

1. Creating Customers table

```
CREATE TABLE Customers (customer_id INT PRIMARY KEY, name VARCHAR(30), phone VARCHAR(10), city VARCHAR(20));
```

2. Creating Orders table

```
CREATE TABLE Orders (order_id INT PRIMARY KEY, customer_id INT, product VARCHAR(20), price FLOAT, FOREIGN KEY(customer_id) REFERENCES Customers(customer_id));
```

3. Inserting data into Customers

```
INSERT INTO Customers VALUES (1, 'Ravi Sharma','9745834678', 'Mumbai');
```

```
INSERT INTO Customers VALUES (2, 'Priya Verma','8345587878', 'Delhi');
```

```
INSERT INTO Customers VALUES (3, 'Amit Kumar','9793258778', 'Pune');
```

```
INSERT INTO Customers VALUES (4, 'Neha Singh','9432885346', 'Chennai');
```

4. Inserting data into Orders

```
INSERT INTO Orders VALUES (101, 1, 'Laptop', 55000.00);
```

```
INSERT INTO Orders VALUES (102, 1, 'Keyboard', 1500.00);
```

```
INSERT INTO Orders VALUES (103, 2, 'Smartphone', 18000.00);
```

```
INSERT INTO Orders VALUES (104, 3, 'Tablet', 12000.00);
```

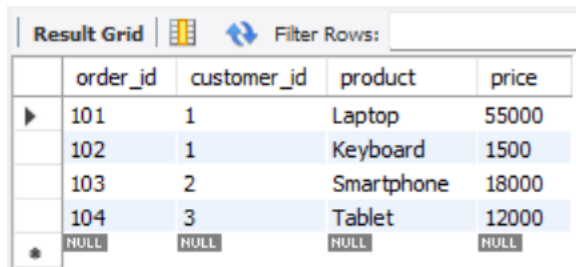
5. Displaying data from Customers table

```
SELECT customer_id, name, phone, city FROM Customers;
```

| Result Grid | | | | |
|--|-------------|-------------|------------|---------|
| Filter Rows: <input type="text"/> Edit | | | | |
| | customer_id | name | phone | city |
| ▶ | 1 | Ravi Sharma | 9745834678 | Mumbai |
| | 2 | Priya Verma | 8345587878 | Delhi |
| | 3 | Amit Kumar | 9793258778 | Pune |
| | 4 | Neha Singh | 9432885346 | Chennai |
| ✱ | NULL | NULL | NULL | NULL |

6. Displaying data from Orders table

SELECT order_id, customer_id, product, price FROM Orders;



The screenshot shows a database interface with a 'Result Grid' tab and a 'Filter Rows' search bar. The grid displays four rows of data from the Orders table. The first row is selected. The last row shows NULL values for all columns.

| | order_id | customer_id | product | price |
|---|----------|-------------|------------|-------|
| ▶ | 101 | 1 | Laptop | 55000 |
| | 102 | 1 | Keyboard | 1500 |
| | 103 | 2 | Smartphone | 18000 |
| | 104 | 3 | Tablet | 12000 |
| ✱ | NULL | NULL | NULL | NULL |

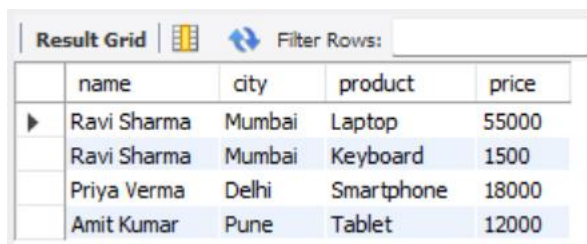
7. INNER JOIN

SELECT C.name, C.city, O.product, O.price

FROM Customers C

INNER JOIN Orders O

ON C.customer_id = O.customer_id;



The screenshot shows a database interface with a 'Result Grid' tab and a 'Filter Rows' search bar. The grid displays four rows of data resulting from an inner join between the Customers and Orders tables. Each row shows a customer's name, city, and the product and price of their order.

| | name | city | product | price |
|---|-------------|--------|------------|-------|
| ▶ | Ravi Sharma | Mumbai | Laptop | 55000 |
| | Ravi Sharma | Mumbai | Keyboard | 1500 |
| | Priya Verma | Delhi | Smartphone | 18000 |
| | Amit Kumar | Pune | Tablet | 12000 |

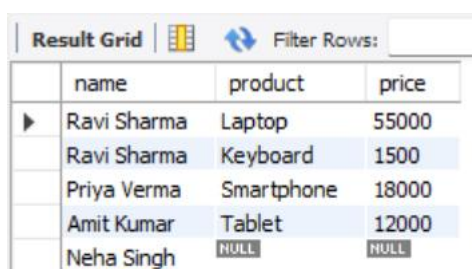
8. LEFT JOIN

SELECT C.name, O.product, O.price

FROM Customers C

LEFT JOIN Orders O

ON C.customer_id = O.customer_id;



The screenshot shows a database interface with a 'Result Grid' tab and a 'Filter Rows' search bar. The grid displays five rows of data resulting from a left join between the Customers and Orders tables. The first four rows are the same as in the inner join, but the fifth row shows a customer (Neha Singh) who has no orders, with NULL values for the product and price columns.

| | name | product | price |
|---|-------------|------------|-------|
| ▶ | Ravi Sharma | Laptop | 55000 |
| | Ravi Sharma | Keyboard | 1500 |
| | Priya Verma | Smartphone | 18000 |
| | Amit Kumar | Tablet | 12000 |
| | Neha Singh | NULL | NULL |

9. RIGHT JOIN

```
SELECT C.name, O.product, O.price
FROM Customers C
RIGHT JOIN Orders O
ON C.customer_id = O.customer_id;
```

| | name | product | price |
|---|-------------|------------|-------|
| ▶ | Ravi Sharma | Laptop | 55000 |
| | Ravi Sharma | Keyboard | 1500 |
| | Priya Verma | Smartphone | 18000 |
| | Amit Kumar | Tablet | 12000 |

10. FULL JOIN

```
SELECT C.name, C.city, C.phone, O.product, O.price
FROM Customers C
LEFT JOIN Orders O
ON C.customer_id = O.customer_id
UNION
SELECT C.name, C.city, C.phone, O.product, O.price
FROM Customers C
RIGHT JOIN Orders O
ON C.customer_id = O.customer_id;
```

| | name | city | phone | product | price |
|---|-------------|---------|------------|------------|-------|
| ▶ | Ravi Sharma | Mumbai | 9745834678 | Keyboard | 1500 |
| | Ravi Sharma | Mumbai | 9745834678 | Laptop | 55000 |
| | Priya Verma | Delhi | 8345587878 | Smartphone | 18000 |
| | Amit Kumar | Pune | 9793258778 | Tablet | 12000 |
| | Neha Singh | Chennai | 9432885346 | NULL | NULL |