EE4350-Database Systems

Mini Project- Database for a Trip Planner App

Group Members

EG/2021/ 4667 – Malsha H.C

EG/2021/4675 – Mewan L.A.D.Y

EG/2021/4671 - Manthilaka M.M.P.L

Contents

Introduction

This report outlines the development process of the "Trip Planning Application Database". The primary objective of the database is to efficiently manage and store the important data related to users who are using the application to plan their trips. The database consists of crucial data such as the user's password and planned trips, including accommodations, destinations, and activities. MySQL was employed as the database management system.

Step 01- Requirement Analysis

The initial phase was to identify the functional and data requirements of the database. This includes understanding the functionality of the app and finalizing the types of information to be stored. After that, the attributes for each entity were identified.

Step 02 – Conceptual Design

The conceptual design process was done based on the data requirements that were identified during the first step. According to the data and functionality requirements, the ER diagram was created using the visual paradigm tool. This diagram is created focusing on the organization of data and the relationships between them. It helps to understand the entities, their attributes, and how they relate to each other. Notations to show structural constraints and role names in relationships were also included. Then the conceptual schema is used to create a logical design of the database. Creating the UML diagram involved defining classes with careful consideration to the Normalization.

Step 03- Implementation

Implementation of the database was done using MySQL client with 2NF of the representational data model. Schema creation, table creation, and Insert, Update, and Delete operations were also done. In this phase, we populated our database with suitable values.

Step 04 – Transactions

In the Transaction phase, we wrote queries to retrieve data from the database. These queries consist of 7 simple queries and 13 Complex queries based on the requirements stated in the project guidelines.

Step 05- Tunning

We focused on improving efficiency and responsiveness of our SQL queries. For this, first, we identified the slow-running queries and then applied optimization techniques like improving filtering, aggregating, indexing, and optimizing joins. We also used parameterization to prevent SQL injection.

1 Chapter 01 – Requirement Analysis

Functional and Data Requirements that were identified for the database are listed below.

1.1 Data Requirements

Entity	Attributes
User	User_Id, User Name, First_name, Last_name, Password, Phone Number, Email, Age, Date of Birth, Gender, Profile Picture, Guide_id
Review	Comments, Ratings
Trip	Trip_Id, Start_date, End_date, Description, Status
Accommodation	Acco_Id, Acco_state, Acco_city, Acco_PhoneNumber, Acco_name, Check_out_date, Check_in_date
Transportation	Transportation_Id, Arrival_date, Departure_date, Location, Mode
Expense	Expense_Id, Amount, Notes, Category
Destination	Destination_Id, Lattitude, Longitude, Dest_Name, Description
Activity	Name, Time

1.2 Functional Requirements

Functional Area	Functional Requirement
User Management	1. Registration- Users can register by providing the necessary information. 2. Login- Registered users should be able to log into the system securely using their password. 3. Profile Management- Users can update their profile information 4. Role Management – Guides are required to register and get a Guide_id.
Trip Management	1. Create Trip: Users should have the ability to create new trips by providing trip details. 2. View Trips- Users should be able to view details of their planned trips.

	3. Edit Trips- Users have access to edit the trip they have planned already. 4. Delete Trip- Users can cancel the trip by deleting it.
Accommodation	1. Book Accommodation- Book Accommodations for the trips. 2. View Accommodation – Users should be able to view accommodation details.
Transportation	1. Book Transportation- Users should be able to book transportation for planned trips. 2. View Transportation- Users should be able to view details of their transportation mode.
Expenses	1. Record Expenses- Users should be able to record trip expenses specifying the amount and other details. 2. View Expenses- Users should be able to view the summary of the expenses for their trip.
Destinations	1. Search destinations- Before planning users should be able to search the destinations based on criteria such as name or locations. 2. View Destination details- Users should be able to view all the details of the destinations they have chosen.
Activities	1. Choose activities- Users should be able to choose activities that are relevant to their destinations. 2. View Activities- Users should be able to view the activities they have chosen
Reviews	1, Submit Reviews- Users have the ability to submit reviews for destinations, accommodations etc. 2. View Reviews- Users should have the ability to view the feedback submitted by themselves and other users.
Security and Access Control	1. Authentication- Ensure a secure login system with username and password.

2 Chapter 02- Conceptual Design

The conceptual data model is created to represent the overall structure of the database. It provides a valuable opportunity to understand the database very easily. We have created the ER diagram using the visual paradigm tool.

