UNIVERSITY INSTITUTE of COMPUTING
Asia's Fastest Growing University

Experiment 4

Student Name: Sumanyu Seth **UID: 25MCI10040**

Branch: MCA (AI & ML) **Section/Group: 25MAM_KAR-1_A**

Semester: II **Date of Performance: 03/02/26**

Subject Name: Technical Training **Subject Code: 25CAP-652**

Title: Implementation of Iterative Control Structures using FOR, WHILE, and LOOP in PostgreSQL

1. **Aim:**
   a. 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.

2. **Software Requirements**:
   a. PostgreSQL (Database Server)
   b. pgAdmin
   c. OS: Windows

3. **Objective:**
   a. To understand why iteration is required in database programming
   b. To learn the purpose and behavior of FOR, WHILE, and LOOP constructs
   c. To understand how repeated data processing is handled in databases
   d. To relate loop concepts to real-world batch processing scenarios
   e. To strengthen conceptual knowledge of procedural SQL used in enterprise systems.

4. **Procedure of the Practical:**
   a. Start the system.
   b. Open pgAdmin.
   c. Create and select the database in which you want to perform the experiment.
   d. Establish connection to the database using Alt+Shift+Q.
   e. Run the following queries given in Experiment Steps (5).

5. Experiment Steps:

Example 1: FOR Loop – Simple Iteration

For Loop (simple iteration):

```
DO $$
BEGIN
  FOR i IN 1..6 LOOP
    RAISE NOTICE 'Number: %', i;
  END LOOP;
END $$;
```

```
NOTICE:  Number: 1
NOTICE:  Number: 2
NOTICE:  Number: 3
NOTICE:  Number: 4
NOTICE:  Number: 5
NOTICE:  Number: 6
DO
```

Example 2: FOR Loop with Query (Row-by-Row Processing)

```
do $$
declare
   emp_record record;
begin
   for emp_record in
      select emp_id, emp_name, salary from employee
   loop
      raise notice 'Employee: %, Salary: %',
      emp_record.emp_name, emp_record.salary;
   end loop;
end;
$$;
```

```
NOTICE:  Employee: Riya, Salary: 45000
NOTICE:  Employee: Karan, Salary: 60000
NOTICE:  Employee: Amit, Salary: 55000
DO
```

Example 3: WHILE Loop – Conditional Iteration

```
do $$
declare
   v_count int := 0;
begin
   while v_count < 3 loop
      raise notice 'Validation attempt: %', v_count;
```

```
        v_count := v_count + 1;
    end loop;
end;
$$;
```

```
NOTICE:  Validation attempt: 0
NOTICE:  Validation attempt: 1
NOTICE:  Validation attempt: 2
DO
```

Example 4: LOOP with EXIT WHEN

```
do $$
declare
    v_count int := 0;
begin
    loop
        raise notice 'Checking schema iteration: %', v_count;
        if v_count = 2 then
            exit;
        end if;
        v_count := v_count + 1;
    end loop;
end;
$$;
```

```
NOTICE:  Checking schema iteration: 0
NOTICE:  Checking schema iteration: 1
NOTICE:  Checking schema iteration: 2
DO
```

Example 5: Salary Increment Using FOR Loop

```
do $$
declare
    emp_record record;
begin
    for emp_record in
        select emp_id, salary from employee
    loop
        update employee
        set salary = salary + 2000
        where emp_id = emp_record.emp_id;
    end loop;
end;
$$;
```

Before:

	emp_id [PK] integer	emp_name character varying (50)	salary integer
1	102	Riya	45000
2	103	Karan	60000
3	101	Amit	55000

After:

	emp_id [PK] integer	emp_name character varying (50)	salary integer
1	102	Riya	47000
2	103	Karan	62000
3	101	Amit	57000

Example 6: Combining LOOP with IF Condition

```
do $$
declare
   order_rec record;
begin
   for order_rec in
      select customer_name, price from orders
   loop
      if order_rec.price > 50000 then
         raise notice 'High value order by %', order_rec.customer_name;
      else
         raise notice 'Normal order by %', order_rec.customer_name;
      end if;
   end loop;
end;
$$;
```

	order_id [PK] integer	customer_name character varying (50)	product character varying (50)	quantity integer	price numeric (10,2)
1	1	amit	laptop	1	65000.00
2	2	neha	mobile	2	40000.00
3	3	rohan	tablet	1	25000.00
4	4	simran	laptop	1	70000.00
5	5	ankit	mobile	3	60000.00
6	6	pooja	headphones	2	5000.00
7	7	rahul	tablet	2	48000.00

```
NOTICE:  High value order by amit
NOTICE:  Normal order by neha
NOTICE:  Normal order by rohan
NOTICE:  High value order by simran
NOTICE:  High value order by ankit
NOTICE:  Normal order by pooja
NOTICE:  Normal order by rahul
DO
```

6. Learning Outcome

a. Understanding of how iterative control structures work in PostgreSQL at a conceptual level.
b. Usage of loops in database systems, such as workflow engines, complex decision cycles, validation loops, etc.
c. Foundational knowledge required for writing procedural logic in enterprise-grade applications.