Mini Project – Hotel Management System

Department of Computer Engineering

Faculty of Engineering

University of Ruhuna

Sri Lanka

On 9th of April 2024

In completing an assignment for the module

EE4350 Database System

By

Group No: 45

MUTHUKUMARI H.M.S.	-EG/2021/4685
NADHA M.J.	-EG/2021/4686
NAFLA M.N.P.	-EG/2021/4687

Contents

Table of Figures

1. Chapter 1 – Requirement Analysis

Introduction

Databases and data management systems are crucial in our daily lives, especially in industries like hospitality. Hotels, being pivotal in the travel and tourism sector, heavily rely on efficient management systems to ensure smooth operations and exceptional guest experiences.

This mini-project aims to address the complexities of managing hotel operations through a comprehensive database system. Well-organized hotel management systems are essential for ensuring guest satisfaction and business success.

Our project focuses on developing a database management system tailored for the hotel industry. Drawing inspiration from the rich cultural heritage and diverse landscapes of our surroundings, we aim to create a system that streamlines hotel operations, enhances guest services, and ultimately contributes to the overall quality of stay for guests.

By studying the organizational and operational characteristics of hotels, we've gathered the necessary information to design a system that meets the unique requirements of hotel management. This system will assist hotel staff in managing room reservations, tracking guest preferences, handling administrative tasks, and providing personalized services.

We believe that implementing this database system will not only improve the efficiency of hotel operations but also enhance the overall guest experience. From front desk personnel to housekeeping staff, everyone involved in hotel management will benefit from this user-friendly and comprehensive system.

Our project adheres to both the functional and data requirements of the hotel industry. Functional requirements define the operations and activities the system should perform, while data requirements identify the necessary information needed to build the system, including attributes of the ER diagram, columns, and records.

In summary, our goal is to develop a hotel management system that revolutionizes the way hotels operate, ensuring seamless experiences for both guests and staff alike

Functional Requirements

Staff Management:

- The system should be capable of managing information about hotel staff, including their names, positions, salary and contact details.
- It should facilitate the addition, updating, and deletion of staff details, ensuring an up-to-date record of all employees.
- The system should provide insights into the availability and status of each staff member, indicating whether they are on duty, on leave, or unavailable for any reason.

Room Management:

- The system should manage information regarding different types of rooms available in the hotel, including their type ID, maximum guest capacity, total number of rooms, room description, and room price.
- The system should provide insights into the availability status of each room type, indicating the number of rooms available for booking and occupancy.

Room Management:

- The system should manage information about individual rooms within the hotel, including their room number, room type, rate, bed type, and cleaning status.
- The system should allow for the assignment of room types to specific room numbers, ensuring proper categorization and organization.
- It should provide insights into the cleaning status of each room, indicating whether it is clean and available for occupancy or in need of cleaning.

Reservation Management:

- The system should manage reservations made by guests, including guest ID, check-in date, check-out date, number of guests, and room number.
- It should facilitate the creation, modification, and cancellation of reservations, ensuring accurate and up-to-date booking records.
- The system should validate reservation details, ensuring that the check-in date is before the check-out date and that the selected room is available for the specified dates.

Payment Management:

- The system should manage payment information associated with guest reservations, including payment ID, guest ID, payment method, payment date, and total amount.
- It should facilitate the recording, updating, and tracking of payments made by guests for their reservations.
- The system should support various payment methods, such as cash, credit card, or online payment services, allowing guests flexibility in settling their bills.

Invoice Management:

- The system should manage invoice information for guest reservations, including guest ID, issue date, due date, tax amount, and total amount.
- It should facilitate the generation, updating, and tracking of invoices for each guest reservation, ensuring accurate billing and payment processing.
- The system should automatically calculate tax amounts based on applicable tax rates and include them in the invoice total.

Guest Management:

- The system should manage guest information, including guest ID, first name, last name, date of birth, age, email address, street address, city, province, and guest password.
- It should facilitate the creation, updating, and deletion of guest profiles, ensuring accurate and up-to-date guest records.
- The system should validate guest information, including ensuring that email addresses are in the correct format and that passwords meet security requirements.

Contact Number Management:

- The system should manage contact number information for guests, including guest ID and contact number.
- The system should validate contact numbers to ensure they are in the correct format and meet any specified criteria.

