Joins practice

```
CREATE TABLE Customers (
  c_id INT PRIMARY KEY AUTO_INCREMENT,
  c_name VARCHAR(50),
  c_city VARCHAR(30),
  c_email VARCHAR(100)
);
INSERT INTO Customers (c_name, c_city, c_email) VALUES
('Vamsi', 'Hyderabad', 'vamsi@shopease.com'),
('Priya', 'Chennai', 'priya@shopease.com'),
('Rahul', 'Delhi', 'rahul@shopease.com'),
('Sneha', 'Mumbai', 'sneha@shopease.com'),
('Arjun', 'Pune', 'arjun@shopease.com'),
('Kiran', 'Hyderabad', 'kiran@shopease.com'),
('Divya', 'Bangalore', 'divya@shopease.com'),
('Sai', 'Vizag', 'sai@shopease.com'),
('Meena', 'Chennai', 'meena@shopease.com'),
('Anil', 'Delhi', 'anil@shopease.com');

CREATE TABLE Orders (
  o_id INT PRIMARY KEY AUTO_INCREMENT,
  c_id INT,
  o_product VARCHAR(100),
  o_amount DECIMAL(10,2),
  o_date DATE,
  FOREIGN KEY (c_id) REFERENCES Customers(c_id)
);
```

INSERT INTO Orders (c_id, o_product, o_amount, o_date) VALUES

(1, 'Mobile', 15000, '2025-10-10'),

(1, 'Headphones', NULL, '2025-10-15'),

(2, 'Laptop', 55000, '2025-09-05'),

(3, 'Smart Watch', 7000, '2025-08-20'),

(3, 'Keyboard', NULL, '2025-08-22'),

(4, 'Television', 42000, '2025-07-18'),

(6, 'Tablet', 18000, '2025-10-02'),

(6, 'Mouse', NULL, '2025-10-10'),

(8, 'Fridge', 30000, '2025-09-09'),

(9, 'AC', 40000, '2025-08-01'),

(9, NULL, 2000, '2025-08-05');

1. Get all customers with their ordered products

SELECT c.c_name, o.o_product, o.o_amount

FROM Customers c

INNER JOIN Orders o ON c.c_id = o.c_id;

| c_name | o_product | o_amount |
|---|---|---|
| Vamsi | Mobile | 15000.00 |
| Vamsi | Headphones | NULL |
| Priya | Laptop | 55000.00 |
| Rahul | Smart Watch | 7000.00 |
| Rahul | Keyboard | NULL |
| Sneha | Television | 42000.00 |
| Kiran | Tablet | 18000.00 |
| Kiran | Mouse | NULL |
| Sai | Fridge | 30000.00 |
| Meena | AC | 40000.00 |
| Meena | NULL | 2000.00 |

2. Find each customer’s city and total purchase amount

SELECT

c.c_name AS CustomerName,

```
    c.c_city AS City,
    SUM(o.o_amount) AS Total_Purchase_Amount
FROM
    Customers c
INNER JOIN
    Orders o
ON
    c.c_id = o.c_id
GROUP BY
    c.c_name, c.c_city;
```

| CustomerName | City | Total_Purchase_Amount |
|---|---|---|
| Vamsi | Hyderabad | 15000.00 |
| Priya | Chennai | 55000.00 |
| Rahul | Delhi | 7000.00 |
| Sneha | Mumbai | 42000.00 |
| Kiran | Hyderabad | 18000.00 |
| Sai | Vizag | 30000.00 |
| Meena | Chennai | 42000.00 |

3. Show all customers who purchased items costing above ₹20,000

```
SELECT c.c_name, o.o_product, o.o_amount
FROM Customers c
INNER JOIN Orders o ON c.c_id = o.c_id
WHERE o.o_amount > 20000;
```

| c_name | o_product | o_amount |
|---|---|---|
| Priya | Laptop | 55000.00 |
| Sneha | Television | 42000.00 |
| Sai | Fridge | 30000.00 |
| Meena | AC | 40000.00 |

