Day 8 Assignment

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, customer.email
FROM
    customer
WHERE
    city = 'New York';
```

Script is added here : Script
Table is added here : SelectQueryDay8 table

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.

```
SELECT customer.customer_id, customer.name, customer.email, customer.city, orders.order_id,
orders.order_date, orders.total_amount
FROM Customer
LEFT JOIN Orders ON customer.customer_id = orders.customer_id
WHERE customer.city = 'New York';
```

Script is added here : Script
Table is added here : LeftJoinCustomer Table

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 c.customer_id, c.name, c.email, c.city, SUM(o.total_amount) AS total_order_amount
FROM Customer c
JOIN Orders o ON c.customer_id = o.customer_id
```

```
GROUP BY c.customer_id
HAVING SUM(o.total_amount) > (
    SELECT AVG(total_amount)
    FROM Orders
) UNION
SELECT c.customer_id, c.name, c.email, c.city, SUM(o.total_amount) AS total_order_amount
FROM Customer c
LEFT JOIN Orders o ON c.customer_id = o.customer_id
GROUP BY c.customer_id
HAVING SUM(o.total_amount) <= (
    SELECT AVG(total_amount)
    FROM Orders
);
```

Script is added here : Script
Table is added here : CustomerAboveAvg table

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.

```
START TRANSACTION;

INSERT INTO Orders (order_id, customer_id, order_date, total_amount, product_id)
VALUES (51, 21, '2024-05-23', 120.00, 6);

COMMIT;

SELECT *
FROM Orders
WHERE order_id = 51;

START TRANSACTION;

UPDATE Product
SET price = 125.00
WHERE product_id = 6;

SELECT *
FROM Product
WHERE product_id = 6;
```

```
ROLLBACK;

SELECT *
FROM Product
WHERE product_id = 6;
```

Script is added here : Script
Table is added here : CommitOrder table , UpdateOrders and RollbackOrders

---

Assignment 5: Begin a transaction, perform a series of INSERTs into 'orders', setting a SAVEPOINT after each, rollback to the second SAVEPOINT, and COMMIT the overall transaction.

```
START TRANSACTION;

INSERT INTO Orders (order_id, customer_id, order_date, total_amount, product_id)
VALUES (101, 1, '2024-05-23', 50.00, 1);

SAVEPOINT SP1;

INSERT INTO Orders (order_id, customer_id, order_date, total_amount, product_id)
VALUES (102, 2, '2024-05-24', 75.00, 2);

SAVEPOINT SP2;

INSERT INTO Orders (order_id, customer_id, order_date, total_amount, product_id)
VALUES (103, 3, '2024-05-25', 100.00, 3);

SAVEPOINT SP3;

SELECT * FROM Orders
ORDER BY order_id DESC
LIMIT 3;

ROLLBACK TO SAVEPOINT SP2;

COMMIT;

SELECT * FROM Orders
ORDER BY order_id DESC
```

LIMIT 2;

Script is added here : Script
Table is added here : SavepointOrder table and SavepointRollBackOrders table

---

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.

Transactional Logs are logs which keep the record of data in a sequential order. It helps monitor the system. If any problem arises due to which data recovery is needed. Transactional logs help establish the safe state at which the database was. It also helps undo and redo changes which might occur due errors, failures, hacks, unplanned shutdown etc.
Let us say Company A is a third party data management vendor which helps a hospital to keep records of the bill. The company handles large volumes of data at high velocity and high velocity and each second thousands of transactions are made at the server.
One day due to an unseen server failure that system stays off for 5 minutes. During the 5 minutes probably the hospital had more than 5000+ transactions.
To ensure that ACID properties of the database remain intact, the company A has decided to check their transaction logs and make changes in the database.
The changes are made according to two criteria, first one is rollback and another one is committed.

- Rollback will occur when the transaction seems incomplete. There is no point in Committing a incomplete transaction.
- Committing transactions which are fully complete and have no issues of incompleteness.

Thus, within a minute they will be able to map the correct data and push it to its original safe state.

Notes
There are some changes in Orders table and Product Table due to requirement of product table in later question.
*Table records have been populated with the help of **ChatGPT** generated records.*

Link to the table :
Customer Table
Orders Table
Products Table