Date: 25-9-25.

TASK 7: PL/SQL procedure for loop.

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 or procedure
2. Use a cursor 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.

section of description

S. NO.

Declarations

1. This section starts with the keyword DECLARE. It is an optional section and defines all variables, errors subprogram and execute element to be used in program execute commands.

2. Execute Commands

The section is ensured between keywords BEGIN and END and it is a mandatory section. It consists of execute PL/SQL

3-Exception Handling

This section Handling starts with keyword exceptions this optional section contains exceptions to handle errors in program.

Simple Program to print a sentence

Syntax:

```
DECARE
< declaration section >
 BEGIN
<executable. Commands (s)>
EXCEPTION
 exception
< exception handling >
 END;
```

Program:

```
DECLARE
   message. varchar 2(20) := 'booking
                              closed';
BEGIN:
  dbms: output put line (message);
END;
```

Static input:

```
SQL > set server output on
SQL > declare
   2 x number(3);
   3 y number(3);
   4 number(4):
```

```
5 begin
6 x := 10;
7 y := 12;
8 z := x+y;
9. dbms_output.put_line
10. end;
11 /
Sum is 22
PL/SQL procedure successfully completed
```

Dynamic input:

```
Set serverout on;
   declare
   x number(5);
   y number(5);
   z number(9);
   begin
    x := 10;
    y := 13;
   z = x+y;
   dbms_output_line
   end;
   /
SQL> declare
```

```
2 val 1 integer; enter value for val1 : 20
3 val 2 integer; old 6: val = &val;
4. val 3 integer; new 6. val;
5. begin.          : enter value for val2:30
6. val1 : &val1;
7 val 2 := &val2;   old 7 : val2 = &val2;
8 val3 := val1 + val2; new 7; val2 := 30;
9. dms : output put- line (val 3);
10. end;
11 /
```

PL/SQL procedure successfully completed

```
DECLARE
 kid number(3) = 100;
 BEGIN
 IF (kid = 10) Then
dbms = output-put-line (value of kid
ELSE IF (kid=20) THEN
dbms-output. put- line (value of kid
ELSE IF (kid =30) Then
dbms- output - line (value of kid is 30),
ELSE
dbms-output - line (invalid value
                    entering)
```

```
END IF;
dbms_output_put_line("Exact value of
hid OS:" || hid);
END;
/
None of the values is matching
exact value of hid U : 100
PL/SQL Procedure successfully completed

DECARE
 hid number(1);
   oid number(1);
BEGIN
  <<Outer loop>>
  FOR hid IN --- loop
dbms_output_put_line hid is :" || hid ||
      and oid is : || oid);
 end loop inner_loop;
 end loop outer_loop;
  END;
hid is : 1 and oid is: 1
hid U : 1 and oid is : 2
hid U : 1 and oid U : 3
hid U : 2 and oid U : 1
hid U : 2 and oid is : 2
```

Nid is 3 and oid is :1
Nid is 3 and old is :2
Nid is 3 and oid is :3.

PL/SQL procedure successfully completed

Sample program for only procedure:

SQL> create or replace procedure ej in formation

-2 <C-id in number1 (-name in varchar2)

3 is

4 begin

5. dbms- output put-line <'id:';; (-id7);

6. dbms - output put-line ('Name: '(-name);

7. End;

8 /

Procedure created

SQL> exec (sinformation <101', 'ram'>;

PL/SQL procedure successfully completed

pQL>set serveroutput on;

SQL> exec information ('01', 'ram');

Name. ram

PL/SQL ram.

PL/SQL procedure successfully completed.

Sample program for only function

SQL> create or replace function(
information (std_id number, c_name
in varchar2)
Return varchar2
IS
Begin
If (-id -> 200 then
Return ("no booking available");
else
Return ('booking open');
End if;
End;
Function created.

