22/9/25

Task 7 : PL/SQL Procedures for loops , Functions

Aim : To implement PL/SQL Procedures , Functions and loops on Number theory and business scenarios.

Procedure :

PL/SQL is a combination of SQL along with the procedural features of programming language. It was developed by Oracle corporation in the early 90's to enhance the capabilities of SQL.

Sections & Descriptions

1. Declaration
This section starts with the keyword DECLARE. It is an optional section and defines all variables

2. Executable Commands
This section is enclosed between the keywords BEGIN and END and it is a mandatory section. It consists of Executable PL/SQL statements of program

3. Exception Handling
This section starts with the keyword Exception.

Simple program to print a sentence:

Syntax :

```
DECLARE
<declaration section>
BEGIN
 < Executable command (s)>
 Exception
 < Execution 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 numbr X+Y;
```

```
dbms-output-line ('sum is '|| z);
End;
/

SQL > declare
  2   Var 1 Integer;
  3   Var 2 integer;
  4   Var 3 integer;
  5   begin
  6    Var1 := & var1;
  7    var 2 := & Var2;
  8    var 3 := var 1 + var 2;
  9    dbms-output.put - line <var 3>;
 10     End;
  11   /

 Enter value for var1 ; 20
 old   6 :   var 1:= & Var1;
 new   6 :   var 1  := 20;
 Enter value for var 2 ; 30
 old   7 :   Var 2 := & var 2 ;
 new   7 :   Var 2 := 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');
  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 values is matching')
```

```
END IF;
dbms_output.put_line ('Excat value of hid is:'|| hid)
END;
/
None of values is matching
Extract value of hid is: 100
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 varchar 2
Is
Begin
If c_id >200 then
Return ('no booking available');
Else
Return ('booking free');
Endif;
End;
Function created.

SQL > declare
 2   mesg varchar 2<200>;
 3   begin
 4   mesg := cs information 2<206,'raam'>;
 5   dbms_output.put_line <mesg>;
 6   End;
 7   /
 No vehicle avaible
PL/SQL procedure successfully 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 procedure for loops and functions has been successfully completed.