Question 1. Retrieve all customers who have placed at least one order.

Answer:

Customers who have placed at least one order are those whose CustomerID appears in the Orders table.

```
SELECT DISTINCT c.CustomerID, c.CustomerName, c.City
FROM Customers c
INNER JOIN Orders o
ON c.CustomerID = o.CustomerID;
```

Answer

This query retrieves all customers who have placed at least one order by matching records between the Customers and Orders tables.

Question 2. Retrieve all customers and their orders, including customers who have not placed any orders.

Answer:

To include customers even if they have not placed any orders, we use a LEFT JOIN.

```
SELECT
  c.CustomerID,
  c.CustomerName,
  c.City,
  o.OrderID,
  o.OrderDate,
  o.Amount
FROM Customers c
LEFT JOIN Orders o
ON c.CustomerID = o.CustomerID;
```

Answer

This query displays all customers along with their order details.
Customers who have not placed any orders will still appear in the result, with NULL values in the order-related columns.

Question 3. Retrieve all orders and their corresponding customers, including orders placed by unknown customers.

Answer:

To include all orders, even those whose `CustomerID` does not exist in the Customers table, we use a RIGHT JOIN (or LEFT JOIN from Orders).

```
SELECT
   c.CustomerID,
   c.CustomerName,
   c.City,
   o.OrderID,
   o.OrderDate,
   o.Amount
FROM Customers c
RIGHT JOIN Orders o
ON c.CustomerID = o.CustomerID;
```

Answer

This query retrieves all orders along with their corresponding customer details.
Orders placed by unknown customers (CustomerID not present in Customers table) are also included, with NULL values for customer information.

Question 4. Display all customers and orders, whether matched or not.

Answer:

MySQL me **FULL OUTER JOIN** directly supported nahi hota, isliye hum **LEFT JOIN + RIGHT JOIN + UNION** ka use karte hain.

```
SELECT
  c.CustomerID,
  c.CustomerName,
  c.City,
  o.OrderID,
  o.OrderDate,
  o.Amount
FROM Customers c
LEFT JOIN Orders o
ON c.CustomerID = o.CustomerID

UNION

SELECT
  c.CustomerID,
  c.CustomerName,
  c.City,
  o.OrderID,
  o.OrderDate,
  o.Amount
FROM Customers c
```

```
RIGHT JOIN Orders o

ON c.CustomerID = o.CustomerID;
```

Answer

This query displays:

- All **customers**, even if they have not placed any orders
- All **orders**, even if they are placed by unknown customers

Matched records appear once, and unmatched records appear with **NULL values**.

Question 5. Find customers who have not placed any orders.

Answer:

Customers who have not placed any orders will have **NULL values** in the Orders table after a **LEFT JOIN**.

```
SELECT
  c.CustomerID,
  c.CustomerName,
  c.City
FROM Customers c
LEFT JOIN Orders o
ON c.CustomerID = o.CustomerID
WHERE o.OrderID IS NULL;
```

Answer

This query retrieves all customers who **have not placed any orders** by checking for NULL values in the Orders table.

Question 6. Retrieve customers who made payments but did not place any orders.

Answer:

We select customers who exist in the Payments table but do not exist in the Orders table.

```
SELECT DISTINCT
  c.CustomerID,
  c.CustomerName,
  c.City
FROM Customers c
INNER JOIN Payments p
ON c.CustomerID = p.CustomerID
LEFT JOIN Orders o
ON c.CustomerID = o.CustomerID
WHERE o.OrderID IS NULL;
```

Answer

This query retrieves customers who have made payments but have not placed any orders.

Question 7. Generate a list of all possible combinations between Customers and Orders.

Answer:

To generate all possible combinations between two tables, we use a CROSS JOIN.

```
SELECT
  c.CustomerID,
  c.CustomerName,
  c.City,
  o.OrderID,
  o.OrderDate,
  o.Amount
FROM Customers c
CROSS JOIN Orders o;
```

Answer

This query produces every possible combination of customers and orders.
Each customer is paired with every order, regardless of whether they are related.

Question 8. Show all customers along with order and payment amounts in one table.

Answer:

To show all customers, along with their order amounts and payment amounts, we use LEFT JOINs so that customers without orders or payments are also included.

```
SELECT
  c.CustomerID,
  c.CustomerName,
  c.City,
  o.OrderID,
  o.Amount AS OrderAmount,
  p.PaymentID,
  p.Amount AS PaymentAmount
FROM Customers c
LEFT JOIN Orders o
ON c.CustomerID = o.CustomerID
LEFT JOIN Payments p
ON c.CustomerID = p.CustomerID;
```

Answer

This query displays all customers along with their order amounts and payment amounts in a single result set.
Customers without orders or payments still appear, with NULL values in the respective columns.

Question 9. Retrieve all customers who have both placed orders and made payments.

Answer:

We select customers who exist in both Orders and Payments tables using INNER JOINs.

```
SELECT DISTINCT
  c.CustomerID,
  c.CustomerName,
  c.City
FROM Customers c
INNER JOIN Orders o
ON c.CustomerID = o.CustomerID
INNER JOIN Payments p
ON c.CustomerID = p.CustomerID;
```

Answer

This query retrieves all customers who have placed at least one order and made at least one payment.