

TASK #:-PL/SQL PROCEDURE, FUNCTIONS, LOOPS

Aim:- To Implement PL/SQL procedure, functions and loops on number theory and business scenarios.

Declaration :-

starts with the keyword DECLARE. It is an optional section and defines all variables, whereas subprograms and other elements to be used in the program.

Executable commands :-

enclosed between the keywords BEGIN and END and it is a main dataoy section.

Exception handling :-

starts with the key word EXCEPTION. This is optional. section contains exception that handle errors in the program.

Syntax :-

DECLARE

&lt;declaration sections&gt;

BEGIN

&lt;executable commands&gt;

exception

&lt;exception handling&gt;

END;

Query :-

```
declare
  message var(CHAR 2(20)) 'Admission open';
BEGIN
  dbms_output.put_line (message);
END;
```

Output:-

Admission open.

Query:-

```
set server output on;
declare
```

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

begin

```
  x := 10;
```

```
  y := 12;
```

```
  z := x * y
```

```
  dbms_output.put_line ("multiplication
of two num is " || z);
```

end;

Output:-

Multiplication of two num is 120.

Query :-

declare

Var 1 integer;

Var 2 integer;

Var 3 integer;

begin

Var 1 := Var 1;

Var 2 := \$Var 2;

Var 3 := Var 1 + Var 2;

dbms - output.put. lput < Var 3 >;

end;

/

Input

enter value for Var 1 : 60

Old 6: Var 1 := \$Var 1;

New 6: Var 1 := 20;

enter value for Var 2 : 30

Old 7: Var 2 := Var 2;

New 7: Var 2 := 30;

90

Declar

hid number(3); = 100;

BEGIN

IF (hid = 10) then

dbms - output put - lput ("Value of  
hid is 10");

Elseif (hid = 20) then

dbms - output.put - lput ("Value of hid is 20");

Elseif (hid = 30) then

else

dbms - output putline ("None of the values  
end if  
is matching");

dbms - output putline ("exact value of hid  
END;  
).

### Output:

None of the value is matching.

exact value of hid is : 100.

Loop

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 putline ("hid is : " || hid || " and  
oid is : " || oid);

end loop inner-loop;

end loop outer-loop;

end;

,

Output:-

hid pi : 1 and oid is : 1  
hid is : 1 and oid is : 2  
hid is : 1 and oid is : 3  
hid pi : 2 and oid is : 1  
hid pi : 2 and oid is : 2  
hid pi : 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

### while loop

Set serial output on:

Create or replace procedure print

- n - prime (n number) i;

- first

v - num. Number := 20;

v - Number := 0;

v - pi - prime boolean:

BEGIN

while v - count in loop

# v - is prime := true;

-- prime check using for loop

for i in ... Trun. (sqrt (v - num))  
loop

# mod (v - num) = 0 then

v - is - prime := false;

exit;

endif;

End loop;

if  $v - 11$  - prime then  
dbms\_output.put\_line ('prime  $11 - num$ ');  
 $v = count := v - count + 1$ ;  
End if;  
 $v = num := v - num + 1$   
end loop;  
end;

2  
3  
5  
7  
11  
13  
17  
19  
23  
29

while loop:-

Create or replace procedure print  
- prime customer is

cursor cur1-cur f.s.

Select sid from student

$v - id$  Number,

$v - pr$  prime boolean

$v - i$  Number;

Begin

loop

    fetch . cust-cur into v-id;

    except when cust-cur% NOT FOUND;

        -- prime check using while loop.

            IF  $v-pd < 2$  then

                v-is-prime := false;

        else

            v-is-prime := true;

            v-p i=2;

        while  $v-i < = \text{true}$  (start ( $v-id$ )) loop

            IF MOD ( $v-id, v-i$ ) = 0 then

                v-is-prime := false;

                exit

            end if;

            v-t i=v-i+1;

        end loop;

        end if;

        If  $v-is-prime$  then

            dbms - output putline ( prime student  
                ID , "||v-pd );

    end if;

End loop;

close cust-cur;

end;

Output:-

prime student ID: 2

prime student ID: 3

prime student ID: 5

VIVA EXAMINATION	EX NO.	
PERFORMANCE (5)	8/5	
RESULT AND ANALYSIS (5)	8	
VIVA VOCE (5)	3	
RECORD (5)	3	
TOTAL (20)	68	
SUMMARY	C	

23/9/15

RESULT:-

Implementation of PL/SQL procedures  
functions and loop on number theory.

has been successfully executed.