**Business Context**

The company wants to analyze **customer purchases, shipments, and non-purchasers**. Using SQL **joins**, you’ll answer practical questions to support analytics and reporting.

**📝 Lab Questions (Practice)**

**Basic INNER JOIN**

1. Write a query to display first\_name, last\_name, and item purchased by customers.
2. Fetch all customers along with their orders **only if they exist in both tables**.

**LEFT & RIGHT JOINS**

1. Write a query to show all customers and their orders, including those who have **not placed any order**.
2. Fetch all orders and their customer details, including orders that **don’t match any customer** (simulate bad data).

**FULL OUTER JOIN**

1. Show a combined list of customers and orders so that you can see **all customers and all orders**, even if unmatched.

**Joins with Different Column Names**

1. Join **Customers** and **Shippings** to display customer names and their shipping status.

**Multiple Common Columns**

1. Suppose we need to join on both customer\_id and country. Write a query that shows customers and their orders but **only if they belong to ‘India’**.

**SELF JOIN**

1. Write a query to display pairs of customers from the **same country**.

**CROSS JOIN**

1. Generate all possible combinations of customers and orders. How many rows do you expect?

**USING Keyword**

1. Rewrite an INNER JOIN between Customers and Orders using the USING(customer\_id) clause.

**WHERE vs HAVING**

1. Find the number of orders placed by customers from India using **WHERE**.
2. Find countries that have **more than 1 order** using **HAVING**.

**SEMI JOIN (EXISTS)**

1. Write a query to display customers who have placed at least one order.

**ANTI JOIN (NOT EXISTS)**

1. Write a query to display customers who have **never placed an order**.

**Practical Business Mix**

1. Display customer name, order item, and shipping status in a single report (joining **all three tables**).

**✅ Lab Questions with Answers**

**1. Show customer name and items purchased (INNER JOIN)**

SELECT c.first\_name, c.last\_name, o.item, o.amount

FROM Customers c

INNER JOIN Orders o

ON c.customer\_id = o.customer\_id;

**2. Fetch only customers who have orders (INNER JOIN again)**

SELECT c.customer\_id, c.first\_name, o.order\_id, o.item

FROM Customers c

INNER JOIN Orders o

ON c.customer\_id = o.customer\_id;

**3. All customers with their orders (LEFT JOIN)**

SELECT c.first\_name, o.item, o.amount

FROM Customers c

LEFT JOIN Orders o

ON c.customer\_id = o.customer\_id;

**4. All orders with customer details (RIGHT JOIN)**

SELECT c.first\_name, o.order\_id, o.item, o.amount

FROM Customers c

RIGHT JOIN Orders o

ON c.customer\_id = o.customer\_id;

**5. All customers and all orders (FULL OUTER JOIN)**

SELECT c.first\_name, o.item, o.amount

FROM Customers c

FULL OUTER JOIN Orders o

ON c.customer\_id = o.customer\_id;

⚠️ In MySQL, simulate with:

SELECT c.first\_name, o.item, o.amount

FROM Customers c

LEFT JOIN Orders o ON c.customer\_id = o.customer\_id

UNION

SELECT c.first\_name, o.item, o.amount

FROM Customers c

RIGHT JOIN Orders o ON c.customer\_id = o.customer\_id;

**6. Customers with shipping status (different column names)**

SELECT c.first\_name, c.last\_name, s.status

FROM Customers c

INNER JOIN Shippings s

ON c.customer\_id = s.customer;

**7. Join on customer\_id + country = 'India'**

SELECT c.first\_name, o.item, o.amount

FROM Customers c

INNER JOIN Orders o

ON c.customer\_id = o.customer\_id

AND c.country = 'India';

**8. Pairs of customers from the same country (SELF JOIN)**

SELECT c1.first\_name AS Customer1, c2.first\_name AS Customer2, c1.country

FROM Customers c1

INNER JOIN Customers c2

ON c1.country = c2.country

AND c1.customer\_id <> c2.customer\_id;

**9. All combinations of customers and orders (CROSS JOIN)**

SELECT c.first\_name, o.item

FROM Customers c

CROSS JOIN Orders o;

**10. INNER JOIN using USING clause**

SELECT c.first\_name, o.item

FROM Customers c

INNER JOIN Orders o USING (customer\_id);

**11. Number of orders placed by Indian customers (WHERE)**

SELECT c.country, COUNT(o.order\_id) AS total\_orders

FROM Customers c

JOIN Orders o ON c.customer\_id = o.customer\_id

WHERE c.country = 'India'

GROUP BY c.country;

**12. Countries with more than 1 order (HAVING)**

SELECT c.country, COUNT(o.order\_id) AS total\_orders

FROM Customers c

JOIN Orders o ON c.customer\_id = o.customer\_id

GROUP BY c.country

HAVING COUNT(o.order\_id) > 1;

**13. Customers who have placed orders (SEMI JOIN via EXISTS)**

SELECT c.\*

FROM Customers c

WHERE EXISTS (

SELECT 1

FROM Orders o

WHERE c.customer\_id = o.customer\_id

);

**14. Customers who never placed orders (ANTI JOIN via NOT EXISTS)**

SELECT c.\*

FROM Customers c

WHERE NOT EXISTS (

SELECT 1

FROM Orders o

WHERE c.customer\_id = o.customer\_id

);

**15. Customer name, order item, and shipping status (3-way JOIN)**

SELECT c.first\_name, o.item, s.status

FROM Customers c

LEFT JOIN Orders o

ON c.customer\_id = o.customer\_id

LEFT JOIN Shippings s

ON c.customer\_id = s.customer;