

Craft a query using an INNER JOIN to combine 'orders' and 'customers' tables for customers in a specified region, and a LEFT JOIN to display all customers including those without orders.

The image displays two screenshots of the MySQL Workbench interface, showing the process of creating and populating a 'customers' table.

Top Screenshot: The 'SQL File Editor' shows the creation of the 'customers' table. The SQL code is as follows:

```
1 CREATE TABLE Customers (  
2   CustomerID INT AUTO_INCREMENT PRIMARY KEY,  
3   FirstName VARCHAR(50) NOT NULL,  
4   LastName VARCHAR(50) NOT NULL,  
5   Email VARCHAR(255) UNIQUE NOT NULL,  
6   Phone VARCHAR(20),  
7   Address TEXT,  
8   City VARCHAR(50),  
9   Country VARCHAR(50)  
10 );  
11 DESC customers;
```

The 'Output' pane shows the execution results:

#	Time	Action	Message	Duration / Fetch
43	15:51:03	INSERT INTO Customers (FirstName, LastName, Email, Phone, Address, City, Country) VAL...	4 row(s) affected Records: 4 Duplicates: 0 Warnings: 0	0.016 sec
44	15:51:52	DESC customers	8 row(s) returned	0.015 sec / 0.000 sec
45	15:51:58	SELECT * FROM customers customers LIMIT 0, 1000	4 row(s) returned	0.015 sec / 0.000 sec
46	15:52:17	CREATE TABLE Customers (CustomerID INT AUTO_INCREMENT PRIMARY KEY, Fir...	Error Code: 1050. Table 'customers' already exists	0.016 sec

Bottom Screenshot: The 'SQL File Editor' shows the insertion of data into the 'customers' table. The SQL code is as follows:

```
1 SELECT * FROM customers customers;  
2 INSERT INTO Customers (FirstName, LastName, Email, Phone, Address, City, Country)  
3 VALUES  
4   ('John', 'Doe', 'john.doe@example.com', '555-1234', '123 Main St', 'New York', 'USA'),  
5   ('Jane', 'Smith', 'jane.smith@example.com', '555-5678', '456 Park Ave', 'London', 'UK'),  
6   ('David', 'Lee', 'david.lee@example.com', '555-9012', '789 Elm St', 'Paris', 'France'),  
7   ('Sarah', 'Jones', 'sarah.jones@example.com', '555-3456', '101 Oak Ave', 'Tokyo', 'Japan');
```

The 'Output' pane shows the execution results:

#	Time	Action	Message	Duration / Fetch
44	15:51:52	DESC customers	8 row(s) returned	0.015 sec / 0.000 sec
45	15:51:58	SELECT * FROM customers customers LIMIT 0, 1000	4 row(s) returned	0.015 sec / 0.000 sec
46	15:52:17	CREATE TABLE Customers (CustomerID INT AUTO_INCREMENT PRIMARY KEY, Fir...	Error Code: 1050. Table 'customers' already exists	0.016 sec
47	15:52:27	INSERT INTO Customers (FirstName, LastName, Email, Phone, Address, City, Country) VAL...	Error Code: 1062. Duplicate entry 'john.doe@example.com' for key 'customers Email'	0.000 sec

MySQL Workbench interface showing a SQL script execution for a database named 'customers'.

SQL Script:

```
1 SELECT * FROM customers.customers;
2 INSERT INTO Customers (Firstname, Lastname, Email, Phone, Address, City, Country)
3 VALUES
4 ('John', 'Doe', 'john.doe@example.com', '555-1234', '123 Main St', 'New York', 'USA'),
5 ('Jane', 'Smith', 'jane.smith@example.com', '555-5678', '456 Park Ave', 'London', 'UK'),
6 ('David', 'Lee', 'david.lee@example.com', '555-9012', '789 Elm St', 'Paris', 'France'),
7 ('Sarah', 'Jones', 'sarah.jones@example.com', '555-3456', '101 Oak Ave', 'Tokyo', 'Japan');
```