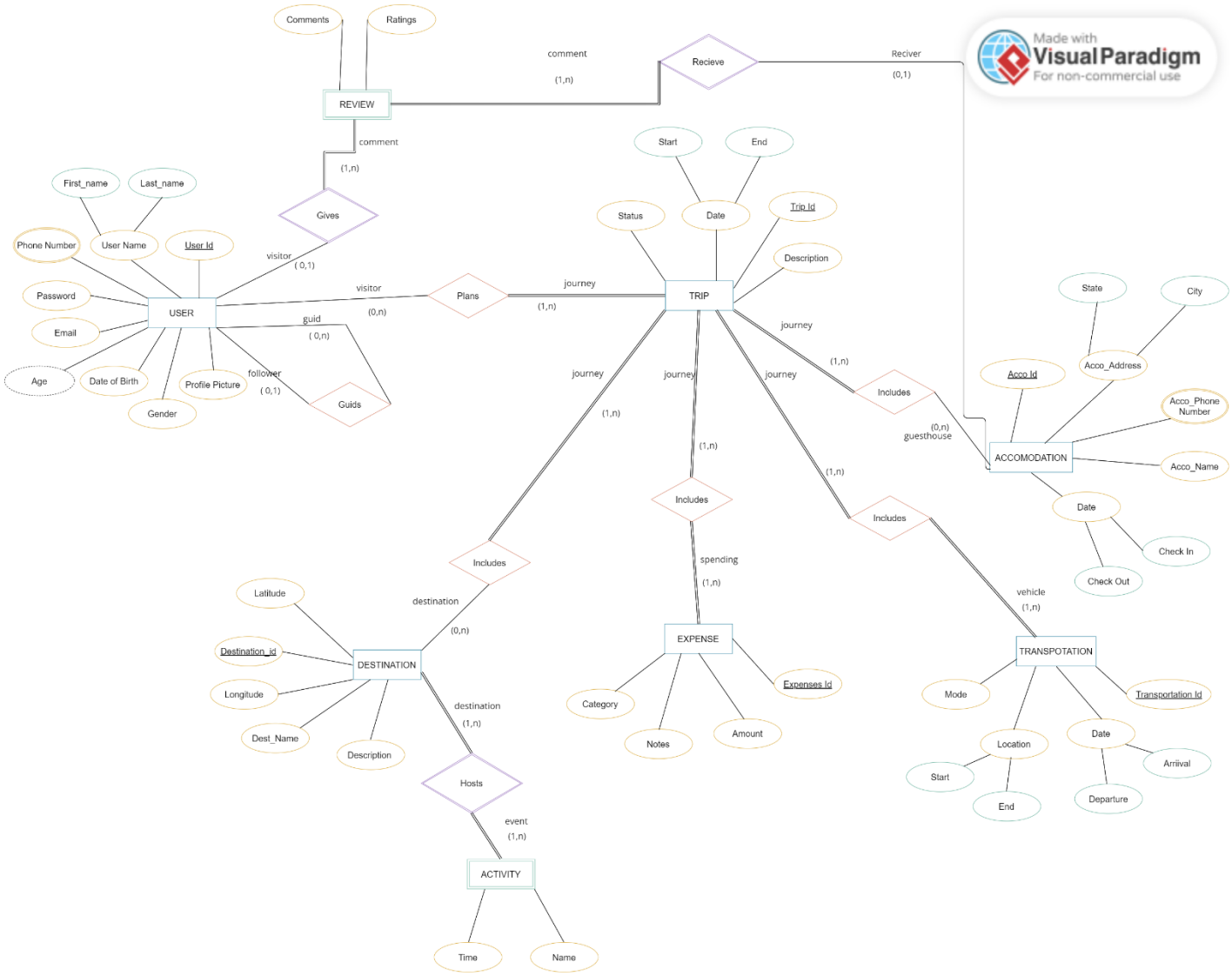


Figure 1: Entity Relationship Diagram

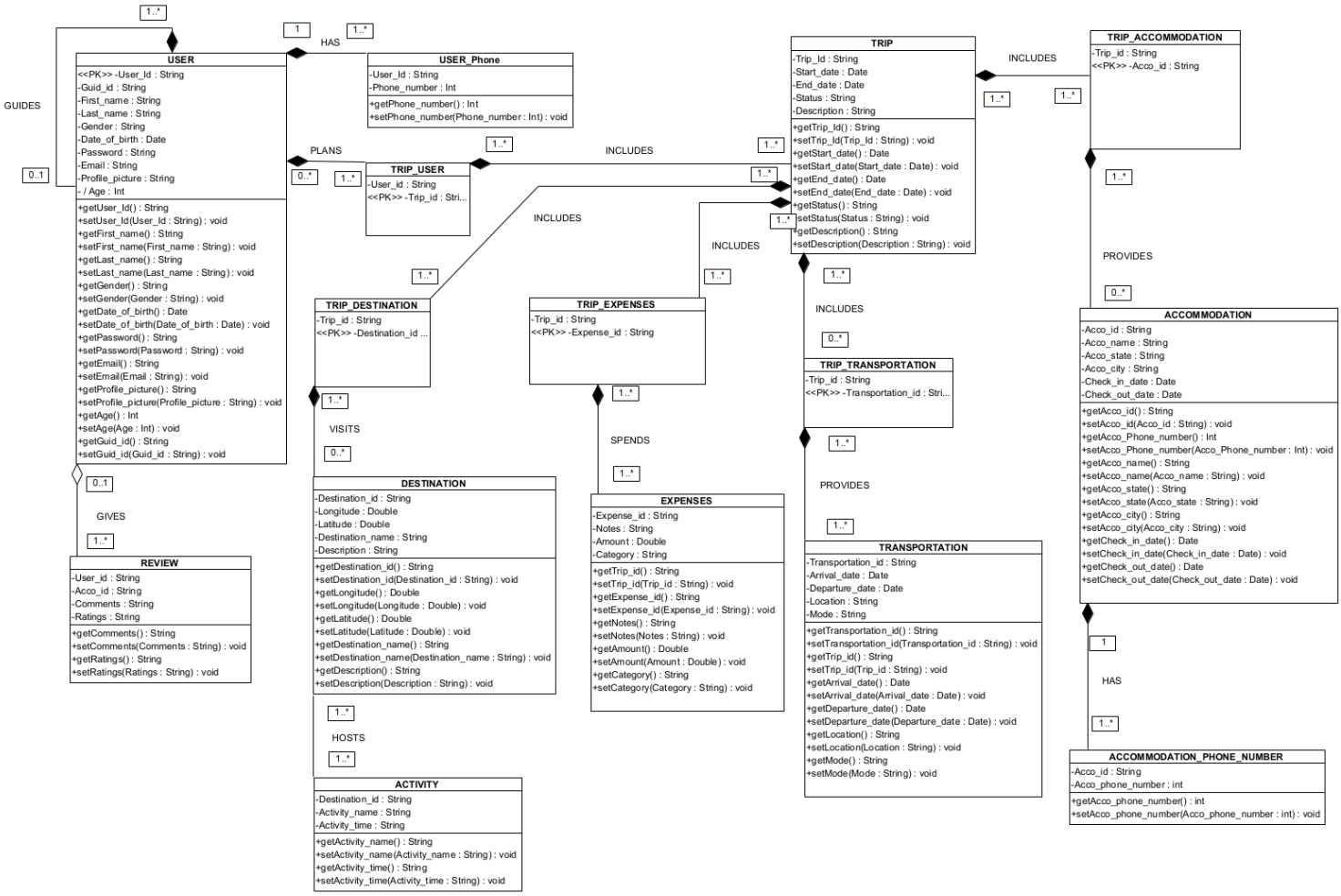


Figure 2: UML Class Diagram

3 Chapter 03- Implementation

3.1 Creating Database

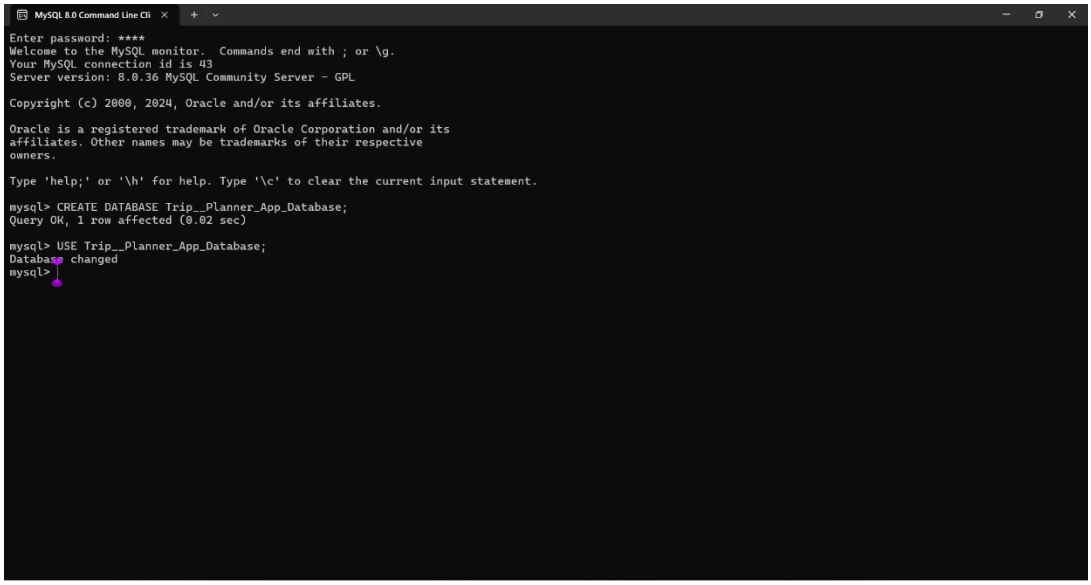


Figure 3: Creating Database

3.2 Creating tables

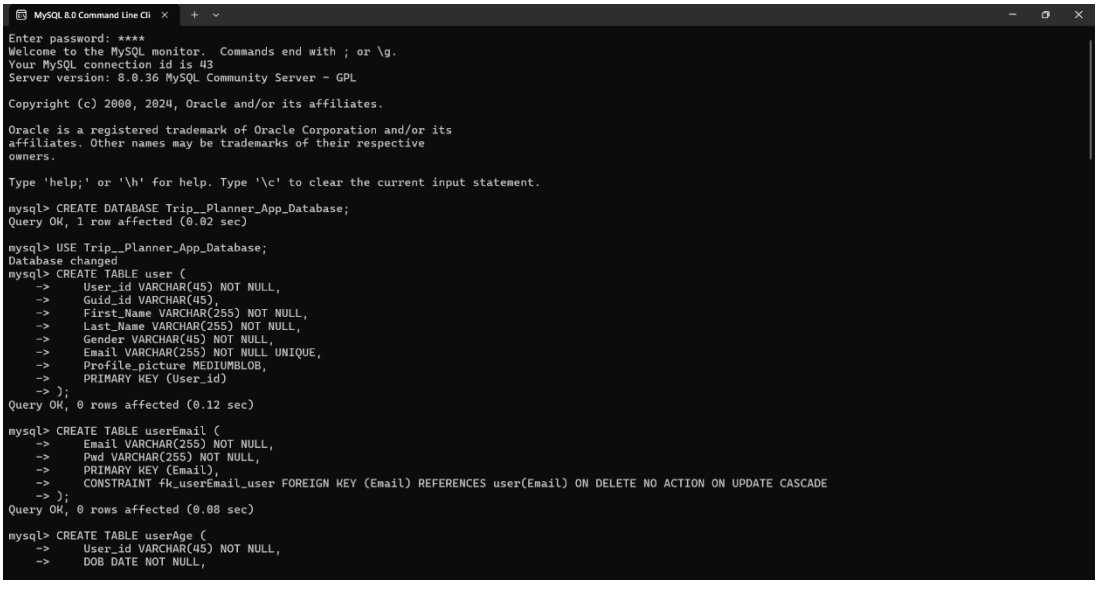


Figure 4: Creating user, userEmail Tables

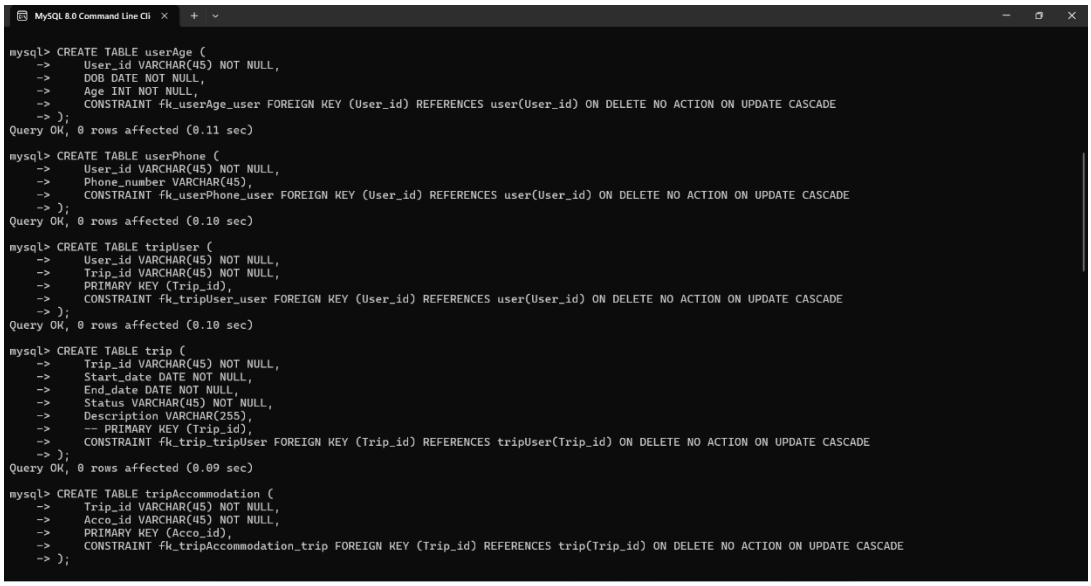


Figure 5: Creating userAge, userPhone, tripUser, trip, tripAccommodation tables

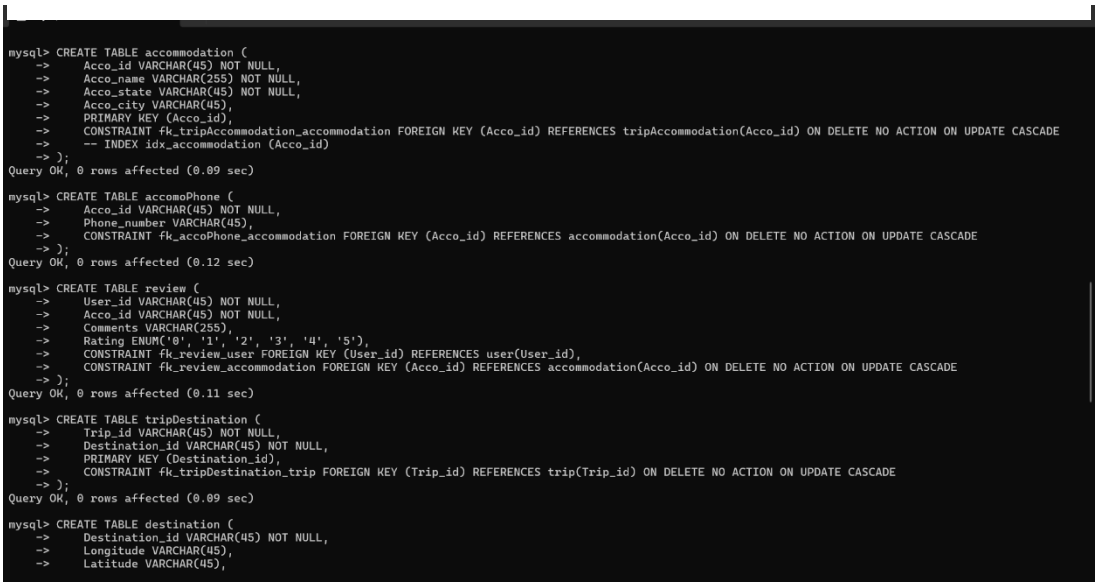


Figure 6: Creating accommodation, accomoPhone, review, tripDestination tables

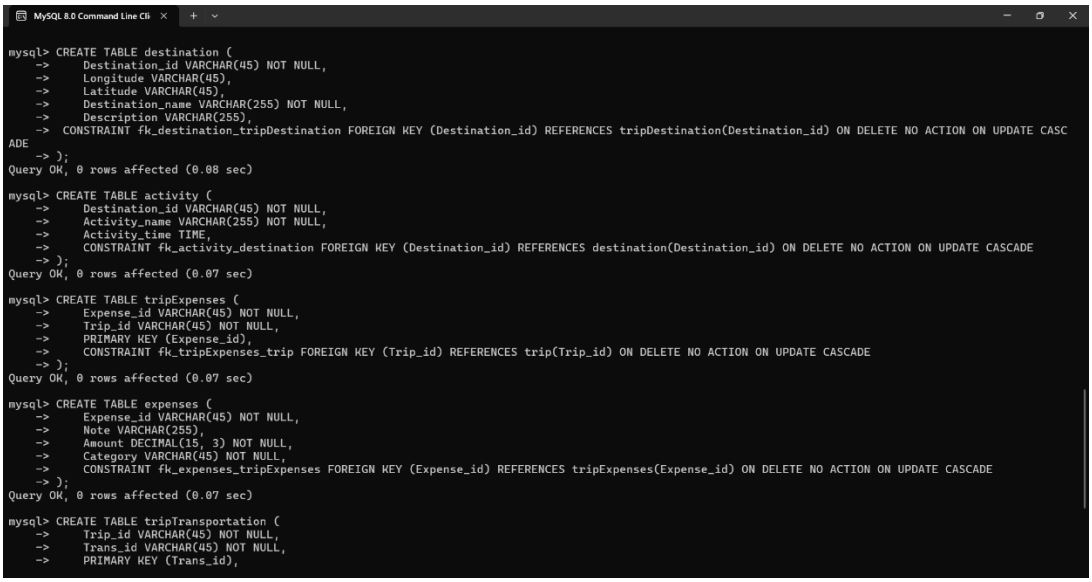


Figure 7: Creating destination, activity, tripExpenses and expenses tables

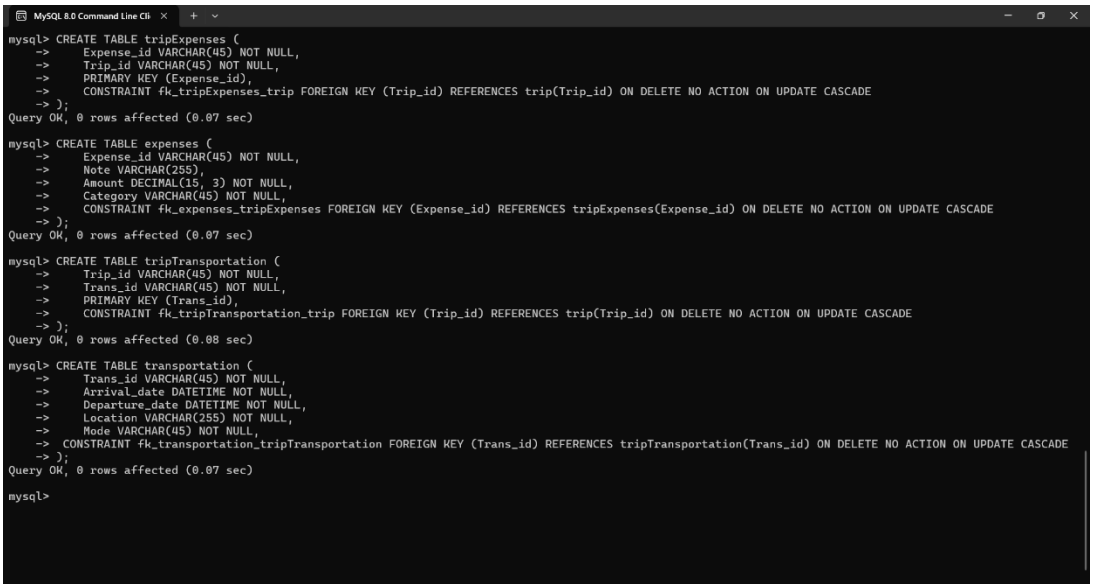


Figure 8: Creating tripTransportation, transportation table

3.3 Insert Data to the Tables

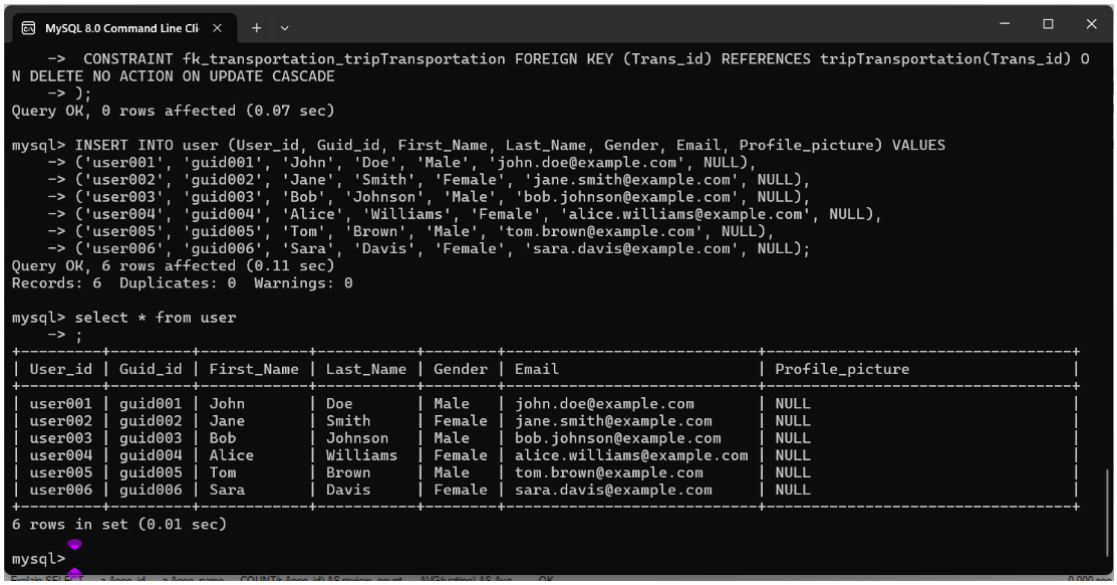


Figure 9: Inserting data into the user table

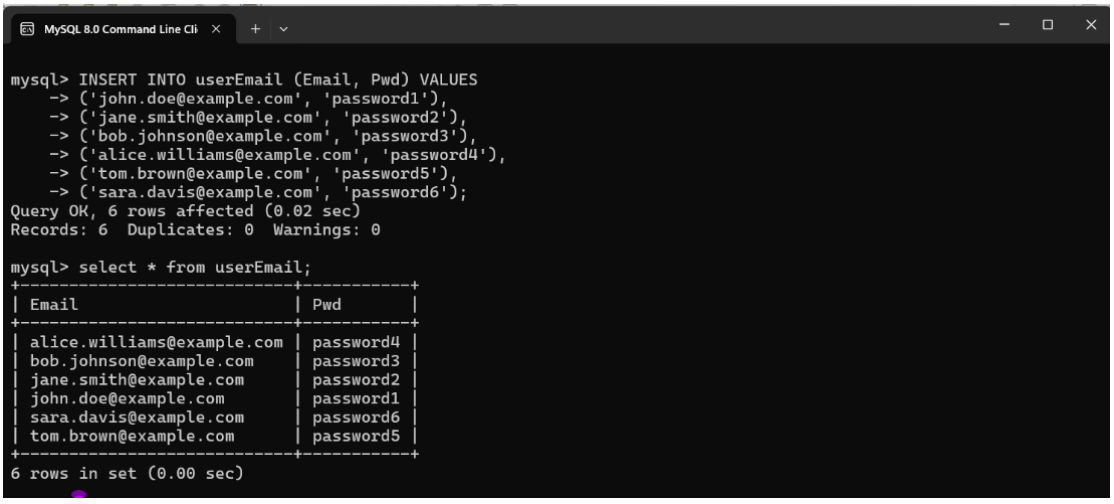


Figure 10: Inserting data into the userEmail

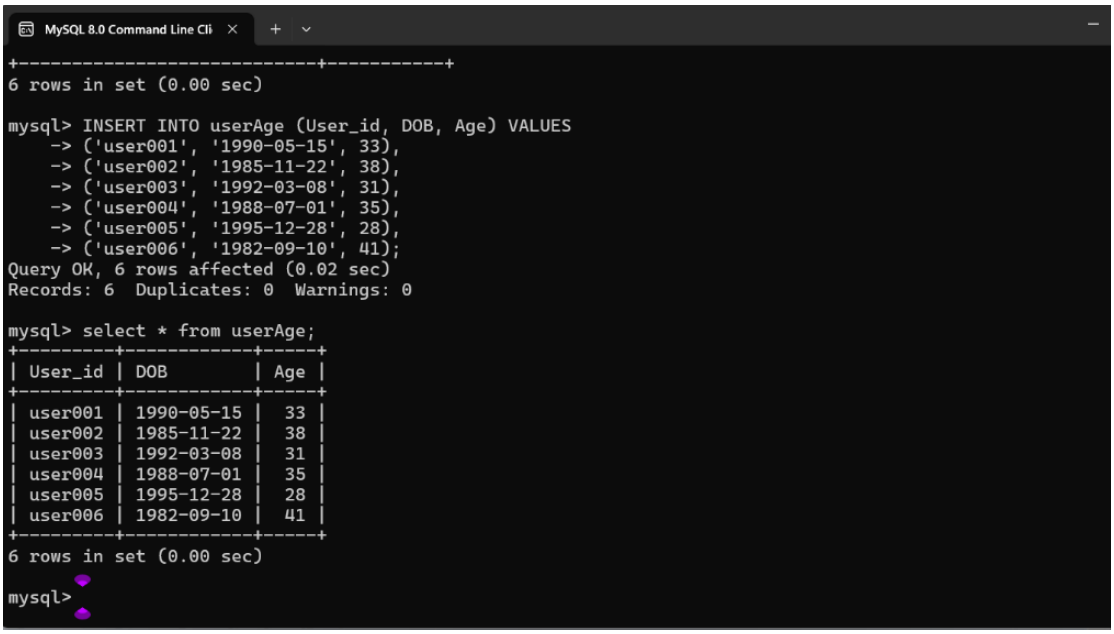


Figure 11: Inserting data into the userAge table

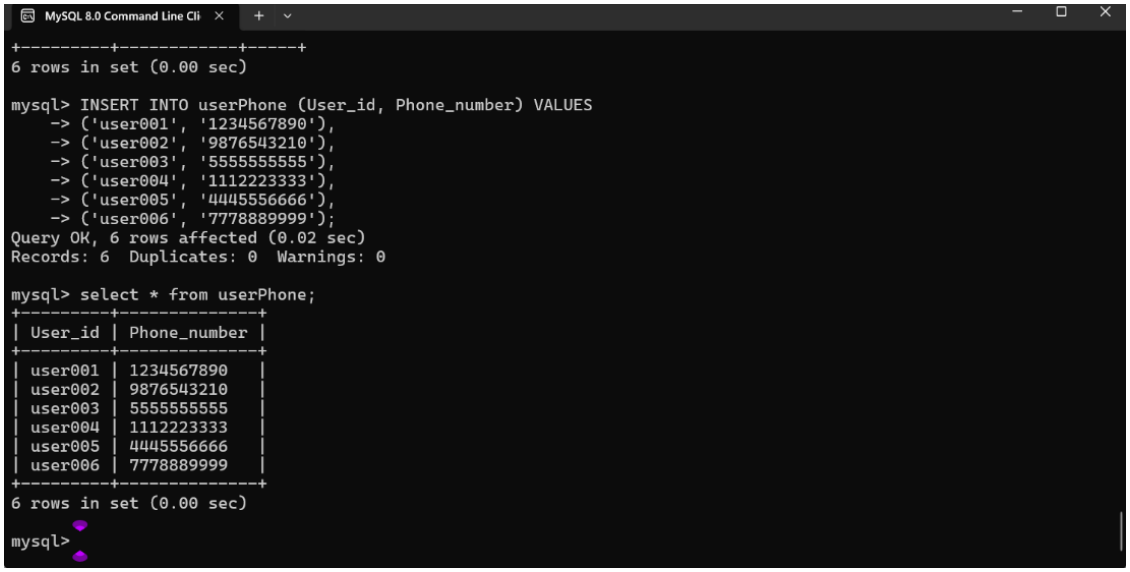


Figure 12: Inserting data into the userPhone table

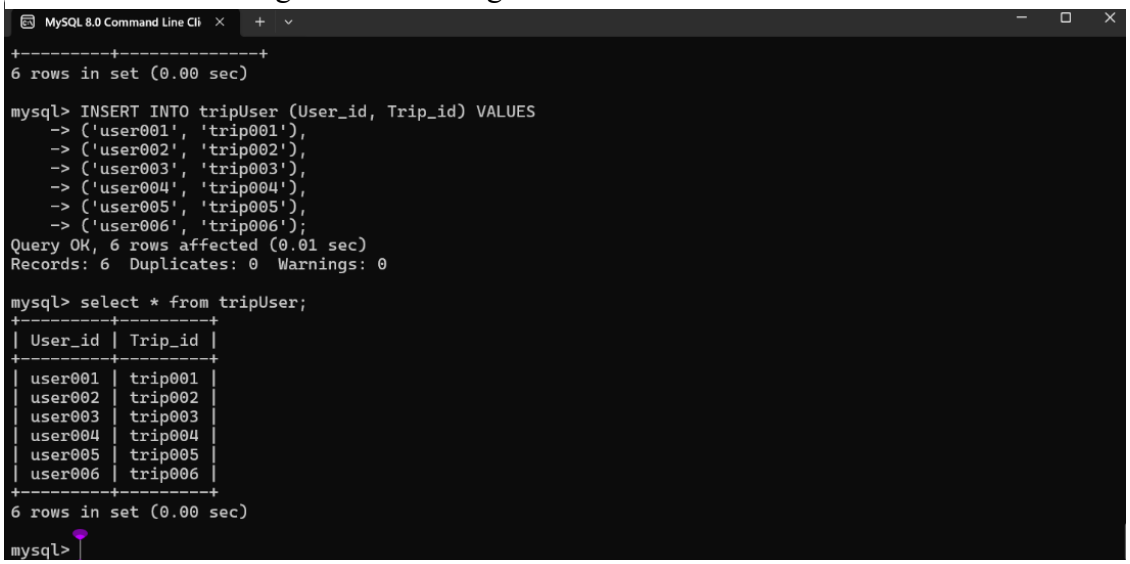
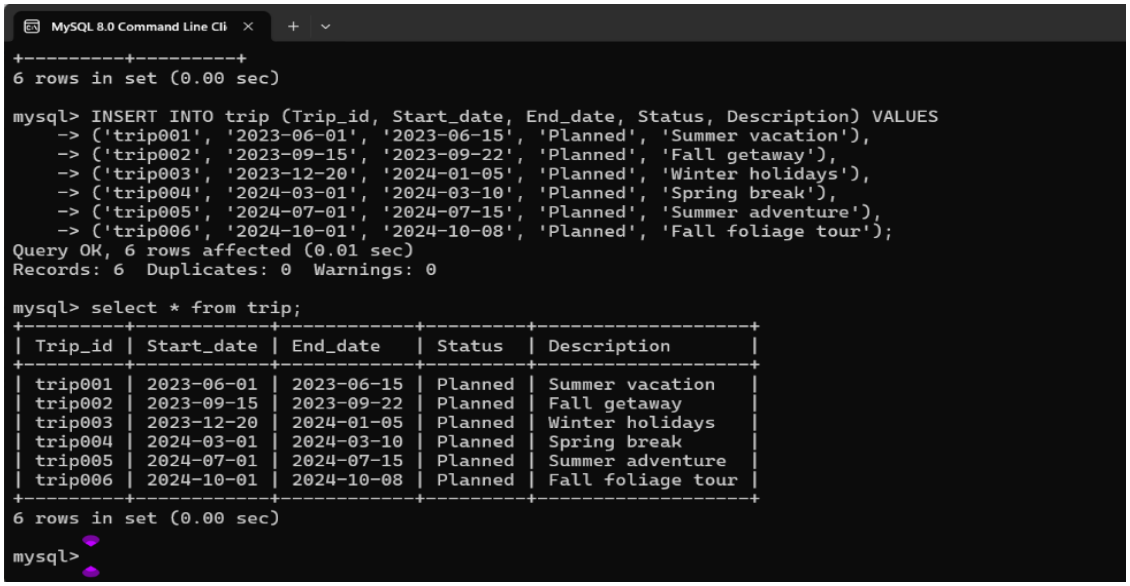


Figure 13: Inserting data into the tripUser table



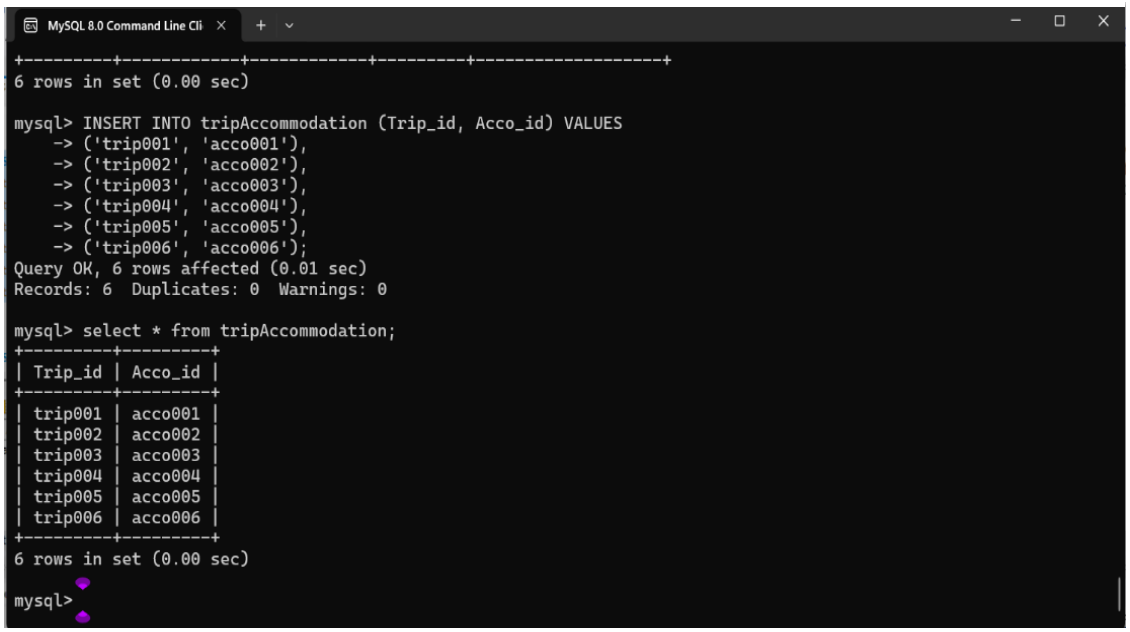
```
MySQL 8.0 Command Line Cli
+---------+---------+
6 rows in set (0.00 sec)

mysql> INSERT INTO trip (Trip_id, Start_date, End_date, Status, Description) VALUES
    -> ('trip001', '2023-06-01', '2023-06-15', 'Planned', 'Summer vacation'),
    -> ('trip002', '2023-09-15', '2023-09-22', 'Planned', 'Fall getaway'),
    -> ('trip003', '2023-12-20', '2024-01-05', 'Planned', 'Winter holidays'),
    -> ('trip004', '2024-03-01', '2024-03-10', 'Planned', 'Spring break'),
    -> ('trip005', '2024-07-01', '2024-07-15', 'Planned', 'Summer adventure'),
    -> ('trip006', '2024-10-01', '2024-10-08', 'Planned', 'Fall foliage tour');
Query OK, 6 rows affected (0.01 sec)
Records: 6  Duplicates: 0  Warnings: 0

mysql> select * from trip;
+---------+------------+------------+---------+-------------------+
| Trip_id | Start_date | End_date   | Status  | Description       |
+---------+------------+------------+---------+-------------------+
| trip001 | 2023-06-01 | 2023-06-15 | Planned | Summer vacation   |
| trip002 | 2023-09-15 | 2023-09-22 | Planned | Fall getaway      |
| trip003 | 2023-12-20 | 2024-01-05 | Planned | Winter holidays   |
| trip004 | 2024-03-01 | 2024-03-10 | Planned | Spring break      |
| trip005 | 2024-07-01 | 2024-07-15 | Planned | Summer adventure  |
| trip006 | 2024-10-01 | 2024-10-08 | Planned | Fall foliage tour |
+---------+------------+------------+---------+-------------------+
6 rows in set (0.00 sec)

mysql>
```

Figure 15: Inserting data into the trip table



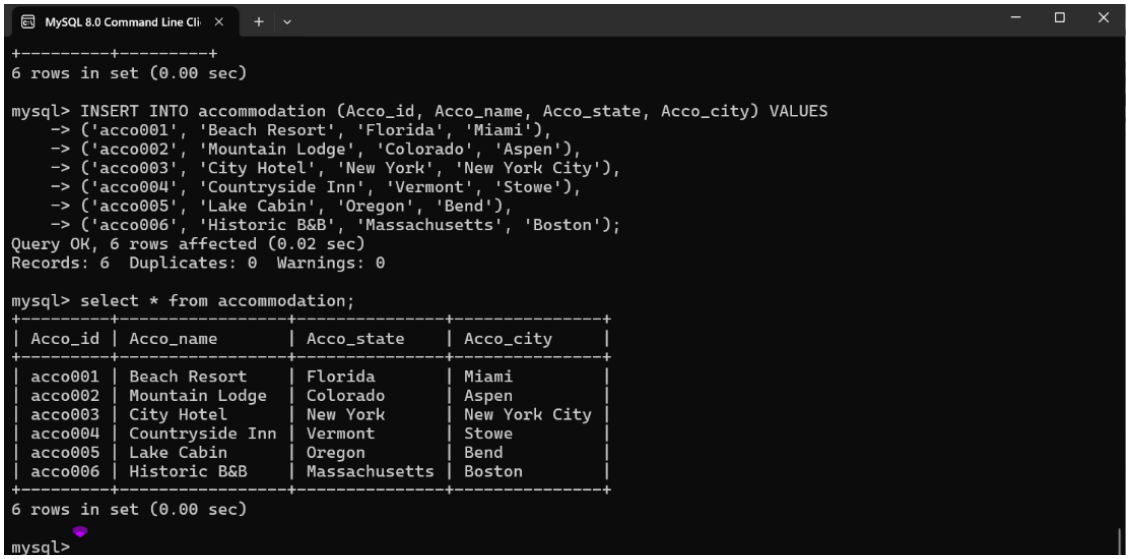
```
MySQL 8.0 Command Line Cli
+---------+------------+------------+---------+-------------------+
6 rows in set (0.00 sec)

mysql> INSERT INTO tripAccommodation (Trip_id, Acco_id) VALUES
    -> ('trip001', 'acco001'),
    -> ('trip002', 'acco002'),
    -> ('trip003', 'acco003'),
    -> ('trip004', 'acco004'),
    -> ('trip005', 'acco005'),
    -> ('trip006', 'acco006');
Query OK, 6 rows affected (0.01 sec)
Records: 6  Duplicates: 0  Warnings: 0

mysql> select * from tripAccommodation;
+---------+---------+
| Trip_id | Acco_id |
+---------+---------+
| trip001 | acco001 |
| trip002 | acco002 |
| trip003 | acco003 |
| trip004 | acco004 |
| trip005 | acco005 |
| trip006 | acco006 |
+---------+---------+
6 rows in set (0.00 sec)

mysql>
```

Figure 14: Inserting data into the tripAccommodation



```
MySQL 8.0 Command Line Cli
+---------+---------+
6 rows in set (0.00 sec)

mysql> INSERT INTO accommodation (Acco_id, Acco_name, Acco_state, Acco_city) VALUES
    -> ('acco001', 'Beach Resort', 'Florida', 'Miami'),
    -> ('acco002', 'Mountain Lodge', 'Colorado', 'Aspen'),
    -> ('acco003', 'City Hotel', 'New York', 'New York City'),
    -> ('acco004', 'Countryside Inn', 'Vermont', 'Stowe'),
    -> ('acco005', 'Lake Cabin', 'Oregon', 'Bend'),
    -> ('acco006', 'Historic B&B', 'Massachusetts', 'Boston');
Query OK, 6 rows affected (0.02 sec)
Records: 6  Duplicates: 0  Warnings: 0

mysql> select * from accommodation;
+---------+----------------+---------------+---------------+
| Acco_id | Acco_name      | Acco_state    | Acco_city     |
+---------+----------------+---------------+---------------+
| acco001 | Beach Resort   | Florida       | Miami         |
| acco002 | Mountain Lodge | Colorado      | Aspen         |
| acco003 | City Hotel     | New York      | New York City |
| acco004 | Countryside Inn | Vermont      | Stowe         |
| acco005 | Lake Cabin     | Oregon        | Bend          |
| acco006 | Historic B&B   | Massachusetts | Boston        |
+---------+----------------+---------------+---------------+
6 rows in set (0.00 sec)

mysql>
```

