Experiment No. 4

Student Name: Shubham Agarwal **UID: 25MCI10091**
Branch: MCA (AI & ML) **Section/Group: 25MAM-1/A**
Semester: 2nd **Date of Performance:03/02/2026**
Subject Name: Technical Training Lab **Subject Code: 25CAH-653**

Aim of the Session:

To understand and implement iterative control structures in PostgreSQL conceptually, including FOR loops, WHILE loops, and basic LOOP constructs, for repeated execution of database logic.

Software Requirements:

- PostgreSQL Database Server
- pgAdmin 4
- Windows Operating System

Objective of the Session:

- To understand why iteration is required in database programming
- To learn the purpose and behaviour of FOR, WHILE, and LOOP constructs
- To understand how repeated data processing is handled in databases
- To relate loop concepts to real-world batch processing scenarios
- To strengthen conceptual knowledge of procedural SQL used in enterprise systems

Practical Experiment Steps:

employee Table Creation :

```
CREATE TABLE employee (
    emp_id SERIAL PRIMARY KEY,
    emp_name VARCHAR(50),
    department VARCHAR(30),
    designation VARCHAR(30),
    salary INT,
    joining_year INT
);
```

```
--Insert data for employee table
INSERT INTO employee (emp_name, department, designation, salary, joining_year) VALUES
('Amit Sharma', 'IT', 'Developer', 30000, 2022),
('Neha Verma', 'HR', 'HR Executive', 35000, 2021),
('Nishant Sharma', 'HR', 'HR Executive', 35000, 2021),
('Nisha Sharma', 'HR', 'HR Executive', 35000, 2021),
('Neeraj Rajput', 'IT', 'Developer', 30000, 2022),
('Neeru Chawan', 'Sales', 'Sales Executive', 37000, 2023),
('Rahul Mehta', 'Finance', 'Accountant', 28000, 2023),
('Sneha Kapoor', 'IT', 'Senior Developer', 50000, 2019),
('Rohit Jain', 'Sales', 'Sales Executive', 32000, 2022),
('Pooja Singh', 'IT', 'Tester', 27000, 2024);
```

	emp_id [PK] integer	emp_name character varying (50)	department character varying (30)	designation character varying (30)	salary integer	joining_year integer
1	1	Amit Sharma	IT	Developer	30000	2022
2	2	Neha Verma	HR	HR Executive	35000	2021
3	3	Nishant Sharma	HR	HR Executive	35000	2021
4	4	Nisha Sharma	HR	HR Executive	35000	2021
5	5	Neeraj Rajput	IT	Developer	30000	2022
6	6	Neeru Chawan	Sales	Sales Executive	37000	2023
7	7	Rahul Mehta	Finance	Accountant	28000	2023
8	8	Sneha Kapoor	IT	Senior Developer	50000	2019
9	9	Rohit Jain	Sales	Sales Executive	32000	2022
10	10	Pooja Singh	IT	Tester	27000	2024

Example 1: FOR Loop – Simple Iteration

```
DO $$
DECLARE
   i INT;
BEGIN
   FOR i IN 1..5 LOOP
      RAISE NOTICE 'Processing batch number: %', i;
   END LOOP;
END $$;
```

```
NOTICE:  Processing batch number: 1
NOTICE:  Processing batch number: 2
NOTICE:  Processing batch number: 3
NOTICE:  Processing batch number: 4
NOTICE:  Processing batch number: 5
DO

Query returned successfully in 1 secs 337 msec.
```

Example 2: FOR Loop – Query Based

```
DO $$
DECLARE
   emp RECORD;
BEGIN
   FOR emp IN SELECT * FROM employee LOOP
      RAISE NOTICE
      'ID: %, Name: %, Dept: %, Salary: %',
      emp.emp_id, emp.emp_name, emp.department, emp.salary;
   END LOOP;
END $$;
```

```
NOTICE:  ID: 1, Name: Amit Sharma, Dept: IT, Salary: 33000
NOTICE:  ID: 2, Name: Neha Verma, Dept: HR, Salary: 38000
NOTICE:  ID: 3, Name: Nishant Sharma, Dept: HR, Salary: 38000
NOTICE:  ID: 4, Name: Nisha Sharma, Dept: HR, Salary: 38000
NOTICE:  ID: 5, Name: Neeraj Rajput, Dept: IT, Salary: 33000
NOTICE:  ID: 6, Name: Neeru Chawan, Dept: Sales, Salary: 40000
NOTICE:  ID: 7, Name: Rahul Mehta, Dept: Finance, Salary: 31000
NOTICE:  ID: 8, Name: Sneha Kapoor, Dept: IT, Salary: 53000
NOTICE:  ID: 9, Name: Rohit Jain, Dept: Sales, Salary: 35000
NOTICE:  ID: 10, Name: Pooja Singh, Dept: IT, Salary: 30000
DO

Query returned successfully in 52 msec.
```

