Assignment1:

Write a SELECT query to retrieve all columns from a 'customers' table, and modify it to return only the customer name and email address for customers in a specific city.

Solution:

1. **Retrieve All Columns from the 'customers' Table SQL**:

SELECT * FROM customers;

2. **Retrieve Only Customer Name and Email Address for Customers in a Specific City** Assuming the columns in the **customers** table include **customer_name**, **email**, and **city**: **SQL**:

SELECT customer_name, email FROM customers

WHERE city = 'London';

Assignment2:

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.

Solution:

1. Using an **INNER JOIN** to combine the **orders** and **customers** tables for customers in a specified region.
2. Using a **LEFT JOIN** to display all customers, including those without orders.

1. **INNER JOIN to Combine 'orders' and 'customers' for Customers in a Specified Region** Assuming the '**customers'** table includes a column '**region'** to specify the customer's region: **SQL**:

SELECT orders.*, customers.customer_name, customers.email FROM orders

INNER JOIN customers ON orders.customer_id = customers.customer_id WHERE customers.region = 'West';

Example:

If the specified region is 'West', the query would be:

SQL:

SELECT orders.*, customers.customer_name, customers.email

FROM orders

INNER JOIN customers ON orders.customer_id = customers.customer_id WHERE customers.region = 'West;

2. LEFT JOIN to Display All Customers Including Those Without Orders SQL:

SELECT customers.customer_name, customers.email, orders.order_id, orders.order_date, orders.amount

FROM customers

LEFT JOIN orders ON customers.customer_id = orders.customer_id;

This query ensures that all customers are listed, including those who do not have any orders. The fields from the **orders** table will be **NULL** for customers without orders.

Combining both requirements, here's how you can use both queries in the same context:

Specified Region 'West' and Including All Customers:

SQL:

SELECT orders.*, customers.customer_name, customers.email FROM orders

INNER JOIN customers ON orders.customer_id = customers.customer_id WHERE customers.region = 'West';

Select customers.customer_name, customers.email, orders.order_id, orders.order_date, orders.amount

FROM customers

LEFT JOIN orders ON customers.customer_id = orders.customer_id;

These queries should help you achieve the desired results using **INNER JOIN** and **LEFT JOIN.**

Assignment3:

Utilize a subquery to find customers who have placed orders above the average order value, and write a UNION query to combine two SELECT statements with the same number of columns.

Solution:

1. Using a subquery to find customers who have placed orders above the average order value.
2. Writing a **UNION** query to combine two **SELECT** statements with the same number of columns.

1. Subquery to Find Customers Who Have Placed Orders Above the Average Order Value

Assuming the **orders** table includes columns **customer_id** and **amount**:

SQL:

--SELECT DISTINCT customer_id FROM orders

WHERE amount > (SELECT AVG(amount) FROM orders

);

To get the customer details (e.g., **customer_name** and **email**) for those who placed orders above the average value, assuming **customers** table includes **customer_id**, **customer_name**, and **email**:

SQL:

--SELECT c.customer_id, c.customer_name, c.email FROM customers c

WHERE c.customer_id IN (SELECT o.customer_id FROM orders o

WHERE o.amount > (SELECT AVG(amount) FROM orders

)

);

2. **UNION Query to Combine Two SELECT Statements with the Same Number of Columns** Assuming we want to combine data from two different regions, let's say **RegionA** and **RegionB**: **SQL**:

SELECT customer_id, customer_name, email, 'RegionA' AS region FROM customers

WHERE region = 'RegionA' UNION

SELECT customer_id, customer_name, email, 'RegionB' AS region FROM customers

WHERE region = 'RegionB';

This **UNION** query combines the results of customers from **RegionA** and **RegionB**, with an additional column indicating the region.

Combined Example

For a complete example, including both the subquery and the **UNION** query:

SQL:

SELECT c.customer_id, c.customer_name, c.email FROM customers c

WHERE c.customer_id IN (SELECT o.customer_id FROM orders o

WHERE o.amount > (SELECT AVG(amount) FROM orders

)

);

--SELECT customer_id, customer_name, email, 'RegionA' AS region FROM customers

WHERE region = 'RegionA' UNION

SELECT customer_id, customer_name, email, 'RegionB' AS region FROM customers

WHERE region = 'RegionB';

These queries demonstrate the use of a subquery to filter data based on a condition involving an aggregate function and the use of **UNION** to combine results from multiple **SELECT** statements.