number theory and business scenario has been executed successfully.

```
DECLARE
        hid  number (1);
        oil  mumber (1);
  BEGIN
      << outer_loop>>
    FOR hid. IN  1...3 Loop
          << Inner-Loop >>
          FOR  oid  IN  1--3  Loop
              dbms_output.putLire ('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
PL/SQL  Procedure  successfully  completed.
```

Ranked 7.2 PL/SQL , Functions , Loops

AIM :- To implement PL/SQL Procedures, functions & Loops on number theory and business scenerio

To print a sentence :

```
Syntax :- Declare
              <declare session>
          Begin
              <executable command>
          Execution
              <exception handling>
          End ;
```

Program :-

```
Declare
    message varchar 2(20) = 'Slot Closed' ;
Begin
    dbms_output.put_line(message)
End ;
```

Output : Slot Closed

Dynamic Input :-

```
Set server output on ;
declare
        X number (5);
        Y number (5);
        Z number(9);
begin
    x := 10 ;
    Y := 12 ;
    Z := x * Y;
```

Sample program for only function :-

```
SQL > Create or replace function CSinformation
(Ch-id in number, C-name in varchar 2)
 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
        mesg varchar 2 <200> ;
        begin
        mesg != Csinformation 2 <102, 'raam');
        dbms :output-put-line <mesg>;
        end ;
        /
```

vehicle available

```
SQL > declare
      mesg varchar2 <200> ;
       begin
       mesg := C Sinformation2 <206; ('raam')) ;
       dbms :output.put-line <mesg>;
       end ;
       /
```

No vehicle available

```
dbms - output . putline ('multiplication of x and y
  || z) ;
    end;
```

Output :- Multiplication of x and y 120

```
Declare
      hid number (3) = 100 ;
Begin
  If (hid = 10) Then
        dbms - out put . put-line ('value of hid is10');
  ElsIf (hid = 50) then
        dbms - output - put-line ('value of hid is 50');
  ElsIf ( hid = 110) then
        dbms - output . put_line ('False');
  Else
        dbms - output . putline ('None');
     END IF;
     dbms output . put-line ('Exact value is : '|| hid);
       END;
    /
```

Output :- None of the values is matching .
Exact value of hid is : 100

PL/SQL, Procedure succesfully completed.