Figure 16: Inserting data into the accommodation

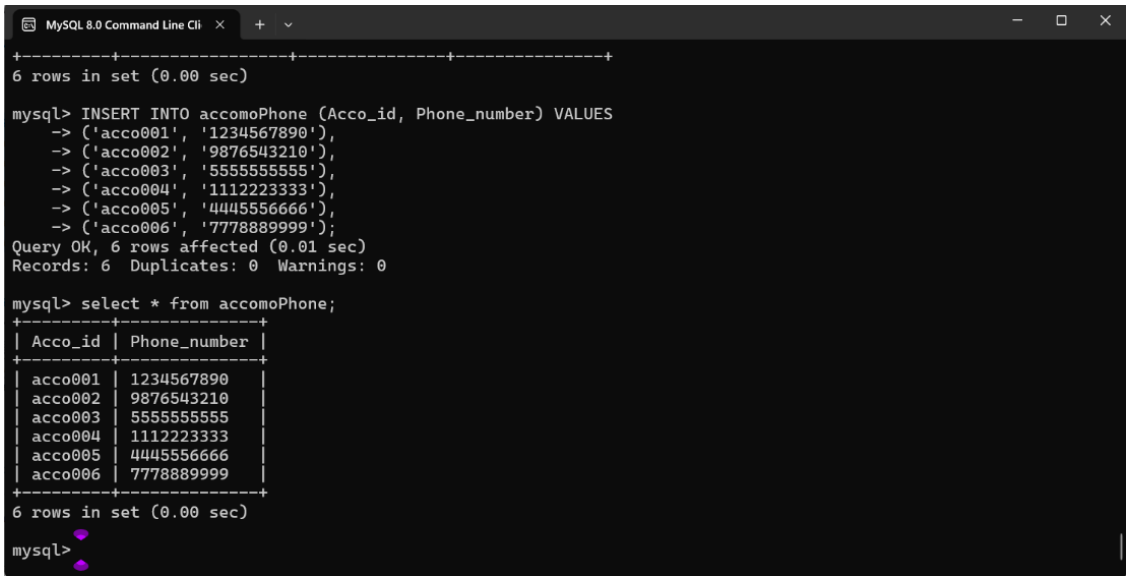


Figure 17: Inserting data into accomoPhone table

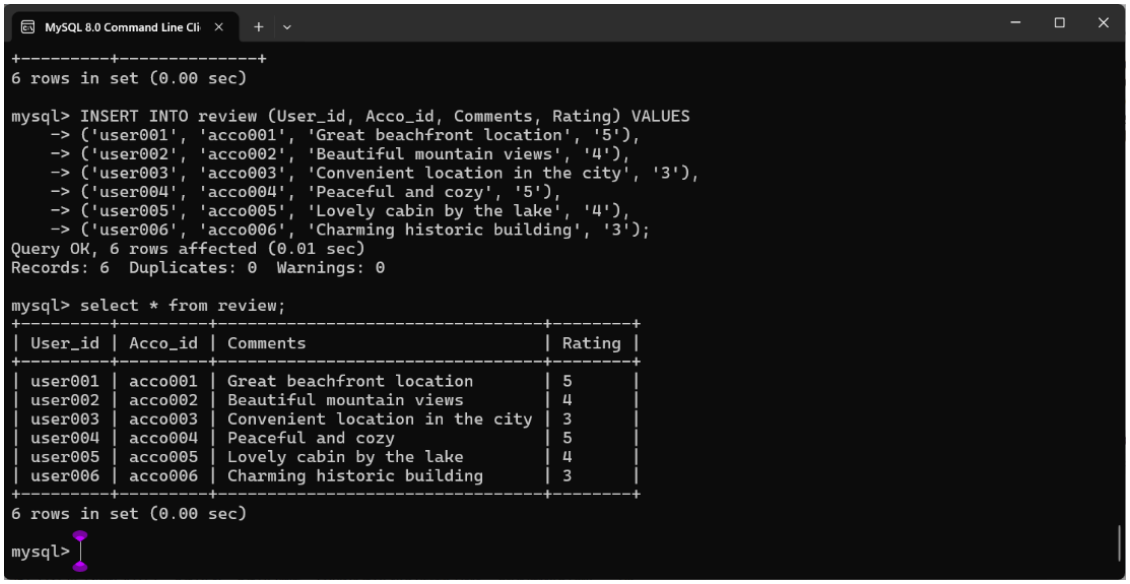


Figure 18: Inserting data into review table

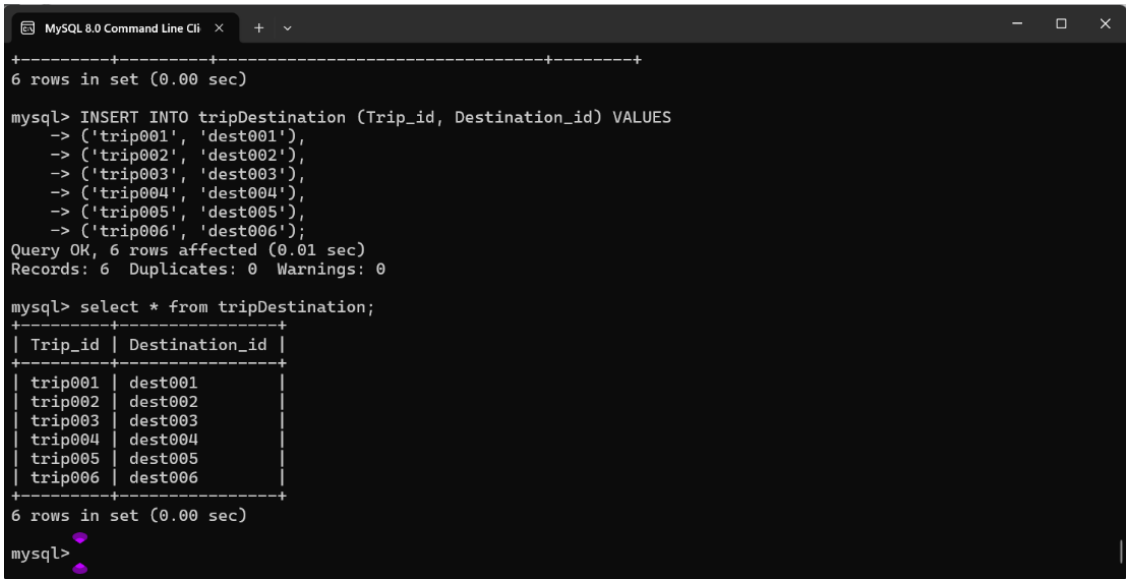


Figure 19: Inserting data into the tripDestination table

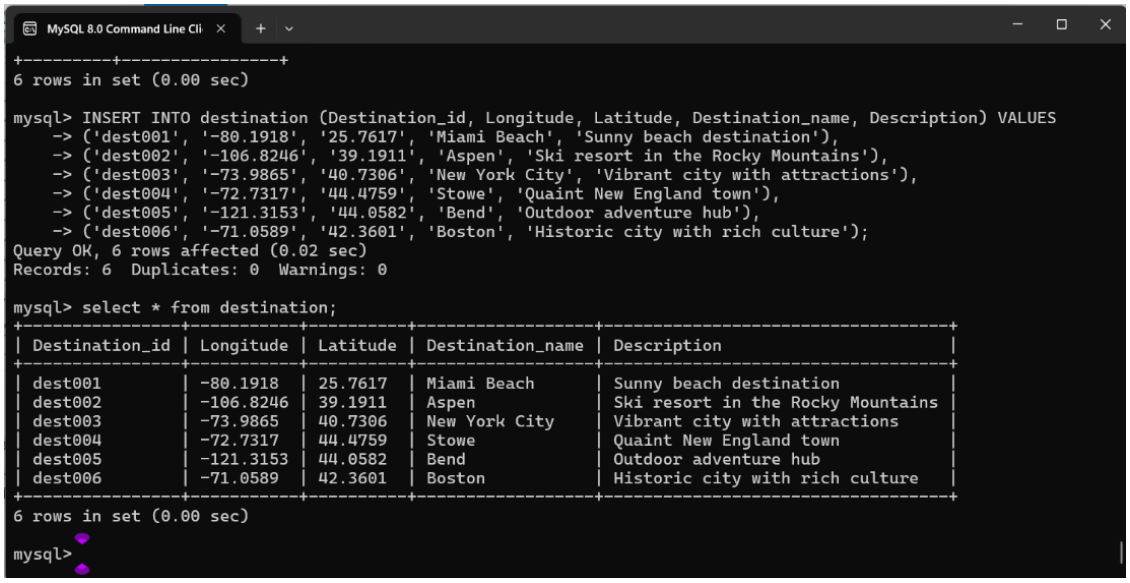


Figure 20: Inserting data into the destination table

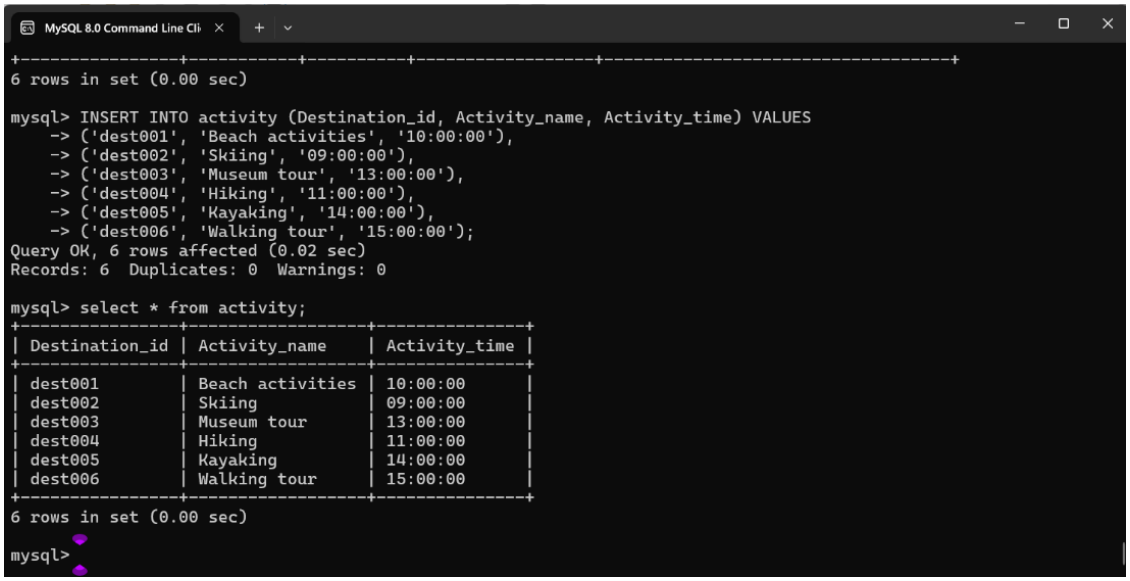


Figure 21: Inserting data into the activity table

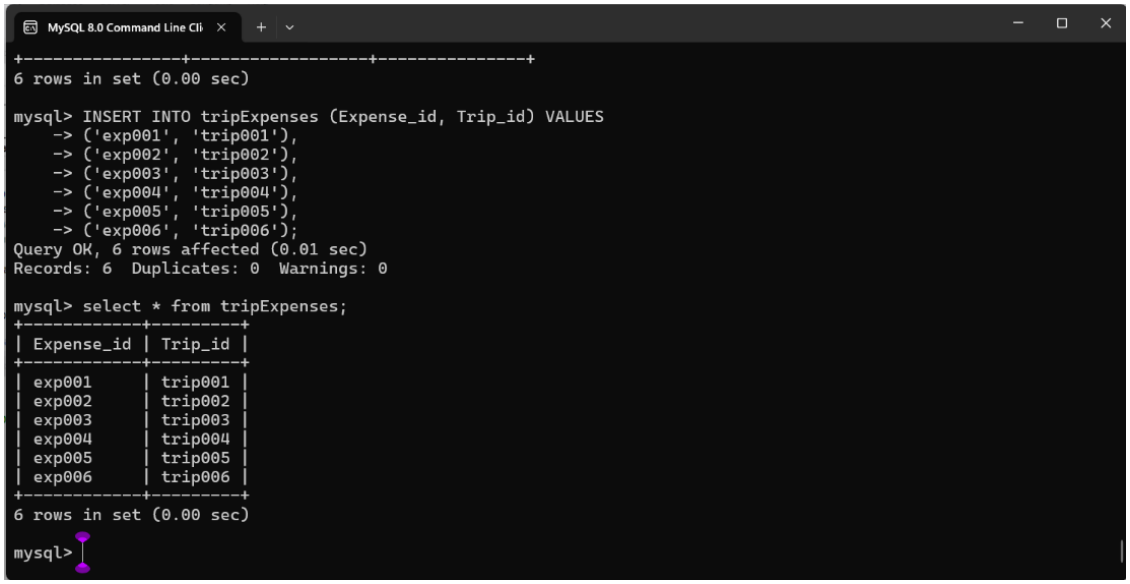


Figure 22: Inserting data into the tripExpenses table

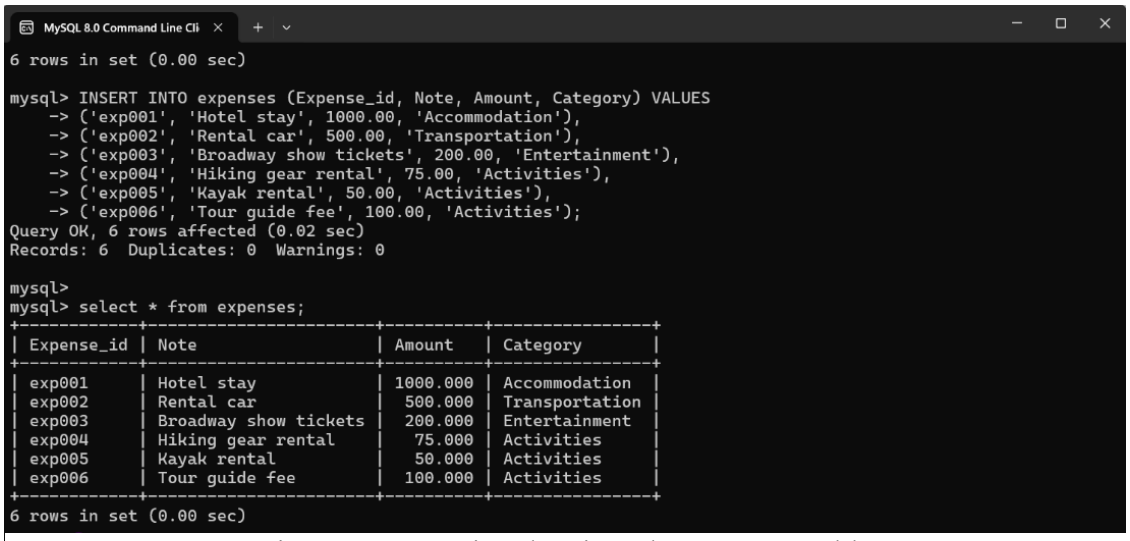


Figure 23: Inserting data into the expenses table

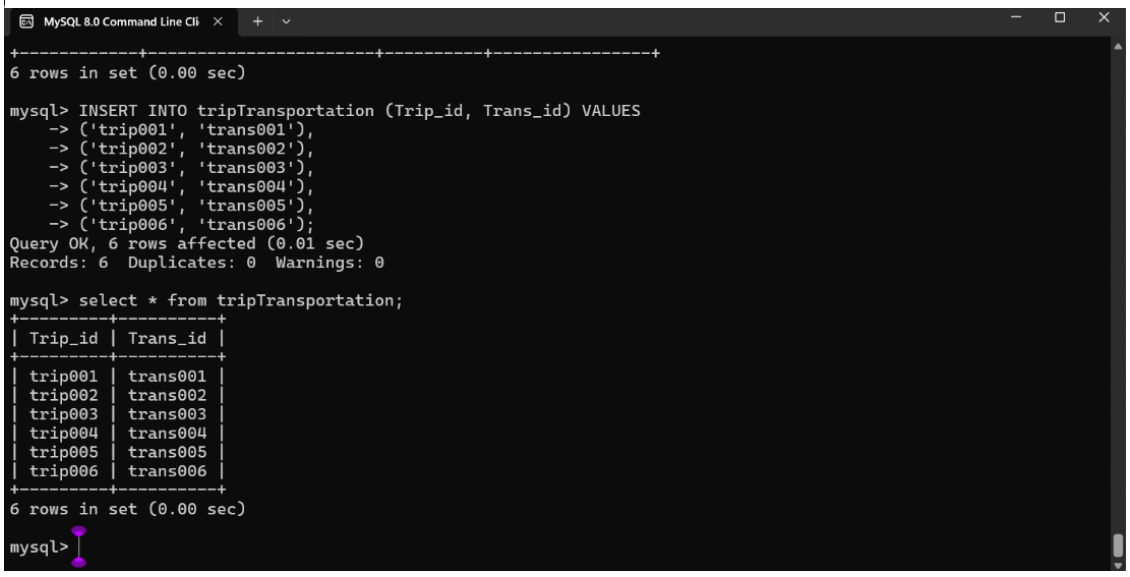


Figure 24: Inserting data into the tripTransportation table

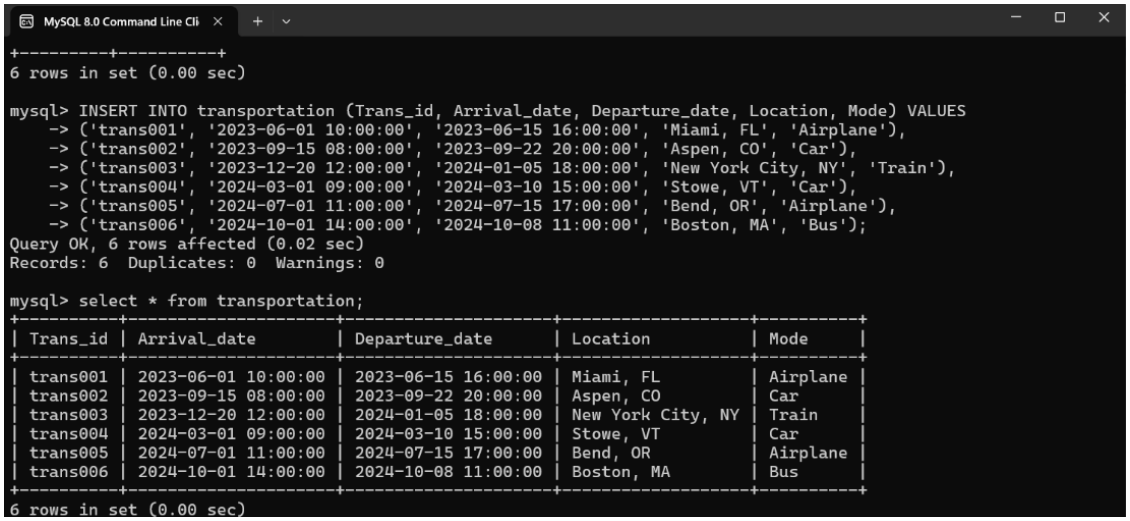


Figure 25: Inserting data into the transportation table

3.4 Update data in the tables

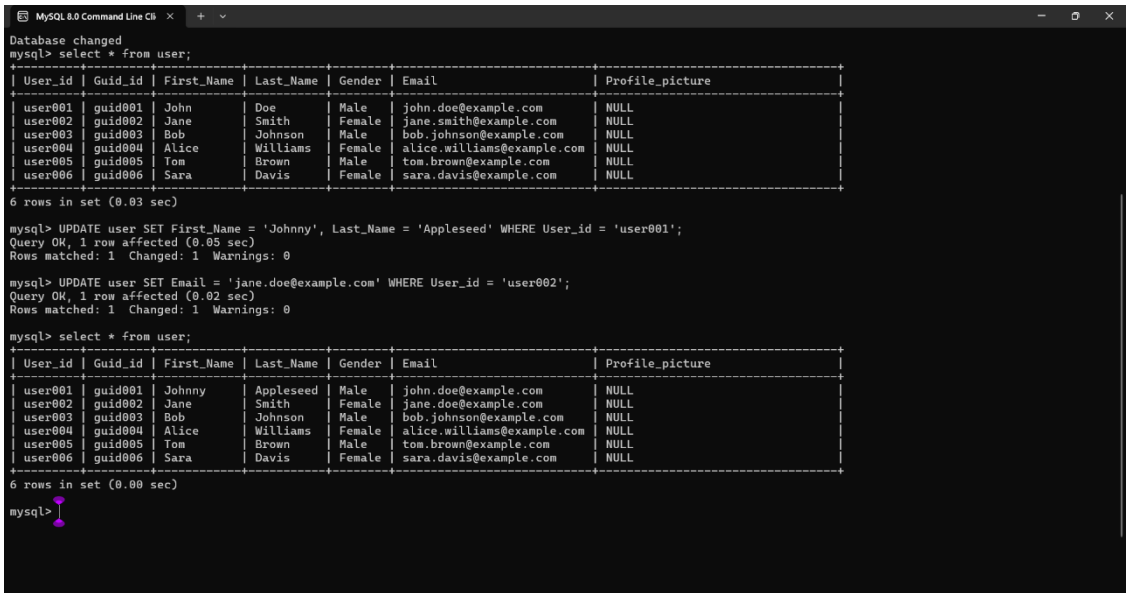


Figure 26: Before and After updating the user table

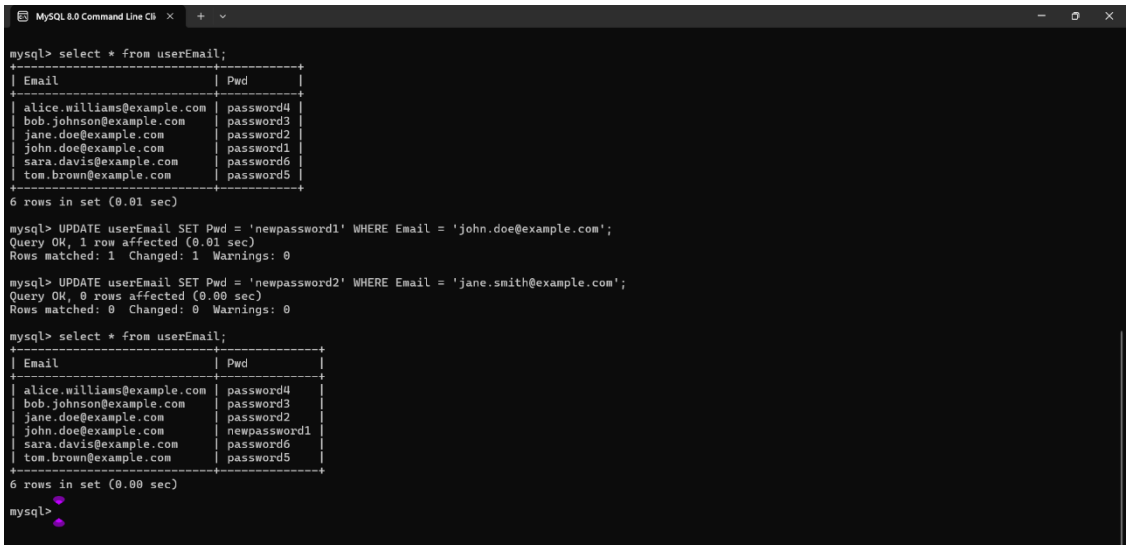


Figure 27: Before and After updating the userEmail table

