

Student Name: Dibyanshu kumar

UID:24BAI71026

Branch: CSE AIML

Section/Group:24AIT-KRG1/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.

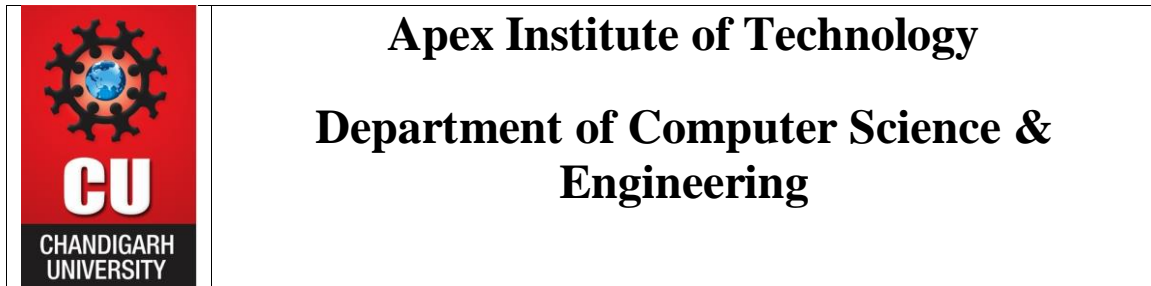
2. Objective of the Session

- 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.

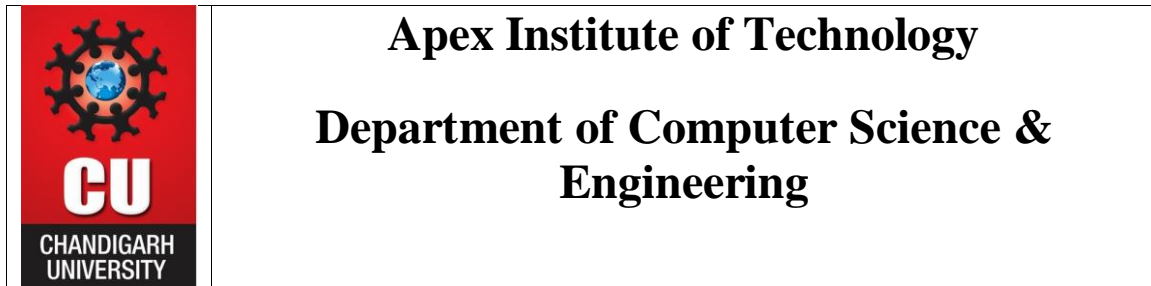
SQL Implementation (PgAdmin / PostgreSQL)

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
(101, 'Arjun Malhotra', 'Data Science', 60000, '2022-01-10'),
(102, 'Pooja Nair', 'Recruitment', 28000, '2021-03-15'),
(103, 'Siddharth Iyer', 'Backend Dev', 42000, '2020-06-20'),
(104, 'Meera Kulkarni', 'Accounts', 65000, '2019-09-05'),
(105, 'Rohan Das', 'HR Operations', 24000, '2023-02-12'),
(106, 'Kavya Menon', 'Auditing', 36000, '2020-11-25'),
(107, 'Nitin Agarwal', 'Business Dev', 48000, '2021-07-30'),
(108, 'Ishita Roy', 'Marketing', 26000, '2022-12-01'),
(109, 'Aditya Kulkarni', 'Cloud Engg', 52000, '2022-04-18');
```

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



Apex Institute of Technology

Department of Computer Science & Engineering

Query

Query History

```
1 CREATE TABLE employee (  
2     emp_id      INT PRIMARY KEY,  
3     emp_name    VARCHAR(50),  
4     department  VARCHAR(50),  
5     salary      NUMERIC(10,2),  
6     joining_date DATE  
7 );  
8  
9 INSERT INTO employee (emp_id, emp_name, department, salary, joining_date) VALUES  
10 (101, 'Arjun Malhotra', 'Data Science', 60000, '2022-01-10'),  
11 (102, 'Pooja Nair', 'Recruitment', 28000, '2021-03-15'),  
12 (103, 'Siddharth Iyer', 'Backend Dev', 42000, '2020-06-20'),  
13 (104, 'Meera Kulkarni', 'Accounts', 65000, '2019-09-05'),  
14 (105, 'Rohan Das', 'HR Operations', 24000, '2023-02-12'),  
15 (106, 'Kavya Menon', 'Auditing', 36000, '2020-11-25'),  
16 (107, 'Nitin Agarwal', 'Business Dev', 48000, '2021-07-30'),  
17 (108, 'Ishita Roy', 'Marketing', 26000, '2022-12-01'),  
18 (109, 'Aditya Kulkarni', 'Cloud Engg', 52000, '2022-04-18');  
19  
20 SELECT * FROM employee;  
21
```

Data Output

Messages

Notifications

SQL

	emp_id [PK] integer	emp_name character varying (50)	department character varying (50)	salary numeric (10,2)	joining_date date
1	101	Arjun Malhotra	Data Science	60000.00	2022-01-10
2	102	Pooja Nair	Recruitment	28000.00	2021-03-15
3	103	Siddharth Iyer	Backend Dev	42000.00	2020-06-20
4	104	Meera Kulkarni	Accounts	65000.00	2019-09-05
5	105	Rohan Das	HR Operations	24000.00	2023-02-12
6	106	Kavya Menon	Auditing	36000.00	2020-11-25
7	107	Nitin Agarwal	Business Dev	48000.00	2021-07-30
8	108	Ishita Roy	Marketing	26000.00	2022-12-01
9	109	Aditya Kulkarni	Cloud Engg	52000.00	2022-04-18



Apex Institute of Technology

Department of Computer Science & Engineering

```
21  
22 SELECT department,AVG(salary) AS avg_salary  
23 FROM employee  
24 GROUP BY department;  
25  
26
```

Data Output Messages Notifications



	department character varying (50)	avg_salary numeric
1	Cloud Engg	52000.000000000000
2	Accounts	65000.000000000000
3	Marketing	26000.000000000000
4	HR Operations	24000.000000000000
5	Recruitment	28000.000000000000
6	Auditing	36000.000000000000
7	Business Dev	48000.000000000000
8	Data Science	60000.000000000000
9	Backend Dev	42000.000000000000



Apex Institute of Technology

Department of Computer Science & Engineering

```
32 SELECT department,AVG(salary) AS avg_salary
33 FROM employee
34 WHERE salary > 20000
35 GROUP BY department
36 HAVING AVG(salary) > 30000;
37
```

Data Output Messages Notifications

	department character varying (50)	avg_salary numeric
1	Cloud Engg	52000.000000000000
2	Accounts	65000.000000000000
3	Business Dev	48000.000000000000
4	Data Science	60000.000000000000
5	Backend Dev	42000.000000000000
6	Auditing	36000.000000000000



Apex Institute of Technology

Department of Computer Science & Engineering

```
37  
38 SELECT department,AVG(salary) AS avg_salary  
39 FROM employee  
40 WHERE salary > 20000  
41 GROUP BY department  
42 HAVING AVG(salary) > 30000  
43 ORDER BY avg_salary DESC;  
44
```

Data Output Messages Notifications

	department character varying (50)	avg_salary numeric
1	Accounts	65000.000000000000
2	Data Science	60000.000000000000
3	Cloud Engg	52000.000000000000
4	Business Dev	48000.000000000000
5	Backend Dev	42000.000000000000
6	Auditing	36000.000000000000