



NAME:Simran Chouhan

SEMESTER:4th

SECTION:24_AIT_KRG-G1

UID:24BAI70124

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 (100, 'SIMRAN', 65000);
```

```
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 = 100;
```

```
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 $$;
```

OUTPUT:

```
Data Output Messages Notifications
NOTICE: Employee ID      : 100
NOTICE: Employee Name    : SIMRAN
NOTICE: Original Salary   : 65000
NOTICE: 10% Bonus Amount : 6500.00
NOTICE: Updated Salary     : 71500.00
DO

Query returned successfully in 100 msec.

Total rows: 1 Query complete 00:00:00.100
```

```
Data Output Messages Notifications
```

The screenshot shows a PostgreSQL pgAdmin interface. At the top, there are three tabs: 'Data Output', 'Messages', and 'Notifications'. Below the tabs is a toolbar with various icons for database management. The main area displays a table with four columns. The first column is empty. The second column contains the primary key 'emp_id' with the value '100'. The third column contains the name 'emp_name' with the value 'SIMRAN'. The fourth column contains the salary 'emp_salary' with the value '65000'. The table has a header row with column names and data types.

	emp_id [PK] integer	emp_name character varying (50)	emp_salary numeric
1	100	SIMRAN	65000

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.