4. Get all customer names with their order dates and product names

```
SELECT c.c_name, o.o_product, o.o_date
FROM Customers c
INNER JOIN Orders o ON c.c_id = o.c_id
ORDER BY o.o_date DESC;
```

| c_name | o_product | o_date |
|---|---|---|
| Vamsi | Headphones | 2025-10-15 |
| Vamsi | Mobile | 2025-10-10 |
| Kiran | Mouse | 2025-10-10 |
| Kiran | Tablet | 2025-10-02 |
| Sai | Fridge | 2025-09-09 |
| Priya | Laptop | 2025-09-05 |
| Rahul | Keyboard | 2025-08-22 |
| Rahul | Smart Watch | 2025-08-20 |
| Meena | NULL | 2025-08-05 |
| Meena | AC | 2025-08-01 |
| Sneha | Television | 2025-07-18 |

5. Find total number of orders placed by each customer

```
SELECT c.c_name, COUNT(o.o_id) AS total_orders
FROM Customers c
INNER JOIN Orders o ON c.c_id = o.c_id
GROUP BY c.c_name;
```

| c_name | total_orders |
|---|---|
| Vamsi | 2 |
| Priya | 1 |
| Rahul | 2 |
| Sneha | 1 |
| Kiran | 2 |
| Sai | 1 |
| Meena | 2 |

```
-- 1. Show all customers and their order details (include customers with no orders)
SELECT c.c_name, o.o_product, o.o_amount
```

FROM Customers c

LEFT JOIN Orders o ON c.c_id = o.c_id;

| c_name | o_product | o_amount |
|---|---|---|
| Vamsi | Mobile | 15000.00 |
| Vamsi | Headphones | NULL |
| Priya | Laptop | 55000.00 |
| Rahul | Smart Watch | 7000.00 |
| Rahul | Keyboard | NULL |
| Sneha | Television | 42000.00 |
| Arjun | NULL | NULL |
| Kiran | Tablet | 18000.00 |
| Kiran | Mouse | NULL |
| Divya | NULL | NULL |
| Sai | Fridge | 30000.00 |
| Meena | AC | 40000.00 |
| Meena | NULL | 2000.00 |

-- 2. Find customers who have NOT placed any order

SELECT c.c_name, c.c_city

FROM Customers c

LEFT JOIN Orders o ON c.c_id = o.c_id

WHERE o.o_id IS NULL;

| c_name | c_city |
|---|---|
| Arjun | Pune |
| Divya | Bangalore |
| Anil | Delhi |

-- 3. Show all customers and total amount they have spent (include zero if no order)

SELECT c.c_name, COALESCE(SUM(o.o_amount), 0) AS total_spent

```
FROM Customers c
LEFT JOIN Orders o ON c.c_id = o.c_id
GROUP BY c.c_name;
```

| c_name | total_spent |
|---|---|
| Vamsi | 15000.00 |
| Priya | 55000.00 |
| Rahul | 7000.00 |
| Sneha | 42000.00 |
| Arjun | 0.00 |
| Kiran | 18000.00 |
| Divya | 0.00 |
| Sai | 30000.00 |
| Meena | 42000.00 |
| Anil | 0.00 |

```
-- 4. Display customers and latest order date (if any)
SELECT c.c_name, MAX(o.o_date) AS latest_order
FROM Customers c
LEFT JOIN Orders o ON c.c_id = o.c_id
GROUP BY c.c_name;
```

| c_name | latest_order |
|---|---|
| Vamsi | 2025-10-15 |
| Priya | 2025-09-05 |
| Rahul | 2025-08-22 |
| Sneha | 2025-07-18 |
| Arjun | NULL |
| Kiran | 2025-10-10 |
| Divya | NULL |
| Sai | 2025-09-09 |
| Meena | 2025-08-05 |
| Anil | NULL |

