Experiment No. 8. Regd. No.

Aim:- Write a PL/SQL Code using Basic Variable, Anchored declarations, and usage of Assignment operation PL/SQL

INTRODUTION.

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.

Oracle uses a PL/SQL engine to process the PL/SQL statements. A PL/SQL Code Can be stored in the Client system (client side) or in the database (server-side).

Advantages of PL/SQL:

- Block structures: PL SQL Consists of blocks of code, which can be nested within each other. Each block forms a unit of a task or a logical module. PL/SQL blocks can be stored in the database and reused.
- Procedural language Capability:- PL SQL Consists of procedure language Constructs such as conditional statements (if else statements) and loops like (FOR loops).
- Better Performance: PL SQL engine process multiple SQL statements simultaneously as a single block, thereby reducing network traffic.
- Error Handling: PL/SQL handles errors or exceptions effectively during the execution of a PL/SQL program.

Experiment No. Regd. No. .

Once an execution is caught, specific actions can be taken.
an exception is caught, specific actions can be taken depending upon the type of the exception or it can be displayed to the user with a message.

Syntax of PL/SQL program:

```
Declare
    Variables
    declaration;
Begin
    Executable statements;
end;
```

PL/SQL Variables Declaration.

→ To declare a variables, you use a variable name followed by the data type and terminated by a semicolon(;).

→ you can also explicitly add a length constraint to the data type within parantheses. The following illustrates.

→ Some examples of declaring variables in a PL/SQL anonymous block:

```
DECLARE
n_employee_id number;
v_first_name varchar2(20);
d_hire_data date;
BEGIN;
NULL;
END;
```

Experiment No. Regd. No.

In PL/SQL program, one of the most common tasks it to select values from columns in a table into a set of variables.

In case the data types of columns of the table changes, you have to change the PL/SQL program to make the types of the variables compatible with the new changes.

PL/SQL provides you with a very useful feature called Variable anchors. It refers to the use of the %TYPE keyword to declare a variable with the data type is associated with a column's data type or a particular column in a table.

```
DECLARE
n_employee_id    EMP.EMPNO%TYPE;
v_first_name     EMP.ENAME%TYPE;
d_hire_date      EMP.HIREDATE%TYPE;
BEGIN
NULL;
END;
/
```

PL/SQL variable assignment: In PL/SQL, to assign a value of a variable to another, you use the assignment operator (:=) which is a colon (:) followed by the equal sign (=)

```
DECLARE
```

Experiment No. Regd. No. .

```
n_employee_id     EMP.EMPNO%TYPE;
v_first_name      EMP.ENAME%TYPE;
d_hire_date       EMP.HIREDATE%TYPE;
BEGIN
v_first_name := 'Balu';
n_employee_id := 8000;
d_hire_date := to_date('19960101','yyyymmdd');
END;
/
:

SET SERVEROUTPUT ON SIZE 100000;
DECLARE
n_employee_id     EMP.EMPNO%TYPE;
v_first_name      EMP.ENAME%TYPE;
d_hire_date       EMP.HIREDATE%TYPE;

BEGIN
SELECT EMPNO, ENAME, HIREDATE INTO
n_employee_id, v_first_name, d_hire_date FROM
emp WHERE EMPNO = 9934;
DBMS_OUTPUT.PUT_LINE('Employee details');
DBMS_OUTPUT.PUT_LINE(n_employee_id);
DBMS_OUTPUT.PUT_LINE(d_hire_date);
END;
/
```

Output:-

Employee Details

7934

MILLER

23-JAN-82

Experiment No. 9.

Regd. No. .

Aims:- Write a PL/SQL Code Bind and Substitution variable printing in PL/SQL.

1. Substitution Variables

The clue here is the name "substitution". It relates to values being substituted into the code before it is submitted to the database.

These substitutions are carried out by the interface being used. In this example were going to use SQL * plus as our interface.

```
Create or replace function myth return varchar2 is
v_dname varchar2(20);
begin
Select dname into v_dname from dept where deptno=
& p_deptno;
return v_dname;
end;
/
```

Output :-

```
SQL>
Enter value for p_deptno : 20
old 7 : where deptno = &p_deptno;
new 7 : where deptno = 20;
```

Experiment No. 10.

