



**University Institute of Engineering**

**Department of Computer Science & Engineering**

## **Experiment: 2**

**Student Name:** Habis Prakash Deka

**UID:** 24BDA70031

**Branch:** CSE

**Section/Group:** AIT-KRG-GP2

**Semester:** 4th

**Date of Performance:** 7/01/26

**Subject Name:** DBMS

**1. Aim of the practical:** To understand and implement SQL SELECT queries using various clauses such as WHERE, ORDER BY, GROUP BY, and HAVING to retrieve and manipulate data efficiently from relational database tables.

**2. Tool Used:**

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

**3. Objective:**

- To practice writing SQL SELECT statements.
- To apply filtering conditions using the WHERE clause.
- To sort query results using the ORDER BY clause.
- To group records using the GROUP BY clause.
- To filter grouped data using the HAVING clause.
- To analyze data using aggregate functions like COUNT(), SUM(), AVG(), MIN(), and MAX().



# University Institute of Engineering

## Department of Computer Science & Engineering

### **4. Practical / Experimental Steps**

- (1) Start the system and log in to the computer.
- (2) Open PgAdmin (PostgreSQL).
- (3) Create or select the required database (e.g., lab\_db).
- (4) Create the EMPLOYEE table using the given schema.
- (5) Insert sample data into the EMPLOYEE table.
- (6) Execute the queries step-by-step according to the practical steps.
- (7) Verify the output after each query execution.
- (8) Capture screenshots of execution and results for record.
- (9) Save the work and upload worksheet (Word + PDF) on GitHub.

### **5. I / O Analysis**

#### **DATABASE DESIGN**

Query to create Table employee :

```
CREATE TABLE employee (
    emp_id      INT PRIMARY KEY,
    emp_name    VARCHAR(50),
    department   VARCHAR(50),
    salary      NUMERIC(10,2),
    joining_date DATE
);
```

#### **DATA MANIPULATION**



# University Institute of Engineering

## Department of Computer Science & Engineering

Insert Sample records in the table

```
INSERT INTO employee (emp_id, emp_name, department, salary, joining_date) VALUES  
(101, 'Amit Sharma', 'IT', 45000, '2022-01-10'),  
(102, 'Neha Verma', 'HR', 22000, '2021-03-15'),  
(103, 'Rahul Singh', 'IT', 30000, '2020-06-20'),  
(104, 'Priya Mehta', 'Finance', 55000, '2019-09-05'),  
(105, 'Karan Gupta', 'HR', 18000, '2023-02-12'),  
(106, 'Sneha Kapoor', 'Finance', 28000, '2020-11-25'),  
(107, 'Rohit Jain', 'Sales', 35000, '2021-07-30'),  
(108, 'Ananya Joshi', 'Sales', 15000, '2022-12-01'),  
(109, 'Vikram Rao', 'IT', 25000, '2022-04-18');
```

```
INSERT 0 9
```

```
Query returned successfully in 91 msec.
```

### DISPLAY THE TABLE

```
SELECT * FROM employee;
```

	emp_id [PK] integer	emp_name character varying (50)	department character varying (50)	salary numeric (10,2)	joining_date date
1	101	Amit Sharma	IT	45000.00	2022-01-10
2	102	Neha Verma	HR	22000.00	2021-03-15
3	103	Rahul Singh	IT	30000.00	2020-06-20
4	104	Priya Mehta	Finance	55000.00	2019-09-05
5	105	Karan Gupta	HR	18000.00	2023-02-12
6	106	Sneha Kapoor	Finance	28000.00	2020-11-25
7	107	Rohit Jain	Sales	35000.00	2021-07-30
8	108	Ananya Joshi	Sales	15000.00	2022-12-01
9	109	Vikram Rao	IT	25000.00	2022-04-18

```
SELECT department, AVG(salary) AS avg_salary  
FROM employee  
GROUP BY department;
```



# University Institute of Engineering

## Department of Computer Science & Engineering

	department character varying (50) 	avg_salary numeric 
1	Finance	41500.0000000000000000
2	Sales	25000.0000000000000000
3	IT	33333.333333333333
4	HR	20000.0000000000000000

```
SELECT department, AVG(salary) AS avg_salary
FROM employee
WHERE salary > 20000
GROUP BY department;
```

	department character varying (50) 	avg_salary numeric 
1	Finance	41500.0000000000000000
2	Sales	35000.0000000000000000
3	IT	33333.333333333333
4	HR	22000.0000000000000000

```
SELECT department, AVG(salary) AS avg_salary
FROM employee
WHERE salary > 20000
GROUP BY department
HAVING AVG(salary) > 30000;
```

	department character varying (50) 	avg_salary numeric 
1	Finance	41500.0000000000000000
2	Sales	35000.0000000000000000
3	IT	33333.333333333333

```
SELECT department, AVG(salary) AS avg_salary
FROM employee
WHERE salary > 20000
GROUP BY department
HAVING AVG(salary) > 30000
ORDER BY avg_salary DESC;
```



# University Institute of Engineering

## Department of Computer Science & Engineering

	department character varying (50)	avg_salary numeric
1	Finance	41500.000000000000
2	Sales	35000.000000000000
3	IT	33333.333333333333

### 6. Learning outcomes (What I have learnt):

- Understood the syntax and usage of SQL SELECT statements.
- Gained practical knowledge of WHERE clause for filtering rows.
- Learned grouping operations using GROUP BY clause.
- Applied HAVING clause to filter grouped results.
- Sorted query outputs using ORDER BY clause.
- Got hands-on experience in PostgreSQL execution using PgAdmin.