Apex Institute of Technology

Department of Computer Science & Engineering

Student Name: Shreyansh Gupta **UID:**24BAI70658

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(30) NOT NULL,
```

```
    department VARCHAR(30) NOT NULL,

    salary INT NOT NULL,

    joining_date DATE

);
```

C) Insert Sample Records:

```
INSERT INTO EMPLOYEE (emp_id, emp_name, department, salary,
joining_date) VALUES

(1,'RAHUL','HR',25000,DATE '2022-01-10'),

(2,'ANITA','HR',32000,DATE '2021-05-12'),

(3,'AMIT','IT',45000,DATE '2020-03-15'),

(4,'NEHA','IT',38000,DATE '2022-07-20'),

(5,'ROHAN','SALES',18000,DATE '2023-02-05'),

(6,'PRIYA','SALES',28000,DATE '2021-11-18'),

(7,'KARAN','FINANCE',52000,DATE '2019-09-30'),

(8,'SIMRAN','FINANCE',48000,DATE '2020-12-25');
```

Step 1 Query:

```
SELECT department, COUNT(emp_id) AS TOTAL_EMPLOYEES

FROM EMPLOYEE

GROUP BY department;
```

Step 2 Query:

```
SELECT department, SUM(salary) AS TOTAL_SALARY

FROM EMPLOYEE

GROUP BY department;
```

Step 3 Query:

```
SELECT department, AVG(salary) AS AVG_SALARY

FROM EMPLOYEE

GROUP BY department;
```

Step 4 Query:

```
SELECT department, MIN(salary) AS MIN_SALARY

FROM EMPLOYEE

GROUP BY department;
```

Step 5 Query:

```
SELECT department, MAX(salary) AS MAX_SALARY

FROM EMPLOYEE

GROUP BY department;
```

Step 6 Query:

```
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

Query Query History

```
1
2  CREATE TABLE EMPLOYEE (
3      emp_id INT PRIMARY KEY,
4      emp_name VARCHAR(30) NOT NULL,
5      department VARCHAR(30) NOT NULL,
6      salary INT NOT NULL,
7      joining_date DATE
8  );
```

Data Output Messages Notifications

```
CREATE TABLE

Query returned successfully in 224 msec.
```

```
9   INSERT INTO EMPLOYEE (emp_id, emp_name, department, salary, joining_date) VALUES
10  (1,'RAHUL','HR',25000,DATE '2022-01-10'),
11  (2,'ANITA','HR',32000,DATE '2021-05-12'),
12  (3,'AMIT','IT',45000,DATE '2020-03-15'),
13  (4,'NEHA','IT',38000,DATE '2022-07-20'),
14  (5,'ROHAN','SALES',18000,DATE '2023-02-05'),
15  (6,'PRIYA','SALES',28000,DATE '2021-11-18'),
16  (7,'KARAN','FINANCE',52000,DATE '2019-09-30'),
17  (8,'SIMRAN','FINANCE',48000,DATE '2020-12-25');
```

Data Output Messages Notifications

```
INSERT 0 8

Query returned successfully in 149 msec.
```

Apex Institute of Technology

Department of Computer Science & Engineering

```
33  SELECT department, AVG(salary) AS AVG_SALARY
34  FROM EMPLOYEE
35  WHERE salary > 20000
36  GROUP BY department
37  HAVING AVG(salary) > 30000
38  ORDER BY AVG_SALARY DESC;
```

Data Output | Messages | Notifications

Showing rows: 1 to 2 | Page No: 1 of 1

	department character varying (30)	avg_salary numeric
1	FINANCE	50000.000000000000
2	IT	41500.000000000000

```
30  SELECT department, MAX(salary) AS MAX_SALARY
31  FROM EMPLOYEE
32  GROUP BY department;
```

Data Output | Messages | Notifications

Showing rows: 1 to 4 | Page No: 1 of 1

	department character varying (30)	max_salary integer
1	FINANCE	52000
2	SALES	28000
3	IT	45000
4	HR	32000

```
27  SELECT department, MIN(salary) AS MIN_SALARY
28  FROM EMPLOYEE
29  GROUP BY department;
```

Data Output | Messages | Notifications

Showing rows: 1 to 4 | Page No: 1 of 1

	department character varying (30)	min_salary integer
1	FINANCE	48000
2	SALES	18000
3	IT	38000
4	HR	25000

Apex Institute of Technology

Department of Computer Science & Engineering

```
24  SELECT department, AVG(salary) AS AVG_SALARY
25  FROM EMPLOYEE
26  GROUP BY department;
```

Data Output | Messages | Notifications

Showing rows: 1 to 4 | Page No: 1 of 1

	department character varying (30)	avg_salary numeric
1	FINANCE	50000.000000000000
2	SALES	23000.000000000000
3	IT	41500.000000000000
4	HR	28500.000000000000

```
21  SELECT department, SUM(salary) AS TOTAL_SALARY
22  FROM EMPLOYEE
23  GROUP BY department;
```

Data Output | Messages | Notifications

Showing rows: 1 to 4 | Page No: 1 of 1

	department character varying (30)	total_salary bigint
1	FINANCE	100000
2	SALES	46000
3	IT	83000
4	HR	57000

```
18  SELECT department, COUNT(emp_id) AS TOTAL_EMPLOYEES
19  FROM EMPLOYEE
20  GROUP BY department;
```

Data Output | Messages | Notifications

Showing rows: 1 to 4 | Page No: 1 of 1

	department character varying (30)	total_employees bigint
1	FINANCE	2
2	SALES	2
3	IT	2
4	HR	2