Result Grid:

CustomerID	Firstname	Lastname	Email	Phone	Address	City	Country
1	John	Doe	john.doe@example.com	555-1234	123 Main St	New York	USA
2	Jane	Smith	jane.smith@example.com	555-5678	456 Park Ave	London	UK
3	David	Lee	david.lee@example.com	555-9012	789 Elm St	Paris	France
4	Sarah	Jones	sarah.jones@example.com	555-3456	101 Oak Ave	Tokyo	Japan

Output Log:

#	Time	Action	Message	Duration / Fetch
45	15:51:58	SELECT * FROM customers.customers LIMIT 0.1000	4 row(s) returned	0.015 sec / 0.000 sec
46	15:52:17	CREATE TABLE Customers (CustomerID INT AUTO_INCREMENT PRIMARY KEY, Fir...	Error Code: 1050. Table 'customers' already exists	0.016 sec
47	15:52:27	INSERT INTO Customers (Firstname, Lastname, Email, Phone, Address, City, Country) VAL...	Error Code: 1062. Duplicate entry 'john.doe@example.com' for key 'customers.Email'	0.000 sec
48	15:52:34	SELECT * FROM customers.customers LIMIT 0.1000	4 row(s) returned	0.000 sec / 0.000 sec

The interface shows the 'customers' schema with tables: customers, employee_details, library_system, books, member, sakila, sys, and world. The 'customers' table is selected in the 'Schema: customers' view.

MySQL Workbench interface showing a database connection to 'cloud4.rpsconsulting.in/console/#/client/MtU3NQBJAG15c3Fs'. The interface displays the 'customers' schema and a query window with the following SQL code:

```
4 ('John', 'Doe', 'john.doe@example.com', '555-1234', '123 Main St', 'New York', 'USA'),
5 ('Jane', 'Smith', 'jane.smith@example.com', '555-5678', '456 Park Ave', 'London', 'UK'),
6 ('David', 'Lee', 'david.lee@example.com', '555-9012', '789 Elm St', 'Paris', 'France'),
7 ('Sarah', 'Jones', 'sarah.jones@example.com', '555-3456', '101 Oak Ave', 'Tokyo', 'Japan');
8 SELECT * FROM customers;
9 SELECT FirstName, LastName, Email
10 FROM customers
11 WHERE City = 'New York';
```

The query results are displayed in the 'Result Grid' window, showing the following data:

CustomerID	FirstName	LastName	OrderID
1	John	Doe	1
1	John	Doe	2
1	John	Doe	6
1	John	Doe	7
2	Jane	Smith	3
2	Jane	Smith	8
3	David	Lee	4
3	David	Lee	9

The 'Output' window shows the execution log, including the following messages:

- 50 16:03:27 SELECT FirstName, LastName, Email FROM customers WHERE City = 'New York' LIMIT 0, 1000 1 row(s) returned
- 51 16:03:43 SELECT * FROM customers LIMIT 0, 1000 4 row(s) returned
- 52 16:03:58 SELECT * FROM customers LIMIT 0, 1000 4 row(s) returned
- 53 16:04:06 SELECT FirstName, LastName, Email FROM customers WHERE City = 'New York' LIMIT 0, 1000 1 row(s) returned

The interface also shows the 'customers' schema structure, including tables, views, stored procedures, and functions. The 'customers' table is highlighted in the 'Schemas' pane.

MySQL Workbench interface showing a database schema and query results.

Schema: customers

Tables: customers, orders, employee_details, library_system, books, member, sakila, sys, world.

