Task 7

23/9/25

PL/SQL Procedures, functions, Loops

Aim:

To implement PL/SQL Procedures, Functions and loops on Number theory and business scenario.

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 embended in the Oracle Database, along with SQL itself and Java.

Section & Description

S.No

1 Declarations
This section starts with the keyword DECLARE. It is an optional section and defines all variables, cursors, subprogram, and another elements to be used in progra.

2 Executable Commands
This section is enclosed btw keywords BEGIN and END and it is a mandatory section. It consists of execut PL/SQL

3 Exception Handling
This section starts with the keyword EXCEPTION. This optional section contains exceptions G) the handle errors in program

Simple program to print a sentence

Syntax:

```
DECLARE
< declaration 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 successfully 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_output.put_line ('sum is '||z);
 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;
```

~~Sum is 22~~

PL/SQL procedure successfully completed.

```
DECLARE
   hid number(3)  := 100;
BEGIN
   IF (hid = 10) THEN
  dbms_output.put_line('Value of hid is 10');
   ELSIF (hid = 20) THEN
  dbms_output.put_line('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 values is matching
Exact values 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 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 C is information
2 < Cio in number, C_home in varchar,
3 < s
4 begin
5 dbms_output_put_ line < 'ID:' ||C_id>;
6 dbms_output_put_line <'Name:' || C-name>;
7 end;
8 /
Procedure Crede;
SOL > exec information <101, 'ram'>
PL/SOL procedure successfully completed.
SOL > set serve output on;
SOL > exeees information < 101, 'raam'>;
ID : 101
Name : raam
PL/SOL procedure successfully compded.
```

Sample program for only function:

```
SOL > Create or replace function cs information
h_icl in number, C_name in varchar2,
Return varchar2
Is
Begin
If C_id > 200 than
Return ('no booking available";
else
Return ('booking open');
Endif;
End;
/
```

function created

```
SQL > declare
  2  mesg varchar 2(200):
  3  begin
  4  mesg = C sinformation 2 clo 2;'raam'>;
  5  dbms = output . put_linec mesg>!
  6  end;
  7  /
```

Vehicle available

```
SQL > declare
  2  mesg varchar 2(200);
  3  begin
  4  mesg = Cs information 2(206,'raam'>;
  5  dbms = output. put_linec mesg>;
  6  end;
  7  /
```

No vehicle avaiable
PL/SQL procedure successfully completed.

Result:- Thus, Implementation of PL/SQL procedure for loops and functions has been successfully Completed.

Task 7
22/9/25

PL/SQL Procedure for Loops

Aim: To write PL/SQL programs using loops for printing prime number customer 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 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%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 CustomerID: ' || v_id);
    END IF;
END LOOP;
CLOSE cust_cur;
END;
```

This procedure checks all customers IDs in ther 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_numi) =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 to a FOR LOOP.

FOR example:

```
BEGIN
    Print_first_n_primes(10);
END;
```

VEL TECH	
EX NO.	7
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
[illegible] (5)	
	15
	23/03/23

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