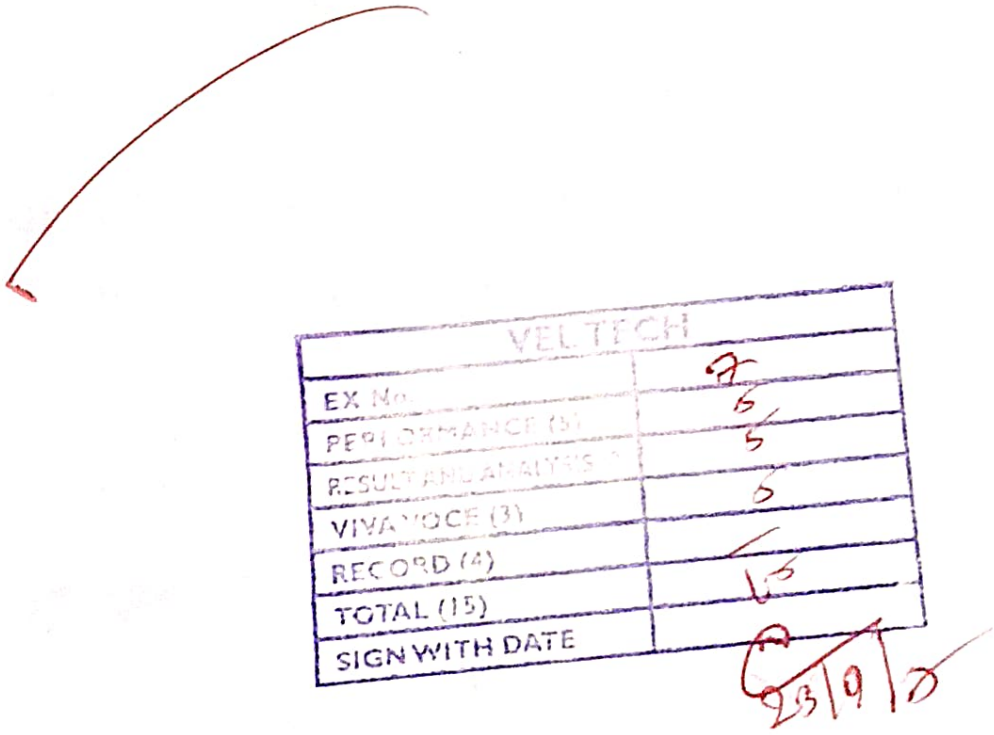
SQL> declare
2. mesg varchar 2 22007;
3 design
4. mesg := (Sinformation 2<102,room7);
5. dbms_output.put_line of mesg7;
6. End;
7. /

vehicle available

```
SQL> declare
2. mesg varchar2(200);
3 begin
4. mesg := 'information :=' 20.6 ', team';
5. dbms_output.put_line(mesg);
6. end;
7 /.
```

No vehicle available

PL/SQL Procedure successfully completed



VELTECH	
EX No.	7
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (3)	5
RECORD (4)	
TOTAL (15)	15
SIGN WITH DATE	23/9/20

Result: Thus, the implementation of PL/SQL Procedure functions and loops on number theory and business scenario has been executed successfully

PL/SQL PROCEDURE FOR LOOPS.

TASK 7-1

Aim: To write PL/SQL programs using loops for printing prime numbers customer IDs and for demonstrating loop control in different scenarios

procedure:

1. Start
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 the loop.
4. Use for loop / while loop to demonstrate prime numbers checking
5. Print the result using - DBMS - OUTPUT PUT-LINE
6. End the block.

Example: Using while loop with cursor

Prime check using while loop

CREATE OR REPLACE PROCEDURE

PRIME1 (first n prime numbers)

IS v-num number: 2;

```
v_count number := 0;
v_is_prime BOOLEAN;
BEGIN
WHILE v_count < n LOOP
v_is_prime := TRUE;
BEGIN: v_count < n LOOP
v_is_prime BO
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
DBMS_OUTPUT.PUT_LINE (v_prime || v_num);
v_count := v_count + 1;
 END IF;
 v_num := v_num + 1;
 END LOOP;
  END;
```

This procedure checks all customer ID's in the table and runs the prime one using a WHILE loop

Example: Using for loop to find N prime numbers

```
CREATE OR REPLACE PROCEDURE PRIME
  (FIRST N PRIMES (NUMBERS) 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 TRUE (SORT(V_NUM))
   LOOP
 IF MOD(V_NUM, I) = 0 THEN
 V_IS_PRIME == FALSE;
   EXIT;
   END LOOP;
IF V_IS_PRIME THEN
DBMS_OUTPUT_PUT_LINE (PRIME :', V_NU
 V_COUNT := V_COUNT + 1);
 END IF;
V_NUM := V_NUM + 1;
 END LOOP;
  END;
```

This procedure prints the first N prime numbers using for loop

```
BEGIN
  print_first_n_primes (10);
END;
```

This procedure prints the first N prime numbers.

VELTECH	
EX No.	7
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	
TOTAL (15)	15
SIGN WITH DATE	23/9/25

Result: Thus implementation of PL/SQL Procedure values function and loops on numbers the ex has been successfully executed.