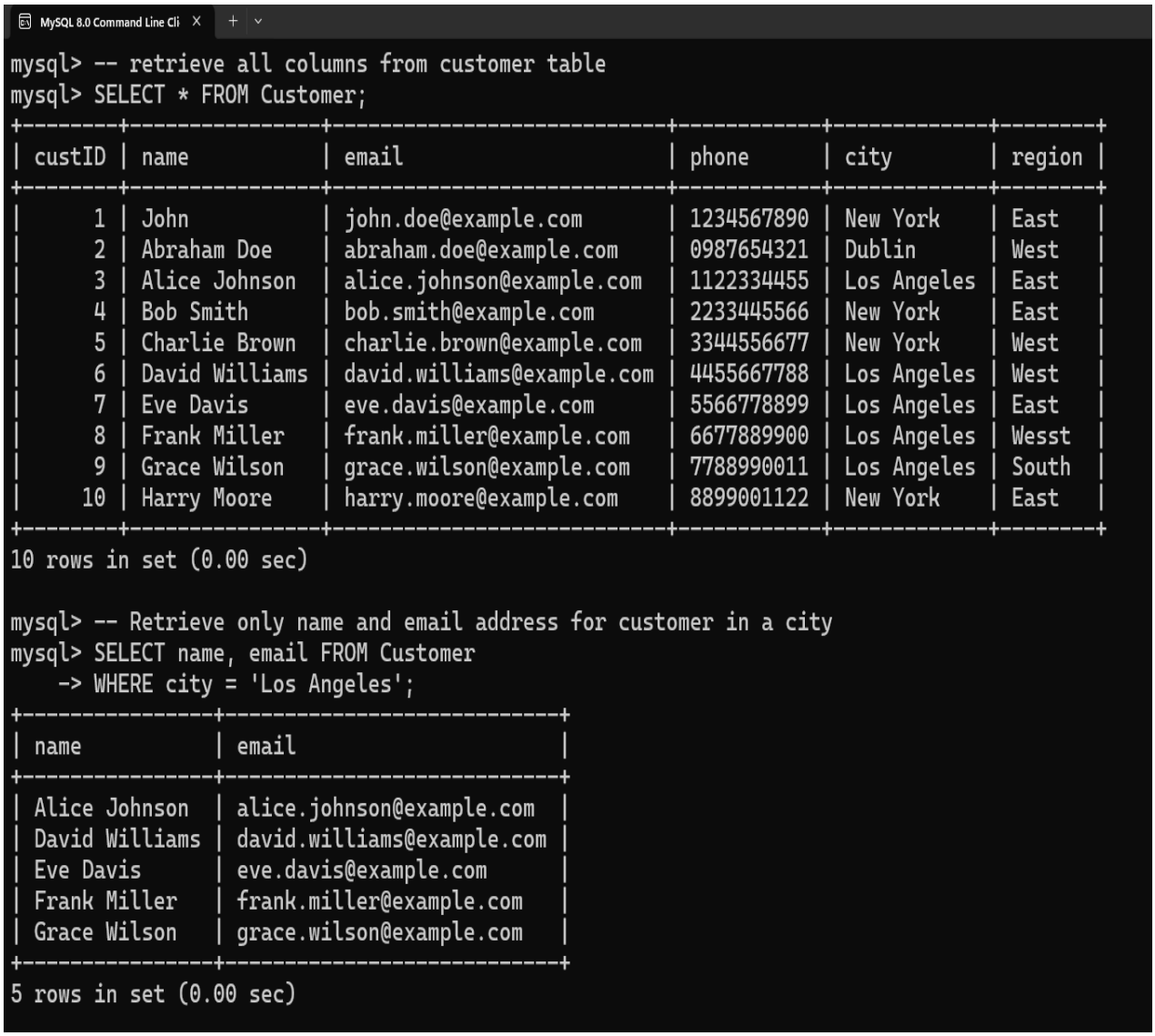
Md Yaseen

DAY-10 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.

