# EXPERIMENT 2

| | |
|---|---|
| **Student Name: Jatin Tasoria** | **UID: 25MCC20054** |
| **Branch: MCA-Cloud Computing** | **Section/Group: MCC 101-A** |
| **Semester: 2nd** | **Date of Performance: 21/01/26** |
| **Subject Name: Technical Training-I** | **Subject Code: 25CAP-652** |

## 1. Aim

The aim of this practical session is to implement SQL SELECT queries in PostgreSQL using filtering, sorting, grouping, and aggregation concepts to understand efficient data retrieval and analytical reporting.

## 2. Objective:

After completing this practical, the learner will be able to:

- Retrieve required data using filtering conditions
- Sort query results using ORDER BY
- Perform aggregation using GROUP BY
- Apply conditions on aggregated data using HAVING
- Understand real-world analytical SQL queries

## 3. Practical / Experiment Steps

a. Create a database and table in PostgreSQL

b. Insert sample records into the table

c. Apply filtering conditions using WHERE

d. Sort data using ORDER BY

e. Group records using GROUP BY

f. Apply conditions on grouped data using HAVING

## 4. Procedure of the Practical

1) Start the system and log in
2) Start the PostgreSQL service
3) Open PostgreSQL client (psql / pgAdmin)
4) Create a new database
5) Create the required table
6) Insert sample records
7) Execute SELECT queries for filtering, sorting, and grouping
8) Verify query execution without focusing on output
9) Save the work for documentation

## 5. SQL Queries Used in the Experiment

a. Database Creation

```
CREATE DATABASE order_management;
```

b. Table Creation

```
CREATE TABLE orders (
order_id SERIAL PRIMARY KEY,
customer_name VARCHAR(50),
product_name VARCHAR(50),
quantity INT,
price NUMERIC(10,2), order_date DATE);
```

c. Inserting Sample Records

```
INSERT INTO orders (customer_name, product_name, quantity, price,
order_date) VALUES
('Amit', 'Laptop', 1, 55000, '2024-01-10'),
('Neha', 'Mobile', 2, 30000, '2024-01-12'),
('Rahul', 'Laptop', 1, 60000, '2024-01-15'),
('Sneha', 'Tablet', 3, 15000, '2024-01-18'),
('Karan', 'Mobile', 1, 28000, '2024-01-20');
```

d. Filtering Data (WHERE Clause)

```
SELECT customer_name, product_name, price
FROM orders WHERE price > 30000;
```

e. Sorting Data (ORDER BY Clause)

```
SELECT customer_name, product_name, price
FROM orders
ORDER BY price DESC;
```

f. Sorting Using Multiple Columns

```
SELECT customer_name, product_name, price
FROM orders
ORDER BY product_name ASC, price DESC;
```

g. Grouping and Aggregation (GROUP BY)

```
SELECT product_name, SUM(price) AS total_sales
```

FROM orders

GROUP BY product_name;

h. Applying Condition on Aggregated Data (HAVING)

SELECT product_name, SUM(price) AS total_sales

FROM orders

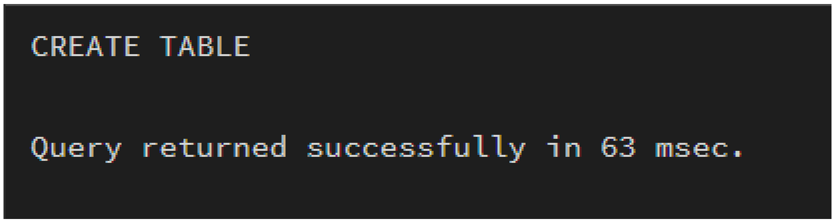
GROUP BY product_name

HAVING SUM(price) > 50000;

### 6. Input / Output Analysis (I/O Analysis)

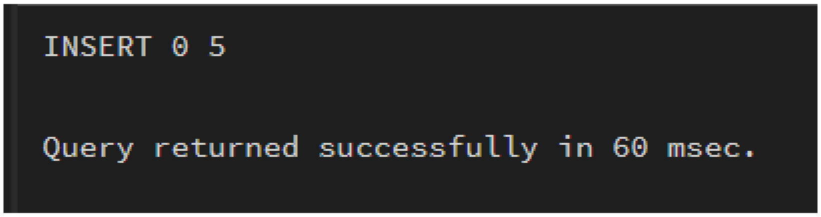
SQL queries for:

- Table creation



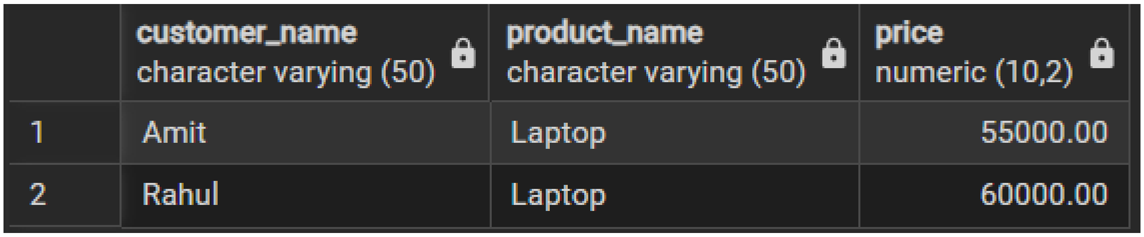
```
CREATE TABLE

Query returned successfully in 63 msec.
```