Query 1:

```
SELECT * FROM customers.orders;
```

Query 2:

```
INSERT INTO Orders (CustomerID, OrderDate, TotalAmount)
VALUES
(1, '2024-07-05', 150.00),
(1, '2024-08-10', 75.50),
(2, '2024-09-20', 200.00),
(3, '2024-10-15', 120.75),
(4, '2024-11-01', 300.00);
```

Result Grid:

OrderID	CustomerID	OrderDate	TotalAmount
1	1	2024-07-05	150.00
2	1	2024-08-10	75.50
3	2	2024-09-20	200.00
4	3	2024-10-15	120.75
5	4	2024-11-01	300.00
6	1	2024-07-05	150.00
7	1	2024-08-10	75.50
8	2	2024-09-20	200.00

Output:

#	Time	Action	Message	Duration / Fetch
59	16:18:57	INSERT INTO Orders (CustomerID, OrderDate, TotalAmount) VALUES (1, '2024-07-05', 150.00)	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.000 sec
60	16:19:43	SELECT c.CustomerID, c.FirstName, c.LastName, o.OrderID FROM Customers c LEFT JOIN Orders o ON c.CustomerID = o.CustomerID	10 row(s) returned	0.000 sec / 0.000 sec
61	16:19:54	SELECT FirstName, LastName, Email FROM customers WHERE City = 'New York' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
62	16:20:17	SELECT * FROM customers.orders LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.016 sec

Query 3:

```
INSERT INTO Customers (FirstName, LastName, Email, Phone, Address, City, Country)
VALUES
('John', 'Doe', 'john.doe@example.com', '555-1234', '123 Main St', 'New York', 'USA'),
('Jane', 'Smith', 'jane.smith@example.com', '555-5678', '456 Park Ave', 'London', 'UK'),
('David', 'Lee', 'david.lee@example.com', '555-9012', '789 Elm St', 'Paris', 'France'),
('Sarah', 'Jones', 'sarah.jones@example.com', '555-3456', '101 Oak Ave', 'Tokyo', 'Japan');
```

Result Grid:

CustomerID	FirstName	LastName	Email	Phone	Address	City	Country
1	John	Doe	john.doe@example.com	555-1234	123 Main St	New York	USA
2	Jane	Smith	jane.smith@example.com	555-5678	456 Park Ave	London	UK
3	David	Lee	david.lee@example.com	555-9012	789 Elm St	Paris	France
4	Sarah	Jones	sarah.jones@example.com	555-3456	101 Oak Ave	Tokyo	Japan

Output:

#	Time	Action	Message	Duration / Fetch
60	16:19:43	SELECT c.CustomerID, c.FirstName, c.LastName, o.OrderID FROM Customers c LEFT JOIN Orders o ON c.CustomerID = o.CustomerID	10 row(s) returned	0.000 sec / 0.000 sec
61	16:19:54	SELECT FirstName, LastName, Email FROM customers WHERE City = 'New York' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
62	16:20:17	SELECT * FROM customers.orders LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.016 sec
63	16:20:35	SELECT * FROM customers customers LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec

MySQL Workbench interface showing a SQL query and its results.

SQL Query:

