

Student Name: DANISHMEET SINGH

UID:24BAI70719

Branch: Computer Science & Engineering (AIML)

Section/Group:24AIT-KRG-1/G2

Semester:4th

Subject Name: DBMS

1. Aim

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. Objective of the Session

- 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().

3. Practical / Experiment Steps

1. Display the department name and the average salary of employees for each department.
2. Consider only those employees whose salary is greater than 20,000.
3. Display only those departments where the average salary is greater than 30,000.
4. Arrange the final output in descending order of average salary.

4. Procedure of the Practical

- (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 (Input / Output Analysis)

Input: SQL commands and queries executed in PgAdmin (table creation, insertion, and SELECT queries).

Output: Result tables displayed in PgAdmin showing department-wise average salary after applying WHERE, HAVING, and ORDER BY clauses.

SQL Implementation (PgAdmin / PostgreSQL)

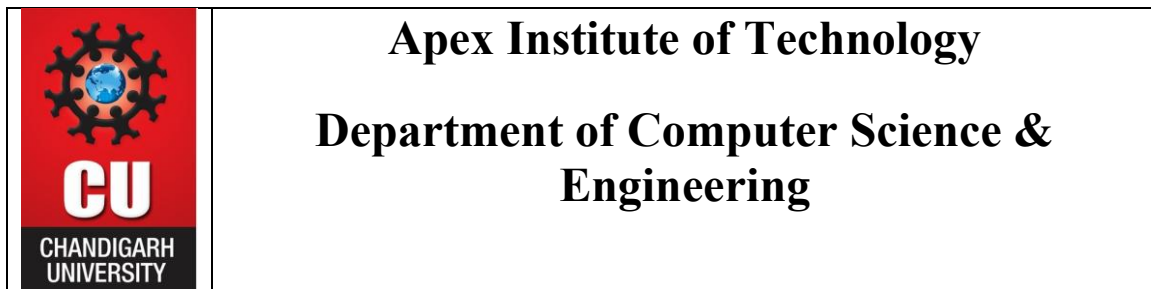
A) Create Database (Optional):

```
CREATE DATABASE lab_db;
```

B) Create Table:

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

C) Insert Sample Records:



```
INSERT INTO employee (emp_id, emp_name, department, salary,
joining_date) VALUES
(201, 'Arjun Malhotra', 'IT', 52000, '2023-02-11'),

(202, 'Meera Nair', 'HR', 26000, '2020-08-19'),

(203, 'Siddharth Rao', 'IT', 34000, '2021-04-10'),

(204, 'Ritika Desai', 'Finance', 60000, '2018-12-05'),

(205, 'Harshit Bansal', 'HR', 17000, '2024-01-22'),

(206, 'Tanya Kapoor', 'Finance', 31000, '2022-05-09'),

(207, 'Gaurav Khanna', 'Sales', 42000, '2020-09-14'),

(208, 'Isha Mathur', 'Sales', 14000, '2023-03-01'),

(209, 'Manoj Patil', 'IT', 29000, '2022-10-18');
```

Step 1 Query:

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

Step 2 Query:

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

Step 3 Query:

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

Step 4 Query (Final Output):

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

```
WHERE salary > 20000
GROUP BY department
HAVING AVG(salary) > 30000
ORDER BY avg_salary DESC;
```

6. Learning Outcome

- 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.

7. Screenshots

```
-- Create employee table
```

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

```
-- Insert new employee data
```

```
INSERT INTO employee (emp_id, emp_name, department, salary, joining_date) VALUES  
(201, 'Arjun Malhotra', 'IT', 52000, '2023-02-11'),  
(202, 'Meera Nair', 'HR', 26000, '2020-08-19'),  
(203, 'Siddharth Rao', 'IT', 34000, '2021-04-10'),  
(204, 'Ritika Desai', 'Finance', 60000, '2018-12-05'),  
(205, 'Harshit Bansal', 'HR', 17000, '2024-01-22'),  
(206, 'Tanya Kapoor', 'Finance', 31000, '2022-05-09'),  
(207, 'Gaurav Khanna', 'Sales', 42000, '2020-09-14'),  
(208, 'Isha Mathur', 'Sales', 14000, '2023-03-01'),  
(209, 'Manoj Patil', 'IT', 29000, '2022-10-18');
```

```
-- Query 1: Display all rows
```

```
SELECT * FROM employee;
```



Apex Institute of Technology

**Department of Computer Science &
Engineering**

```
-- Query 2: Average salary by department
SELECT department,AVG(salary) AS avg_salary
FROM employee
GROUP BY department;
```

```
-- Query 3: Average salary (salary > 20000)
SELECT department,AVG(salary) AS avg_salary
FROM employee
WHERE salary > 20000
GROUP BY department;
```

```
-- Query 4: Average salary > 30000 after filter
SELECT department,AVG(salary) AS avg_salary
FROM employee
WHERE salary > 20000
GROUP BY department
HAVING AVG(salary) > 30000;
```



Apex Institute of Technology

Department of Computer Science & Engineering

```
-- Query 5: Same as above + order desc
SELECT department,AVG(salary) AS avg_salary
FROM employee
WHERE salary > 20000
GROUP BY department
HAVING AVG(salary) > 30000
ORDER BY avg_salary DESC;
```