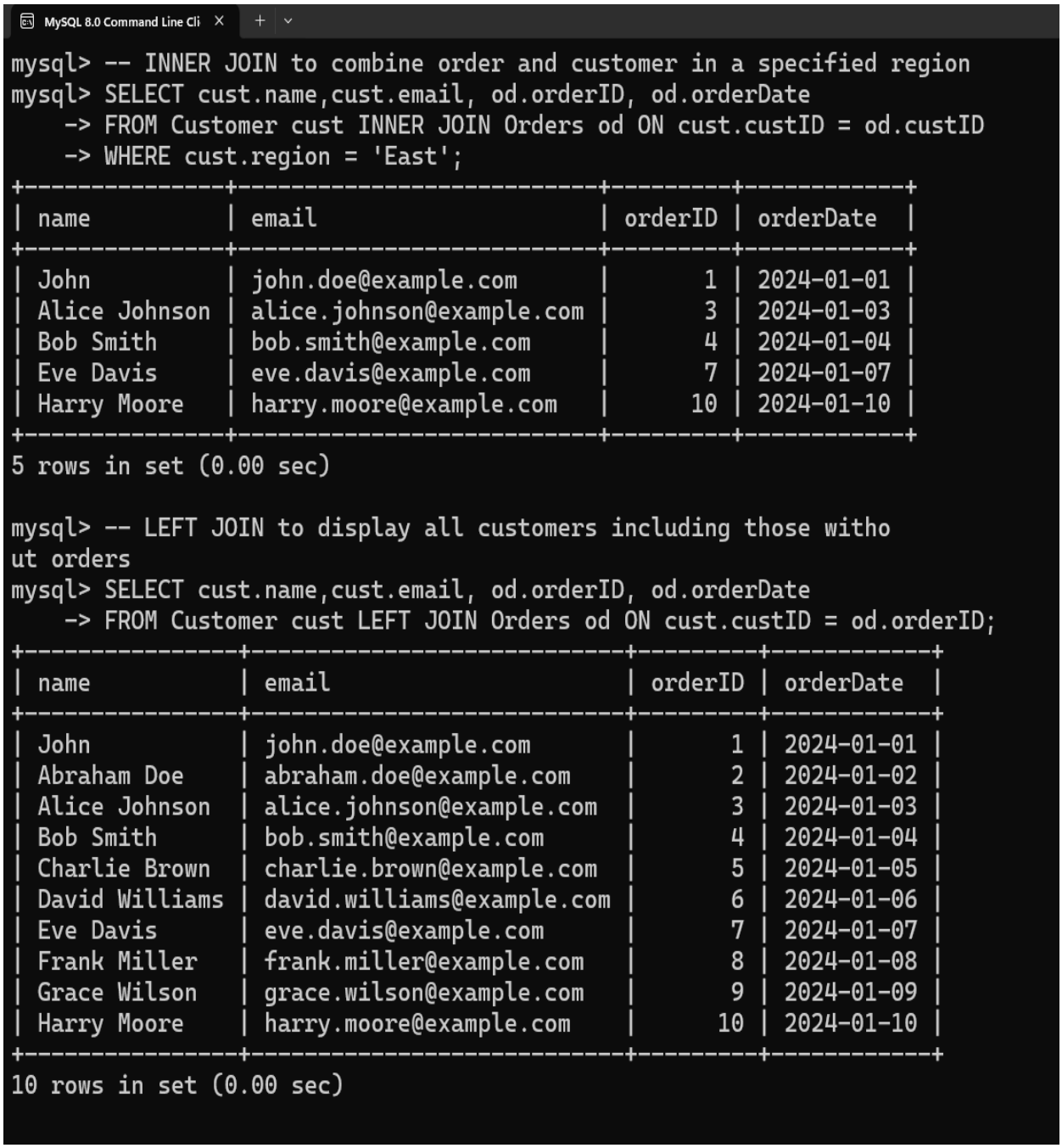
SOLUTION:



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.

