

## Experiment 2

**Student Name: Aradhya Sharma**

**UID: 25MCC20042**

**Branch: MCA (CCD)**

**Section/Group: 25MCC-101A**

**Semester: 2**

**Date of Performance: 21 jan 2026**

**Subject Name: Technical training**

**Subject Code: 25CAP-652**

### **Implementation of SELECT Queries with Filtering, Grouping, and Sorting in PostgreSQL**

---

#### **1. AIM:**

To implement and analyze SQL SELECT queries using filtering, sorting, grouping, and aggregation concepts in PostgreSQL for efficient data retrieval and analytical reporting.

---

#### **2. OBJECTIVES:**

- To retrieve specific data using filtering conditions
- To sort query results using single and multiple attributes
- To perform aggregation using grouping techniques
- To apply conditions on aggregated data

### **3. SOFTWARE REQUIREMENTS:**

To perform this experiment, the following software is required:

#### **1. Operating System**

- Windows 10 / 11, Linux, or macOS

#### **2. Database Management System (DBMS)**

- PostgreSQL (version 12 or higher)

#### **3. SQL Client / Interface**

- pgAdmin 4 (for PostgreSQL)

**OR**

- Command Line Interface (psql)

### **4. PRACTICAL/ EXPERIMENT STEPS:**

- Create a sample table representing customer orders
- Insert realistic records into the table
- Retrieve filtered data using WHERE clause
- Sort query results using ORDER BY
- Group records and apply aggregate functions
- Apply conditions on grouped data using HAVING
- Analyze execution order WHERE and HAVING clauses

## 5. PROCEDURE OF EXPERIMENT:

### STEP 1: Database and Table Preparation

```
create table customer_orders(  
id serial primary key,  
customer_name varchar(20),  
product varchar(20),  
quantity int,  
price numeric(10,2),  
order_date date  
);
```

	id [PK] integer	customer_name character varying (20)	product character varying (20)	quantity integer	price numeric (10,2)	order_date date
1	1	Aradhya	Laptop	2	64000.00	2026-01-03
2	2	Rohan	Speaker	1	32000.00	2026-01-07
3	3	Raj	Headset	10	2000.00	2026-01-10
4	4	Hanna	Laptop	5	80000.00	2026-01-12
5	5	Harry	Keyboard	3	15000.00	2026-01-15
6	6	Mary	Mobile	1	28000.00	2026-01-15
7	7	Chandra	Charger	3	1500.00	2026-01-23

Created a realistic dataset with 10 records suitable for analytical queries.

---

## STEP 2: Filtering Data Using Conditions

```
select id, customer_name, product, quantity, price  
from customer_orders  
where price > 30000;
```

	id [PK] integer	customer_name character varying (20)	product character varying (20)	quantity integer	price numeric (10,2)
1	1	Aradhya	Laptop	2	64000.00
2	2	Rohan	Speaker	1	32000.00
3	4	Hanna	Laptop	5	80000.00

---

## STEP 3: Sorting Query Results

-- Sorting Query Results (Ascending Order)

```
select id, customer_name, product, quantity, price  
from customer_orders  
where price > 30000  
order by price;
```

	id [PK] integer	customer_name character varying (20)	product character varying (20)	quantity integer	price numeric (10,2)
1	2	Rohan	Speaker	1	32000.00
2	1	Aradhya	Laptop	2	64000.00
3	4	Hanna	Laptop	5	80000.00

-- Sorting Query Results (Decending Order)

```

select id, customer_name, product, quantity, price
from customer_orders
where price > 30000
order by price desc;

```

	id [PK] integer	customer_name character varying (20)	product character varying (20)	quantity integer	price numeric (10,2)
1	4	Hanna	Laptop	5	80000.00
2	1	Aradhya	Laptop	2	64000.00
3	2	Rohan	Speaker	1	32000.00

---

#### STEP 4: Grouping Data for Aggregation

```

select product, count(*) as total_sales
from customer_orders
group by product;

```

	product character varying (20)	total_sales bigint
1	Charger	1
2	Mobile	1
3	Speaker	1
4	Keyboard	1
5	Laptop	2
6	Headset	1

---

## STEP 5: Applying Conditions on Aggregated Data

```
select product, sum(quantity * price) as total_revenue  
from customer_orders  
group by product  
having sum(quantity * price) > 60000;
```

	product character varying (20) 🔒	total_revenue numeric 🔒
1	Laptop	528000.00

---

## STEP 6: Conceptual Understanding – Filtering VS Aggregation

```
select product, sum(quantity * price) as total_revenue  
from customer_orders  
where order_date >= '2026-01-01'  
group by product  
having sum(quantity * price) > 60000;
```

	product character varying (20) 🔒	total_revenue numeric 🔒
1	Laptop	528000.00

---

## 6. LEARNING OUTCOMES

- **Filtering Skills:** Learned to retrieve relevant records using the WHERE clause with various conditions.
- **Sorting Techniques:** Understood ordering results using ORDER BY with single and multiple columns.
- **Grouping & Aggregation:** Gained proficiency in summarizing data using GROUP BY with aggregate functions such as SUM, AVG, and COUNT.
- **Conditional Aggregation:** Clearly differentiated between WHERE (row-level filtering) and HAVING (group-level filtering).
- **Real-World Application:** Built confidence in writing analytical SQL queries used in dashboards and business reports.
- **Interview Readiness:** Prepared for SQL placement questions on filtering, grouping, aggregation, and analytical functions.