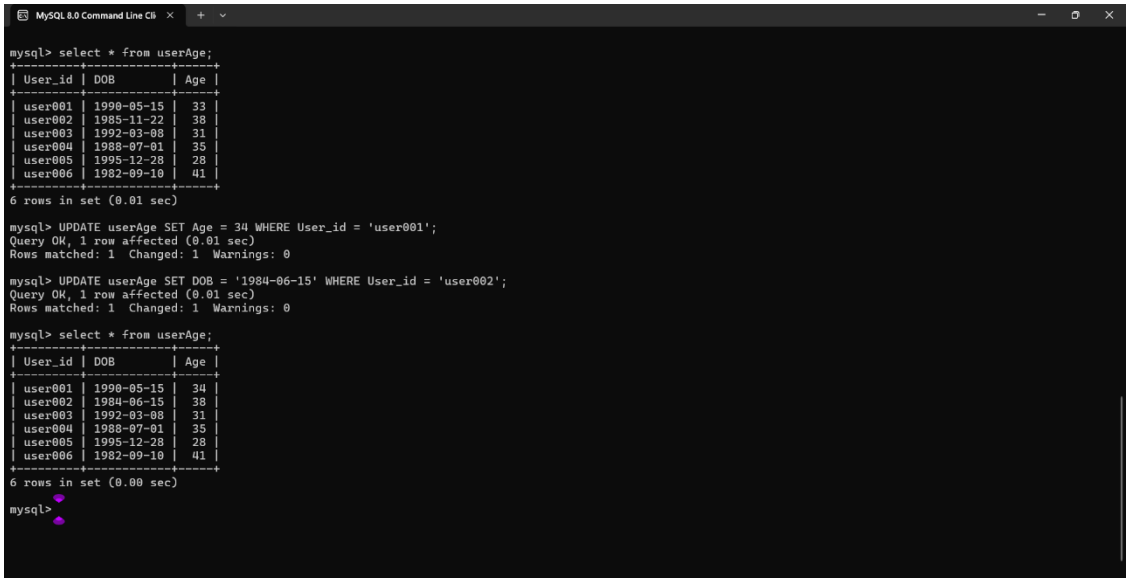


Figure 28: Before and After updating the userAge table

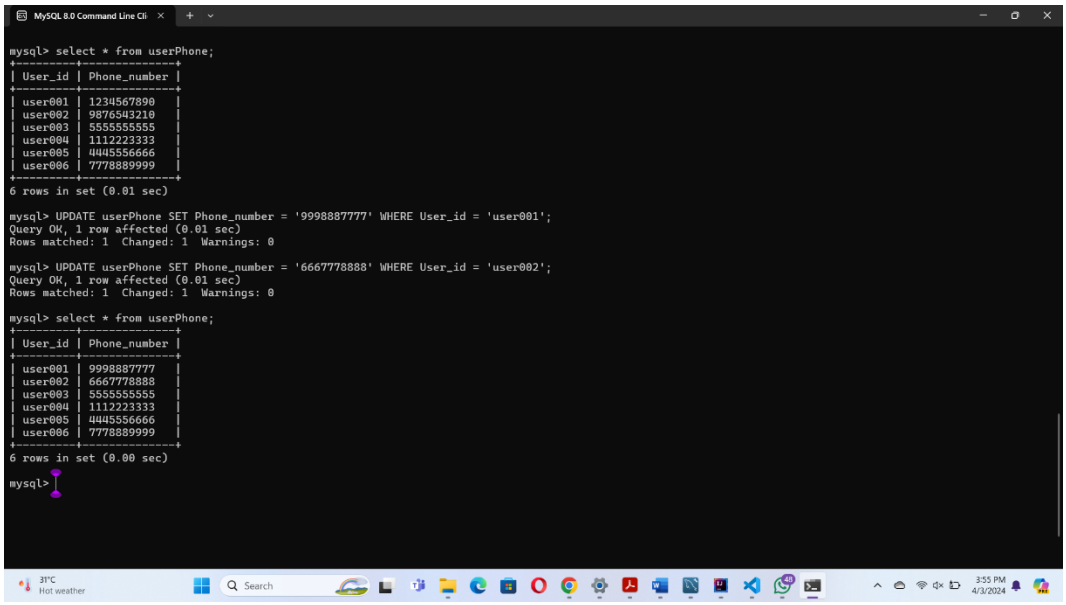


Figure 29: Before and After updating the userPhone table

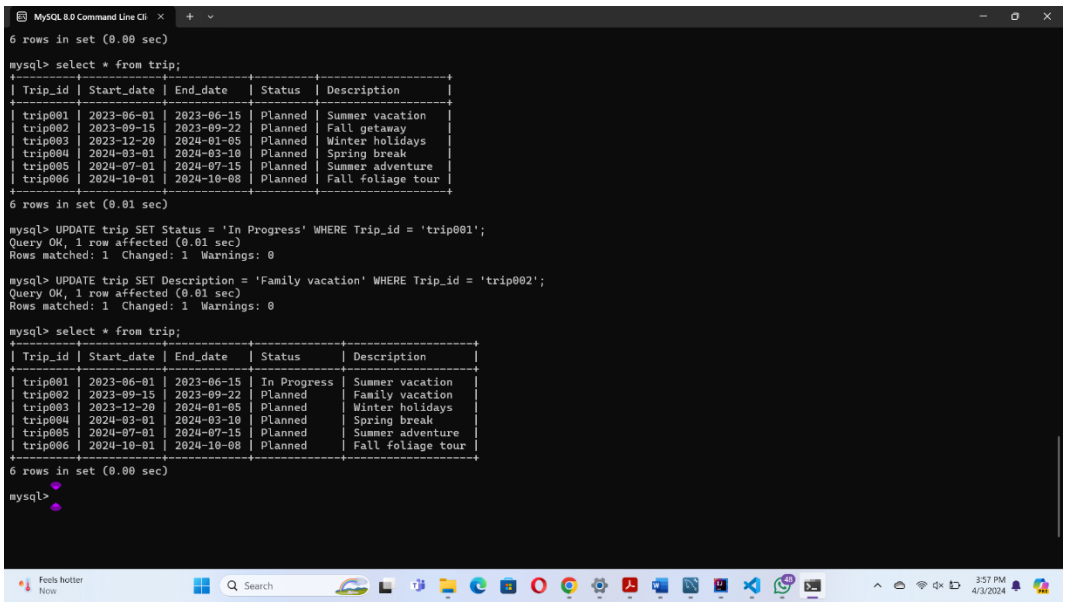


Figure 30: Before and After updating the trip table

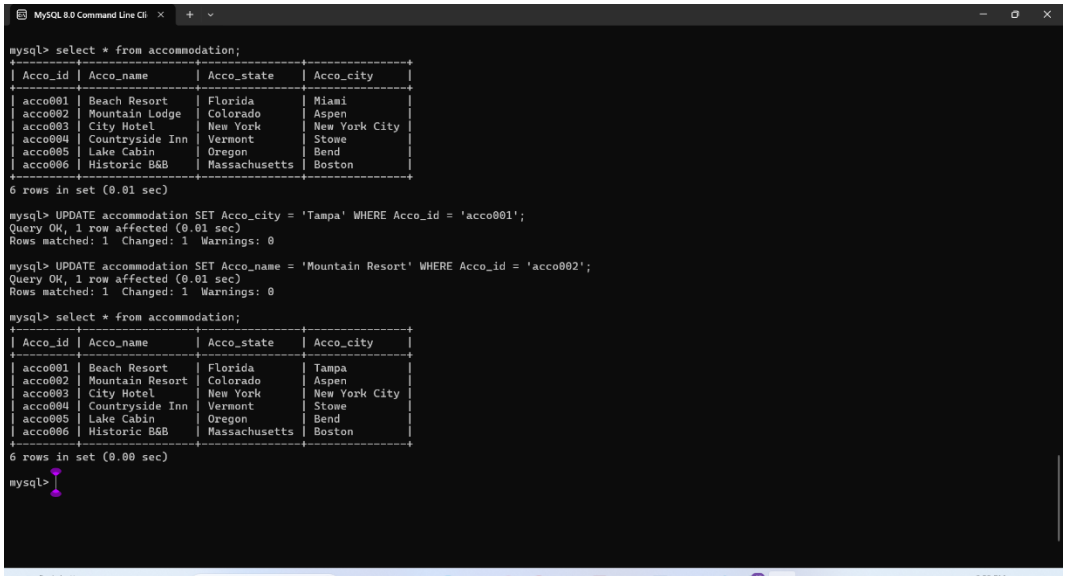


Figure 31: Before and After updating the accommodation table

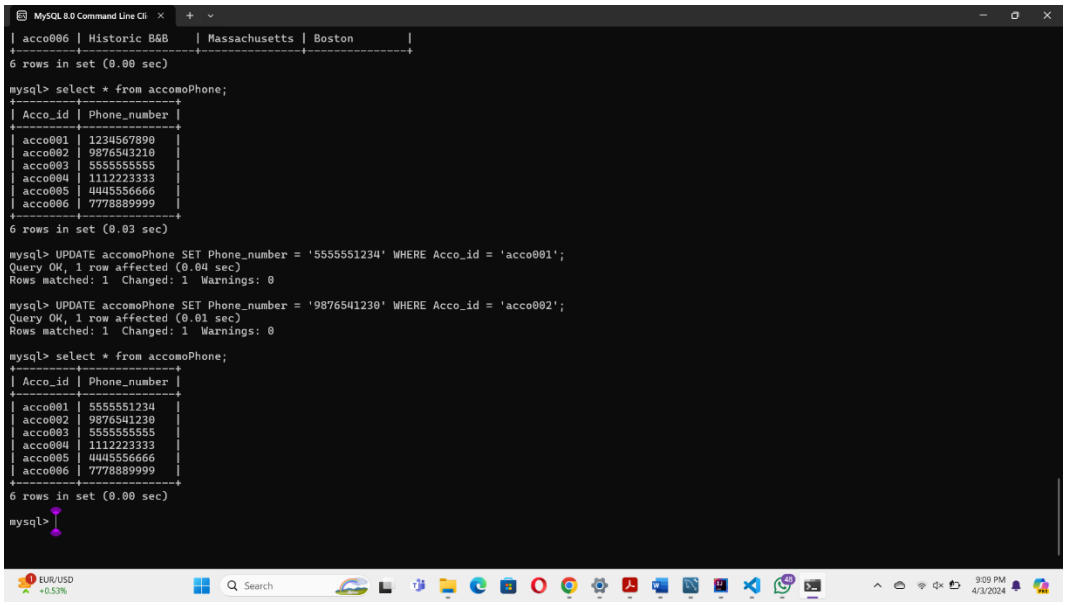


Figure 32: Before and After updating the accomoPhone table

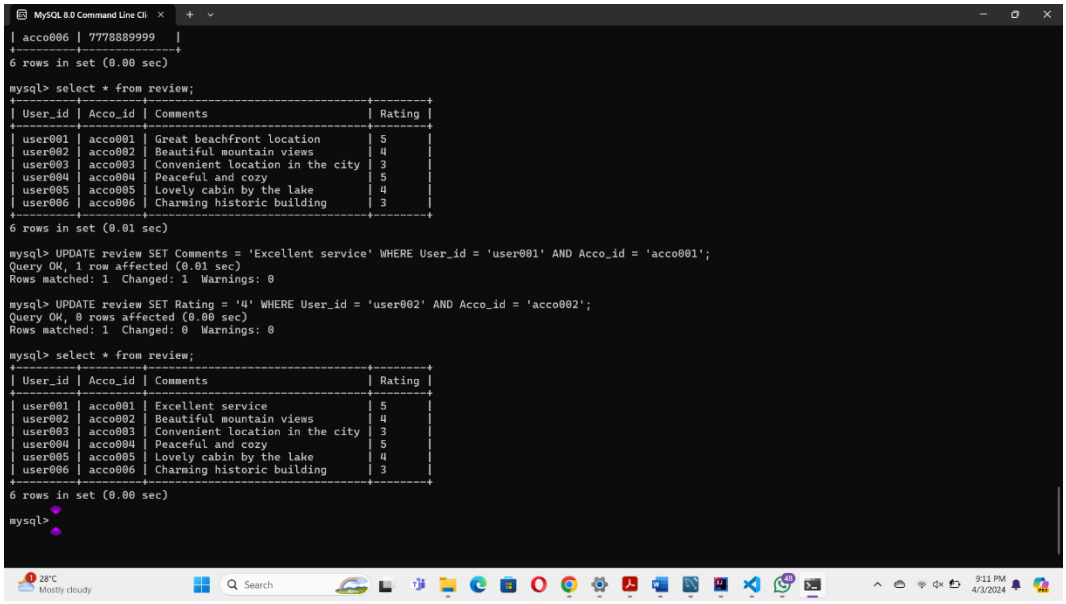


Figure 33: Before and After updating the review table

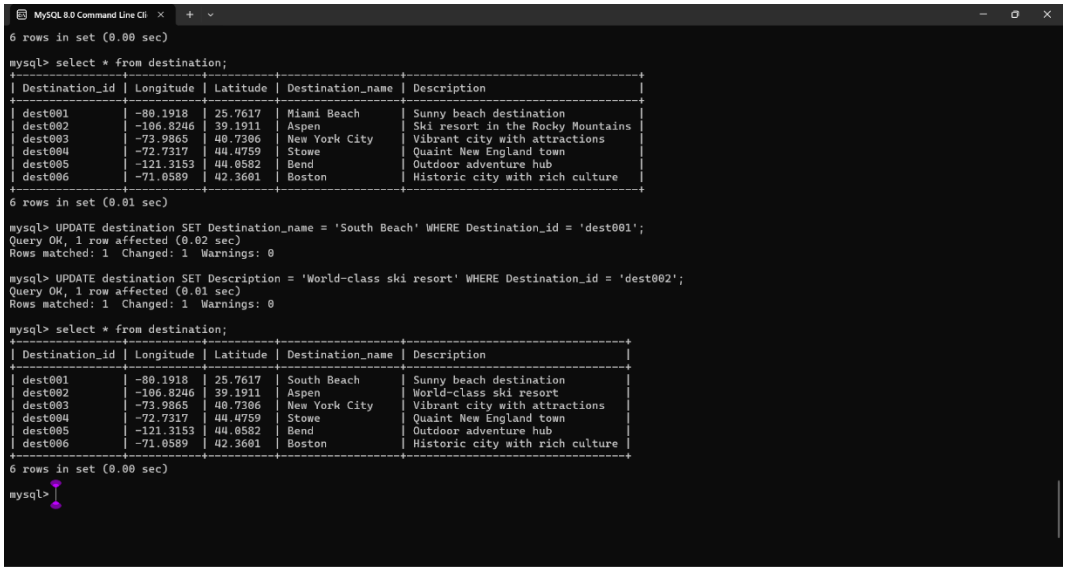


Figure 34: Before and After updating the destination table

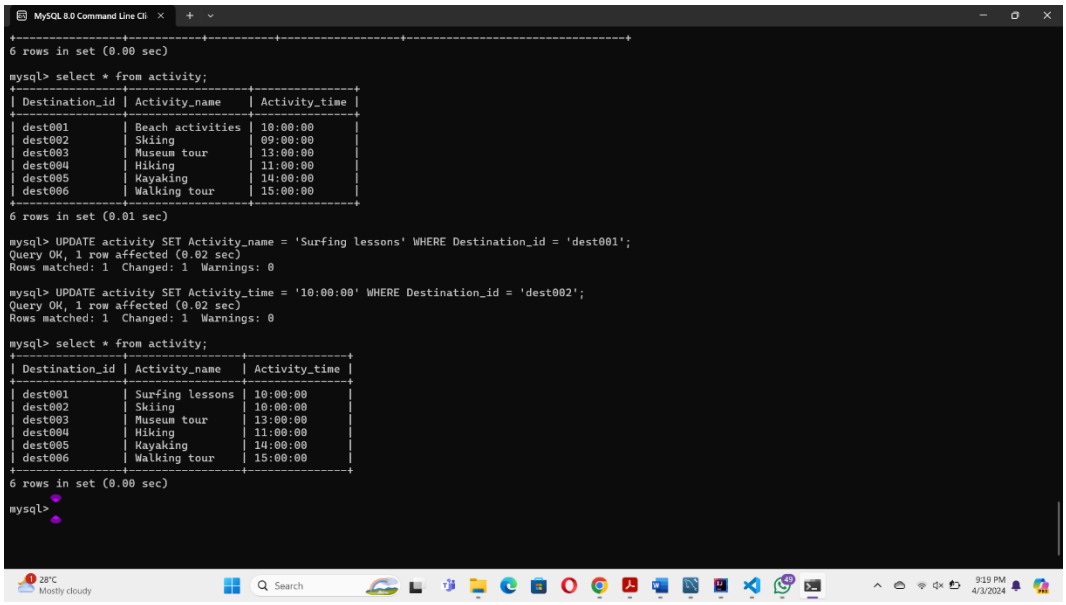


Figure 35: Before and After updating the activity table

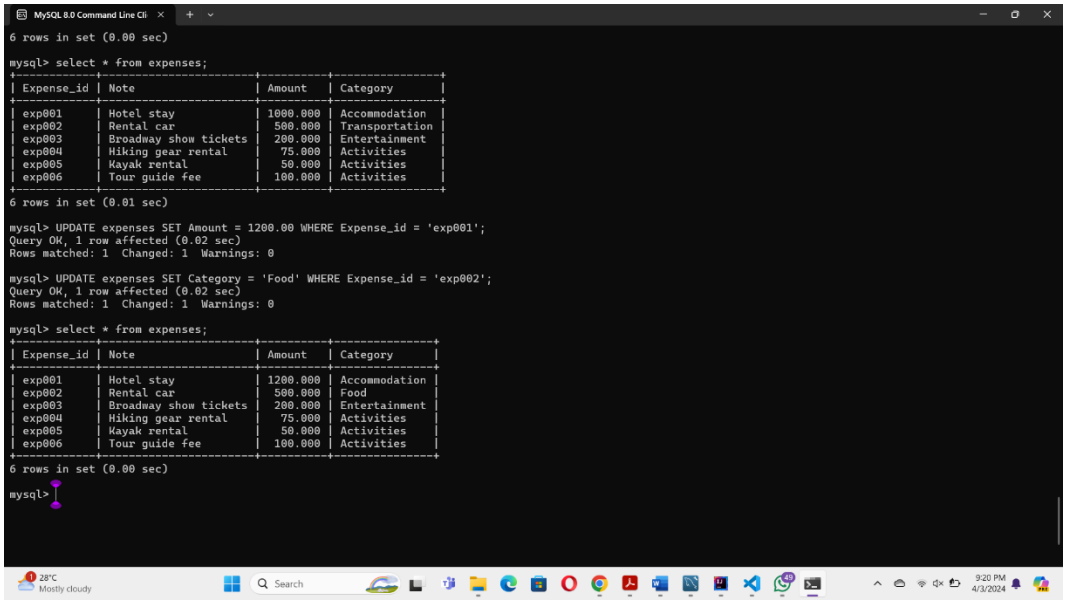


Figure 36: Before and After updating the accommodation table

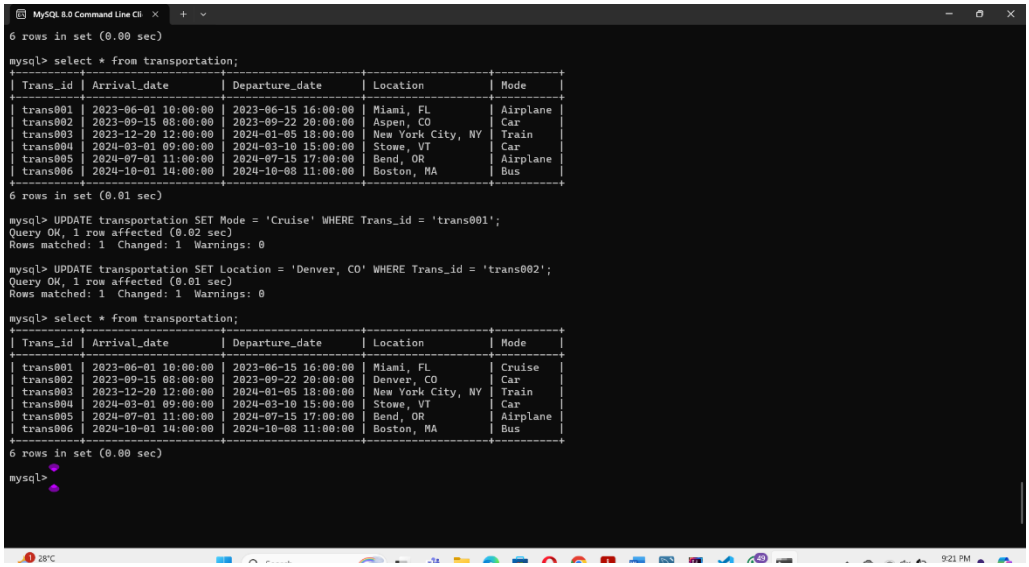


Figure 37: Before and After updating the transportation table

3.5 Delete data in the tables

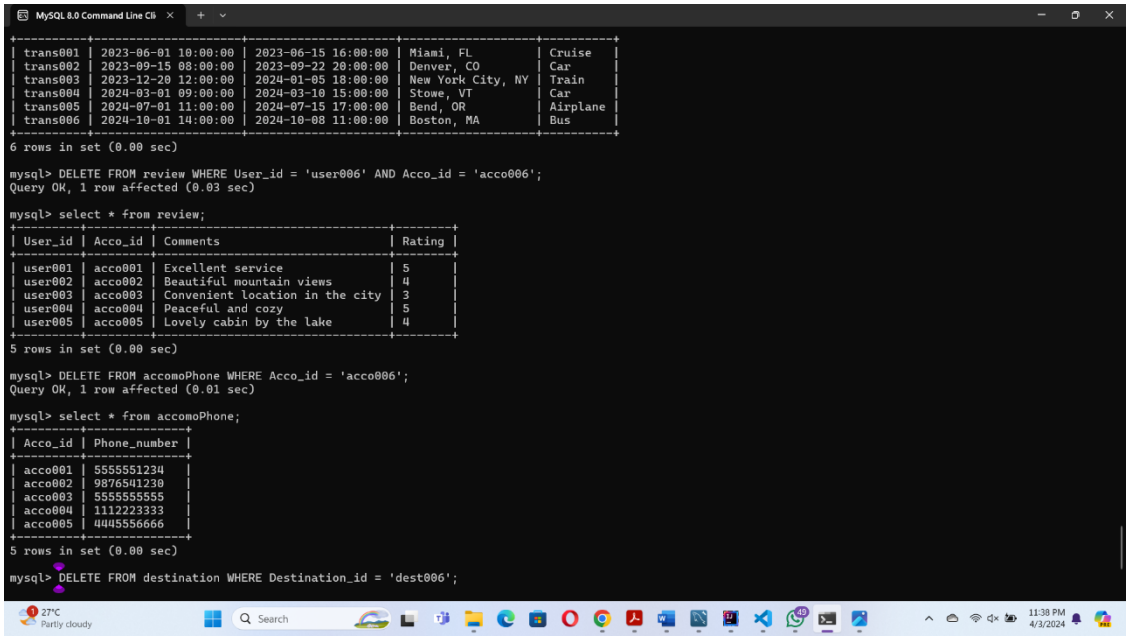


