



# DEPARTMENT OF COMPUTER

Discover. Learn. Empower.

## Experiment 2

**Student Name:**Riya Kashyap

**Branch:** MCA-UIC

**Semester:** 2<sup>nd</sup>

**Subject Name:** Technical Training

**UID:** 25MCA20033

**Section/Group:** 1

**Date of Performance:**13/1/2026

**Subject Code:** 25CAP-652

**1. Aim:**The aim of this project is to create and manage an orders database using SQL and to perform various database operations such as inserting data, retrieving records, sorting results, grouping data, and applying aggregate and conditional functions to analyze sales information effectively.

**2. Objective:**The objective of this project is to gain practical knowledge of SQL by creating a table, inserting records, and executing queries using WHERE, ORDER BY, GROUP BY, HAVING, aggregate functions, and CASE statements to generate meaningful summaries and insights from the stored data.

### **3. Practical/Experiment Steps:**

- Design an orders table with appropriate attributes
- Create the table using SQL commands
- Insert multiple records into the table
- Retrieve records based on conditions
- Sort the retrieved data in descending order
- Group data and calculate total sales
- Apply aggregate and conditional functions for analysis

### **4. Procedure of the Practical:**

```
CREATE TABLE orders (
    order_id    INT,
    customer_name VARCHAR(50),
    product      VARCHAR(50),
    quantity     INT,
    price        NUMERIC(8,2),
    order_date   DATE
);
```

```
INSERT INTO orders VALUES
```

```
(1, 'Amit', 'Laptop', 1, 55000, '2024-01-10'),
(2, 'Riya', 'Mobile', 2, 22000, '2024-01-12'),
(3, 'Rahul', 'Laptop', 1, 60000, '2024-01-15'),
(4, 'Sneha', 'Tablet', 3, 15000, '2024-01-18'),
(5, 'Ankit', 'Mobile', 1, 25000, '2024-01-20'),
(6, 'Pooja', 'Laptop', 2, 58000, '2024-01-22'),
(7, 'Karan', 'Tablet', 1, 18000, '2024-01-25');
```

```
SELECT customer_name, product, price
FROM orders
WHERE price > 20000;
```

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

```
SELECT product,
       SUM(quantity * price) AS total_sales,
       COUNT(*) AS total_orders
FROM orders
GROUP BY product;
```

```
SELECT product,
       SUM(quantity * price) AS total_sales
FROM orders
GROUP BY product
HAVING SUM(quantity * price) > 50000
ORDER BY total_sales DESC;
```

```
SELECT product,
       SUM(CASE
           WHEN price > 20000
           THEN quantity * price
           ELSE 0
           END) AS high_value_sales
FROM orders
GROUP BY product
ORDER BY high_value_sales DESC;
```

## 5. Output:



The screenshot shows a SQL database interface with a toolbar at the top containing various icons for file operations, a search bar, and a SQL button. Below the toolbar is a table with five rows of data. The table has four columns: customer\_name, product, and price, all of which are locked (indicated by a padlock icon). The fifth column is a primary key (indicated by a key icon).

	customer_name	product	price
1	Amit	Laptop	55000.00
2	Riya	Mobile	22000.00
3	Rahul	Laptop	60000.00
4	Ankit	Mobile	25000.00
5	Pooja	Laptop	58000.00

	customer_name character varying (50)	product character varying (50)	price numeric (8,2)
1	Rahul	Laptop	60000.00
2	Pooja	Laptop	58000.00
3	Amit	Laptop	55000.00
4	Ankit	Mobile	25000.00
5	Riya	Mobile	22000.00
6	Karan	Tablet	18000.00
7	Sneha	Tablet	15000.00

	product character varying (50)	total_sales numeric	total_orders bigint
1	Mobile	69000.00	2
2	Tablet	63000.00	2
3	Laptop	231000.00	3

	product character varying (50)	total_sales numeric
1	Laptop	231000.00
2	Mobile	69000.00
3	Tablet	63000.00

## 6. Learning Outcome:

- Understood the concept of relational database tables and data types.
- Learned to create tables and insert records using SQL commands.
- Gained knowledge of data retrieval using SELECT and WHERE clauses.
- Learned how to sort data using the ORDER BY clause.
- Understood the use of aggregate functions such as SUM() and COUNT().
- Learned to group data using the GROUP BY clause.
- Understood the difference between WHERE and HAVING clauses.
- Gained practical experience in using CASE statements for conditional data analysis.

