

22/9/25 PL/SQL, Functions, Loops

Aim: To implement PL/SQL procedures, functions + Loops on number theory and business scenario.

To print a sentence:

Syntax: Declare
 <declare session>
 Begin
 <executable command>
 Execution
 <exception handling>
 End;

Program:

Declare
 message varchar2(20) := 'slot closed';
 Begin
 dbms_output.put_line(message)
 End;

Output: slot closed

Dynamic Input: set serveroutput on
 declare

x number(5);
 y number(5);
 z number(9);
 begin
 n := 10;
 y := 12;
 z := n * y;
 dbms_output.put_line('multiplication of n and y || z');
 end;

EX NO	DATE	PERFORMANCE (%)	RESULT AND ANALYSIS (%)	VIVA VOCE (%)	RECORD (%)	TOTAL (%)
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						

Result: The implementation of trigger + view on the database has been successfully completed and working.

output: multiplication of n and y 120.

Declare

wid number (3) := 100;

Begin

If (wid = 10) Then

dbms_output.put_line('value of wid is 10');

Else If (wid = 50) then

dbms_output.put_line('value of wid is 50');

Else If (wid = 110) then

dbms_output.put_line('False');

Else

dbms_output.put_line('None');

END IF;

dbms_output.put_line('Exact value is: ' || wid);

END;

Output:

- None
Exact value is 100

PL/SQL procedure successfully completed

Loops:

Declare

wid number (1);

old number (1);

Begin

<< outer-loop >>

FOR wid IN 1..2 LOOP

<< inner-loop >>

FOR old IN 1..2 LOOP

dbms_output.put_line('wid is: ' ||

wid || ' and old is: ' || old);

END LOOP inner-loop;

END LOOP outer-loop;

END;

wid is : 1 and old is : 1
 wid is : 2 and old is : 2
 wid is : 2 and old is : 1
 wid is : 2 and old is : 2

PL/SQL procedure successfully completed.

Function:

SQL> create or replace function slotbook
 (st-id in number, st-name in varchar2,
 slot-name in varchar2)
 Return varchar2

is
 Begin
 If st-id > 200 then
 Return('No slot available');
 Else
 Return('Slot open');
 End If;
 End;
 /

Function created :-

SQL> declare
 msg varchar2(200);
 begin
 msg := slotbook(206, 'woman');
 dbms_output.put_line(msg);
 end;

Output: No slot available

PL/SQL Procedure successfully completed.

Using while loop with cursor
 odd check using while loop (Positive odds)
 sql > create or replace procedure print-odd nos is
 cursor num-us is
 select student from students;
 v-id NUMBER;
 v-is-odd BOOLEAN;
 v-i NUMBER;
 Begin
 open num-us;
 LOOP
 FETCH num-us into v-id;
 EXIT when num-us % NOT FOUND;
 IF v-id < 0 then
 v-is-odd := False;
 Else
 v-is-odd := True;
 v-i := 2;
 while v-i <= TRUNC(SQRT(v-id)) LOOP
 IF MOD(v-id, v-i) = 0 Then
 v-is-odd := False;
 Exit;
 END LOOP;
 END IF;
 If v-is-odd Then
 dbms_output.put_line('odd student id:
 || v-id);
 END IF;
 END LOOP;
 CLOSE num-us;
 END;

SQL>

create or replace procedure

print_n (n NUMBER) IS
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-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;

begin

print_n(4);

end;

Result:

Output: 1
2
3

thus, the implementation of PL/SQL Function and Loops on the database has completed successfully.

VEL TECH	
EX NO.	7
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	5
TOTAL (20)	25
SIGN WITH DATE	23/9/22