Figure 38: Deleting data from review and accomoPhone table

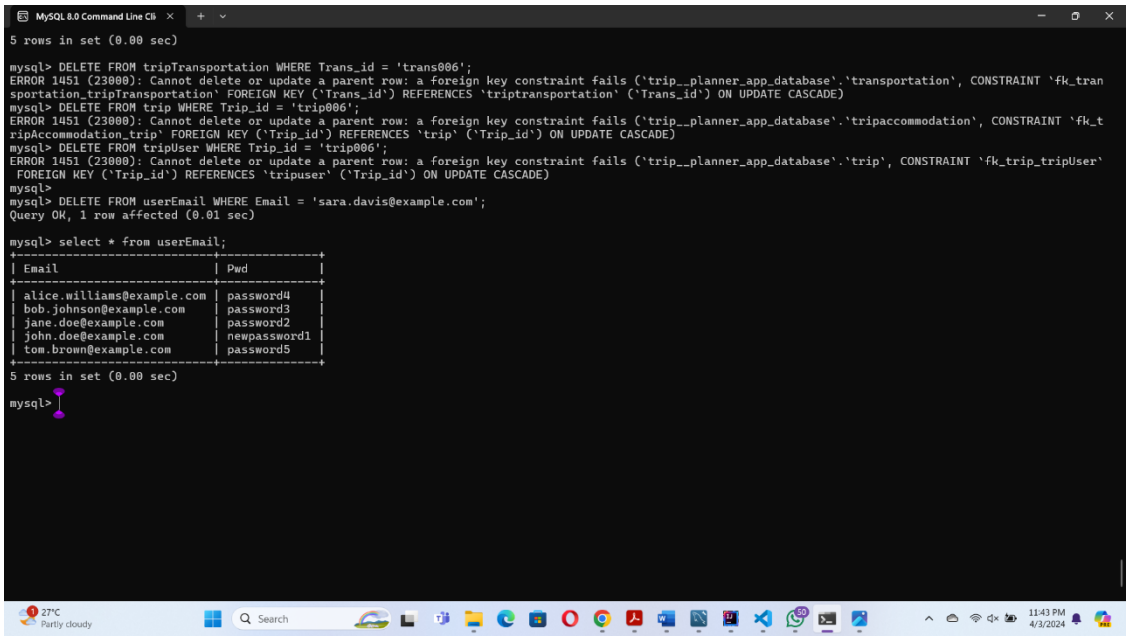


Figure 39: Deleting data from userEmail table

4 Chapter 04 – Transactions

4.1 Simple Queries

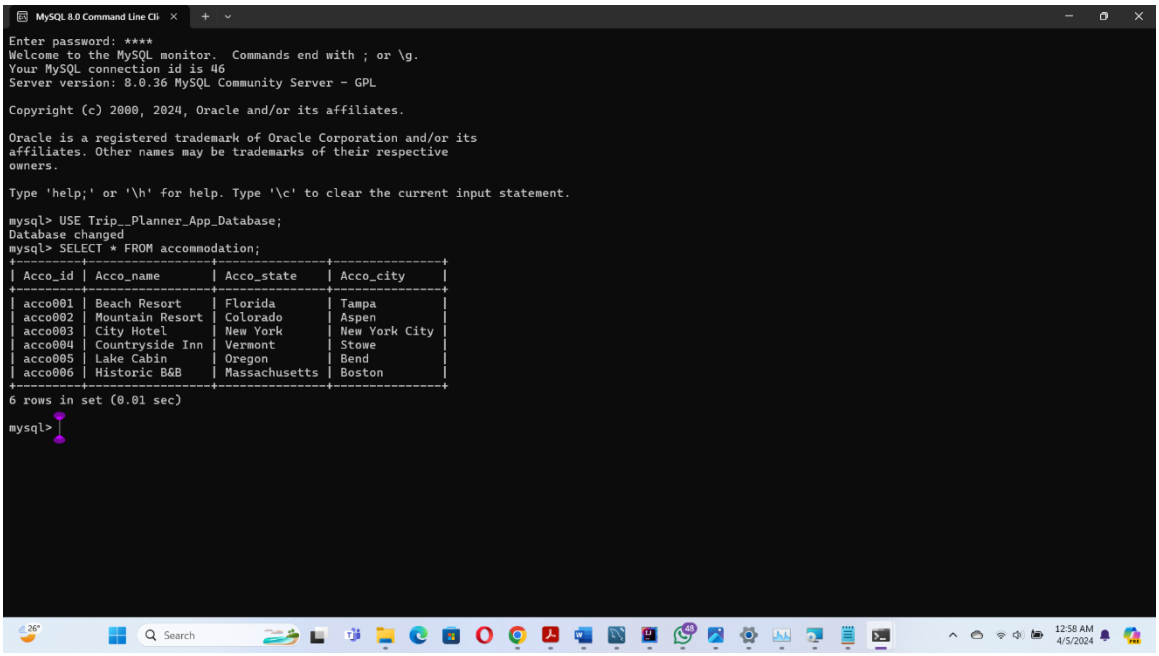


Figure 40: Simple Query demonstrating Select Operation

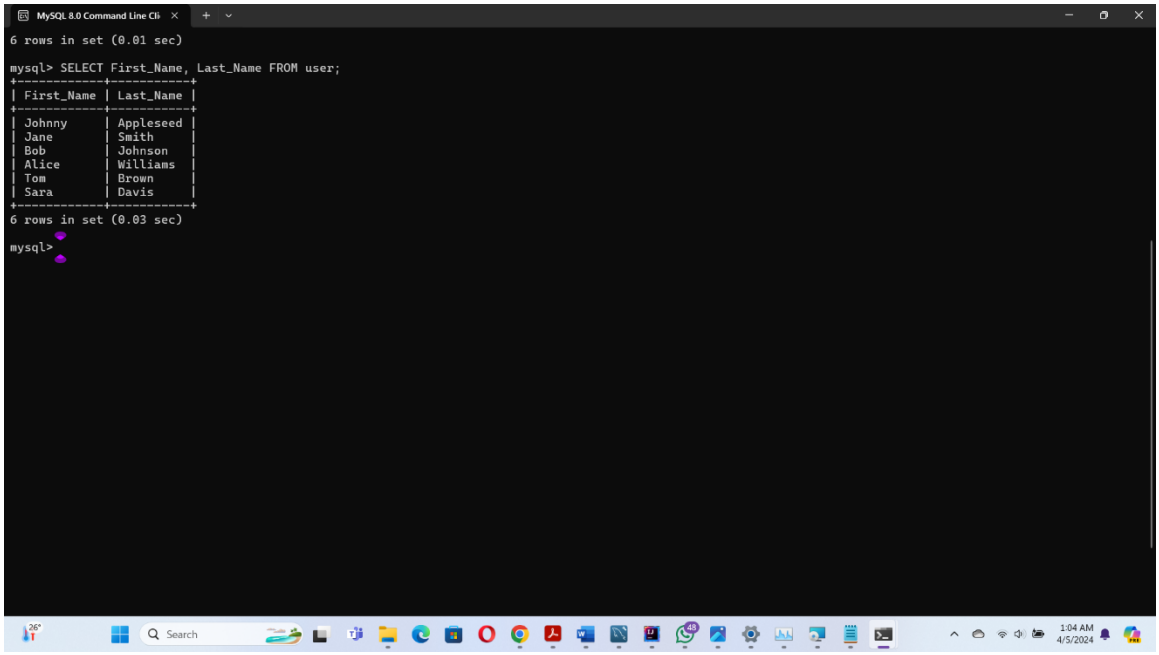


Figure 41: Simple Query demonstrating Select Operation

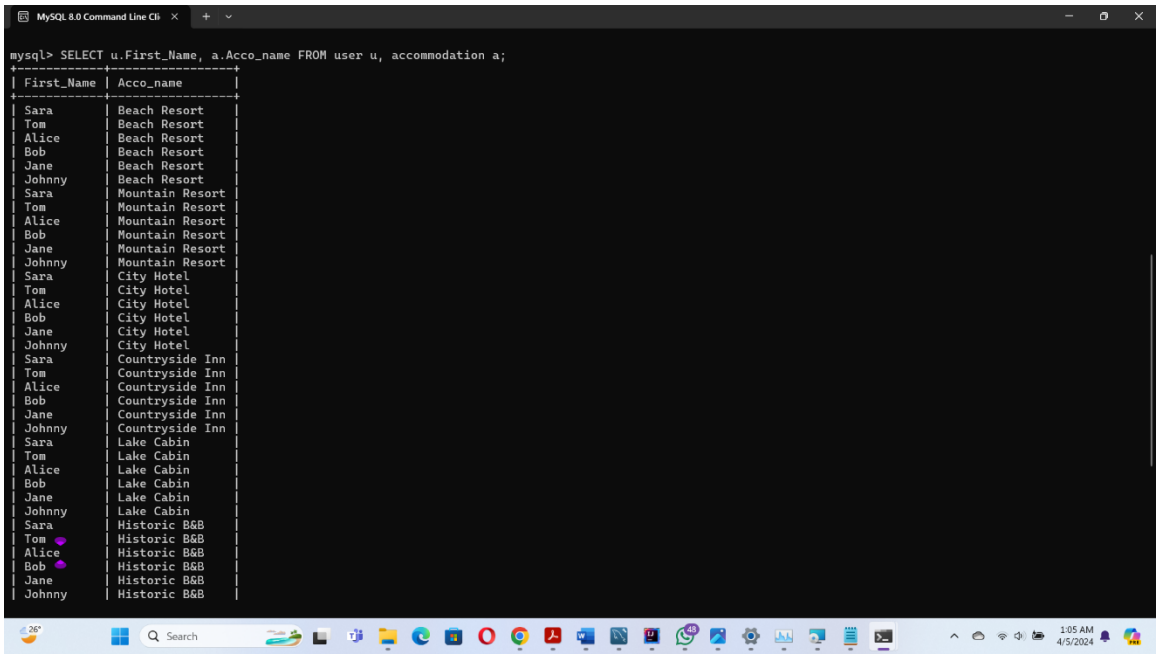


Figure 42: Simple Query demonstrating select operation

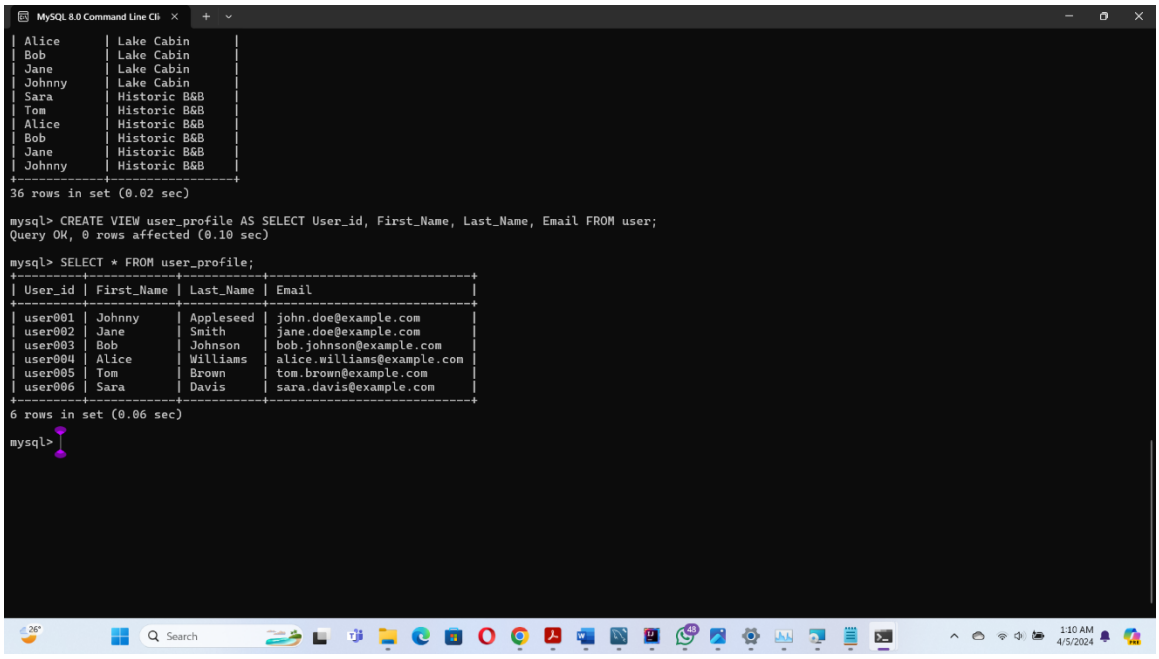


Figure 43: Simple Query to create a User View

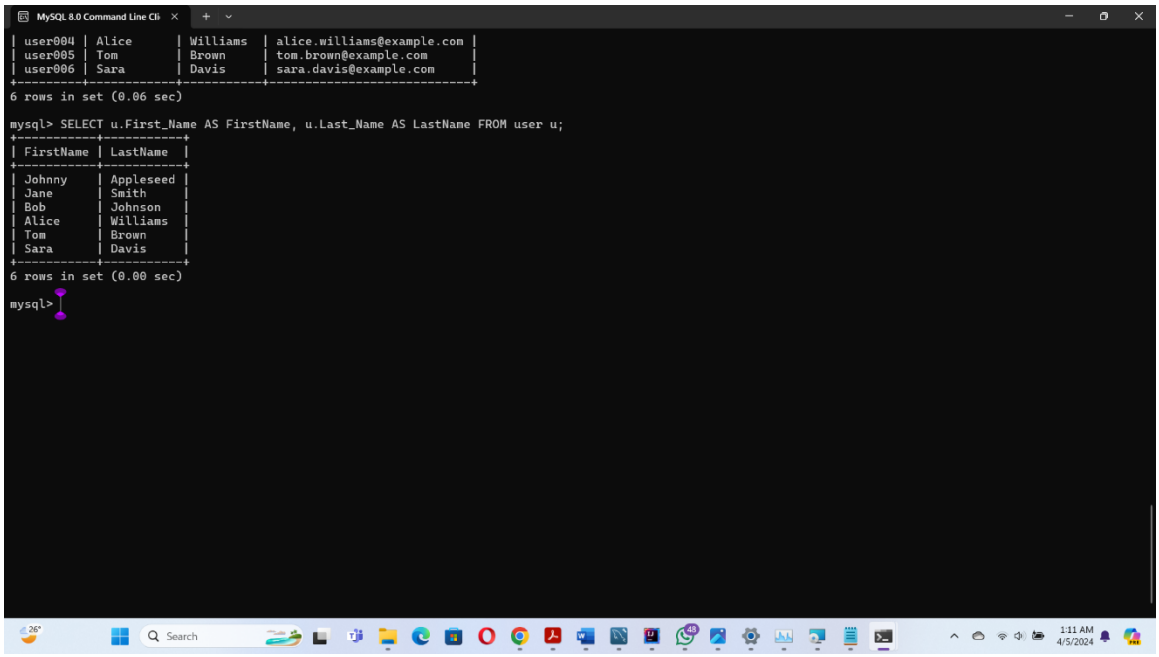


Figure 44: Simple query to retrieve data from a table (from user table)

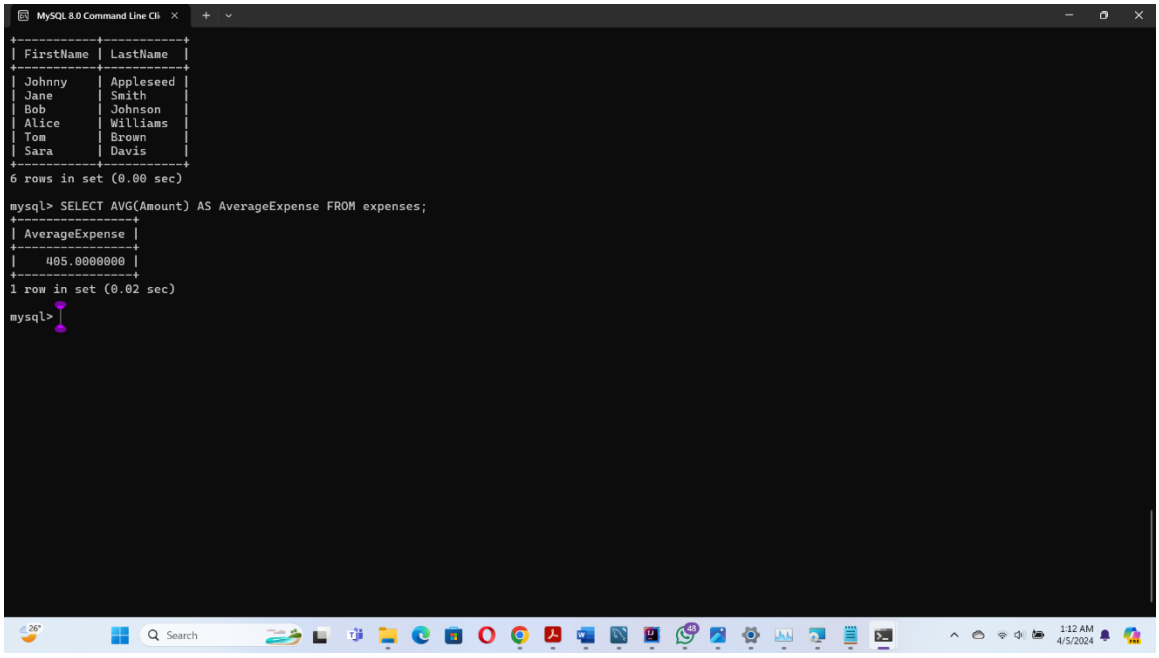


Figure 45: Simple Query demonstrating the use of an aggregation function(Average)

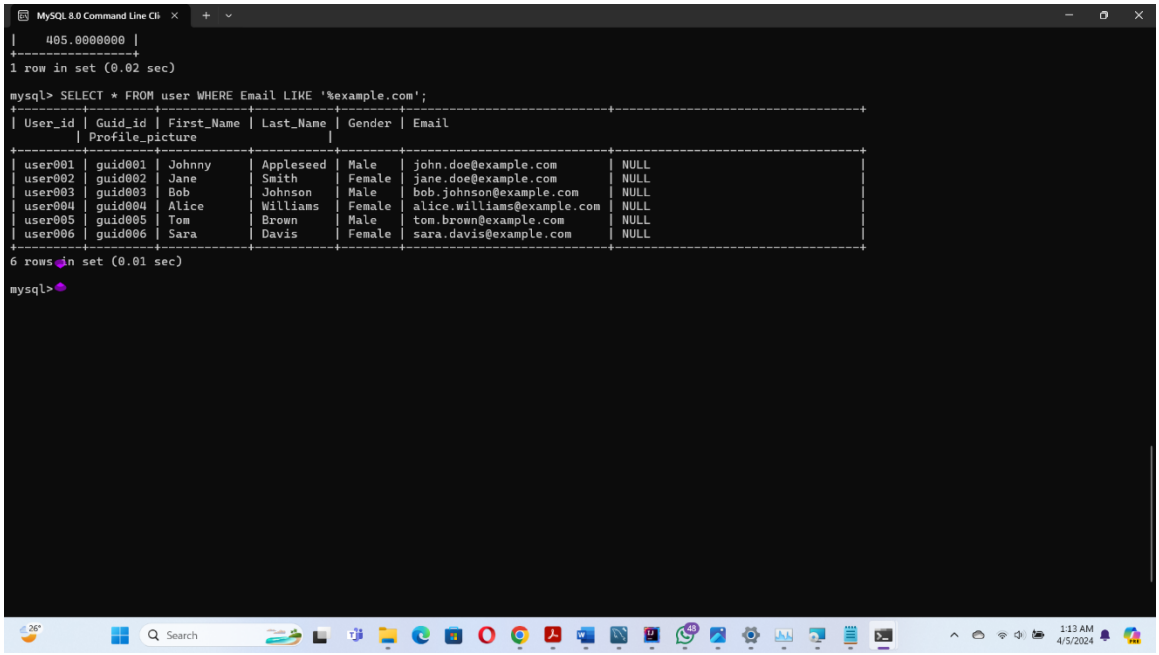


Figure 46: Simple Query to Demonstrate the use of Like keyword

4.2 Complex Queries

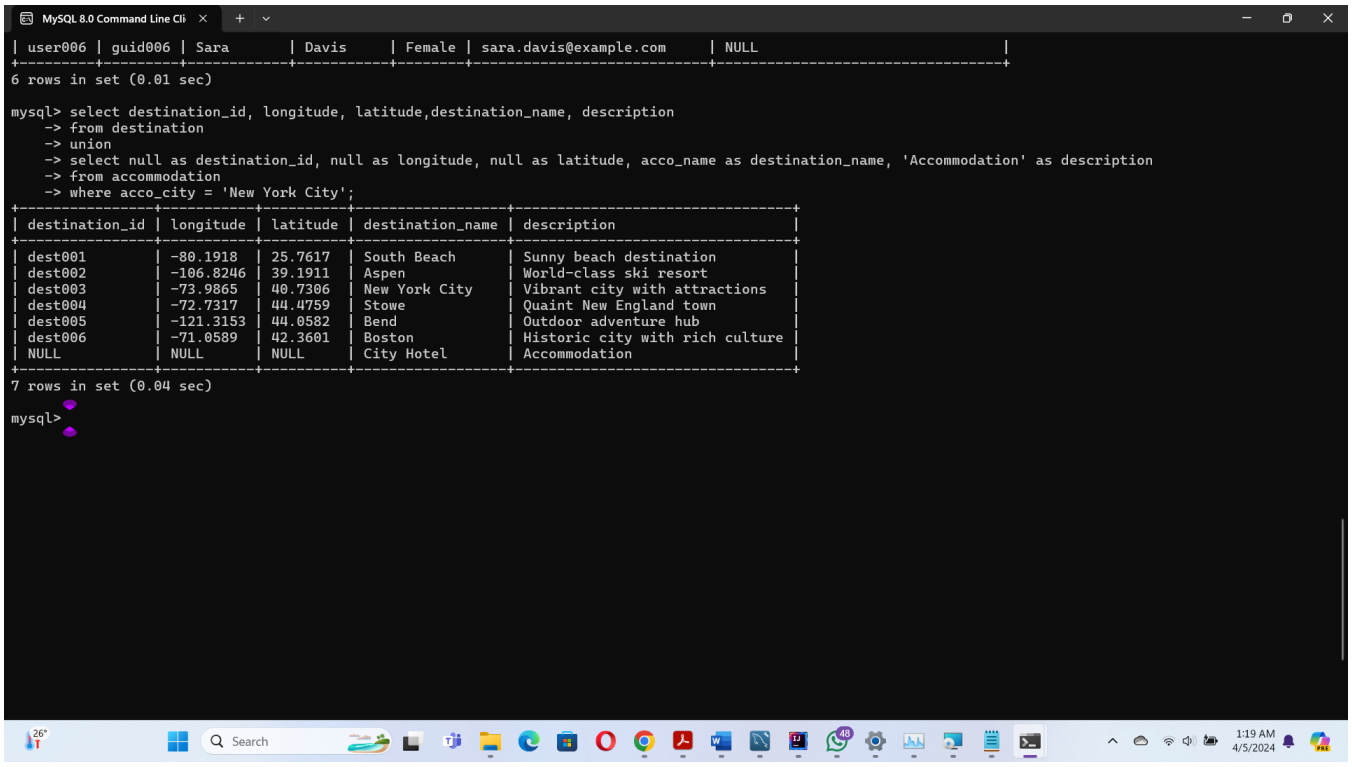


Figure 47: Complex Query 1 (Basic set operation Union)

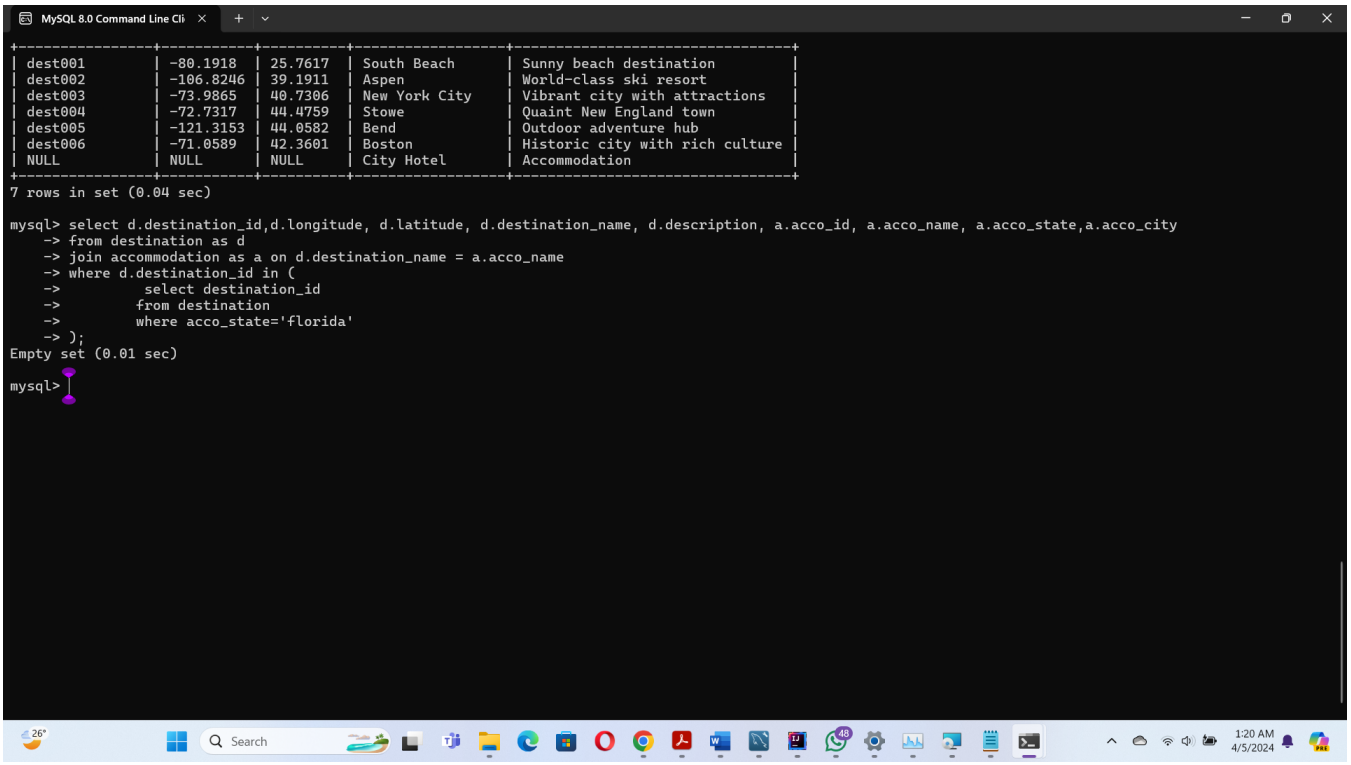


Figure 48: Complex Query 2 (Basic set operation Intersection)

