

Task No:- 3 PL/SQL Procedures, functions and loops

Date :- 23/09/25

Aim:- To implement PL/SQL procdes, functions and 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 is one of three key programming languages embedded in oracle database along with SQL it self and Java.

Simple Program to Print a Sentence:

Syntax:-

```
DECLARE:  
    < declaration section >  
BEGIN  
    < executable sections  
exception  
    < exception handling >  
END;
```

Program:-

```
DECLARE  
    message varchar2(20) := 'booking closed';  
BEGIN  
    dbms-output-put-line (message);  
END;
```

Dynamic Input:-

Set Serveroutput on;

declare

```
x number(5);  
y number(5);  
z number(9);
```

```
begin
    x := 10;
    y := 10;
    z := x+y;
    dbms-output.put-line('sum is '||z);
end;
/
```

Output :- Sum is 20

declare

```
var1 integer;
var2 integer;
var3 integer;
```

begin

```
var1 := &var1;
var2 := &var2;
var3 := var1+var2;
```

```
dbms-output.put-line(var3);
end;
/
```

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

DECLARE

hid Number(3) := 100;

Begin

if (hid = 10) then

dbms-output.put-line ('Value of hid is 10');

else if (hid = 20) then

dbms-output.put-line ('Value of hid is 20');

else if (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;
/
Output:

```

None of the values is matching
exact value of hid is :100

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 :||

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

Program for only Procedure

create or replace Procedure cInformation

<c-id in number, c-name in varchar2>

is

begin

dbms-output.Put-line ('DO: ||c-id|');

```
dbms_output.put_line('Name'||c_name);
end;
/
```

Procedure created

```
exec csinformation(101,'ram');
```

PL/SQL Procedure successfully completed.

```
Set serveroutput on;
```

```
exec csinformation(101,'ram');
```

```
DD : 101
```

```
Name : ram.
```

Program for only functions

```
Create or replace function csinformation
(h_id in number, c_name in varchar2)
```

```
Return varchar2
```

```
Is
```

```
Begin
```

```
if(h_id > 200 then)
```

```
Return ('no booking available');
```

```
else
```

```
Return ('booking open');
```

```
end if;
```

```
END;
```

function created

```
declare
```

```
msg := csinformation(102,'ram');
```

```
dbms_output.put_line(msg);
```

```
end;
```

```
/
```

Vehicle available.

Output: booking open

```
declare
mesg varchar2(200);
begin
mesg := dbms_information(206, 'ram');
dbms_output.put_line(mesg);
end;
/
```

~~After vehicle available~~

Output:- no booking available.

Date 23-9-25

PL/SQL LOOPS

Procedure :-

- Start a PL/SQL block or Procedure
- Use a cursor (if required) to fetch customer IDs from a table.
- for each ID, check whether it is a prime number using a loop.
- use for loop/while loop to demonstrate prime number checking.
- Print the result using dbms-output.put-line.
- End the block.

Using while loop with cursor

Create or replace procedure Print-Prime customers%
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;  
        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;  
        if V-is-Prime then  
            dbms-output.put-line(V-id);  
        end if;  
    end loop;  
end;
```

(writing on model 102) It is a good
example of recursive definition (using P) based on some
initial values and a recursive rule.

Let's define a sequence of numbers a_n ($n \in \mathbb{N}$) defined by $a_0 = 1$, $a_1 = 2$, and for $n \geq 2$, $a_n = a_{n-1} + a_{n-2}$.

So, we have $a_0 = 1$, $a_1 = 2$, $a_2 = 3$, $a_3 = 5$, $a_4 = 8$, $a_5 = 13$, $a_6 = 21$, $a_7 = 34$, $a_8 = 55$, $a_9 = 89$, $a_{10} = 144$, etc.

Output

Prime: 2

Prime: 3

Prime: 5

Prime: 7

Prime: 11

Prime: 13

Prime: 17

Prime: 19

Prime: 23

Prime: 29

```

        END LOOP;
    END IF;
    IF v-is-prime THEN
        DBMS-OUTPUT-PUT-LINE('Prime customer
                                ID:' || v-id);
    END IF;
END LOOP;
CLOSE-CUST-CUR;
END;

```

procedure created.

using for loop for first N Prime Numbers.

Create or replace procedure Print-first-n-primes

```

V-num NUMBER := 2;           (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;

```

Procedure created.

ex :- Begin

end; Print-first-n-primes(10);

VEL TECH	
EX NO.	1
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	-
TOTAL (20)	15
SIGN WITH DATE	R 23/9/20

Result:- The ~~PLC~~ Procedures, functions and loops were successfully implemented.