Cancellation Management:

- The system should manage cancellation information for guest reservations, including cancellation ID, guest ID, payment ID, reason for cancellation, refund amount, and cancellation date.

Data Requirements

Data requirements are the data that provide a detailed description of the data model that the system must use to fulfil its functional requirements. There are many entities in Er diagram and the attributes of those entities shows the data we need to build this database of a Hotel Management System. So, the data requirements can be listed as follows.

List of entities

1. Staff
2. Room Type
3. Room
4. Reservation
5. Payment
6. Invoice
7. Guest
8. Contact Number
9. Cancellation

Attribute List

Entity	Attributes (data Requirements)
Staff	Staff ID
	Staff Name
	Position
	Salary
	Hire date
Room Type	Type ID
	Maximum Guest
	Total Rooms
	Room Description
	Room Price
Room	Room Number
	Room Type
	Rate
	Bed Type
	Cleaning Status
Reservation	Guest ID
	Check In Date
	Check Out Date
	Number of Guests

	Room Number
Payment	Payment ID
	Guest ID
	Payment Method
	Payment Date
	Total Amount
Invoice	Guest ID
	Issue Date
	Due Date
	Tax Amount
	Total Amount
Guest	Guest ID
	First Name
	Last Name
	Date of Birth
	Age
	Email
	Street
	City
	Province
	Guest Password
Contact Number	Guest ID
	Contact Number
Cancellation	Cancellation ID
	Guest ID
	Payment ID
	Reason
	Refund Amount
	Cancellation Date

Chapter 2 – Conceptual Design (ER Diagram)