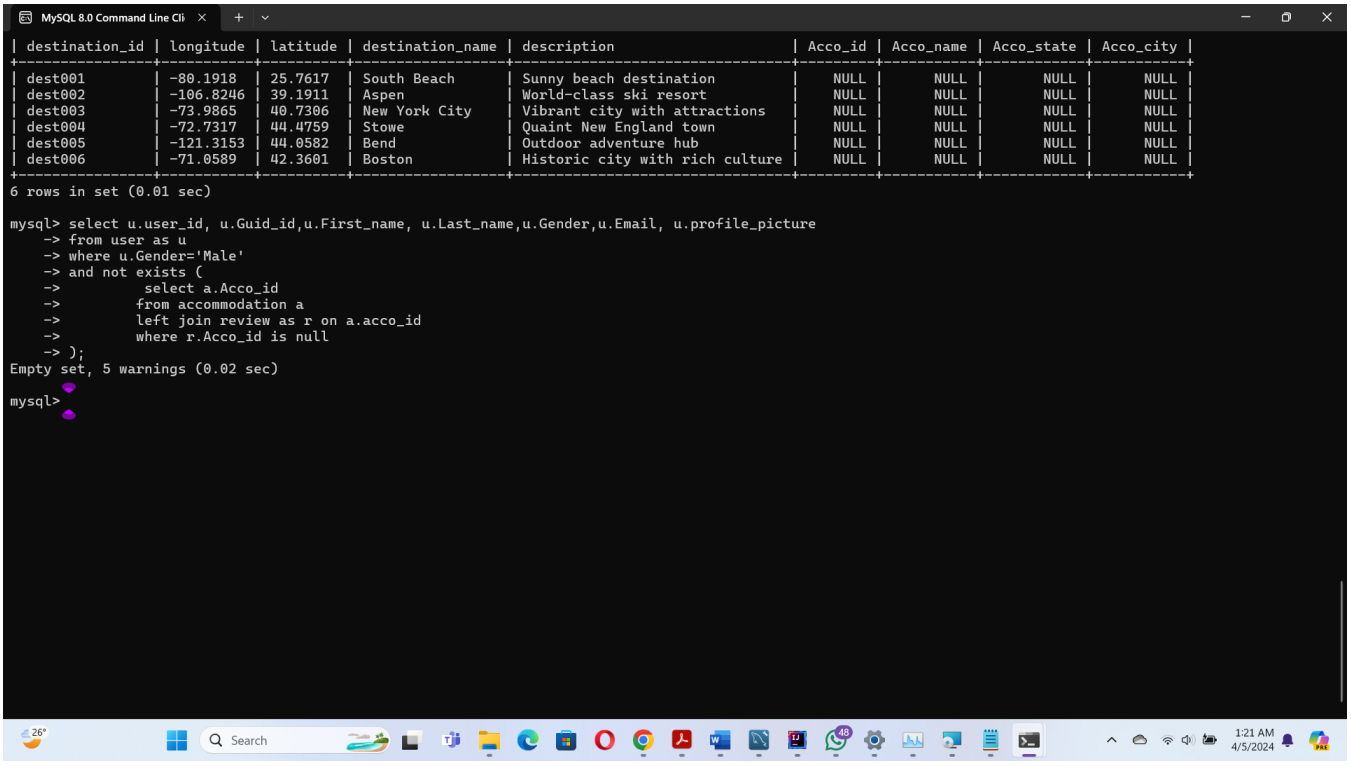


Figure 49: Complex Query 3 (Basic set operation division)

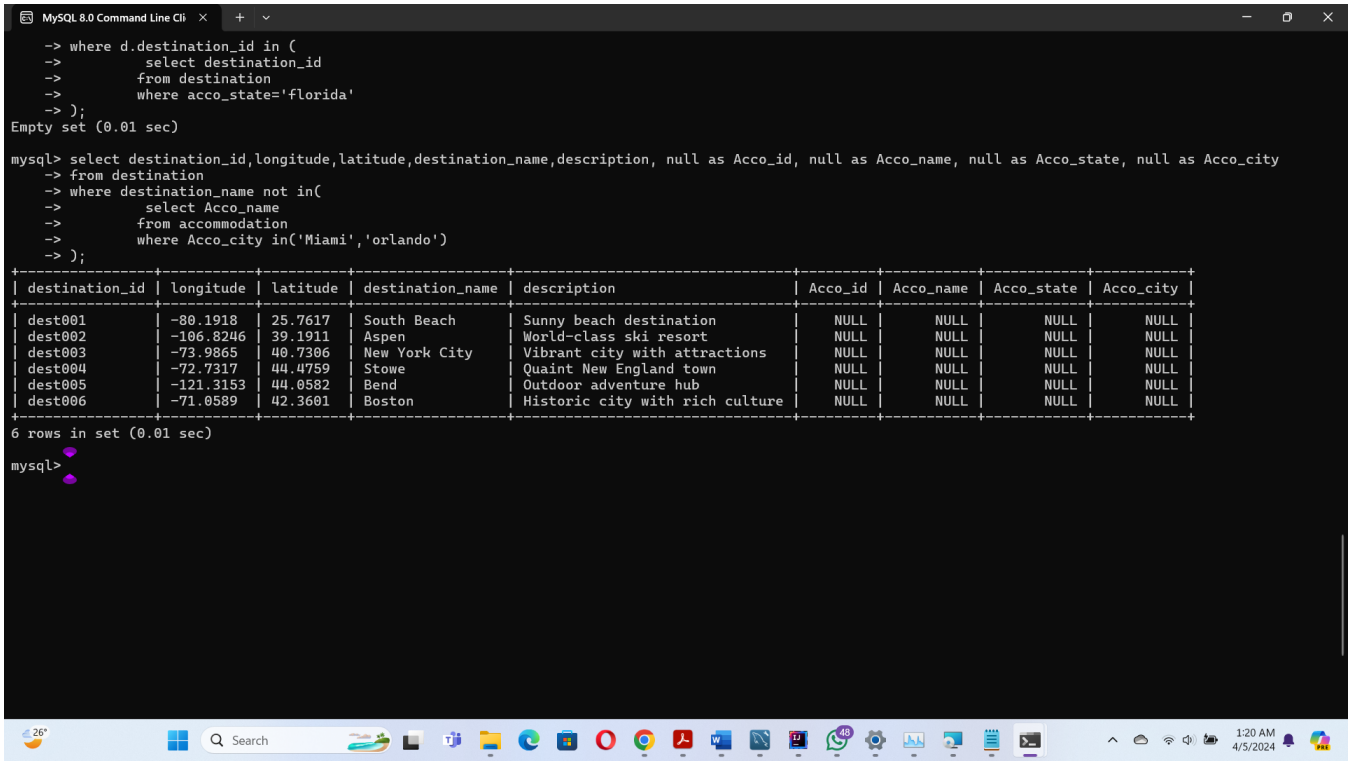


Figure 50: Complex Query 4 (Basic set operation set difference)

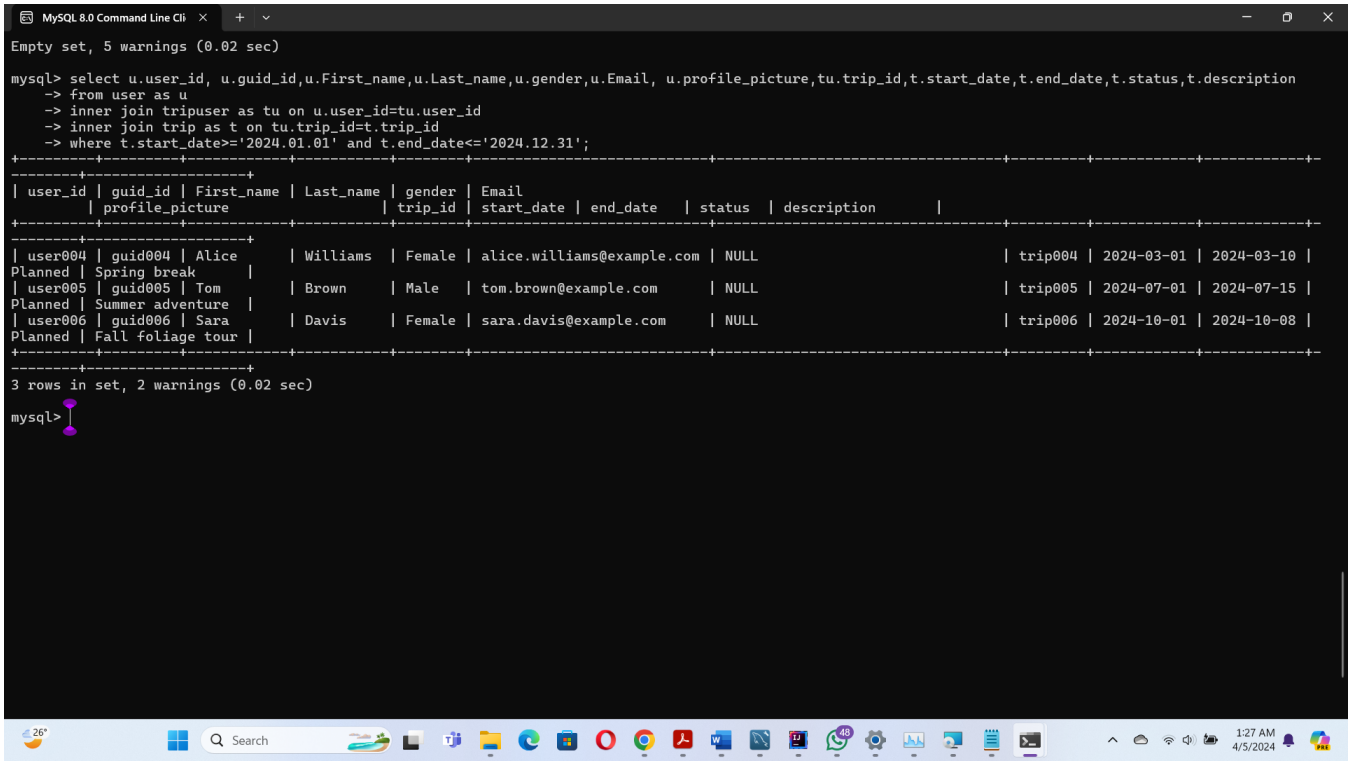


Figure 51: Complex Query 5 (inner join)

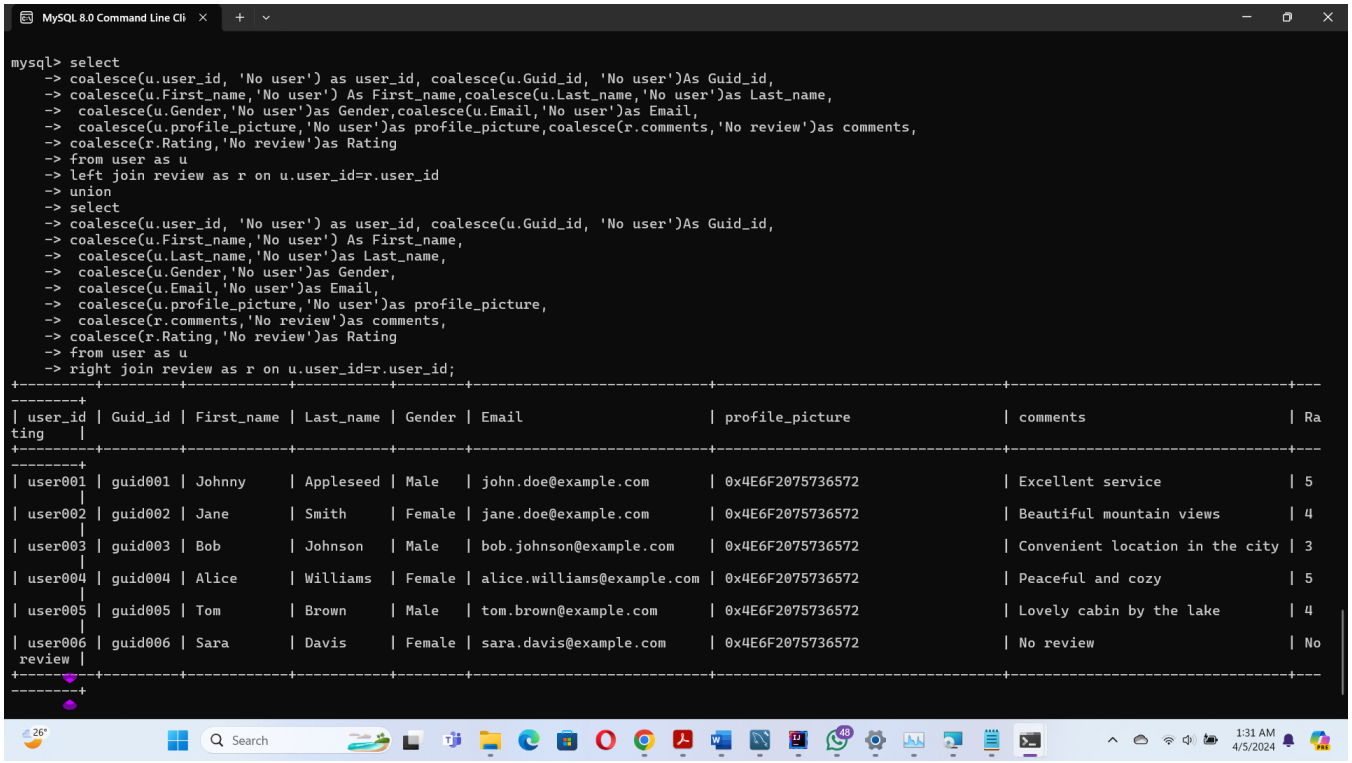


Figure 52: Complex Query 6 (full outer join)

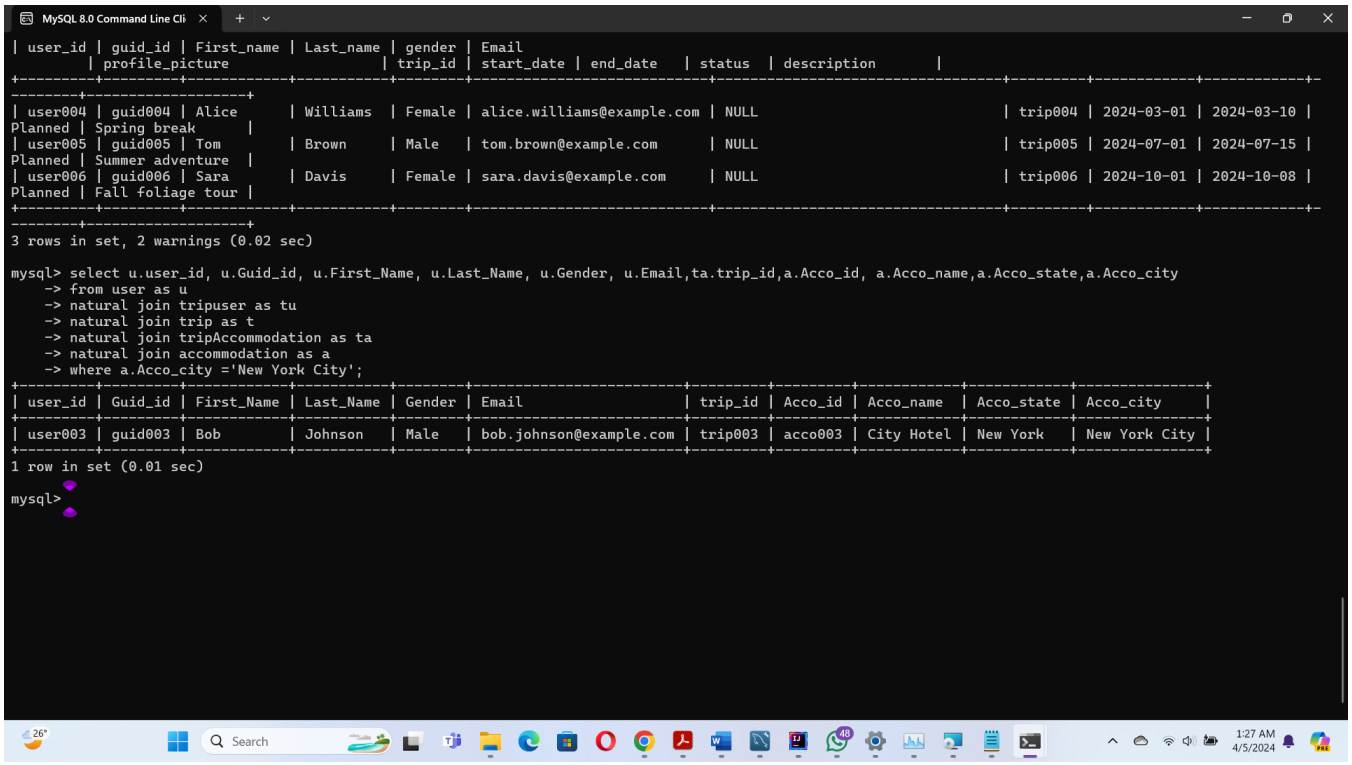


Figure 53: Complex Query 7 (natural join)

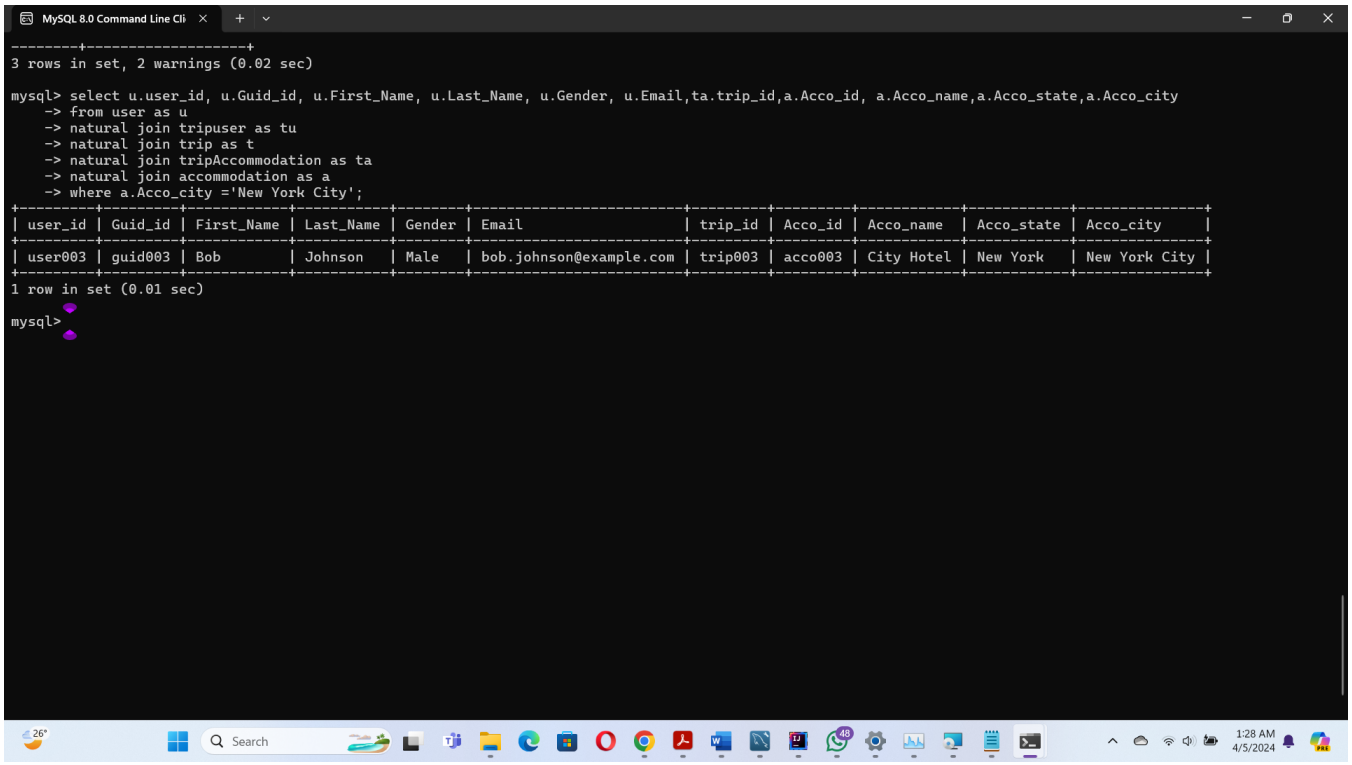


Figure 54: Complex Query 8

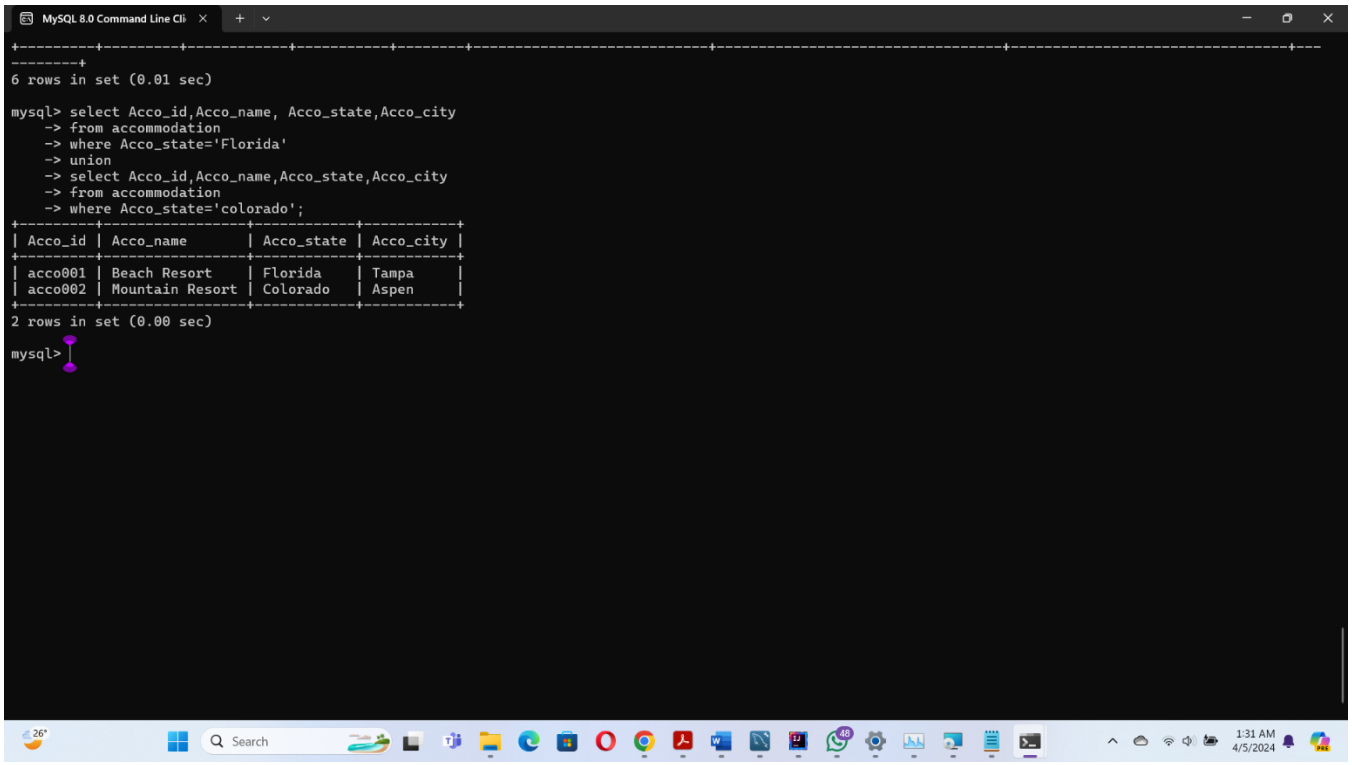


Figure 55: Complex Query 9 (outer join)

