Task 7 - PL/SQL, Procedures, Loops

Aim:

To implement PL/SQL Procedures, functions and Loops

Procedure:

PL/SQL is a combination of SQL along with procedural 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 embedded 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 other elements to be used in the program.
2.	**Executable commands:** This section is enclosed between the keywords BEGIN and END and it is a mandatory section. It consist of the executable PL/SQL statements of the program. It should have atleast one executable line of code, which may be just a NULL command to indicate that nothing should be executed
3.	**Exception handling:** This section starts with the Keyword EXCEPTION. This optional contains exception(s) that handle errors in the program.

Syntax:

```
DECLARE
    <declarations section>
BEGIN
    < executable command(s)>
EXCEPTION
    <exception handling >
END;
```

Simple program to print a message:

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  /
PL/SQL procedure successfully completed
sum is 22
```

Dynamic Input:

```
set serveroutput on:
declare
        x number(5);
        y number (5);
        z number (x);
begin
        x:=10;
        y:12;
        z:=x+y:
        dbms_output.put_line(sum is'||z);
end;
/
SQL> declare
  2  var1 integer;
  3  var2 integer;
  4  var3 integer;
  5  begin
  6  var1 = &var1;
  7  var2 = &var2;
  8  var3 = var1+var2;
  9  dbms_output.put_line (var3);
 10  end;
 11  /
 Enter value for var1 : 20
 old 6: var1:= &var1;
 new 6: var1:=20;
 Enter value for var2 : 30
 old 7: var2:= &var2;
 new 7: var2:=30;

 50
 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
                                ("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 is matching");

END IF
      dbms_output.put_line ("Exact value of hid is:'||hid);

END;
/
```

```
None of the value is matching
Exact values of hid:100

PL/SQL procedure successfully completed
```

```
DECLARE
    hid number(1);
    oid number(1);
BEGIN
    FOR hid IN 1..3 LOOP
         FOR oid 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> 1 create or replace procedure is information
  2 c_id in number, c_name in varchar2)
  3 is
  4 begin
  5 dbms_output.put_line ('ID:'||c_id);
```

```
7 end;
8 /
Procedure executed
SQL> exec csinformation (101, 'raam');
PL/SQL procedure successfully completed
SQL> set serveroutput on;
SQL> exec csinformation (101, 'raam');
ID no 1
Name: raam
PL/SQL procedure successfully completed
```

<u>Sample program for only function</u>:

```
SQL> create or replace fution csinformation
(ch-id in number, c_name in varchar 2)
Return varchar2
Is
Begin
If cid>200 then
Return ('no booking available');
Else
Return ('booking open');
End if;
End;
/
Function created
SQL> declare
  2  mesg varchar2 (200);
  3  begin
  4  mesg := csinformation 2(102, 'raam');
  5. dbms_output.put_line (mesg);
  6  end;
  7  /
Vehicle available
```

```
SQL> declare
  2  mesg varchar2<200>;
  3  begin
  4  ~~begin~~
  4  mesg := <<information2<206,'raam'>;
  5  dbms_output.put_line <mesg>;
  6  end;
  7  /
```

No vehicle available

PL/SQL procedure successfully completed.

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 when cust_cur%NOTFOUND;
        If v_id < 1 Then
            v_is_prime := false;
        Else
            v_is_prime := True;
            V_i =2;
            while v_i <= Trunc(sqrt(v_id)) Loop
```

~~Result:~~

Thus the implementation of PL/SQL

procedures and

```
          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;
```

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(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 procedare prints the first N prime numbers using a FOR LOOP

for example:

```
BEGIN
     print_first_nprimes(10);
END;
```

VEL TECH	
EX NO.	
PERFORMANCE (5)	7
RESULT AND ANALYE'S (5)	5
VIVA VOCE (5)	[illegible]
RECORD (5)	[illegible]
TOTAL (20)	–
SIGN WITH DATE	✓

[illegible] 23/9

Result:

Thus the implementation of the PL/SQL Procedures and Loops was executed successfully.