

# PL/SQL Procedure, functions, Loops

## Aim:

To implement PL/SQL procedures, Functions and loops on number theory & business scenarios.

S/no	Section & Description
1	<u>Declarations</u> This section starts with the keyword DECLARE. It is an optional section & defines all variables, cursors, subprograms and other elements to be used in the program.
2	<u>Executable Command</u> This section is enclosed between the keywords BEGIN & END it is a mandatory section. It consists of the executable statements of the program.
3	<u>Exception Handling</u> This section starts with the keyword <del>EXCEPTION</del> .

Simple program to print a sentence:

SYNTAX: DECLARE

< section >

BEGIN

<executable section>

Exception

<Exception handling>

END;

⌞

## Program

Declare

message varchar2(20) := 'booking closed';

BEGIN

dbms\_output.put\_line(message);

END;

## Static input

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.

★ 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.put\_line('value of hid is 20');



ELSEIF (hid = 30) Then

dbms\_output.put\_line('value of hid is 30');

ELSE

dbms\_output.put\_line('None of the value is matching');

END IF;

dbms\_outputline('Exact value of hid is:');

END;

/

PL/SQL procedure successfully completed.

DECLARE

hid number(1);

oid number(1);

BEGIN

<< outer loop >>

FOR oid IN 1..3 LOOP

dbms\_output.put\_line('hid is: ' || hid || 'and  
oid is: ' || oid);

END loop inner-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

## sample program for only function

```
SQL> create or replace function csinformation  
  (c-id in number, c-name in varchar2)  
  Return varchar 2  
  Is  
  Begin  
  If c-id > 200 then  
  Return ('no booking available');  
  Else  
  Return ('booking open')  
  End if;  
  End;
```

### Function created

```
SQL> declare  
  2 msg varchar 2(200);  
  3 begin  
  4 msg = csinformation 2 (206, 'raam');  
  5 dbms_output.put_line (msg);  
  6 end;  
  7 /
```

No vehicle available.

VEL TECH	
EX No.	
PERFORMANCE (5)	
RESULT AND ANALYSIS (10)	
VIVA VOCE (5)	
RECORD (5)	
TOTAL (20)	
SIGN WITH DATE	

Result: Implementation of PL & SQL procedures  
functions & loops on number theory and  
business has been done successfully.



PL/SQL Procedure for loopsAim:

To write PL/SQL programs using loops for printing prime numbers customers IDs and for demonstrating loop control in different scenarios

Procedure

- 1) Start a PL/SQL block or procedure
- 2) Use a cursor (if required) to fetch customers IDs
- 3) For each ID, check whether it is a prime number ~~using~~ checking.
- 4) USE FOR LOOP/ WHILE LOOP to demonstrate prime numbers checking.
- 5) Print the result using DBMS-OUTPUT.PUT-LINE
- 6) End the block

Example 1: Using WHILE LOOP cursor

Prime check using WHILE loop

```

Create OR REPLACE PROCEDURE print_prime_customers IS
  Cursor cust_cur IS
    SELECT cusmer_id FROM customers;

  v_id Number;
  v-is-prime Boolean;
  v-i NUMBER;

```

BEGIN

open cust-cur;

Loop

FETCH cust-cur INFO v-id;

EXIT WHEN cust-cur % NOT FOUND;

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 ;

ENDLOOP;

ENDIF

IF v-is-prime THEN

DBMS\_OUTPUT.PUT\_LINE('Prime Customer  
ID: ' ||

v-id);

END IF;

ENDLOOP;

CLOSE cust-cur;

END;

This procedure checks all customers IDs in the table and prints the prime one using a WHILE LOOP.

Example 2: Using For Loop for first N prime Numbers

Create OR REPLACE PROCEDURE print\_first\_n  
primes

v-num NUMBER := 2;

v-count NUMBER := 0;

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;

END IF

v-num := v-num + 1;

END LOOP;

END;



output

Prime: 2

Prime: 3

Prime: 5

Prime: 7

Prime: 11

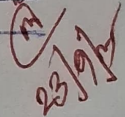
Prime: 13

Prime: 17

Prime: 19

Prime: 23

Prime: 29

VEL TECH	
EX NO.	7
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	5
TOTAL (20)	20
SIGN WITH DATE	 23/9/24

Result: thus, the implementation of PL/SQL procedures, functions & loops on number theory has been successfully completed.