- Data insertion



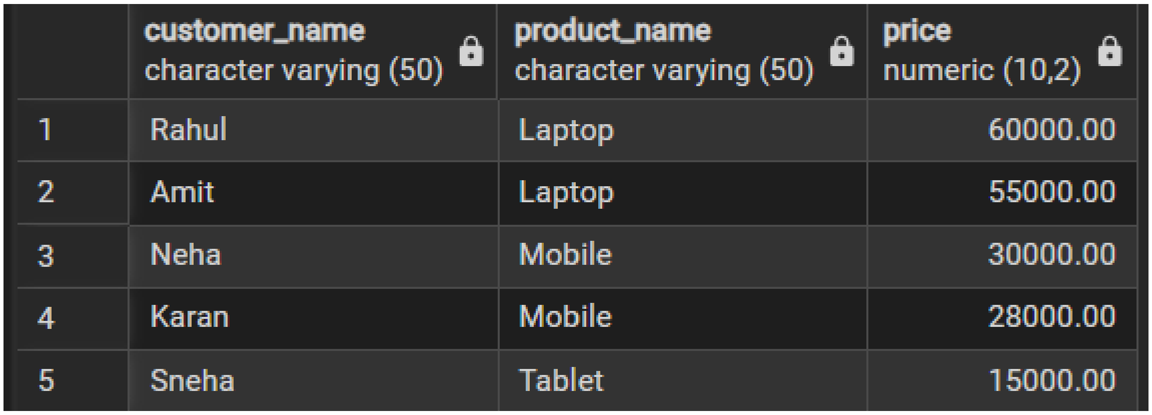
```
INSERT 0 5

Query returned successfully in 60 msec.
```

- Filtering



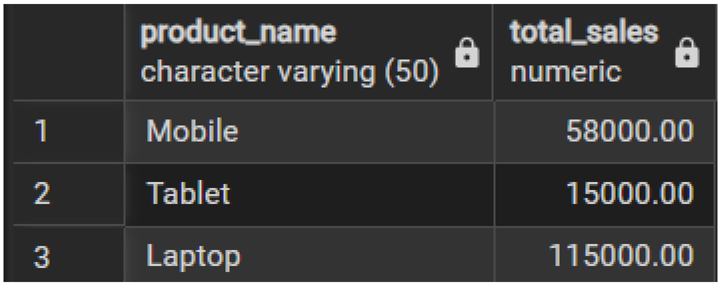
| | customer_name<br>character varying (50) | product_name<br>character varying (50) | price<br>numeric (10,2) |
|---|---|---|---|
| 1 | Amit | Laptop | 55000.00 |
| 2 | Rahul | Laptop | 60000.00 |

- Sorting



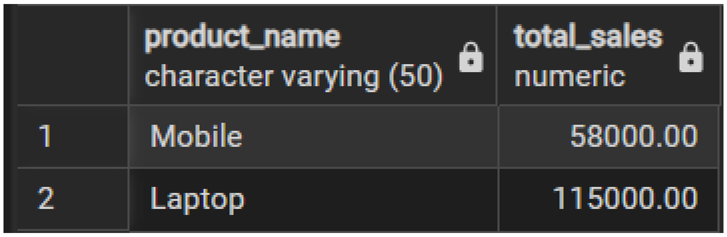
| | customer_name<br>character varying (50) | product_name<br>character varying (50) | price<br>numeric (10,2) |
|---|---|---|---|
| 1 | Rahul | Laptop | 60000.00 |
| 2 | Amit | Laptop | 55000.00 |
| 3 | Neha | Mobile | 30000.00 |
| 4 | Karan | Mobile | 28000.00 |
| 5 | Sneha | Tablet | 15000.00 |

- Grouping



| | product_name<br>character varying (50) | total_sales<br>numeric |
|---|---|---|
| 1 | Mobile | 58000.00 |
| 2 | Tablet | 15000.00 |
| 3 | Laptop | 115000.00 |

- Aggregation



| | product_name<br>character varying (50) | total_sales<br>numeric |
|---|---|---|
| 1 | Mobile | 58000.00 |
| 2 | Laptop | 115000.00 |

## 7. Learning Outcome

Concepts Understood:

- Filtering using WHERE
- Sorting using ORDER BY
- Grouping and aggregation using GROUP BY and HAVING

Skills Developed:

- Writing structured PostgreSQL queries
- Designing analytical SQL statements
- Understanding query execution flow

Practical Exposure Gained:

- Hands-on experience with PostgreSQL
- Exposure to interview-oriented SQL queries