In [1]:
```
# Step 1: Import libraries
import sqlite3
import pandas as pd

# Step 2: Connect to SQLite DB (it will create a file)
conn = sqlite3.connect("sales_data.db")
cursor = conn.cursor()

# Step 3: Create sample tables
cursor.execute("""
CREATE TABLE IF NOT EXISTS Customers (
    customer_id INTEGER PRIMARY KEY,
    name TEXT,
    city TEXT
)
""")

cursor.execute("""
CREATE TABLE IF NOT EXISTS Products (
    product_id INTEGER PRIMARY KEY,
    product_name TEXT,
    price REAL
)
""")

cursor.execute("""
CREATE TABLE IF NOT EXISTS Orders (
    order_id INTEGER PRIMARY KEY,
    customer_id INTEGER,
    product_id INTEGER,
    quantity INTEGER,
    order_date TEXT,
    FOREIGN KEY(customer_id) REFERENCES Customers(customer_id),
    FOREIGN KEY(product_id) REFERENCES Products(product_id)
)
""")

# Step 4: Insert sample data
cursor.executemany("INSERT INTO Customers VALUES (?, ?, ?)", [
    (1, 'Alice', 'Delhi'),
    (2, 'Bob', 'Mumbai'),
    (3, 'Charlie', 'Delhi'),
    (4, 'David', 'Bangalore')
])

cursor.executemany("INSERT INTO Products VALUES (?, ?, ?)", [
    (1, 'Laptop', 80000),
    (2, 'Mouse', 500),
    (3, 'Keyboard', 1500),
    (4, 'Monitor', 12000)
])

cursor.executemany("INSERT INTO Orders VALUES (?, ?, ?, ?, ?)", [
    (1, 1, 1, 1, '2025-08-01'),
    (2, 2, 2, 3, '2025-08-02'),
    (3, 3, 4, 2, '2025-08-03'),
    (4, 1, 3, 1, '2025-08-04'),
    (5, 4, 1, 1, '2025-08-05')
```

```
])

conn.commit()
```

In [2]:
```
query1 = """
SELECT * FROM Customers
WHERE city = 'Delhi'
ORDER BY name ASC;
"""
pd.read_sql(query1, conn)
```

Out[2]:

| | customer_id | name | city |
|---|---|---|---|
| **0** | 1 | Alice | Delhi |
| **1** | 3 | Charlie | Delhi |

In [3]:
```
query2 = """
SELECT Orders.order_id, Customers.name, Products.product_name, Orders.quantity
FROM Orders
INNER JOIN Customers ON Orders.customer_id = Customers.customer_id
INNER JOIN Products ON Orders.product_id = Products.product_id;
"""
pd.read_sql(query2, conn)
```

Out[3]:

| | order_id | name | product_name | quantity |
|---|---|---|---|---|
| **0** | 1 | Alice | Laptop | 1 |
| **1** | 2 | Bob | Mouse | 3 |
| **2** | 3 | Charlie | Monitor | 2 |
| **3** | 4 | Alice | Keyboard | 1 |
| **4** | 5 | David | Laptop | 1 |

In [4]:
```
query3 = """
SELECT name FROM Customers
WHERE customer_id IN (
    SELECT customer_id FROM Orders WHERE quantity > 1
);
"""
pd.read_sql(query3, conn)
```

Out[4]:

| | name |
|---|---|
| **0** | Bob |
| **1** | Charlie |

In [5]:
```
query4 = """
SELECT Customers.city, SUM(Products.price * Orders.quantity) AS total_sales
FROM Orders
JOIN Customers ON Orders.customer_id = Customers.customer_id
JOIN Products ON Orders.product_id = Products.product_id
GROUP BY Customers.city;
```

```
"""
pd.read_sql(query4, conn)
```

Out[5]:

| | city | total_sales |
|---|---|---|
| 0 | Bangalore | 80000.0 |
| 1 | Delhi | 105500.0 |
| 2 | Mumbai | 1500.0 |

In [6]:

```
cursor.execute("""
CREATE VIEW IF NOT EXISTS sales_summary AS
SELECT Customers.name, Products.product_name, Orders.quantity, (Products.price *
FROM Orders
JOIN Customers ON Orders.customer_id = Customers.customer_id
JOIN Products ON Orders.product_id = Products.product_id;
""")

pd.read_sql("SELECT * FROM sales_summary", conn)
```

Out[6]:

| | name | product_name | quantity | total_price |
|---|---|---|---|---|
| 0 | Alice | Laptop | 1 | 80000.0 |
| 1 | Bob | Mouse | 3 | 1500.0 |
| 2 | Charlie | Monitor | 2 | 24000.0 |
| 3 | Alice | Keyboard | 1 | 1500.0 |
| 4 | David | Laptop | 1 | 80000.0 |

In [7]:

```
sql_queries = """
-- a. SELECT, WHERE, ORDER BY
SELECT * FROM Customers WHERE city = 'Delhi' ORDER BY name ASC;

-- b. INNER JOIN
SELECT Orders.order_id, Customers.name, Products.product_name, Orders.quantity
FROM Orders
INNER JOIN Customers ON Orders.customer_id = Customers.customer_id
INNER JOIN Products ON Orders.product_id = Products.product_id;

-- c. Subquery
SELECT name FROM Customers
WHERE customer_id IN (SELECT customer_id FROM Orders WHERE quantity > 1);

-- d. Aggregate Functions
SELECT Customers.city, SUM(Products.price * Orders.quantity) AS total_sales
FROM Orders
JOIN Customers ON Orders.customer_id = Customers.customer_id
JOIN Products ON Orders.product_id = Products.product_id
GROUP BY Customers.city;

-- e. Create View
CREATE VIEW sales_summary AS
SELECT Customers.name, Products.product_name, Orders.quantity, (Products.price *
FROM Orders
JOIN Customers ON Orders.customer_id = Customers.customer_id
```

```
JOIN Products ON Orders.product_id = Products.product_id;

-- f. Indexes
CREATE INDEX idx_city ON Customers(city);
CREATE INDEX idx_order_date ON Orders(order_date);
"""

with open("sql_queries.sql", "w") as f:
    f.write(sql_queries)
```

In [ ]:

In [ ]: