**Ex.No .7** **STUDY OF PL/SQL PROGRAMS**

**AIM:**

To Study about PL/SQL program structures and implement simple programs using PL/SQL.

**DESCRIPTION:**

**PL/SQL**

* PL/SQL stands for Procedural Language extension of SQL.
* 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.

**A Simple PL/SQL Block**

Each PL/SQL program consists of SQL and PL/SQL statements which from a PL/SQL

block.

**PL/SQL Block consists of three sections**

* The Declaration section (optional).
* The Execution section (mandatory).
* The Exception (or Error) Handling section (optional).

**Declaration Section**

The Declaration section of a PL/SQL Block starts with the reserved keyword DECLARE. This section is optional and is used to declare any placeholders like variables, constants, records and cursors, which are used to manipulate data in the execution section. Placeholders may be any of Variables, Constants and Records, which stores data temporarily. Cursors are also declared in this section.

**Execution Section**

The Execution section of a PL/SQL Block starts with the reserved keyword BEGIN and ends with END. This is a mandatory section and is the section where the program logic is written to perform any task. The programmatic constructs like loops, conditional statement and SQL statements from the part of execution section.



**Exception Section**

The Exception section of a PL/SQL Block starts with the reserved keyword EXCEPTION. This section is optional. Any errors in the program can be handled in this section, so that the PL/SQL Blocks terminates gracefully. If the PL/SQL Block contains exceptions that cannot be handled, the Block terminates abruptly with errors.

Every statement in the above three sections must end with a semicolon ; . PL/SQL blocks can be nested within other PL/SQL blocks. Comments can be used to document code.

A Sample PL/SQL Block Looks like: DECLARE

Variable declaration

BEGIN

Program Execution

EXCEPTION Exception handling

END;

**PL/SQL Block Structure:**

DECLARE

v\_variable VARCHAR2(5);

BEGIN

SELECT column\_name

INTO v\_variable

FROM table\_name;

EXCEPTION

WHEN exception\_name THEN

...

END;

HELLO PROGRAM

SQL> edit hello

SQL> declare

2 msg varchar2(10);

3 begin

4 msg:='hello';

5 dbms\_output.put\_line('message='||msg);

6 end;

7 /

OUTPUT

message=hello

PL/SQL procedure successfully completed.

ODD OR EVEN

SQL> edit oddeven

SQL> declare

2 n number(3);

3 begin

4 n:=&n;

5 if (mod(n,2)=0) then

6 dbms\_output.put\_line(n||'even number');

7 else

8 dbms\_output.put\_line(n||'odd number');

9 end if;

10 end;

11 /

OUTPUT

Enter value for n: 5

old 4: n:=&n;

new 4: n:=5;

5odd number

PL/SQL procedure successfully completed.

ADDITION OF TWO NUMBER

SQL>edit add

SQL> declare

2 a number(3);

3 b number(3);

4 c number(3);

5 begin

6 a:=&a;

7 b:=&b;

8 c:=a+b;

9 dbms\_output.put\_line('the addition is '||c);

10 end;

11 /

OUTPUT

Enter value for a: 2

old 6: a:=&a;

new 6: a:=2;

Enter value for b: 3

old 7: b:=&b;

new 7: b:=3;

the addition is 5

PL/SQL procedure successfully completed.

FACTORIAL

SQl>edit fact

SQL> declare

2 n number(3);

3 fact number(3);

4 begin

5 n:=&n;

6 fact:=1;

7 for i in 1..n

8 loop

9 fact:=fact\*i;

10 end loop;

11 dbms\_output.put\_line('the factorial is'||fact);

12 end;

13 /

OUTPUT

Enter value for n: 5

old 5: n:=&n;

new 5: n:=5;

the factorial is120

PL/SQL procedure successfully completed.

UPDATING THE SALARY

edit emp\_5

SQL> declare

2 newsal number(8,2);

3 a number(8,2);

4 begin

5 select salary into a from emp\_5 where empno=101;

6 newsal:=a+(a\*10/100);

7 update emp\_5 set salary=newsal where empno=101;

8 end;

9 /

PL/SQL procedure successfully completed.

OUTPUT

SQL> select \* from emp\_5;

EMPNO EMPNAME DEPT SALARY

---------- ---------- ---------- ----------

101 ragu it 55000

102 ram cse 60000

103 ragavan ece 40000

1 rathika eee 30000

2 yokesh eee 30000