Regd. No. .

Write a PL/SQL block using SQL and control structures in PL/SQL.

Conditional statements in PL/SQL :-

As the name implies, PL/SQL supports programming language features like conditional statements, iterative statements. The programming constructs are similar to how you see in programming languages like java and c++.

IF THEN ELSE STATEMENT :

1) IF Condition THEN Statement 1;
ELSE
statement 2;
END IF ;

2) IF Condition 1 THEN statement 1; statement 2;
ELSIF Condition 2 THEN
Statement 3; ELSE
Statement 4; END

LOOPS in PL/SQL :- There are three types of loops in SQL/PL.

1. simple loop
2. while loop
3. FOR loop.

simple LOOP : A simple loop is used when a set of statements is to be executed at least one before the loop terminates. An Exit Condition must be specified in the loop.

Experiment No. Regd. No. .

Syntax:

LOOP Exit: statements;
{or EXIT WHEN Condition;}

2. While loop: A WHILE LOOP is used when a set of statements has to be executed as long as a condition is true. The condition is evaluate at the beginning of each iteration. The iteration until the condition becomes false.

Syntax:- WHILE <condition>
LO
OP
Statement
S; END
LOOP;

3. FOR LOOP: A FOR LOOP is used to execute a set of statements for a pre-determined no. of times. Iteration occurs b/w the start and end interger values given.

Syntax:

FOR Counter IN val1..val2
LO
OP
statements;
END LOOP;

Experiment No. Regd. No. .

Example - 1:

Write a PL/SQL program to find the Arithmetic operations by using if, else if, else

```
SET SERVEROUTPUT ON
SET VERIFY OFF
DCLARE
a number(3):=&a;
b number(3):=&b;
ch number(3):=&ch;
r number(4);
BEGIN
IF ch=1 THEN
r:=a+b;
DBMS_OUTPUT.PUT_LINE('Addition:'||r);
ELSIF ch=2 THEN
r:=a-b;  (# r:=a-b;)
DBMS_OUTPUT.PUT_LINE('subtraction:'||r);
ELSIF ch=3 THEN
r:=a*b;
DBMS_OUTPUT.PUT_LINE('Multiplication:'||r);
ELSIF ch=4 THEN
r:=a/b;
DBMS_OUTPUT.PUT_LINE('Division:'||r);
ELSIF ch=5 THEN
r:=mod(a,b);
```

Experiment No. Regd. No. .

```
DBMS_OUTPUT.PUT_LINE('Remainder : '||r);
ELSE
DBMS_OUTPUT_LINE ('Invalid choice :... ");
END IF;
END;
/
```

Eg-2: Write a PL/SQL Program on Simple LOOP.

```
SET SERVEROUTPUT ON
SET VRIFY OFF
DCLARE
i number(3);
n number(3):=&n;
BEGIN
i:=1;
LOOP
        DBMS_OUTPUT.PUT_LINE(i);
        i:=i+1;
EXIT WHEN i>n;
END LOOP;
END;
/
```

Output :-

```
Enter value for a:10
Enter value for b:8
Enter value for ch:1
Addition :18.
```

Output :-

```
Enter value for n:5
1
2
3
4
```

Experiment No. Regd. No. .

Eg-3:- Write a PL/SQL program on while loop.

```
SET SERVEROUTPUT ON
SET VERIFY OFF
DCLARE
    ; number(3);
    n number(3):= &n;
BEGIN
    i:=1;
    WHILE i<=n
    LOOP
        DBMS_OUTPUT.PUT_LINE(i);
        i:=i+i;    (# i:= i+1);
    END LOOP;
    END;
/
```

Example-4 : Write a PL/SQL program on for loop.

```
SET SERVEROUTPUT ON
SET VERIFY OFF
DECLARE
        n number(3):=&n;
BEGIN
        FOR i IN 1..n
LOOP    DBMS_OUTPUT.PUT_LINE(i);
    END LOOP;
END;
/
```

Output:-

```
Enter value for n: 3
1
2
3
```

Output:-

```
Enter value for n: 3
1
2
3
```