Date: 23/9/25

Task 7 : PL/SQL : procedure for loops

Aim:- To implement : PL/SQL : procedure functions and loops on number theory and business scenarios.

Procedure:-

PL/SQL is combination of SQL along with the procedual features of programming languages. It was developed by oracle corporation in the early 90's to enhance the capabilities of SQL. PL/SQL is one of three key programming languages embeneked in the oracle Database, along with SQL itself and Java

S.NO.	Section & Description
1.	Declarations This section starts with the keyword DECLARE. It is an optional section and defines all variables, cursors, subprograms, and another elements to be used in program
2.	Executable Commands This section is enclosed between keywords BEGIN and END and it is a mandatory.

Section :- It consists execute PL/SQL

3. Exception Handling

This section starts with keyword Exception this optional section contains exceptions that handle errors in program

Simple program to print a sentence

Syntax:

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

Program :-

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

Static input:

```
SQL > set serveroutput on
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

PL/SQL procedure successfully
                Completed.
```

Dynamic input:

```
set serveroutput on;
declare
  x number(5);
  y number(5);
  z number(9);
begin
  x:=10;
```

```
z= x+y;
dbms_output.put_line('sum is' || z);
end;
/
```

```
SQL> declare
  2  val1 integer;
  3  val2 integer;
  4  val3 integer;
  5  begin
  6  val1 := &val1;
  7  val2 := &val2;
  8  val3 := val1+val2;
  9  dbms_output.put_line(val3);
 10  end;
 11  /
```

Enter value for val1: 20
old 6: val1 := &val1;
new 6: val1 :
Enter value for val2: 30
old 7: val2 := &val2;
new 7: val2 := 30;

PL/SQL procedure successfully completed

```
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');
ELSE IF(hid = 30) THEN
dbms_output.put_line ('value of hid is 30');
ELSE
dbms_output.put_line ('none of the values
                       is matching');
END IF;
dbms_output.put_line ('Exact value of
    hid. is :'|| hid);
END;
/
none of the values is matching
Exact value of hid. is :100.
PL/SQL procedure successfully
                       completed.
DECLARE.
hid number(1);
    sid number(1);
BEGIN.
    << outer loop >>
FOR hid IN 1 .. 3 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 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.

Sample program for only procedure:

SQL > create or replace procedure es in
-formation
-2 (e_id in number, e_name in
varchar2)

```
4 begin
5. dbms_output.put_line('ID: '||c_id);
6. dbms_output.put_line('name: '||c_name);
7. end;
8 /
```

Procedure created.

```
SQL>exec c_information(101,'room');
```

PL/SQL procedure successfully completed.

```
SQL > set serveroutput on;
SQL > exec c_information(101,'roam');
name : raam
```

PL/SQL procedure successfully completed.

Sample program for only function

```
SQL> create or Replace function c_informa-
     tion (c_id in number, c_name
     in varchar2)
  Return varchar2.
  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 design
4. mesg := (information 2(102, 'raam');
5 dbms-output-put-line (: mesg);
6. End;
7)

vehicle avaiable

SQL > declare
2 mesg varchar 2(200);
3 begin.
4 mesg := (' information 2(206, 'raam');
5 dbms-output-put-line (mesg);
6_ end;
7.1.

no vehicle available

PL/SQL procedure successfully completed

VEL TECH	
EX NO.	7
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	
TOTAL (20)	15

23/9/20

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

Table 9-1 PL/SQL Procedure for Loops Date: 23/9/25

Aim:- To write PL/SQL programs using loops for printing prime numbers between 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: 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(SORT(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 ID's 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 (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 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 for loop.

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

This procedure prints the first n prime numbers.

VEL TECH	
EX NO.	11
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	
TOTAL (20)	20

Result:- Thus implementation of PL/SQL procedures functions and loops on number theory has been successfully executed.

23/9/22