# Experiment 2

Student Name: Pushpit kumar gaur
UID: 24BAI70534
Branch: CSE - AIML
Section/Group: 24AIT_KRG G1
Semester: 4
Date of Performance: 16/1/26
Subject Name: DBMS
Subject Code: 24CSH-298

## 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.

## Software Requirements

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

## Objectives

- 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().

## Problem Statement

An organization maintains an EMPLOYEE table to store details of its employees. The structure of the table is as follows:

| Column Name | Data Type |
|---|---|

| | |
|---|---|
| emp_id | NUMBER |
| emp_name | VARCHAR |
| Department | VARCHAR |
| Salary | NUMBER |
| joining_date | DATE |

# Code

```
CREATE TABLE EMPLOYEE(
EMP_ID NUMERIC PRIMARY KEY,
EMP_NAME VARCHAR(20),
DEPARTMENT VARCHAR(20),
SALARY NUMERIC(10,2),
JOINING_DATE DATE
)

SELECT * FROM EMPLOYEE


INSERT INTO EMPLOYEE VALUES(1, 'Aman', 'IT', 30000, '2023-05-23');
INSERT INTO EMPLOYEE VALUES(2, 'Sam', 'IT', 25000, '2016-05-23');
INSERT INTO EMPLOYEE VALUES(3, 'Neha', 'HR', 18000, '2025-09-19');
INSERT INTO EMPLOYEE VALUES(4, 'Suman', 'Finance', 20000, '2021-11-06');
INSERT INTO EMPLOYEE VALUES(5, 'Rohan', 'Finance', 50000, '2023-10-23');
NSERT INTO EMPLOYEE VALUES(6, 'Sarika', 'HR', 28000, '2020-05-16');
INSERT INTO EMPLOYEE VALUES(7, 'Dardon', 'IT', 26000, '2021-07-07')

SELECT DEPARTMENT, AVG(SALARY)::NUMERIC(10,2) AS AVG_SAL FROM EMPLOYEE
GROUP BY DEPARTMENT

SELECT EMP_ID, EMP_NAME, SALARY
FROM EMPLOYEE
GROUP BY EMP_ID
HAVING SALARY>20000

SELECT DEPARTMENT, AVG(SALARY)::NUMERIC(10,2) AS AVG_SAL FROM EMPLOYEE
GROUP BY DEPARTMENT
HAVING AVG(SALARY)>30000

SELECT DEPARTMENT, AVG(SALARY)::NUMERIC(10,2) AS AVG_SAL FROM EMPLOYEE
GROUP BY DEPARTMENT
ORDER BY AVG(SALARY) DESC
```

# Output

Table created

Data Output | Messages | Notifications

```
CREATE TABLE

Query returned successfully in 109 msec.
```

Records inserted

Data Output | Messages | Notifications

```
INSERT 0 1

Query returned successfully in 110 msec.
```

Employees with salaries greater than 20,000

Data Output | Messages | Notifications

Showing rows: 1 to 5 | Page No: 1 of 1

| | emp_id [PK] numeric | emp_name character varying (20) | salary numeric (10,2) |
|---|---|---|---|
| 1 | 1 | Aman | 30000.00 |
| 2 | 6 | Aditi | 28000.00 |
| 3 | 7 | Aanya | 26000.00 |
| 4 | 2 | Sam | 25000.00 |
| 5 | 5 | Rohan | 24500.00 |

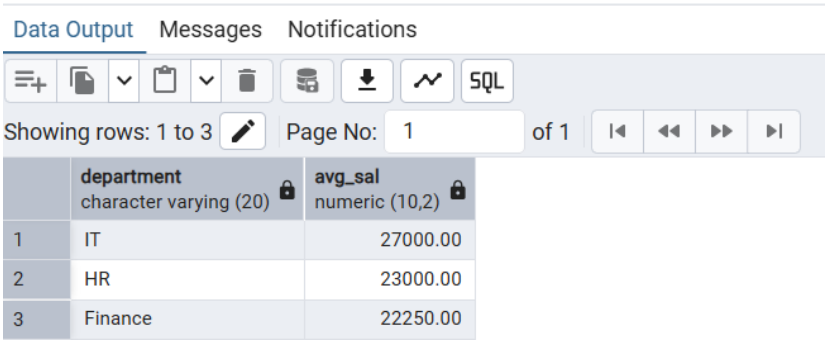
Average salaries of department

Data Output | Messages | Notifications

Showing rows: 1 to 3 | Page No: 1 of 1

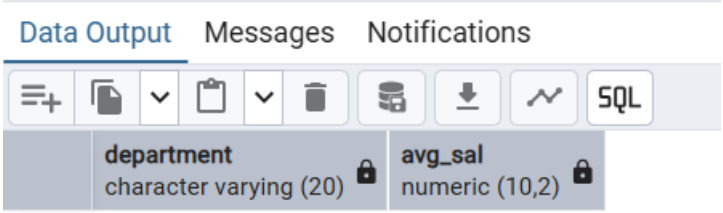
| | department character varying (20) | avg_sal numeric (10,2) |
|---|---|---|
| 1 | IT | 27000.00 |
| 2 | HR | 23000.00 |
| 3 | Finance | 22250.00 |

Sorting average salaries in descending order:




| | department character varying (20) | avg_sal numeric (10,2) |
|---|---|---|
| 1 | IT | 27000.00 |
| 2 | HR | 23000.00 |
| 3 | Finance | 22250.00 |

Departments with average salary more than 30,000 (empty because none)




## Learning Outcomes

- Learn to filter records using the WHERE clause.
- Group records using GROUP BY.
- Apply conditions on grouped data using HAVING.
- Sort query results using ORDER BY.