29|4|25 Task7: PL/SQL Procedures, Function Loops

Aim:

To implement PL/SQL Procedures, functionand loops on Number theory and business scenarios

Procedure:

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

section of Description

S.NO Delarations

1 Thissection Starts with the keyword DECLARE. It is an optional section and defines all variables, cursors subprogram, and another elements to be used in program

2 Executable commands

This section is enclosed b/w keyword BEGIN and END & it is a mandatoy section. It consists of execute PL/SQL

3 Exception Handling

This section starts with the keyword Exception This optional Sections contains exceptions(s) the handle errors in program.

Simple program to print a senctence

Syntax:

```
        DECLARE
        <declarations section>
      BEGIN
        <executable command(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 Succesfully completed
```

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_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_outpcet.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 procedurce successfully completed.
```

```
DECLARE
    hid number(3) :=100;
BEGIN
    IF (hid=10) THEN
    dbms_output.put -line('value of hid is 10');
    EISIF (hid =20) THEN
     dbms-output.pud -line('value of hid is 20');
    EISIF (hid >30) THEN
    dbms_ output.put_line ('Value of hid is 30');
  EISE
    dbms -outped.put - line('None of the values is matching');
```

```
ENDIF;
dbms-output.put_line('None of the values is matching');
 ENDIF;
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);
     oid   number(1);
  BEGIN
    <<outer_loop>>
    For hid IN 1...3 LOOP
      <<inner_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:1
   hid is:2 and oid:2
   hid is:2 and oid:3
   hid is:3 and oid:1
   hid is:3 and oid:2
   hid is:3 and oid:3
 PL/SQL procedure successfully completed.
```

Sample program for only procedure

```
SQL>create or replace procedure cs 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 created:
SQL> exec cs information(101,'raam'):
PL/SQL procedure successfully completed:
SQL>set serveroutput on:
SQL>exec cs information(101,'raam'):
ID:101
Name:raam
PL/SQL procedure successfully completed
```

Sample program for only function:

```
SQL> create or replace function cs information
 (h-id in number, c- name in varchar2)
 Return varchar2
IS
 Begin
 If c-id>200 then
 Return('no booking available'):
 else
 Return('booking open');
 Endif;
End;
/
```

```
functioncreated
SQL> declare
 2  mesg varchar2<200>;
 3  begin
 4  mesg:=csinformation2<102!'raam'>;
 5  dbms-output.put-line<mesg>;
 6  end;
 7  /
 vehicle available

SQL> declore
 2  mesg varchar2<200>;
 3  begin
 4  mesg :=cs information2<206;'raam'>;
 5  dbms-output.put -line <mesg>;
 6  end;
 7  /
 No vechicle available
 PL/SQL procedere successfuly completed.
```

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

Result: Thus, Implementation of PL/SQL procedures for loops and functions has been successfully completed.

Task 7.1: PL/SQL procedarce for loops

Aim:

To write PL/SQL program using loops forprintin primenamber customer ID and for demonstrating loop control in different scenarios

Procedere

1) Start a PL/SQL block 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 demostrate primenanber checking
5) print the result using DBMS-output.put-line.
6) End the block

Ex:1 Using whileloop with cursor

Prime check using while loop

```
Create OR Replace procedure print-prime custome
  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;
    --.Prime check using while loop
  If v-is-   < 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 ~~If~~;
           END If;

           v-i :=v-i+1;
         ENDloop;
       END If;
     If  v-is-prime Then
       DBMS-output.put-line(Prime customerID'|| v-id);
     End If;
   END loop;
  close cust-cur
  END;
```

This procedure checks all customer IDs in the table and prints the prime ones using a while Loop.

Ex:2 Using for loop for first N prime Number

Create or Replace procedure print first_n_prime

```
      (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;
  -- Prime check using for loop
  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+;
   END loop;
 END;
```

This procedure prints the first N prime numbers using a for loop.

for example:

```
  Begin
      print_first_n_primes(10); --prints first 10
                                   prime numbers


  END;
```

VEL TECH	
X NO.	7
'ERFORMANCE (5)	5
'ESULT AND ANALYS'S (5)	5
VIVA VOCE (5)	5
RECORD (5)	-
TOTAL (20)	15
GN WITH DATE	M 23/9

Result: Thus, Implementation of PL/SQL Procedure functions and loops on number theory has been successfully executed.