TASK-7 PL/SQL Procedures, function, Loops 23/9/25

Aim:

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

Procedure Procedure:

PL/SQL is combination of SQL along with the Procedural features of Programing language It was developed by oracle corporation in the early 90's to enhance the capabilities of SQL PL/SQL is one of three key Programing language embedded in the oracle Database along with SQL itself and Java.

Selection & Description

2 NO

1. Declarations

This section starts with the keyword DECLARE. It is an optional section and defines all variable cursors subProgram and another element to be used in Program

2 Executable commands.

this section is enclosed blw keywords BEGIN and and it is a mandatory section It contains at exec PL/SQL

3. Exception Handling

This section starts with keyword EXCEPTION this optional section contains exception (s) the handle error 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(6);
  5   begin
  6   x:= 10;
  7.  y:= 12;
  8   z:= x+y;
  9   dbms_output.put_line('sum is' :: z);
 10  end;
  4  /
```

som is 22

PL/SQL Procedure successfully completed.

Dynamic Input

```
set serveroutput on;
declare
x number(5);
y number(5);
z number(9);
being
x:=10;
y:=12;
z:=x+y;
dbms_output.put_line('som 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
10. end
11 /
```

Enter value for var1:20
old 6: var1 = var1;
new 6: var1:= 1;
Enter value for var2:20
new 7: var2:=&var2;
new 7: var2:=30;
var 3.

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 value 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
Simple Program for only Procedure
SQL> create or replace Procedure c is information
 2< cis in number c-home n varchar
 3 <s
 4 begin
 5 dbms_output_put_line <'ID', ||c-id';
 6. dbms_ output_put_line < Name ||c-names>;
 7 end;
 8 /
Procedure create
SQL> exec information <10, 'ram'>
PL/SQL Procedure successfully completed
```

SQL > set serve outful on;

SQL > execes information < 101 ram '>'

ID :101

Name :Raam

PL/SQL Procedure successfully completed

Simple Program for only function.

SQL > create or replace function cs information h-icl
in number c-name in varchar, Return varchar2

Is.

Begin

If c-id > 20dbms

Return (no booking available";

else

Return (booking open);

End if;

End;

function created

SQL > declare

2. mesg varchar 2 <200>

3. begin

4. mesg : c S information 2 clo2 'room',.

5. dbms - outful .Put - line <mes4>.

6. end;

7. /

vehicle available

```
SQL > declare
2 mesg varchar2(100);
3 begin
4. mesg := 'information 206; room';
5 dbms_output.put_line(mesg);
6. end;
7 /
```

No vehicle available

PL/SQL Procedure successfully completed.

VEL TECH	
EX No.	6
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	
TOTAL (20)	20
SIGN WITH DATE	

16/9/25

Result: Thus implementation of PL/SQL Procedure for loops and function has been successfully completed

Task-7.1 PL/SQL Procedure for loops

Aim: To write PL/SQL Program using loops for Printing Prime number customer IDs and for demonstration loop control in difficult scenarios.

Procedure:

1. Start a PL/SQL block or Procedure
2. use a cursor (if required) to fetch customer IDs for 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 6: using WHILE loop with cursor

Prime check using WHILE loop

```
CREATE OR REPLACE PROCEDURE Print_Prime_customer is
    CURSOR cust_cur IS
        u_id Number;
        u_is_Prime BOOLEAN;
        u_i NUMBER;
BEGIN
    OPEN cust_cur;
    LOOP
        FETCH cust_cur INTO u_id;
```

```
EXIT WHERE cust_cur%NOT FOUND;
IF u.id <2 THEN.
   u_is_Prime := FALSE;
ELSE
  u_is_Prime := TRUE;
  u_i := 2;
   WHERE u_i <= TRUE (SQRT(u.id)) LOOP
     IF MOD(u.id, u_i)=0 THEN
          u_is_Prime = FALSE;
           EXIT;
        END IF;
        u_i := u_i+1;
      END LOOP;
      END IF;
   IF u_is_Prime THEN
       DBMS_OUTPUT.PUT_LINE (Prime customer ID: ||uid
  ENDIF;
 END LOOP;
 CLOSE cust_cur;
 END;
```

this Procedure checks all customer IDs in the table and prints the Prime ones using a while loop

Ex2 using FORLOOP for first N Prime Number

```
CREATE OR REPLACE PROCEDURE Print_First_n_Prime (n number)
    v_num    NUMBERS := 2;
    v_count  NUMBERS := 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 Procedure Prints the first N Prime number using a for loop

FOR example

```
BEGIN
    Print_First_n_Primes(10);
END;
```

VEL TECH - CSE	
EX NO.	6
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	-
TOTAL (20)	20
SIGN WITH DATE	16/9/24

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