Task-5

Objective:

Learn how to **merge data** from two or more tables using:

- **INNER JOIN**
- **LEFT JOIN**
- **RIGHT JOIN** (Note: Not supported in SQLite)
- **FULL JOIN** (Not directly supported in SQLite – but can be simulated)

Tools:

- **DB Browser for SQLite** (for local SQLite database work)
- **MySQL Workbench** (for working with MySQL databases)

Deliverables:

SQL scripts that:

1. Create two related tables (e.g., `Customers`, `Orders`)
2. Insert sample data into them
3. Perform queries using all JOIN types

Hints / Mini Guide:

1. Create Tables

```
CREATE TABLE Customers (
    CustomerID INTEGER PRIMARY KEY,
    Name TEXT
);

CREATE TABLE Orders (
    OrderID INTEGER PRIMARY KEY,
    CustomerID INTEGER,
    Product TEXT,
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
);
```

2. Insert Sample Data

```
INSERT INTO Customers (CustomerID, Name) VALUES
(1, 'Alice'), (2, 'Bob'), (3, 'Charlie');

INSERT INTO Orders (OrderID, CustomerID, Product) VALUES
(101, 1, 'Laptop'), (102, 1, 'Mouse'), (103, 2, 'Keyboard');
```

3. Perform JOIN Queries

INNER JOIN – shows only matches in both tables:

```
SELECT Customers.Name, Orders.Product
FROM Customers
INNER JOIN Orders ON Customers.CustomerID = Orders.CustomerID;
```

LEFT JOIN – all customers, even if they have no orders:

```
SELECT Customers.Name, Orders.Product
FROM Customers
LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID;
```

RIGHT JOIN – all orders, even if customer is missing (MySQL only):

```
SELECT Customers.Name, Orders.Product
FROM Customers
RIGHT JOIN Orders ON Customers.CustomerID = Orders.CustomerID;
```

FULL OUTER JOIN – all records from both tables (MySQL):

```
SELECT Customers.Name, Orders.Product
FROM Customers
FULL OUTER JOIN Orders ON Customers.CustomerID = Orders.CustomerID;
```

☑ *SQLite doesn't support RIGHT JOIN or FULL OUTER JOIN directly, but we can simulate them using `UNION`.*

Outcome:

By practicing these, We'll:

- Understand how to retrieve and merge related data.
- Learn how each join type affects the result set.
- Be able to choose the appropriate join for your real-world SQL queries.

Here's a complete `.sql` script that includes:

1. Creating the `Customers` and `Orders` tables
2. Inserting sample data
3. Demonstrating `INNER JOIN`, `LEFT JOIN`, and simulated `FULL JOIN` (since SQLite doesn't support RIGHT/FULL OUTER JOIN natively)

SQL Script:

```
-- 1. Drop existing tables if they exist
DROP TABLE IF EXISTS Orders;
DROP TABLE IF EXISTS Customers;

-- 2. Create Customers table
CREATE TABLE Customers (
    CustomerID INTEGER PRIMARY KEY,
    Name TEXT
);

-- 3. Create Orders table
CREATE TABLE Orders (
    OrderID INTEGER PRIMARY KEY,
    CustomerID INTEGER,
    Product TEXT,
    FOREIGN KEY (CustomerID) REFERENCES CustomersCustomerID)
);

-- 4. Insert sample data into Customers
INSERT INTO Customers (CustomerID, Name) VALUES
(1, 'Alice'),
(2, 'Bob'),
(3, 'Charlie'),
(4, 'Diana');

-- 5. Insert sample data into Orders
INSERT INTO Orders (OrderID, CustomerID, Product) VALUES
(101, 1, 'Laptop'),
(102, 1, 'Mouse'),
(103, 2, 'Keyboard'),
(104, 5, 'Monitor'); -- This order has no matching customer (simulates
orphaned data)

-- 6. INNER JOIN
SELECT 'INNER JOIN' AS JoinType, c.Name, o.Product
FROM Customers c
INNER JOIN Orders o ON c.CustomerID = o.CustomerID;

-- 7. LEFT JOIN
SELECT 'LEFT JOIN' AS JoinType, c.Name, o.Product
FROM Customers c
LEFT JOIN Orders o ON c.CustomerID = o.CustomerID;

-- 8. Simulated RIGHT JOIN (using LEFT JOIN with tables reversed)
SELECT 'SIMULATED RIGHT JOIN' AS JoinType, c.Name, o.Product
FROM Orders o
LEFT JOIN Customers c ON o.CustomerID = c.CustomerID;
```

```
-- 9. Simulated FULL OUTER JOIN (UNION of LEFT and RIGHT)
SELECT 'FULL OUTER JOIN' AS JoinType, c.Name, o.Product
FROM Customers c
LEFT JOIN Orders o ON c.CustomerID = o.CustomerID

UNION

SELECT 'FULL OUTER JOIN' AS JoinType, c.Name, o.Product
FROM Orders o
LEFT JOIN Customers c ON o.CustomerID = c.CustomerID
WHERE c.CustomerID IS NULL;
```