-- 5. Find customers who ordered NULL-priced products

```
SELECT c.c_name, o.o_product, o.o_amount
FROM Customers c
LEFT JOIN Orders o ON c.c_id = o.c_id
WHERE o.o_amount IS NULL;
```

| c_name | o_product | o_amount |
|---|---|---|
| Vamsi | Headphones | NULL |
| Rahul | Keyboard | NULL |
| Arjun | NULL | NULL |
| Kiran | Mouse | NULL |
| Divya | NULL | NULL |
| Anil | NULL | NULL |

---

### 🧩 3️⃣ RIGHT JOIN (All Orders, Even Without Matching Customers)

**Business goal:** See all orders, including those without valid customers.

-- 1. Show all orders and their corresponding customer names

```
SELECT c.c_name, o.o_product, o.o_amount
FROM Customers c
RIGHT JOIN Orders o ON c.c_id = o.c_id;
```

| | c_name | o_product | o_amount |
|---|---|---|---|
| ▶ | Vamsi | Mobile | 15000.00 |
| | Vamsi | Headphones | NULL |
| | Priya | Laptop | 55000.00 |
| | Rahul | Smart Watch | 7000.00 |
| | Rahul | Keyboard | NULL |
| | Sneha | Television | 42000.00 |
| | Kiran | Tablet | 18000.00 |
| | Kiran | Mouse | NULL |
| | Sai | Fridge | 30000.00 |
| | Meena | AC | 40000.00 |
| | Meena | NULL | 2000.00 |

-- 2. Find all orders that belong to non-existing customers

```
SELECT o.o_id, o.o_product, o.o_amount
FROM Customers c
RIGHT JOIN Orders o ON c.c_id = o.c_id
WHERE c.c_id IS NULL;
```

| | o_id | o_product | o_amount |
|---|---|---|---|

-- 3. Display all orders with their customer's city (if available)

```
SELECT o.o_product, o.o_amount, c.c_city
FROM Customers c
RIGHT JOIN Orders o ON c.c_id = o.c_id;
```

| | o_product | o_amount | c_city |
|---|---|---|---|
| ▶ | Mobile | 15000.00 | Hyderabad |
| | Headphones | NULL | Hyderabad |
| | Laptop | 55000.00 | Chennai |
| | Smart Watch | 7000.00 | Delhi |
| | Keyboard | NULL | Delhi |
| | Television | 42000.00 | Mumbai |
| | Tablet | 18000.00 | Hyderabad |
| | Mouse | NULL | Hyderabad |
| | Fridge | 30000.00 | Vizag |
| | AC | 40000.00 | Chennai |
| | NULL | 2000.00 | Chennai |

-- 4. List all orders placed after August 2025, even if customer data is missing

```
SELECT o.o_id, o.o_product, o.o_amount, c.c_name
FROM Customers c
RIGHT JOIN Orders o ON c.c_id = o.c_id
WHERE o.o_date > '2025-08-01';
```

| | o_id | o_product | o_amount | c_name |
|---|---|---|---|---|
| ▶ | 13 | Mobile | 15000.00 | Vamsi |
| | 14 | Headphones | NULL | Vamsi |
| | 15 | Laptop | 55000.00 | Priya |
| | 16 | Smart Watch | 7000.00 | Rahul |
| | 17 | Keyboard | NULL | Rahul |
| | 19 | Tablet | 18000.00 | Kiran |
| | 20 | Mouse | NULL | Kiran |
| | 21 | Fridge | 30000.00 | Sai |
| | 23 | NULL | 2000.00 | Meena |

## 🧩 4️⃣ SELF JOIN (Within the Same Table)

-- 1. Find all pairs of customers living in the same city

```
SELECT a.c_name AS customer1, b.c_name AS customer2, a.c_city
FROM Customers a
JOIN Customers b ON a.c_city = b.c_city
WHERE a.c_id <> b.c_id;
```

| customer1 | customer2 | c_city |
|---|---|---|
| Kiran | Vamsi | Hyderabad |
| Meena | Priya | Chennai |
| Anil | Rahul | Delhi |
| Vamsi | Kiran | Hyderabad |
| Priya | Meena | Chennai |