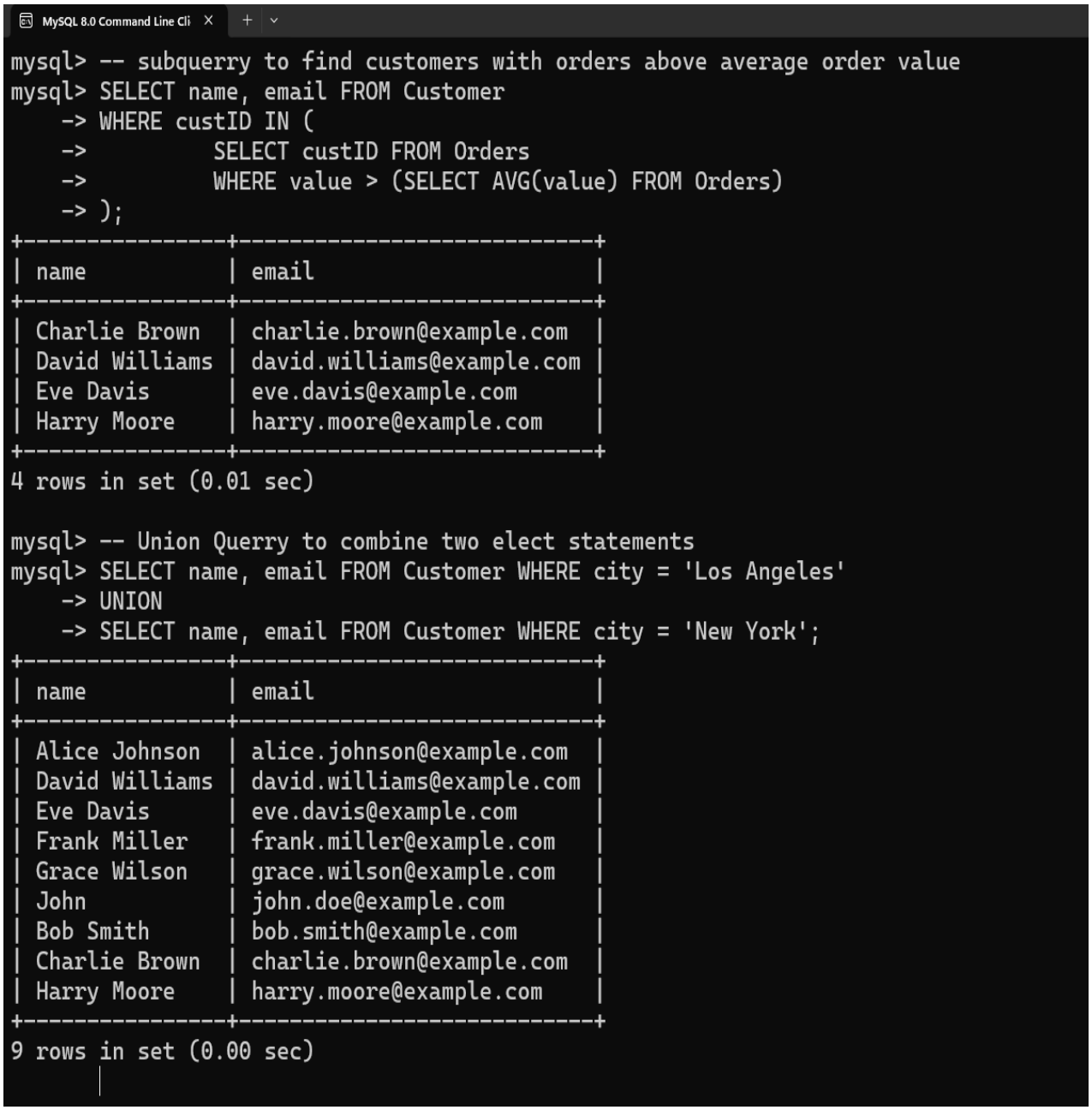
SOLUTION:



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.