Example 3: WHILE Loop – Conditional Iteration

```
DO $$
DECLARE
   i INT := 0;
BEGIN
   WHILE i <= 7 LOOP
      RAISE NOTICE '%',i;
      i := i + 1;
   END LOOP;
END $$;
```

```
NOTICE:  Loop execution count: 1
NOTICE:  Loop execution count: 2
NOTICE:  Loop execution count: 3
NOTICE:  Loop execution count: 4
NOTICE:  Loop execution count: 5
DO

Query returned successfully in 85 msec.
```

Example 4: LOOP with EXIT WHEN

```
DO $$
DECLARE
   counter INT := 1;
BEGIN
   LOOP
      RAISE NOTICE 'Loop execution count: %', counter;
      counter := counter + 1;

      EXIT WHEN counter > 5;
   END LOOP;
END $$;
```

```
NOTICE:  0
NOTICE:  1
NOTICE:  2
NOTICE:  3
NOTICE:  4
NOTICE:  5
NOTICE:  6
NOTICE:  7
DO

Query returned successfully in 82 msec.
```

Example 4: LOOP with EXIT WHEN

```
DO $$
DECLARE
   counter INT := 1;
BEGIN
   LOOP
      RAISE NOTICE 'Loop execution count: %', counter;
      counter := counter + 1;

      EXIT WHEN counter > 5;
   END LOOP;
END $$;
```

```
NOTICE:  Loop execution count: 1
NOTICE:  Loop execution count: 2
NOTICE:  Loop execution count: 3
NOTICE:  Loop execution count: 4
NOTICE:  Loop execution count: 5
DO

Query returned successfully in 85 msec.
```

Example 5: Salary Increment using FOR Loop

```
DO $$
DECLARE
   emp RECORD;
BEGIN
   FOR emp IN SELECT emp_id FROM employee LOOP
      UPDATE employee
```

```
 SET salary = salary + 3000
      WHERE emp_id = emp.emp_id;
   END LOOP;
END $$;
```

	emp_id [PK] integer	emp_name character varying (50)	department character varying (30)	designation character varying (30)	salary integer	joining_year integer
1	1	Amit Sharma	IT	Developer	33000	2022
2	2	Neha Verma	HR	HR Executive	38000	2021
3	3	Nishant Sharma	HR	HR Executive	38000	2021
4	4	Nisha Sharma	HR	HR Executive	38000	2021
5	5	Neeraj Rajput	IT	Developer	33000	2022
6	6	Neeru Chawan	Sales	Sales Executive	40000	2023
7	7	Rahul Mehta	Finance	Accountant	31000	2023
8	8	Sneha Kapoor	IT	Senior Developer	53000	2019
9	9	Rohit Jain	Sales	Sales Executive	35000	2022
10	10	Pooja Singh	IT	Tester	30000	2024

Example 6: LOOP + IF Condition

```
DO $$
DECLARE
   emp RECORD;
BEGIN
   FOR emp IN SELECT emp_name, salary FROM employee LOOP
      IF emp.salary >= 40000 THEN
         RAISE NOTICE '% is a Senior Employee', emp.emp_name;
      ELSE
         RAISE NOTICE '% is a Junior Employee', emp.emp_name;
      END IF;
   END LOOP;
END $$;
```

```
NOTICE:  Amit Sharma is a Junior Employee
NOTICE:  Neha Verma is a Junior Employee
NOTICE:  Nishant Sharma is a Junior Employee
NOTICE:  Nisha Sharma is a Junior Employee
NOTICE:  Neeraj Rajput is a Junior Employee
NOTICE:  Neeru Chawan is a Senior Employee
NOTICE:  Rahul Mehta is a Junior Employee
NOTICE:  Sneha Kapoor is a Senior Employee
NOTICE:  Rohit Jain is a Junior Employee
NOTICE:  Pooja Singh is a Junior Employee
DO

Query returned successfully in 81 msec.
```

I/O Analysis

Input:

- Employee Data

Output:

- Employee records were processed repeatedly using **FOR loops** to display and update data.
- **Query-based FOR loop** classified and handled each employee record individually.
- **WHILE loop** executed until the given condition became false, showing conditional iteration.
- **LOOP with EXIT** demonstrated controlled termination of repeated execution.
- Salary values were **updated iteratively**, simulating payroll processing.
- Conditional checks inside loops produced **procedural validation messages** during execution.

Learning Outcomes

- Understand the concept of iterative control structures in PostgreSQL and explain their role in procedural database programming.
- Identify scenarios in database systems where loops such as FOR, WHILE, and LOOP are required.
- Differentiate between various loop constructs in PostgreSQL based on their behavior and use cases.
- Apply iterative logic to simulate real-world database operations such as batch processing, record traversal, and conditional execution.
- Develop a foundational understanding of PL/pgSQL required for implementing procedural logic in enterprise-grade database applications.