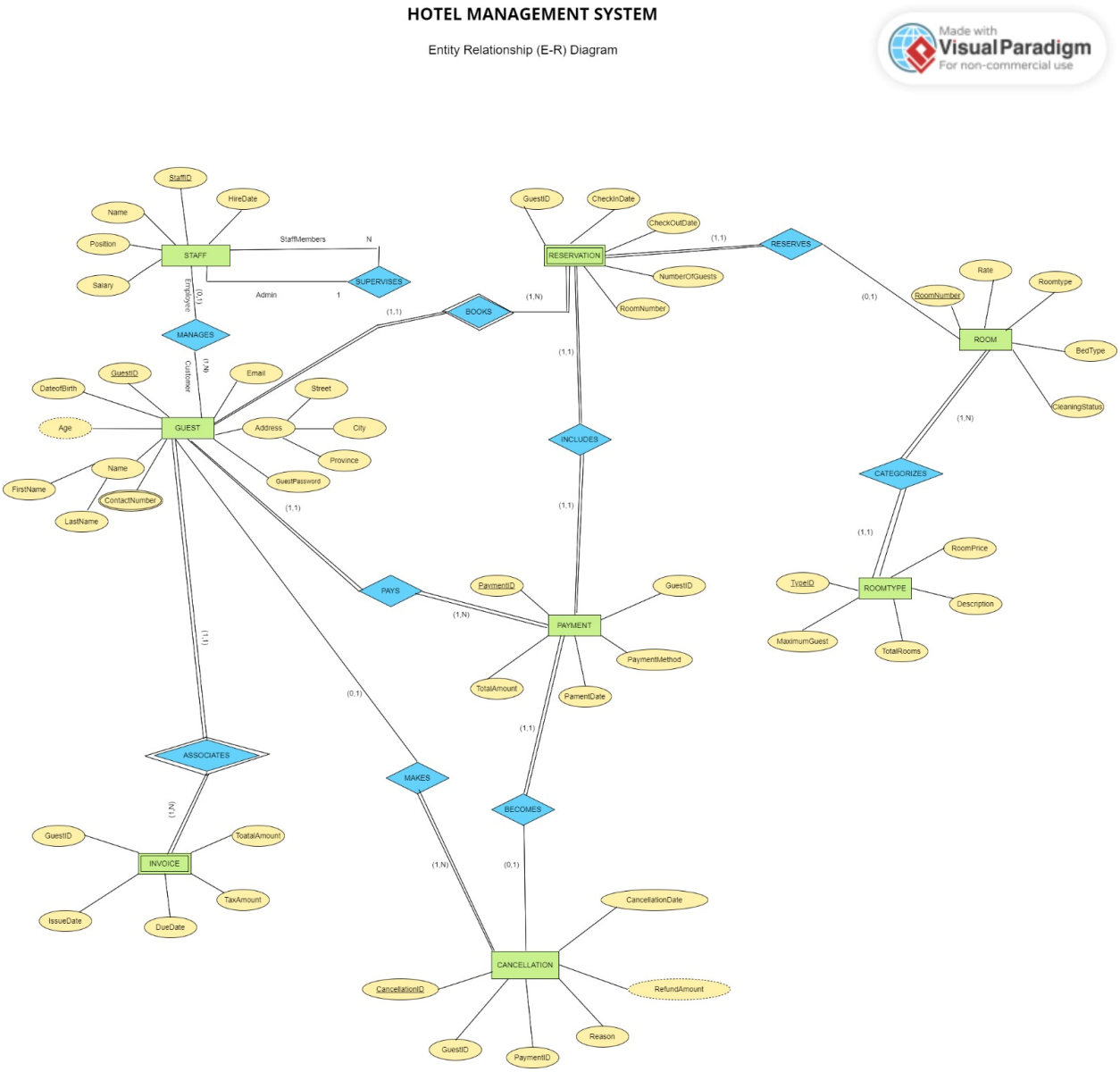


Figure 1: ER Diagram

Conceptual design (UML Diagram)

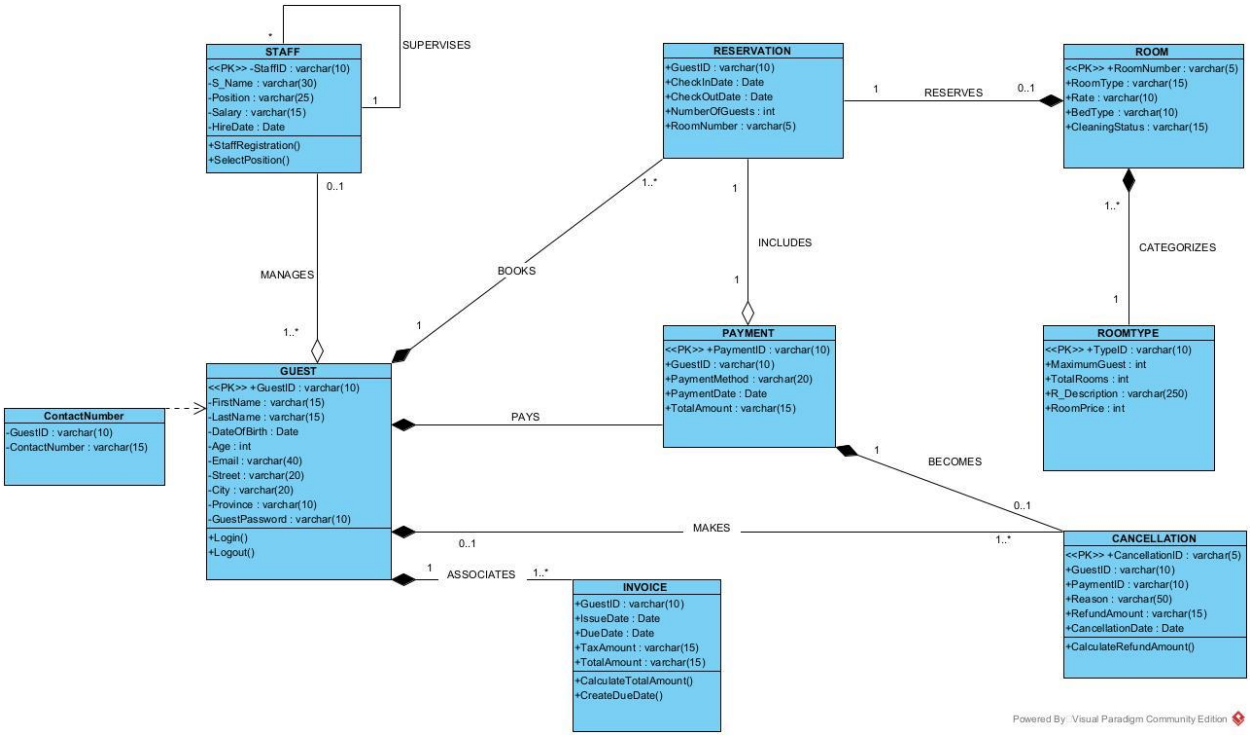


Figure 2: UML Diagram

Relationships of ER Diagram

- Guest-Reservation Relationship: A guest can make multiple reservations, but each reservation is made by one guest.
- Reservation-Rooms Relationship: A reservation can include multiple rooms, and each room can be included in multiple reservations.
- Room-Type of Rooms Relationship: Each room belongs to one type, and each type can have multiple rooms.
- Service Requested by Guests Relationship: Each service is requested by one guest, but a guest can request multiple services.
- Bill-Paid by Guest Relationship: Each bill is paid by one guest, but a guest can pay multiple bills.
- Bill-Service Charges Relationship: A bill includes charges for multiple services, and each service charge can be included in multiple bills.

