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.

```
SELECT customer_name, email

FROM customers

WHERE city = 'CityName';
```

Modifying data:

```
UPDATE customers

SET customer_name = “Narayan More” where id  =3,

and

   email = “narayanmore2019@gmail.com”

WHERE city = 'New York';
```

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.

INNER JOIN to combine 'orders' and 'customers' tables for customers in a specified region:

```
SELECT c.customer_id, c.customer_name, c.email, c.region, o.order_id,
o.order_date, o.amount

FROM customers c ,orders o

INNER JOIN

ON c.customer_id = o.customer_id

WHERE c.region = 'West';
```

```
SELECT c.customer_id, c.customer_name, c.email, c.region, o.order_id,
o.order_date, o.amount

FROM customers c ,orders o

LEFT JOIN

ON c.customer_id = o.customer_id;
```

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.

```
SELECT  customer_name, c.email,

FROM customers c, orders o

INNER JOIN

ON  c.customer_id = o.customer_id
```

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

Assignment 4: Compose SQL statements to BEGIN a transaction, INSERT a new record into the 'orders' table, COMMIT the transaction, then UPDATE the 'products' table, and ROLLBACK the transaction.

Step-by-Step Instructions

1. *BEGIN a transaction*
2. *INSERT a new record into the table*
3. *COMMIT the transaction*
4. *BEGIN a new transaction*
5. *UPDATE the products table*
6. *ROLLBACK the transaction*

Begin the first transaction

1] BEGIN;

INSERT INTO orders (order_id, order_date, customer_id, amount)

VALUES (101, '2024-05-23', 1, 150.00);

2] Commit the transaction

COMMIT;

3] Begin the second transaction

BEGIN;

4] Update the 'products' table

UPDATE products

SET stock = stock - 10

WHERE product_id = 1;

5]Rollback the transaction

ROLLBACK;

Assignment 6: Draft a brief report on the use of transaction logs for data recovery and create a hypothetical scenario where a transaction log is instrumental in data recovery after an unexpected shutdown.

Transaction logs are a critical component of database management systems (DBMS). They record all changes made to the database, including additions, modifications, and deletions of data. The primary purpose of transaction logs is to ensure data integrity and provide a mechanism for data recovery in the event of system failures, such as unexpected shutdowns, crashes, or other disasters**.**

1. Initial Assessment: Upon restarting the database server, the DBMS detects that an unexpected shutdown occurred and automatically initiates the recovery process.

2. Redo Phase:
    - The DBMS begins reading the transaction log from the last checkpoint.
    - It re-applies all changes recorded in the log for transactions that were committed up to the point of failure. This ensures that all committed transactions are reflected in the database.
3. Undo Phase:
    - The DBMS identifies any transactions that were active but not committed at the time of the failure.
    - It reverses the changes made by these uncommitted transactions using the pre-change data recorded in the transaction log. This ensures that any incomplete transactions do not affect the database's consistency.

Outcome: After completing the recovery process, the database is restored to a consistent state, reflecting all committed transactions and excluding any partial changes from uncommitted transactions. The retail company's database is now accurate and reliable, ready to continue operations without data loss or corruption.