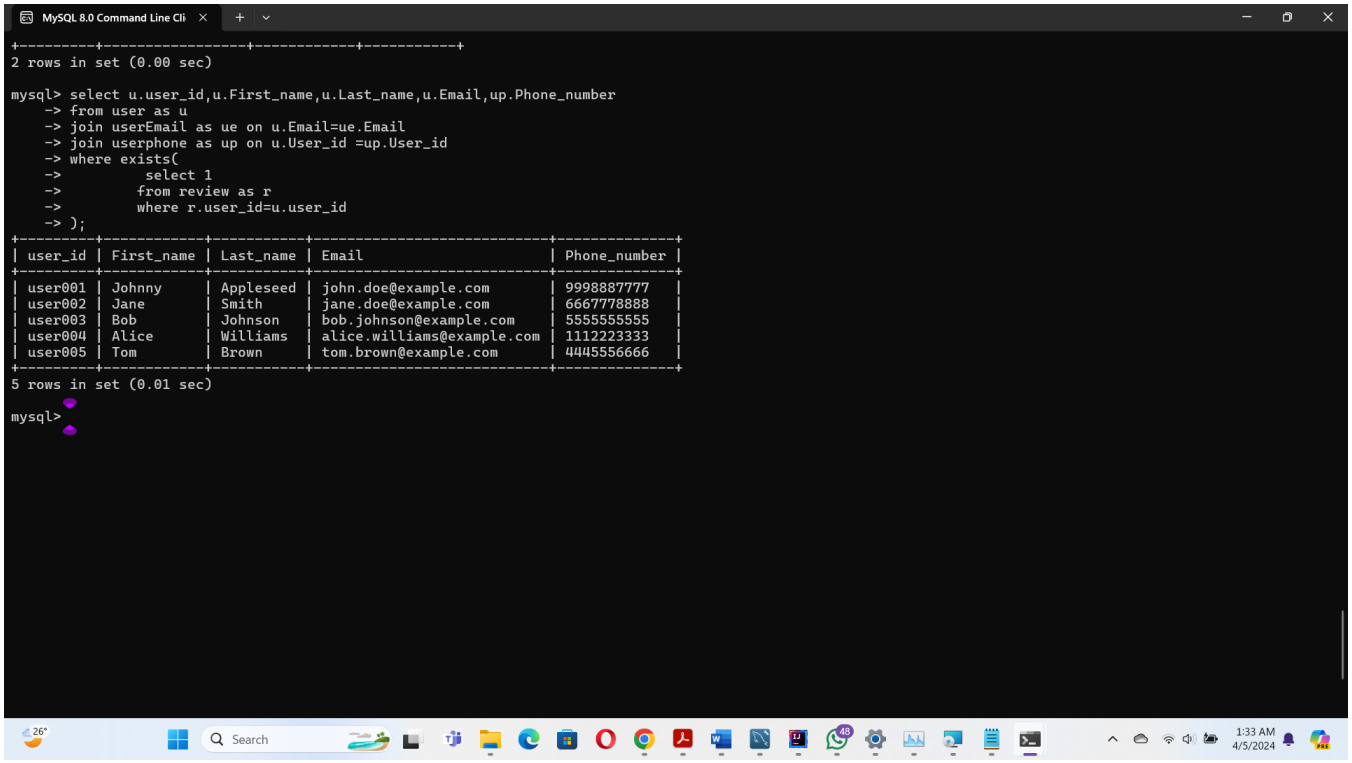


Figure 56: nested query with subquery

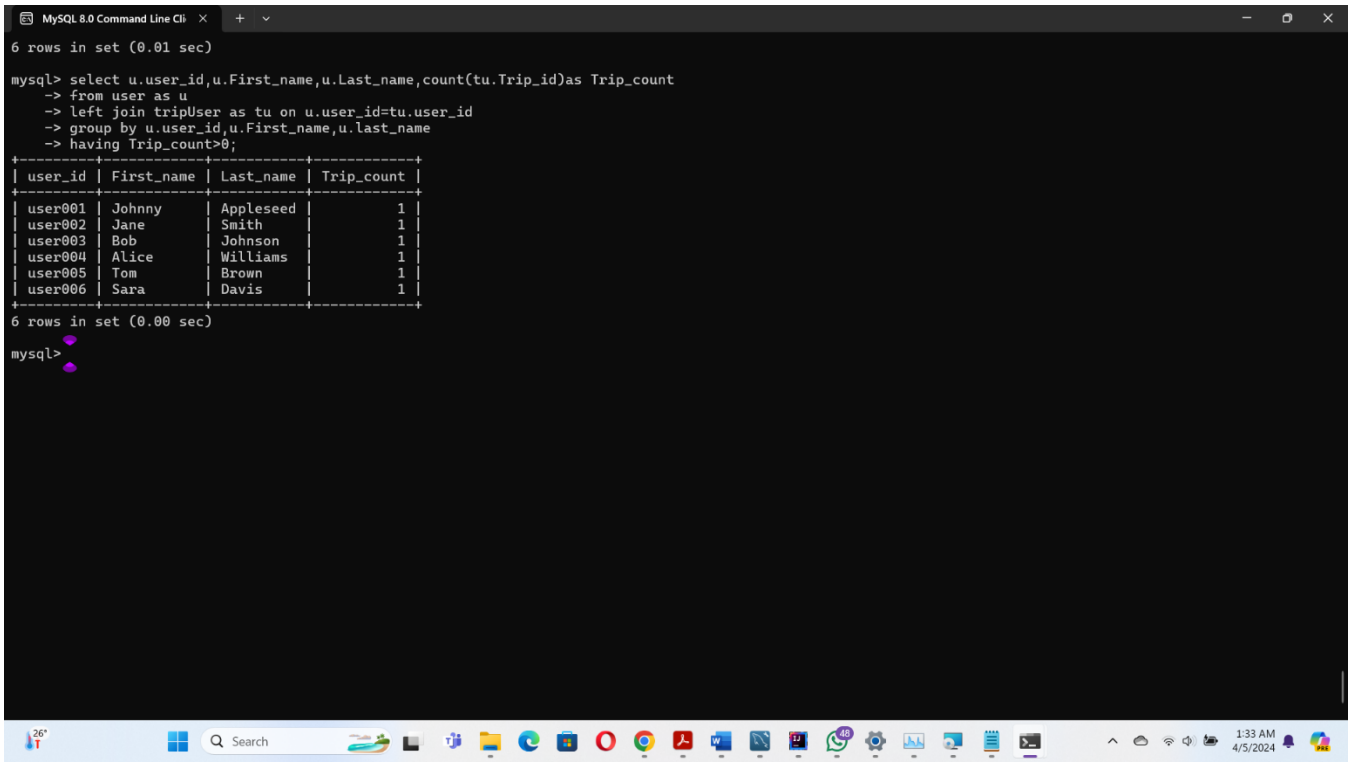


Figure 57: nested query with subquery and conditional count

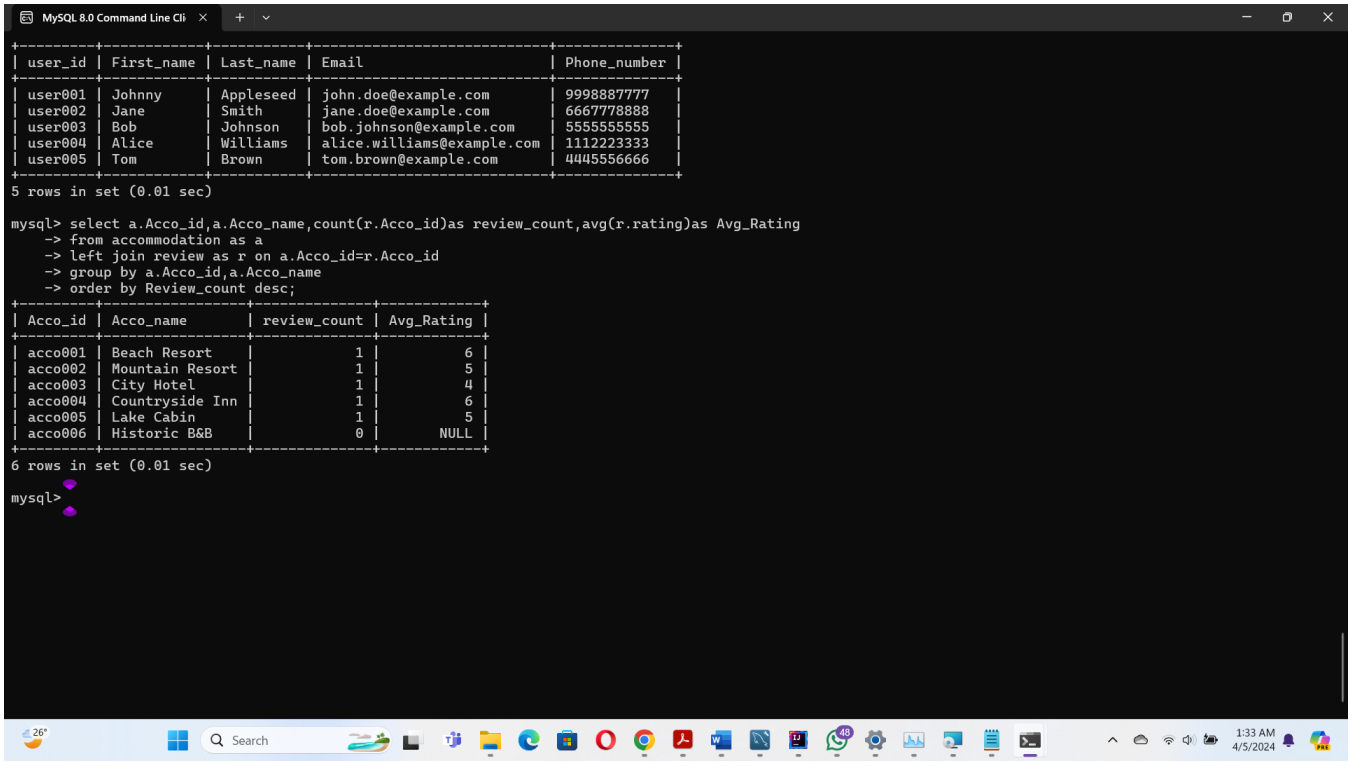


Figure 58: nested query with aggregate function

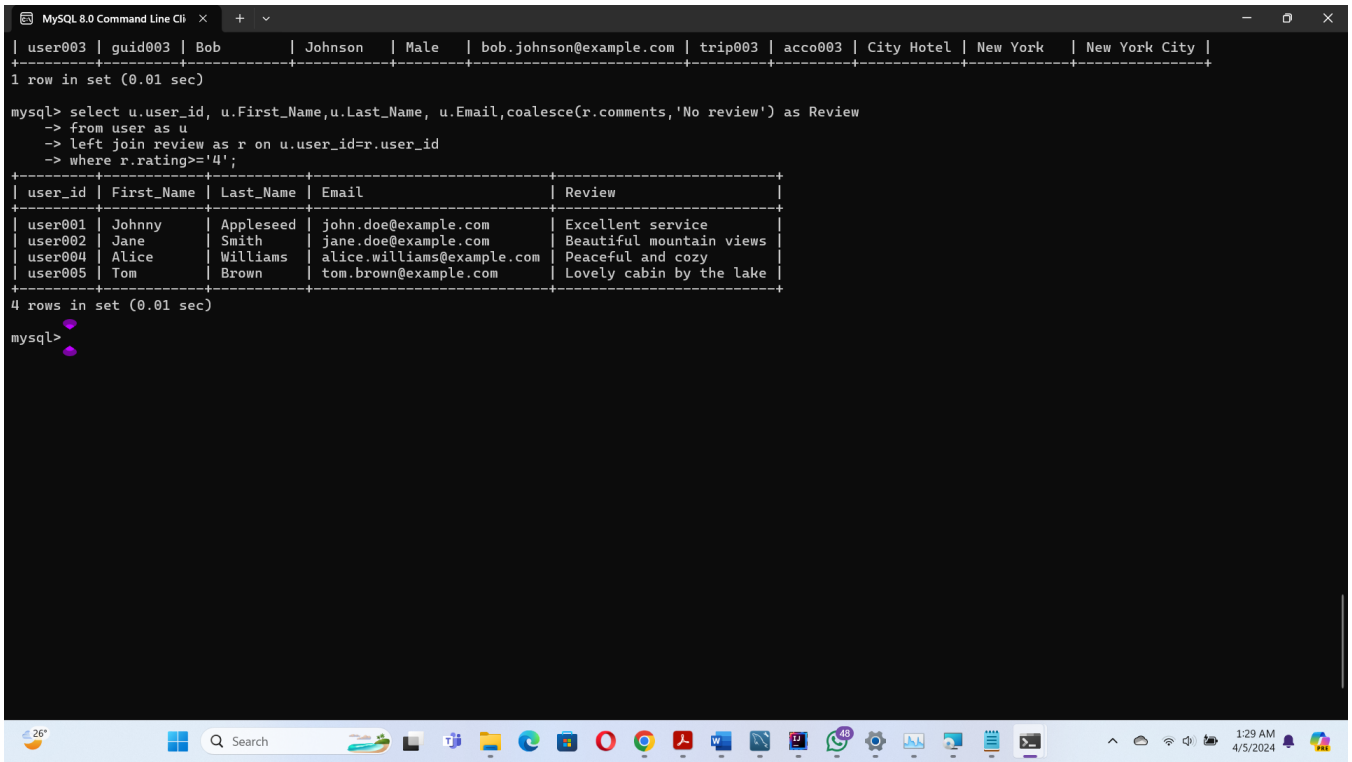


Figure 59: complex query – left outer join

5 Chapter 05- Database Tuning

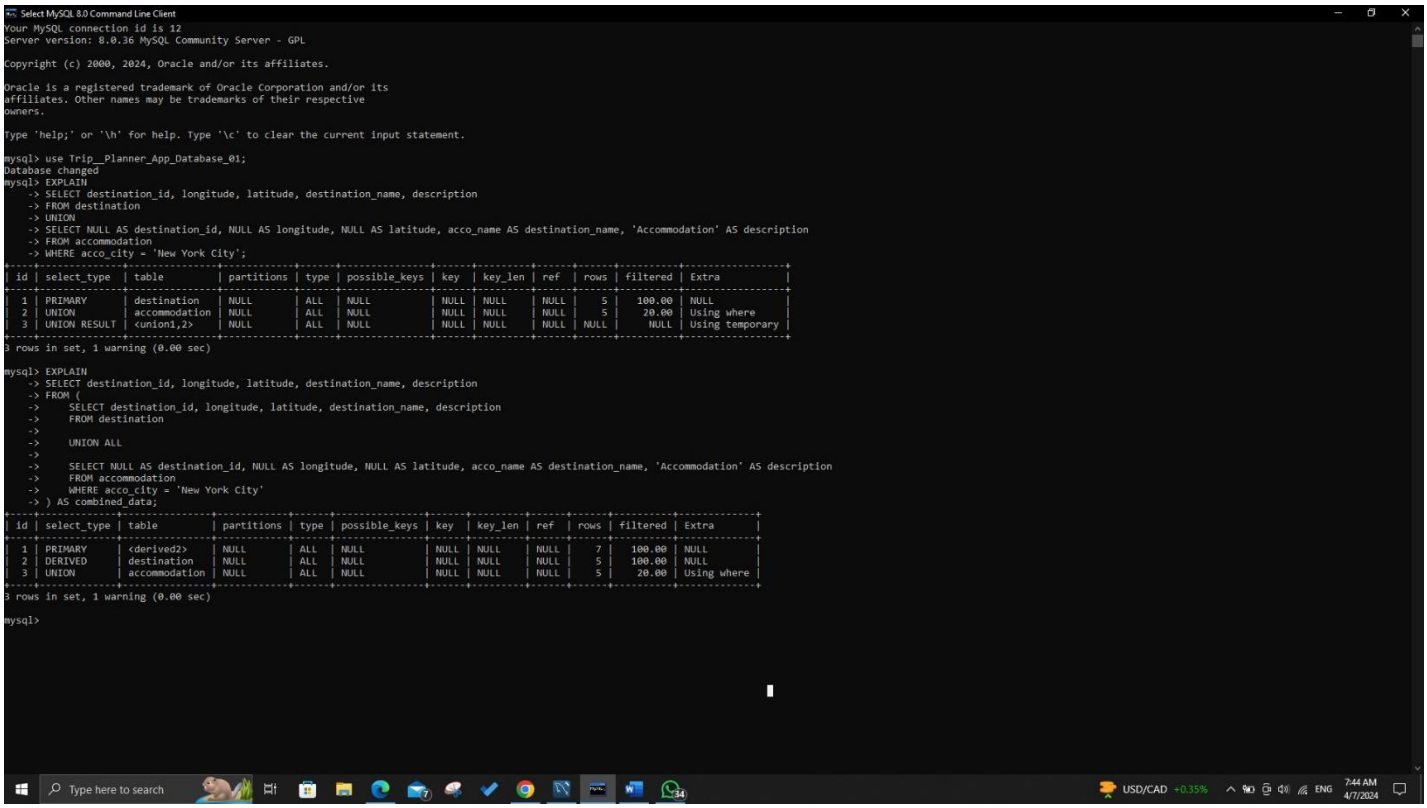


Figure 60: Tuned and explained query of basic set operation union

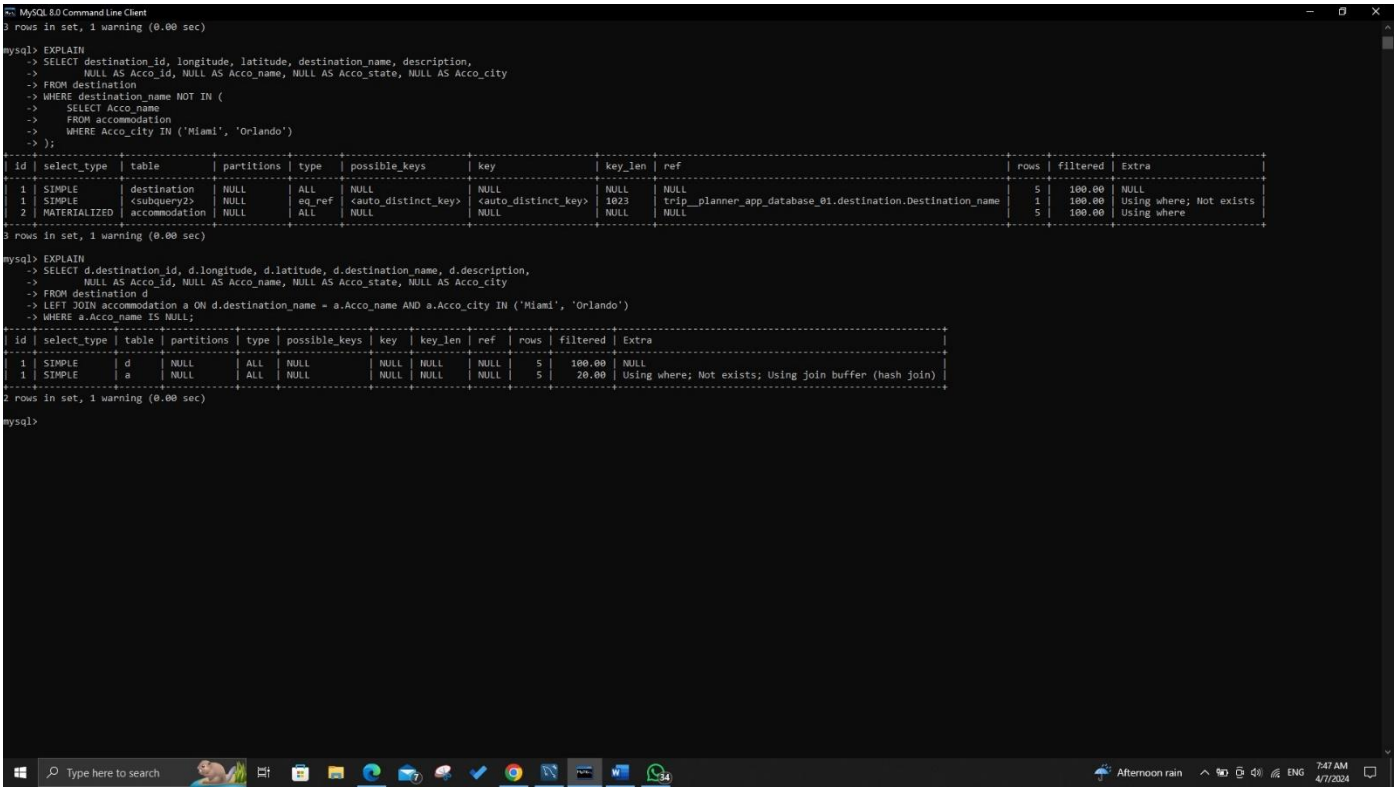


Figure 61: Tuned and explained query of basic set operation set difference

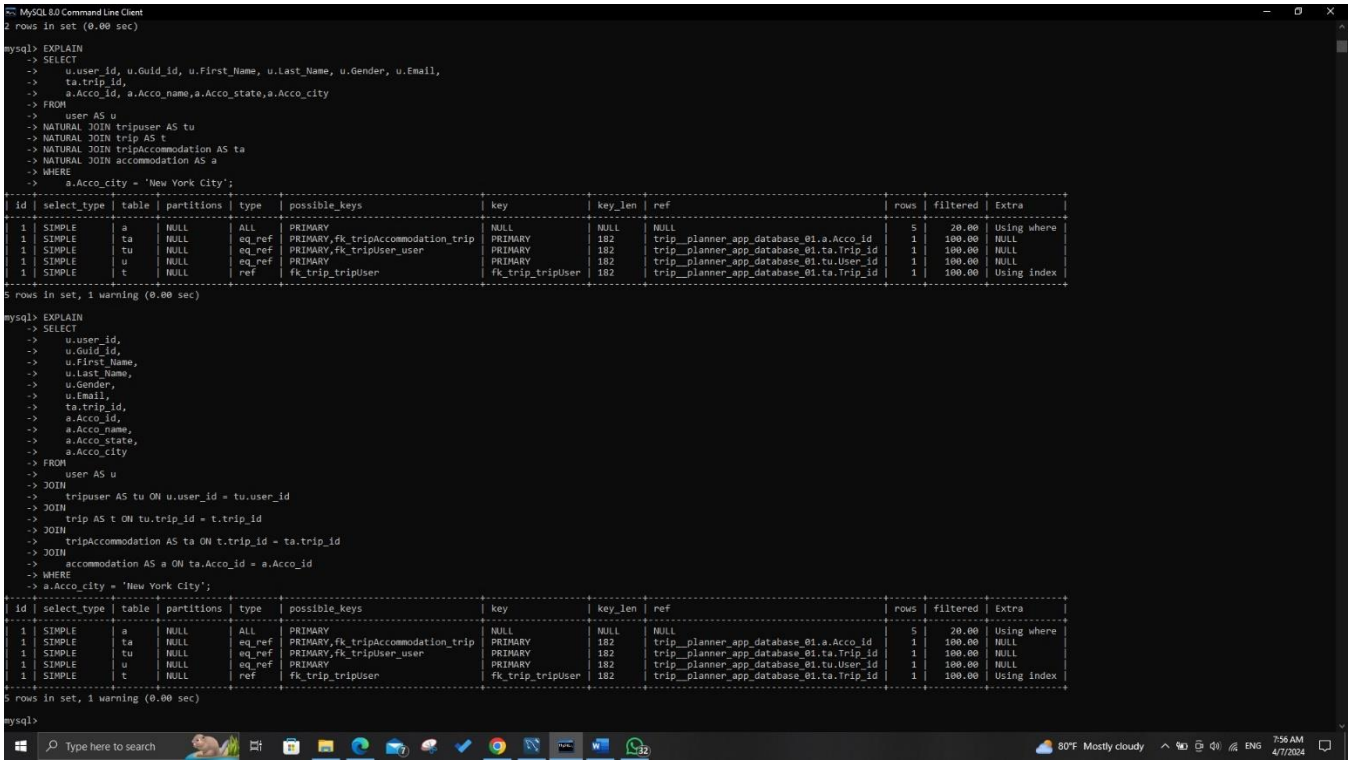


Figure 62: Tuned and explained query of natural join

