Task7:-

Date:

PL/SQL procedure, functions, Loops

Aim:- To implement PL/SQL procedure functions and loops number. Theory and business scenarios.

Declaration:

start 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 EXCEPTION This is optional section contains exception that handle errors in the program.

Exception Handling: keyword Exception This is optional section.

Syntax:

```
DECLARE
 <declaration section>
BEGIN.
 <Executable commands>
 Exception.
 <exception handlings>
 END;
```

Query:

```
DECLARE
 message varchar2(20) := 'booking closed';
BEGIN.
  dbms_output.put_line (message);
END;
```

Query

```
set serveroutput on;
declare
 x number(5);
 y number(5);
 z number(a);
```

```
begin.
x := 10;
y := 12;
z := x+y;
dbms_output.line ('sum is' || z);
end;
/
```

output : sum is 22.

Query

```
declare
var1 integer ;
var2 integeer;
var3 integeer ;
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

PL/SQL procedure successfully completed

```
DECLARE
  hid number(3) := 100;
BEGIN
  IF(hid = 10) THEN.
  dbms_output.put_line ('value of hid is 10');
```

```
ELSEIF (hid=20) THEN
  dbms_output.putline('value of hid is 20');
ELSEIF (hid=30) THEN
  dbms_output.putline('value of hid is 30');
ELSE
  dbms_output.put_line('none of the values is matching');
ENDIF
  dbms_output.putline ('Exact value of hid is:'||hid);
END;
/
```

none of the value is in matching
Exact value 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;
/
```

output :

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

PL/SQL procedure successfully completed.

sample program for only procedure.

```
create or replace procedure csinformation
(c_id in number, c_name in varchar2)
is
begin
dbms_output.put_line ('ID' :: c_id);
dbms_output.put_line ('name' : c_name);
end;
/
```

procedure created:

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

procedure succesfully completed

```
set serveroutput on;
exec csinformation (101 : 'raam');
```

ID : 101
name : raam
procedure succesfully completed

sample query for only function:

```
create or replace function csinformation
(h_id in number, c_name in varchar2)
Return varchar2.
IS
Begin.
If c_id > 100 then
```

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

Else
Return ("booking open');
End if;
End;
/

declare
mesg varchar. 2(200 ?;
begin.
mesg := csinformation 2 (102,'raam'?;
dbms- output.put-line (mesg ?
end;
/
```

vehicle available

```
declare
mesg varchar.(200?;
begin
mesg :=csinformation 2 (206,'raan');
dbms_output.put line (mesg?;
end;
/
```

No vehicle available
PL/SQL procedure successfully completed

while loop:

```
Set serve. output on:
create or replace procedure printfirst-n Primes (nnumber)
is
    v-NUM Number := 20:
    v- Number:=0:
    v-is-primeBoolean,:
```

```
Begin
while v_count <= n loop
  v_is_prime := True;
-- prime check using for loop
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;
/
```

output:

```
Exec print_first_n_primes(10);
2
3
5
7
11
13
17
19
23
27.
```

while loop.

```
Create or replace  procedure print_prime customer is
cursor cust_cur is
  select sin from student;
  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%not found;
-- prime check using while loop
If v_id <2 then
   v_is_prime := False;
Else
v_is_prime := True;
v_i := 2;
while v_i < True (sqrt(v_id)) loop
if MOD(v_id, v_i)=0 then
   v_is_prime := False;
   Exit;
   Endif;
   v_i := v_i+1;
Endloop;
Endif;
If v_is_prime then
dbms_output_line('prime student ID: '||v_id);
End if;
Endloop;
close cust_cur;
End;
/
```

~~For loop.~~

Output:

prime student ID: 2

prime student ID: 3

prime student ID: 5

FOR LOOP

```
CREATE OR REPLACE PROCEDURE print_first_n_prime(n NUMBER) IS
  v_num NUMBER := 2;
  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_first_n_primes(10);
END;
```

output:

```
Prime: 2
Prime: 3
Prime: 5
Prime: 7
Prime: 11
Prime: 13
Prime: 17
Prime: 19
Prime: 23
Prime: 29.
```

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

Result: Implementation of PL/SQL procedures, functions and loop on number theory has been successfully executed.