Task: 7

Date: 23/9/25

PL/SQL Procedures, function, loops.

Aim:

To implement PL/SQL procedure, Function and loops on Number theory and business scenarios.

Procedure:

PL/SQL is a combination of SQL along with procedure features of programming languages. It was developed by Oracle. Corporation in the early ii'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, sub programs 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
            table command i)>
```

```
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(9);
begin x:=10;
    y:=12;
    z:=x+y;
    dbms.output.put_line('sum is"||z);
end;
/
```

```
SQL > declare
 2  var1 integer;
 3  Var 2 integer;
 4  Var 3 integer;
 5   begin
 6  var1 = &var1;
 7  Var 2 = & Var 2;
 8   Var 3 =  var 1 + var 2
 9  dbms_output.put_line (var 3);
10  end;
11  /

  Ender value for var1: 20
       old 6: Var1 := & Var1:2;
       new6: var1: = 20;
  Enter value for var2: 30
       old 7: Var 2:= &Var2;
       new 7: Var 2:= 30;
       50
    PL/SQL procedure successfully completed.
```

```
DELCRE
    hid number (3]::=100;
 BEGIN
    If (hid=10) THEN
    dbms_output.put_line ('Value of hid is10");
    ELSE If (hid=20) THEN
            dbms_output.put_line(" Value of hid is 20"):
    ELSIF (hid=30) THEN
            dbms-output .put_line (" Value of hid= 30"):
    ELSE
         dbms.output.put_line ("None of the Value is matching");
    END If
         dbms.output.put_line ("Exact value of hid is :" || hid);

    END;

    /

    None of the Value is matching E
    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 : 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 complete.
```

Sample program for only procedure:

```
SQL> 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 ),'
  6 dbms_output .put_line ('Name= || c.name);
  7 End;
  8 /
```

Procedure executed:

```
SQL >exec cs information<101, 'raam');
PL / SQL procedure Succesfully completed.
SQL > set serveroutput on,;
SQL, Exec cs information <101, 'raam);
ID no: 101
  Name : raam
  PL/SQL procedure successfully completed
```

Sample program for only function:

```
SQL> Create or replace function csinformation.
c hid in number , C_name in Varchar 2)
```

```
IS
Begin
 If C_id >200 then
Return ('no boxing available');
Else :
Return ('booking open');
End if ;
End ;
/
```

Function created

```
SQL > declare
  2  mesg  Varchar 2 <200>;
  3  begin
  4  mesg := (Sinformation 2<102; 'raon" >;
  5  dbms_output . put_ line (mesg>;
  6  end;
  7  /
```

Vachile available

```
SQL > declare
  2 mesg varchar 2<200>;
  3 begin .
  4. mesg:= <Sinformation2<206, 'raan'>;
  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 ore replace procedure point-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 % NOT FOUND;
        If v_id < 2 Then
                v_is_prime := false
         Else
             v_is_prime := True;
              v_i := 2;
             while v_i <= Trunc(sqrt(v_id)) loop.
         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- 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

for example:

```
     BEGIN
        print_first -nprimes (10);
     END;
```

VEL TECH

EX NO.	7
PERFORMANCE (5)	6
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	
TOTAL (20)	16
SIGN WITH DATE	23/9/2

Result:

Thus the implementation of the PL/SQL procedure and loops was executed successfully.