Recursive Relationship:
The recursive relationship for the STAFF attributes with the SUPERVISES relationship can be described as follows:

Entity: Staff Attribute:

- StaffID (Primary key)
- Name
- Position
- Salary
- Hiredate

Relationships:

- Admin (foreign key referencing StaffID) - This field represents the admin of the current staff member.
- Staff Members - This field represents the Staff members who are working in the hotel
- Explanation: In this recursive relationship, the Staff entity relates to itself through the SUPERVISES relationship. Each staff member can act as both an admin and a staff member.

This recursive relationship allows for the representation of hierarchical structures within the staff organization, where staff members can admin and can themselves act as staff members for other staff members. This can be applied to both regular staff members and administrative staff.

Chapter 3 – Implementation

Schema Creation

Creating Database

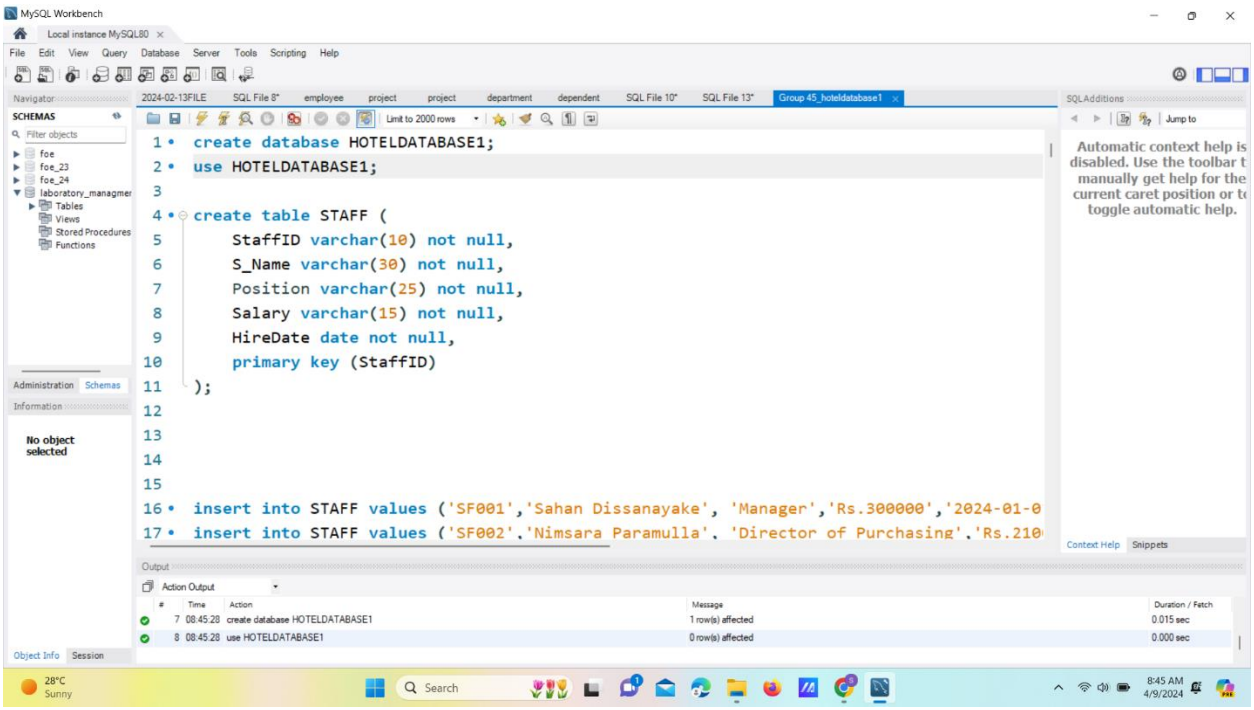
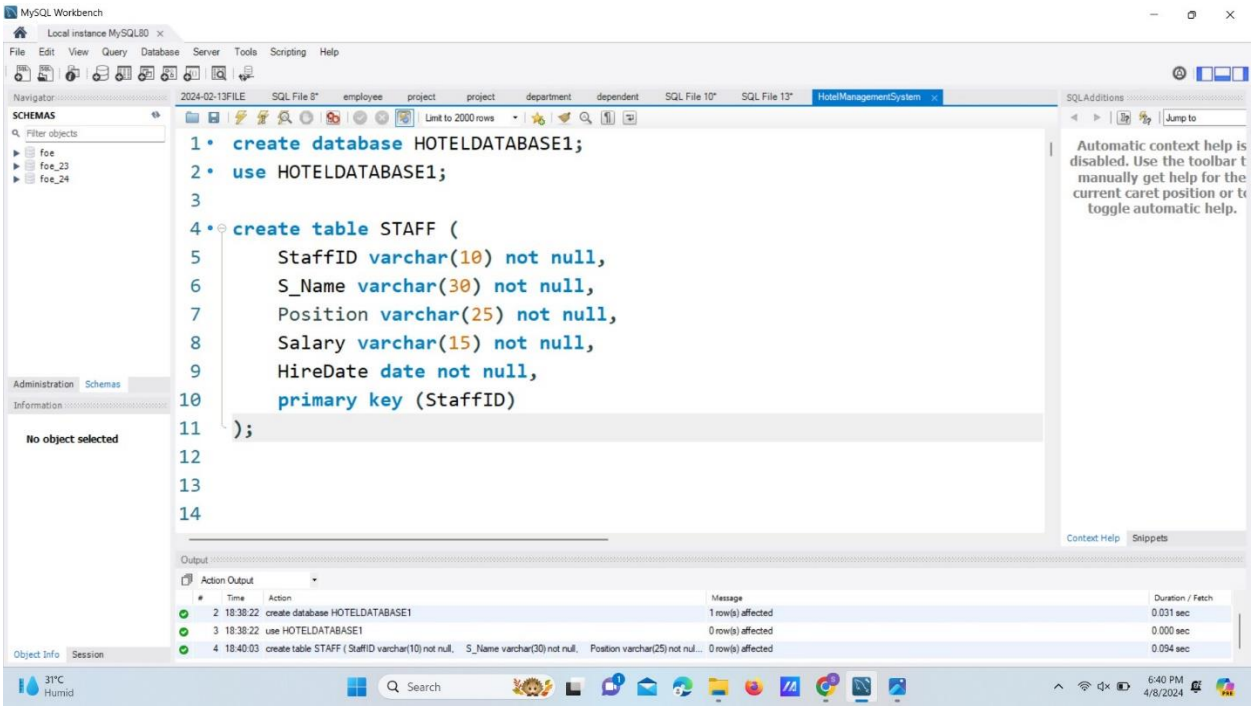
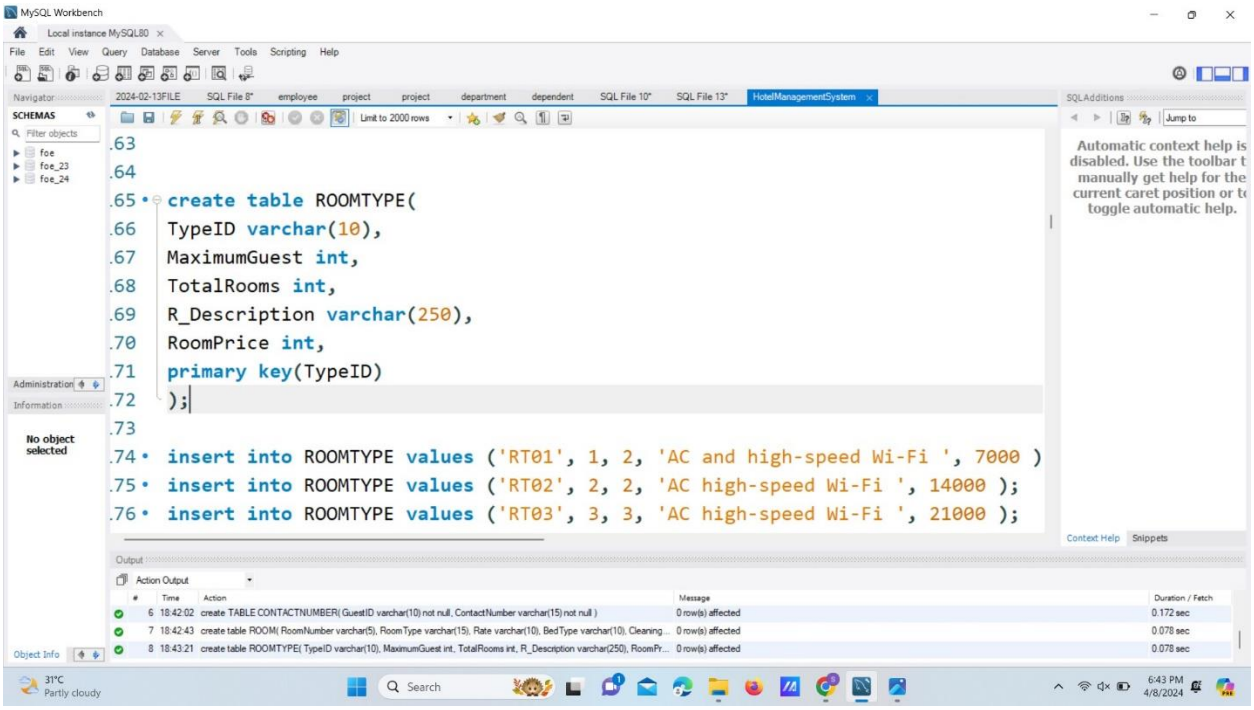


