Assignment 1:

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.

Sol:

The basic SQL query to retrieve all columns from a `customers` table looks like this:

SELECT * FROM customers;

To modify this query to return only the customer name and email address for customers in a specific city, you'll need to add a `WHERE` clause to filter by city and specify the columns you want to retrieve. Assuming the columns for customer name and email address are named `customer_name` and `email`, and the column for the city is named `city`, the query would be:

SELECT customer_name, email

FROM customers

WHERE city = 'Specific City';

Replace `'Specific City'` with the actual city you are interested in "New York".

SELECT customer_name, email

FROM customers

WHERE city = 'New York';

Here is the general form of the query:

SELECT customer_name, email

FROM customers

WHERE city = 'YourCityName';

Just replace ` with the desired YourCityName`city's name.

==========

Assignment 2:

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.

Sol:

Here are the Two SQL queries:

1. INNER JOIN to combine `orders` and `customers` tables for customers in a specified region.

2. LEFT JOIN to display all customers, including those without orders.

Query 1: INNER JOIN for Customers in a Specified Region

Let's assume the `customers` table has columns like `customer_id`, `customer_name`, `email`, and `region`, and the `orders` table has columns like `order_id`, `order_date`, `customer_id`, and `order_amount`.

Here's how you can write an `INNER JOIN` query to get the orders for customers in a specific region:

```
SELECT c.customer_name, c.email, o.order_id, o.order_date, o.order_amount
FROM customers c
INNER JOIN orders o ON c.customer_id = o.customer_id
WHERE c.region = 'Specified Region';
```

Replace `'Specified Region'` with the actual region you are interested in. For example, if the region is 'North', the query would be:

```
SELECT c.customer_name, c.email, o.order_id, o.order_date, o.order_amount
FROM customers c
INNER JOIN orders o ON c.customer_id = o.customer_id
WHERE c.region = 'North';
```

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

Here's how you can write a `LEFT JOIN` query to get all customers, including those without any orders:

```
SELECT c.customer_name, c.email, o.order_id, o.order_date, o.order_amount
FROM customers c
LEFT JOIN orders o ON c.customer_id = o.customer_id;
```

This query will return all customers, and for those without any orders, the `order_id`, `order_date`, and `order_amount` fields will contain `NULL`.

By combining these queries, you can retrieve the information as needed for different scenarios.

==========

Assignment 3:

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.

Sol:

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

To find customers who have placed orders above the average order value, we first need to calculate the average order value. Then, we use a subquery to filter out customers whose orders exceed this average.

```
SELECT c.customer_name, c.email
FROM customers c
WHERE c.customer_id IN (
    SELECT o.customer_id
    FROM orders o
    WHERE o.order_amount > (
```

SELECT AVG(order_amount)

FROM orders));

UNION Query to Combine Two SELECT Statements with the Same Number of Columns

A UNION query combines the results of two or more SELECT statements. Each SELECT statement within the UNION must have the same number of columns, and the corresponding columns must have compatible data types.

Here are two example SELECT statements that can be combined using UNION:

1. The first SELECT statement retrieves customer names and emails from customers who are located in 'Region A'.

2. The second SELECT statement retrieves customer names and emails from customers who have placed orders above a certain amount.

-- First SELECT statement: Customers from 'Region A'

SELECT customer_name, email

FROM customers

WHERE region = 'Region A'

UNION

-- Second SELECT statement: Customers with orders above a certain amount

SELECT c.customer_name, c.email

FROM customers c

JOIN orders o ON c.customer_id = o.customer_id

WHERE o.order_amount > 100;

In this example, replace `'Region A'` with the desired region and `100` with the order amount threshold you are interested in.

Combined Explanation

-Subquery: Used within the `WHERE` clause to find customers who have placed orders above the average order value. The subquery calculates the average order value, and the main query filters customers based on this value.

-UNION: Combines the results of two SELECT statements with the same number of columns and compatible data types. The `UNION` operator removes duplicate rows by default. If you want to include duplicates, use `UNION ALL`.

These queries demonstrate how to perform complex data retrieval tasks using subqueries and UNION operations in SQL.