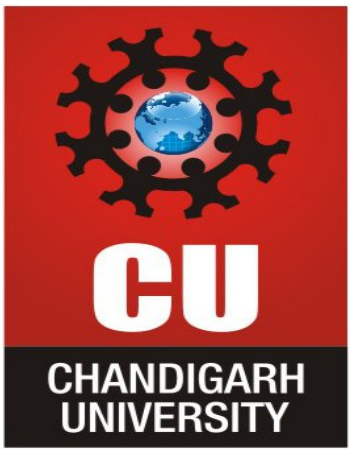


NAME: Manan Verma **SEMESTER:** 4th

SECTION: 24_AIT_KRG-G1 **UID:** 24BAI70008

SUBJECT CODE: 24CSH-298

FACULTY'S NAME: Mr. SHALABH BHATIA

Topic : Experiment 3

AIM :

To understand the basic structure of a PL/SQL program by creating and executing a simple PL/SQL block that includes **declaration** and **execution** sections, and to display output using built-in procedures.

Software Requirements

- **Database Management System:**
    - PostgreSQL
- **Database Administration Tool:**
    - pgAdmin

Objective:

To create a simple PL/SQL program demonstrating **Declaration Section** and **Execution Section**.

Problem Statement:

Design and implement a simple PL/SQL program that demonstrates the basic structure of a PL/SQL block. The program should include a declaration section to define variables and an execution section to perform operations using those variables and display the results using appropriate output statements.

1. Declaration Section (DECLARE)

- Variables are declared and initialized:
    - **emp_id** → Employee ID
    - **emp_name** → Employee Name
    - **emp_salary** → Employee Salary

2. Execution Section (BEGIN ... END)

- **DBMS_OUTPUT.PUT_LINE** is used to display output.

Procedure:

1. Open pgAdmin and connect to the PostgreSQL database server.
2. Create an employee table with employee ID, name, and salary.
3. Insert sample data into the employee table.
4. Write a PL/pgSQL anonymous block using the DO $$ command.
5. Declare variables in the **DECLARE** section to store employee details.

6.In the **BEGIN…END** section:

- Retrieve employee data from the table into variables.
- Perform a basic operation (calculate 10% salary increment).
- Display employee details and calculated salary using output statements.

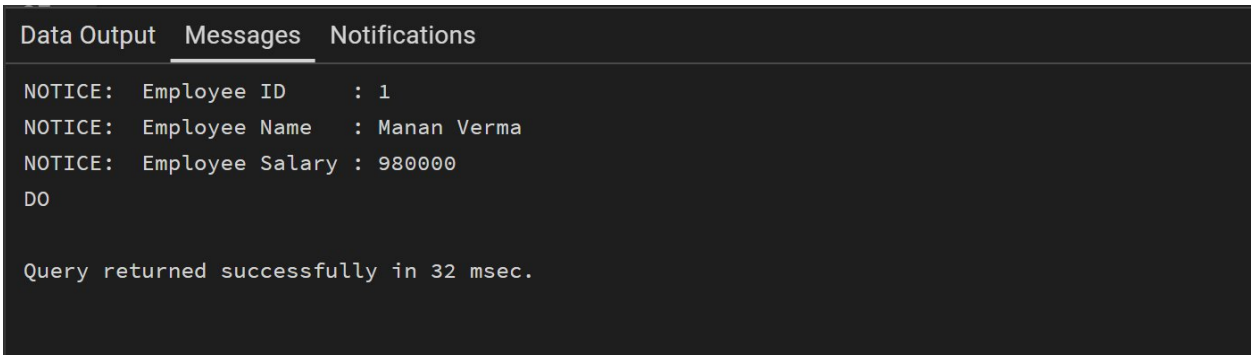
7.Execute the block and observe the output in the Messages/Notices panel of pgAdmin.

CODE:

```
CREATE TABLE employee (
   emp_id INTEGER PRIMARY KEY,
   emp_name VARCHAR(50),
   emp_salary NUMERIC
);


INSERT INTO employee VALUES (01, Manan, 980000);
DO $$
DECLARE
   v_id INTEGER;
   v_name VARCHAR(50);
   v_salary NUMERIC;
   v_bonus NUMERIC;
BEGIN
   SELECT emp_id, emp_name, emp_salary
   INTO v_id, v_name, v_salary
```

```
FROM employee

WHERE emp_id = 01;


v_bonus := v_salary * 0.10;


RAISE NOTICE 'Employee ID        : %', v_id;

RAISE NOTICE 'Employee Name      : %', v_name;

RAISE NOTICE 'Original Salary    : %', v_salary;

RAISE NOTICE '10%% Bonus Amount  : %', v_bonus;

RAISE NOTICE 'Updated Salary     : %', v_salary + v_bonus;

END $$;
```

Outputs:



Data Output | Messages | Notifications

	emp_id [PK] integer	emp_name character varying (50)	emp_salary numeric
1	1	Manan Verma	980000

violation/postgres@PostgreSQL 18*

violation/postgres@PostgreSQL 18

No limit

Query | Query History

```
1  CREATE TABLE employee (
2      emp_id INTEGER PRIMARY KEY,
3      emp_name VARCHAR(50),
4      emp_salary NUMERIC
5  );
6
7  INSERT INTO employee VALUES (01, 'Manan Verma', 980000);
8
9
10 DO $$
11 DECLARE
12     v_emp_id INTEGER;
13     v_emp_name VARCHAR(50);
14     v_emp_salary NUMERIC;
15 BEGIN
16
17     SELECT emp_id, emp_name, emp_salary
18     INTO v_emp_id, v_emp_name, v_emp_salary
19     FROM employee
20     WHERE emp_id = 01;
21
22
23     RAISE NOTICE 'Employee ID     : %', v_emp_id;
24     RAISE NOTICE 'Employee Name   : %', v_emp_name;
25     RAISE NOTICE 'Employee Salary : %', v_emp_salary;
26 END $$;
27
```

Data Output | Messages | Notifications

```
NOTICE:  Employee ID     : 1
NOTICE:  Employee Name   : Manan Verma
NOTICE:  Employee Salary : 980000
DO

Query returned successfully in 53 msec.
```

Learning Outcomes:

- Understand the structure of a PL/pgSQL program.

- Use the declaration section to define variables.
- Retrieve data from a table into variables.
- Perform basic arithmetic operations using variables.