



Apex Institute of Technology

Computer Science & Engineering

Worksheet 2

Name: Krish Latawa

UID: 24BAI70541

Branch: B.E. CSE (AIML)

Section: 24AIT_KRG-G1

Semester: 4

Date of Performance: 16.01.2026

**Subject Name: Database
Management System**

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.

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()**.

SOFTWARE REQUIREMENTS:

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



Apex Institute of Technology

Computer Science & Engineering

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

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**.

CODE:

```
CREATE TABLE employee (  
    emp_id NUMERIC,  
    emp_name VARCHAR(50),  
    department VARCHAR(50),  
    salary NUMERIC,  
    joining_date DATE  
);
```

```
INSERT INTO employee VALUES  
(1, 'Amit', 'IT', 45000, '2022-01-10'),  
(2, 'Neha', 'HR', 28000, '2021-05-12'),
```



Apex Institute of Technology

Computer Science & Engineering

```
(3, 'Rohit', 'IT', 65000, '2020-03-18'),  
(4, 'Pooja', 'Finance', 35000, '2019-07-25'),  
(5, 'Karan', 'HR', 22000, '2023-02-14'),  
(6, 'Sneha', 'Finance', 18000, '2021-11-30');
```

```
SELECT department,  
       AVG(salary)::NUMERIC(10,2) AS avg_sal  
FROM employee  
GROUP BY department;
```

```
SELECT emp_id, emp_name, salary  
FROM employee  
WHERE 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;
```

I/O ANALYSIS:

- The EMPLOYEE table was created successfully.

emp_id	emp_name	department	salary	joining_date
[PK] numeric	character varying (20)	character varying (20)	numeric (10,2)	date

- Records inserted into EMPLOYEE table.



Apex Institute of Technology

Computer Science & Engineering

	emp_id [PK] numeric	emp_name character varying (20)	department character varying (20)	salary numeric (10,2)	joining_date date
1	1	Aman	IT	30000.00	2023-05-23
2	2	Sam	IT	25000.00	2016-05-23
3	3	Neha	HR	18000.00	2025-09-19
4	4	Suman	Finance	20000.00	2021-11-06
5	5	Rohan	Finance	50000.00	2023-10-23

- Department-wise average salary was calculated and displayed.

	department character varying (20)	avg_sal numeric (10,2)
1	Finance	35000.00
2	IT	27500.00
3	HR	18000.00

- Employees having salary greater than 20,000 were retrieved.

	emp_id [PK] numeric	emp_name character varying (20)	salary numeric (10,2)
1	1	Aman	30000.00
2	2	Sam	25000.00
3	5	Rohan	50000.00

- Departments with average salary above 30,000 were displayed.

	department character varying (20)	avg_sal numeric (10,2)
1	Finance	35000.00

- The final output was sorted in descending order of average salary.



Apex Institute of Technology

Computer Science & Engineering

	department character varying (20) 🔒	avg_sal numeric (10,2) 🔒
1	Finance	35000.00
2	IT	27500.00
3	HR	18000.00

LEARNING OUTCOMES:

1. Filter records using the **WHERE** clause.
2. Group records using **GROUP BY**.
3. Apply conditions on grouped data using **HAVING**.
4. Sort query results using **ORDER BY**..