-- 2. Find customers who share a city with 'Vamsi'

```
SELECT a.c_name, a.c_city
FROM Customers a
JOIN Customers b ON a.c_city = b.c_city
WHERE b.c_name = 'Vamsi' AND a.c_name <> 'Vamsi';
```

| c_name | c_city |
|---|---|
| Kiran | Hyderabad |

1️⃣

Find all pairs of customers who live in the same city.

```
SELECT
  c1.c_name AS Customer1,
  c2.c_name AS Customer2,
  c1.c_city
FROM Customers c1
JOIN Customers c2
  ON c1.c_city = c2.c_city
 AND c1.c_id < c2.c_id;
```

| Customer1 | Customer2 | c_city |
|---|---|---|
| Vamsi | Kiran | Hyderabad |
| Priya | Meena | Chennai |
| Rahul | Anil | Delhi |

---

2

Find customers who share the same city with **‘Vamsi’**.

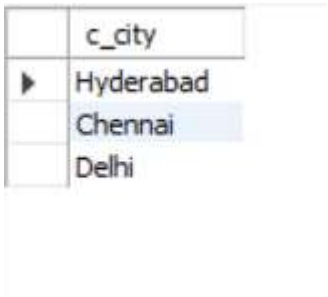
```
SELECT c2.c_name, c2.c_city
FROM Customers c1
JOIN Customers c2
  ON c1.c_city = c2.c_city
 AND c1.c_id <> c2.c_id
WHERE c1.c_name = 'Vamsi';
```

| c_name | c_city |
|---|---|
| Kiran | Hyderabad |

4️⃣

List cities that have more than one customer.

```
SELECT DISTINCT c1.c_city
FROM Customers c1
JOIN Customers c2
 ON c1.c_city = c2.c_city
AND c1.c_id <> c2.c_id;
```



| c_city |
| --- |
| Hyderabad |
| Chennai |
| Delhi |

---

5️⃣

Show the total number of customers per city (only for cities with more than one person).

```
SELECT
 c1.c_city,
 COUNT(DISTINCT c1.c_id) AS total_customers
FROM Customers c1
JOIN Customers c2
 ON c1.c_city = c2.c_city
AND c1.c_id <> c2.c_id
GROUP BY c1.c_city;
```

| | c_city | total_customers |
|---|---|---|
| ▶ | Chennai | 2 |
| | Delhi | 2 |
| | Hyderabad | 2 |

---

6

Display customer pairs **only from Hyderabad**.

```
SELECT
  c1.c_name AS Customer1,
  c2.c_name AS Customer2,
  c1.c_city
FROM Customers c1
JOIN Customers c2
  ON c1.c_city = c2.c_city
 AND c1.c_id < c2.c_id
WHERE c1.c_city = 'Hyderabad';
```

| | Customer1 | Customer2 | c_city |
|---|---|---|---|
| ▶ | Vamsi | Kiran | Hyderabad |

---

7

Find customers who don’t share their city with anyone else.

```
SELECT c1.c_name, c1.c_city
FROM Customers c1
LEFT JOIN Customers c2
  ON c1.c_city = c2.c_city
 AND c1.c_id <> c2.c_id
```

WHERE c2.c_id IS NULL;

| c_name | c_city |
|---|---|
| Sneha | Mumbai |
| Arjun | Pune |
| Divya | Bangalore |
| Sai | Vizag |

Find customers whose names start with the same letter.

```
SELECT
 c1.c_name AS Customer1,
 c2.c_name AS Customer2,
 LEFT(c1.c_name, 1) AS Starting_Letter
FROM Customers c1
JOIN Customers c2
 ON LEFT(c1.c_name, 1) = LEFT(c2.c_name, 1)
 AND c1.c_id < c2.c_id;
```

| Customer1 | Customer2 | Starting_Letter |
|---|---|---|
| Sneha | Sai | S |
| Arjun | Anil | A |