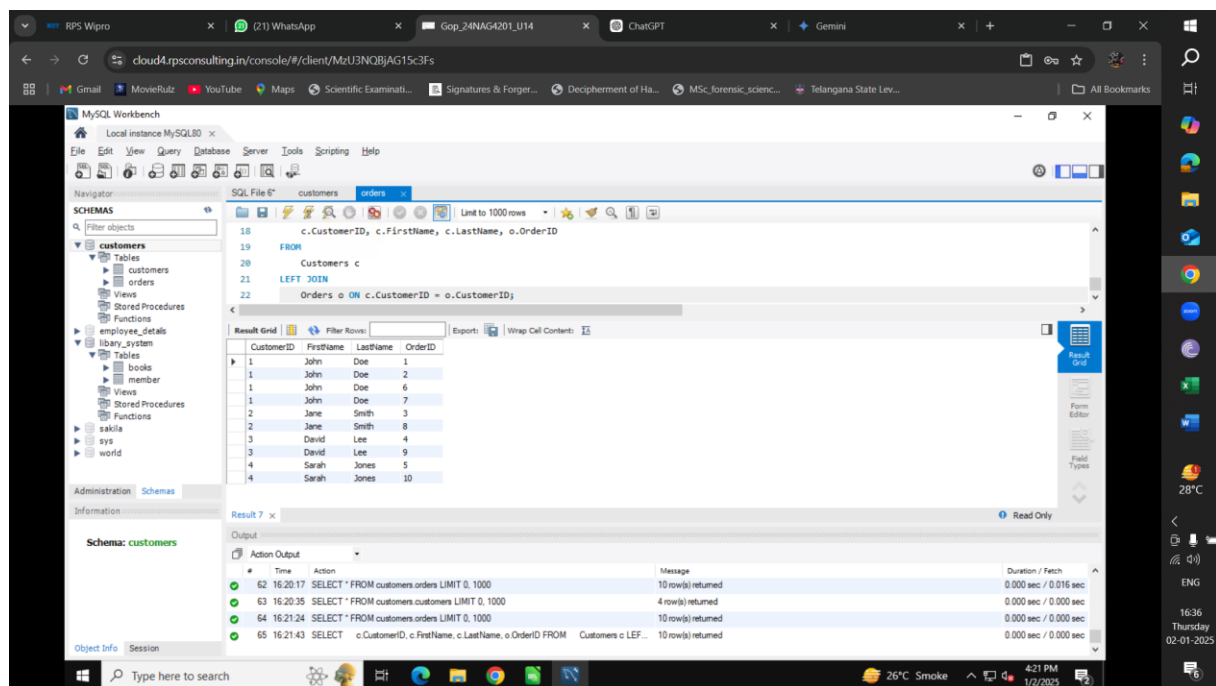
```
SELECT  
  c.CustomerID, c.FirstName, c.LastName, o.OrderID  
FROM  
  Customers c  
INNER JOIN  
  Orders o ON c.CustomerID = o.CustomerID  
WHERE  
  c.Region = 'West';
```

Result Grid:

OrderID	CustomerID	OrderDate	TotalAmount
1	1	2024-07-05	150.00
2	1	2024-08-10	75.50
3	2	2024-09-20	200.00
4	3	2024-10-15	120.75
5	4	2024-11-01	300.00

Output:

#	Time	Action	Message	Duration / Fetch
60	16:19:43	SELECT	c.CustomerID, c.FirstName, c.LastName, o.OrderID FROM Customers c LEFT JOIN Orders o ON c.CustomerID = o.CustomerID WHERE c.Region = 'West';	10 row(s) returned 0.000 sec / 0.000 sec
61	16:19:54	SELECT	First Name, Last Name, Email FROM customers WHERE City = 'New York' LIMIT 0, 1000	1 row(s) returned 0.000 sec / 0.000 sec
62	16:20:17	SELECT	* FROM customers.orders LIMIT 0, 1000	10 row(s) returned 0.000 sec / 0.016 sec
63	16:20:35	SELECT	* FROM customers.customers LIMIT 0, 1000	4 row(s) returned 0.000 sec / 0.000 sec



INNER JOIN:

This query retrieves CustomerID, FirstName, LastName, and OrderID for customers who have placed at least one order.

INNER JOIN combines rows from the Customers table (c) and the Orders table (o) based on the matching CustomerID values.

The WHERE clause filters the results to include only customers from the "West" region.

LEFT JOIN:

This query retrieves CustomerID, FirstName, LastName, and OrderID for all customers, including those who have not placed any orders.

LEFT JOIN returns all rows from the Customers table and matches them with the corresponding rows in the Orders table. If no matching order is found for a customer, the OrderID column will be NULL.

Note:

This assumes that your Customers table has a Region column.

Replace "West" with the actual region name in the WHERE clause of the INNER JOIN query.