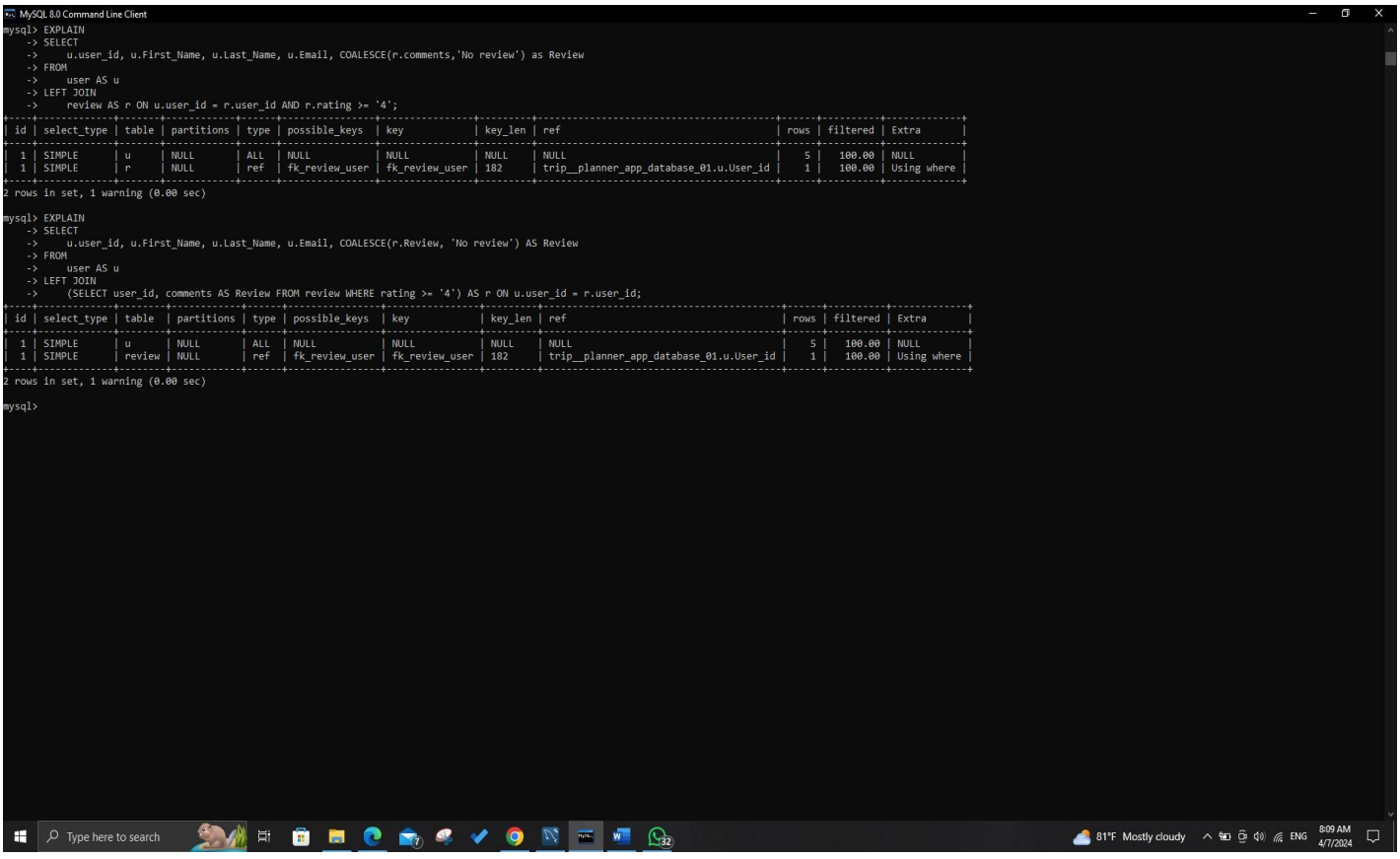


Figure 63: Tuned and explained query of left outer join

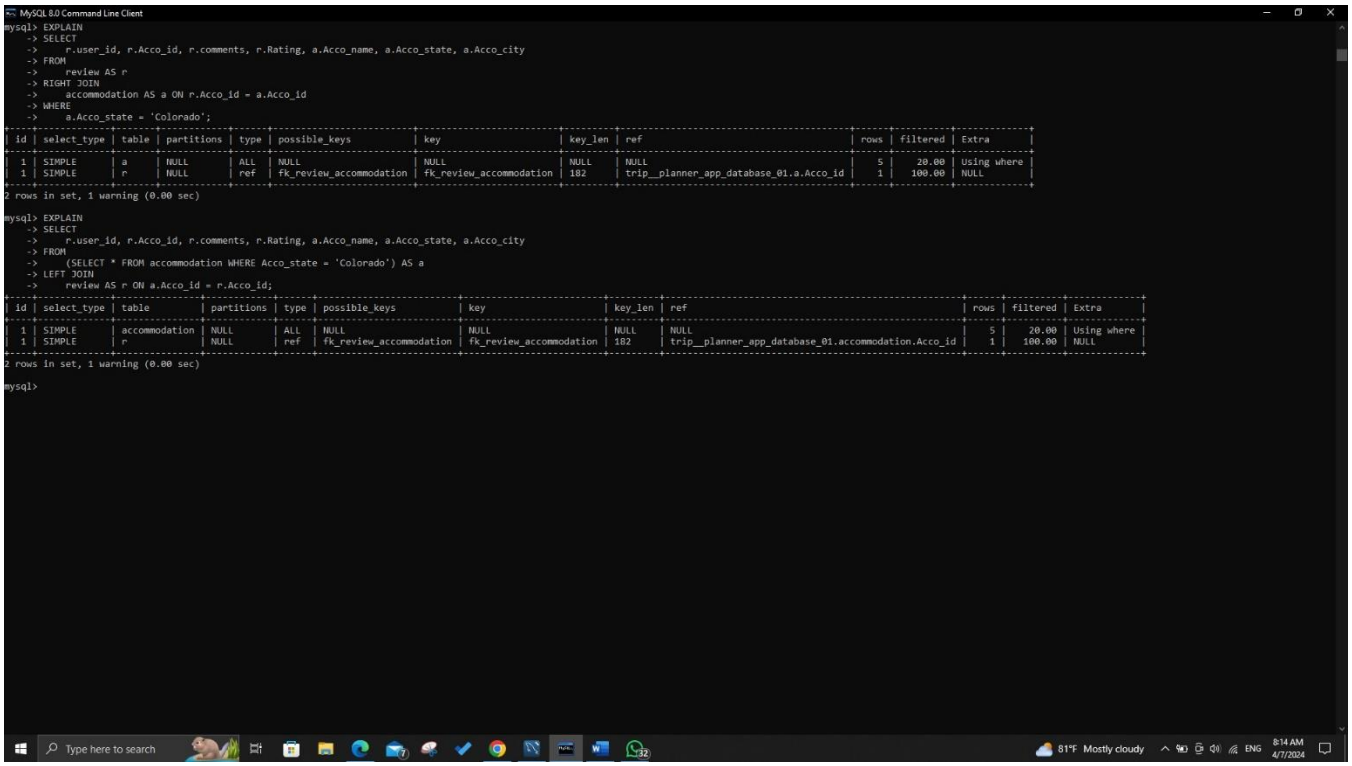


Figure 64: Tuned and explained query of right outer join

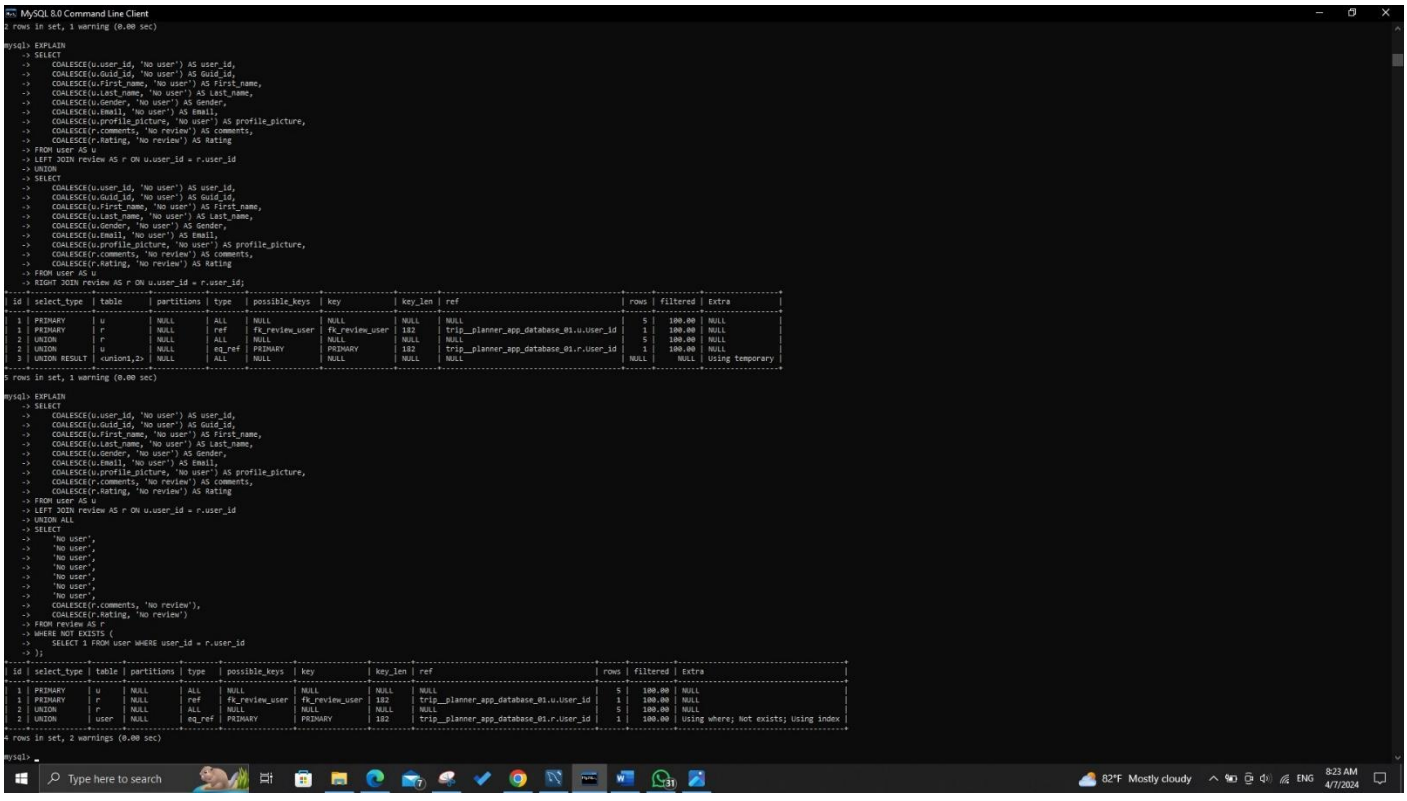


Figure 65: Tuned and explained query of full outer join

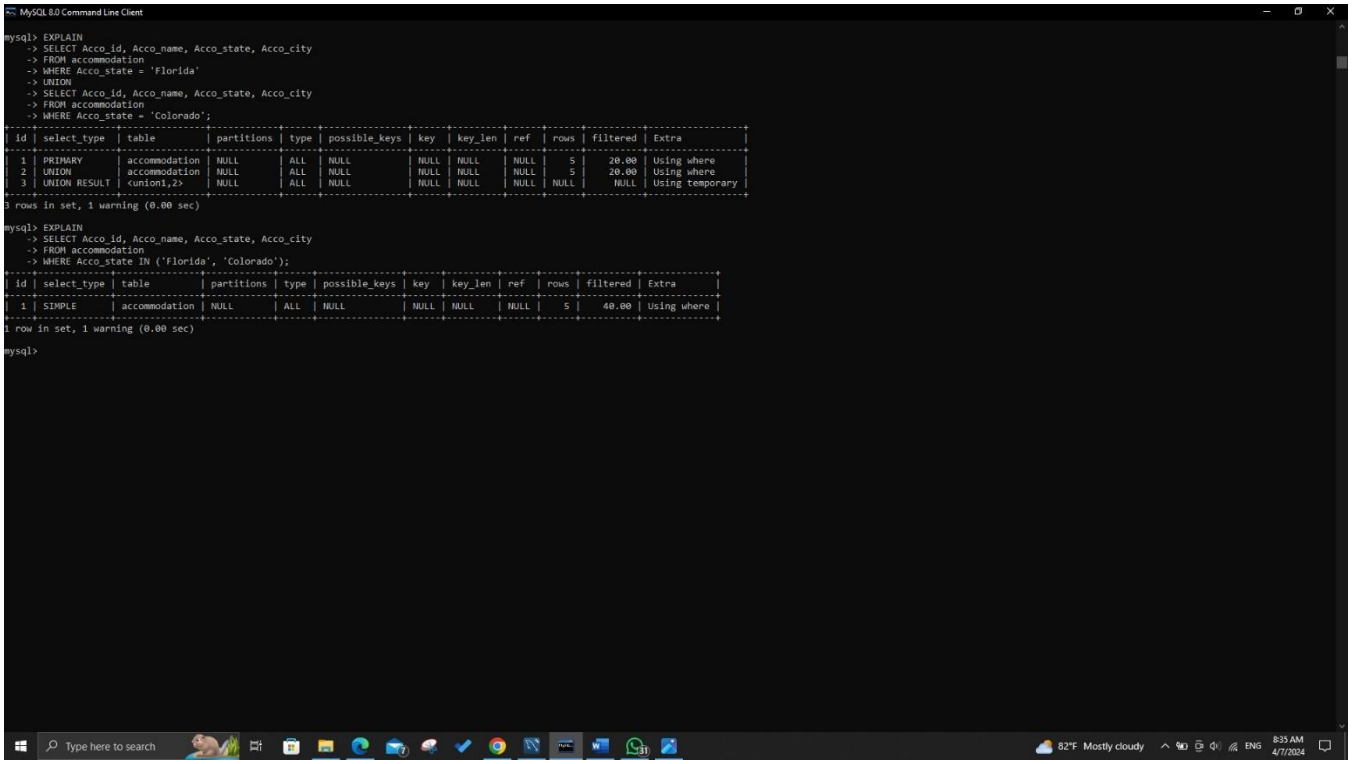


Figure 66: Tuned and explained query of outer join

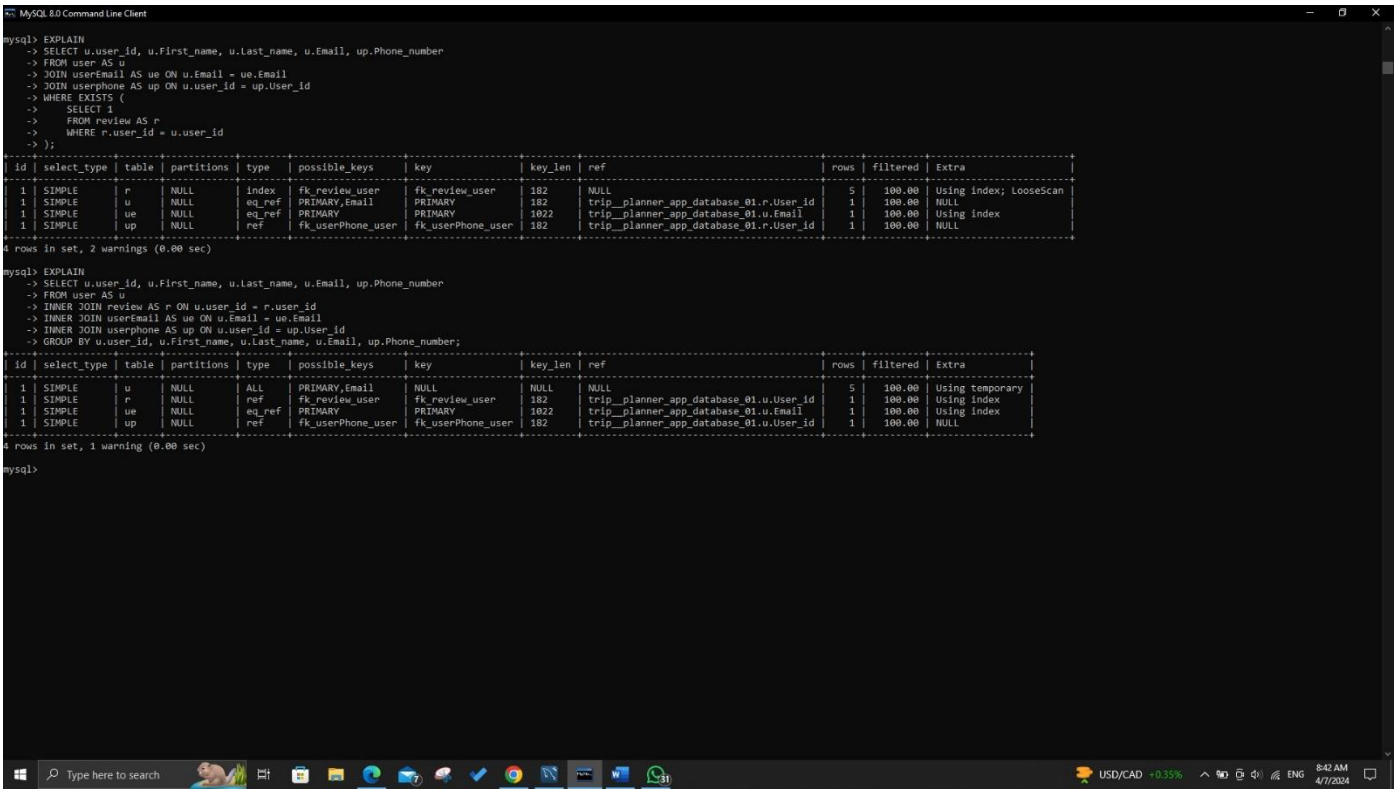


Figure 67: Tuned and explained query of nested query with subquery

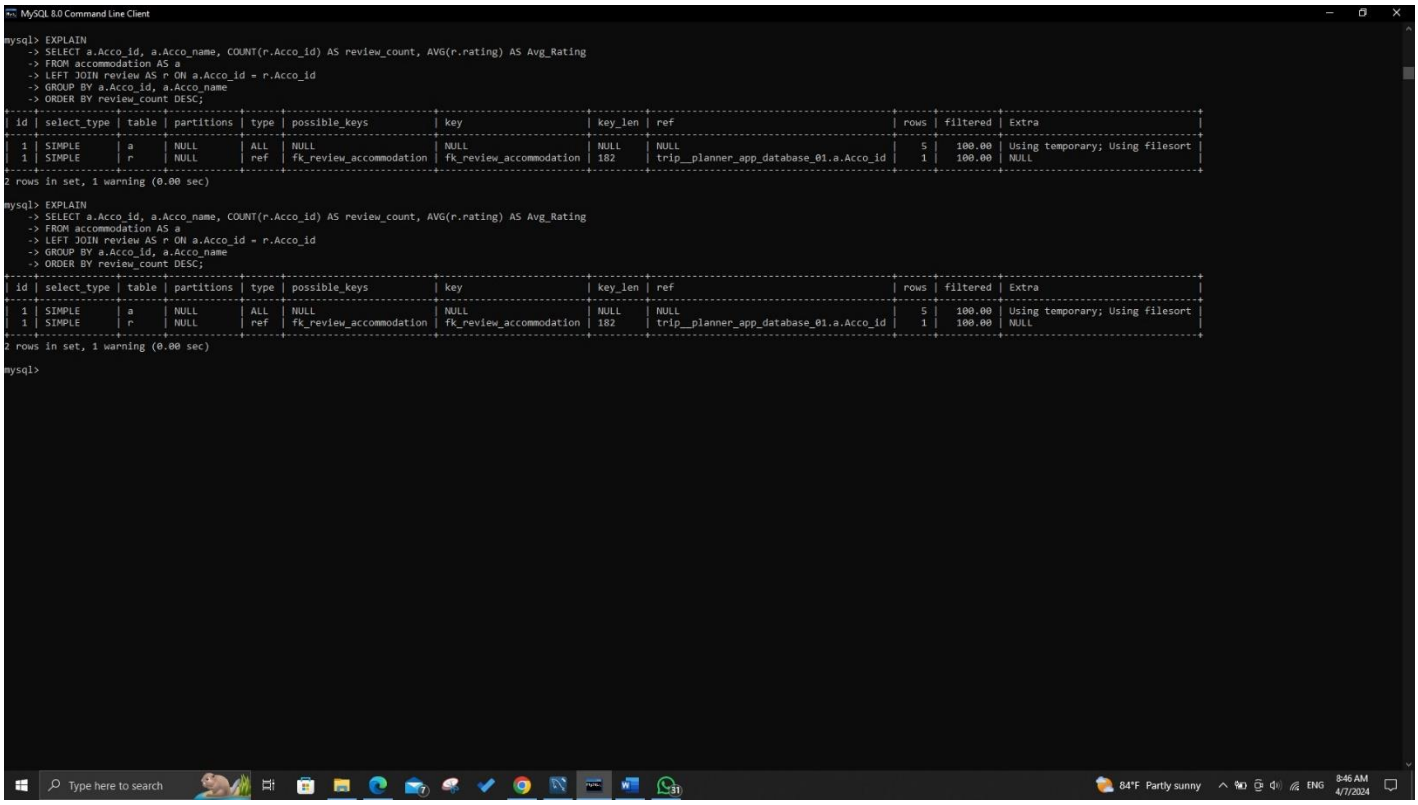


Figure 68: Tuned and explained query of nested query with aggregate function

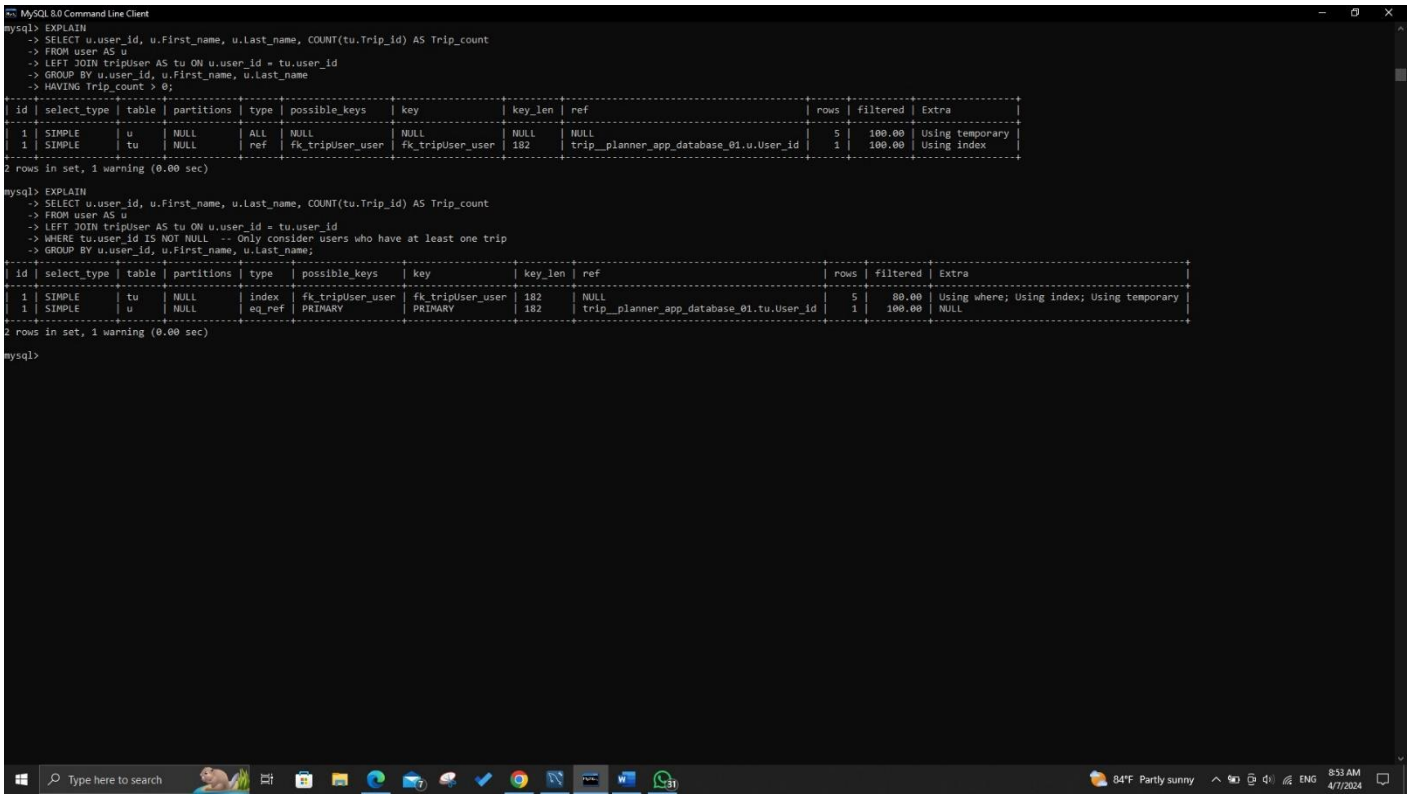


Figure 69: Tuned and explained query of nested query with subquery and conditional count