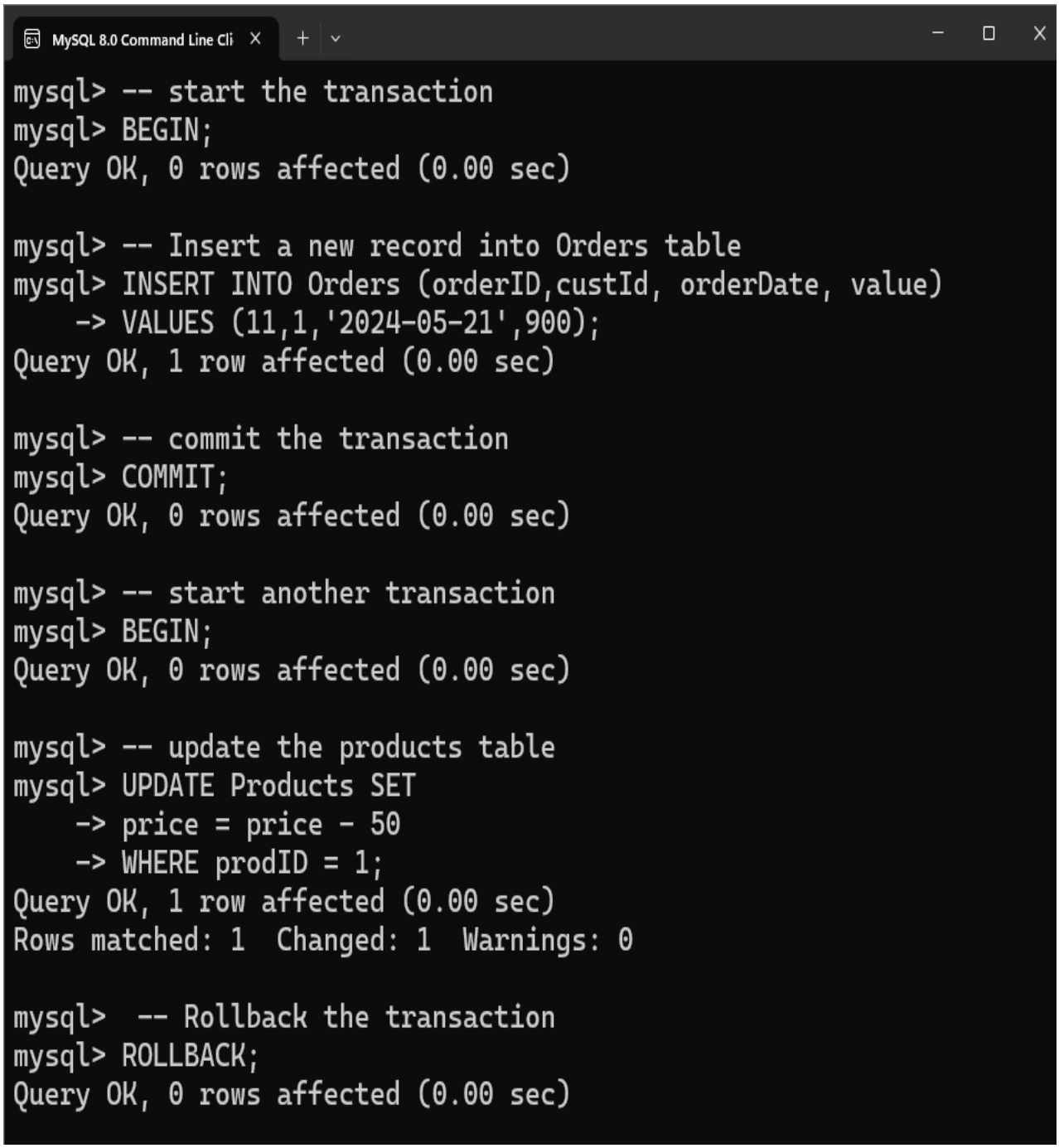
SOLUTION:



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.

SOLUTION:



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.

SOLUTION:

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.

SOLUTION:

Transaction Logs for Data Recovery:

Transaction logs are critical for ensuring data integrity and enabling data recovery in case of system failures. These logs record all changes made to the database, including INSERT, UPDATE, DELETE operations, and transaction control commands like COMMIT and ROLLBACK. By keeping a detailed record of all transactions, transaction logs allow the database to:

- **Recover from Crashes:** In the event of an unexpected shutdown or crash, the database can use the transaction log to recover to the last known consistent state. Any uncommitted transactions can be rolled back, and committed transactions can be replayed to ensure no data loss.
- **Point-in-Time Recovery:** Transaction logs enable point-in-time recovery, allowing the database to be restored to a specific moment before an error or data corruption occurred.

Hypothetical Scenario

Imagine a retail company's database server crashes unexpectedly due to a power outage. The database was handling numerous transactions at the time, including new orders and updates to inventory levels. Upon restarting the server, the database uses the transaction log to:

- **Identify Uncommitted Transactions:** Any transactions that were not committed at the time of the crash are identified and rolled back, ensuring that partial or corrupt data is not retained.

- **Replay Committed Transactions:** Transactions that were committed but not yet written to the main database files are replayed from the log, ensuring that all customer orders and inventory updates are accurately reflected.

This process allows the company to resume operations with confidence that the database is in a consistent and accurate state, minimizing downtime and potential data loss.