Table Definitions

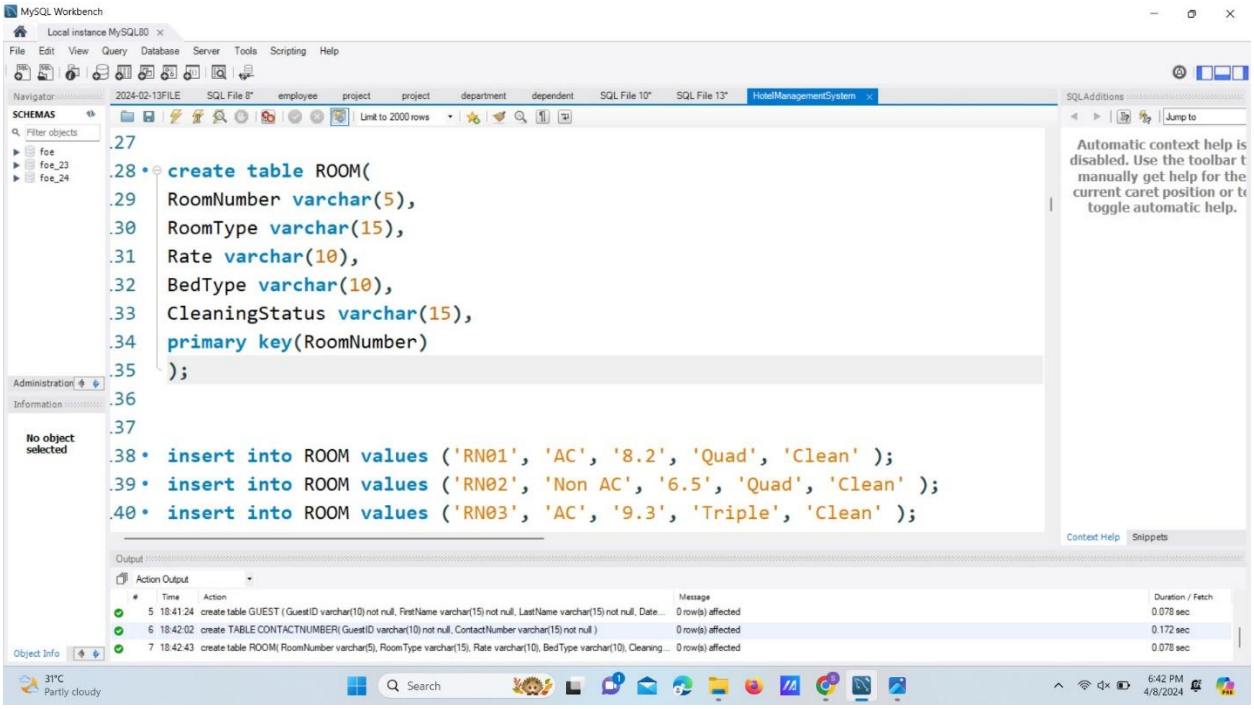
Staff Table



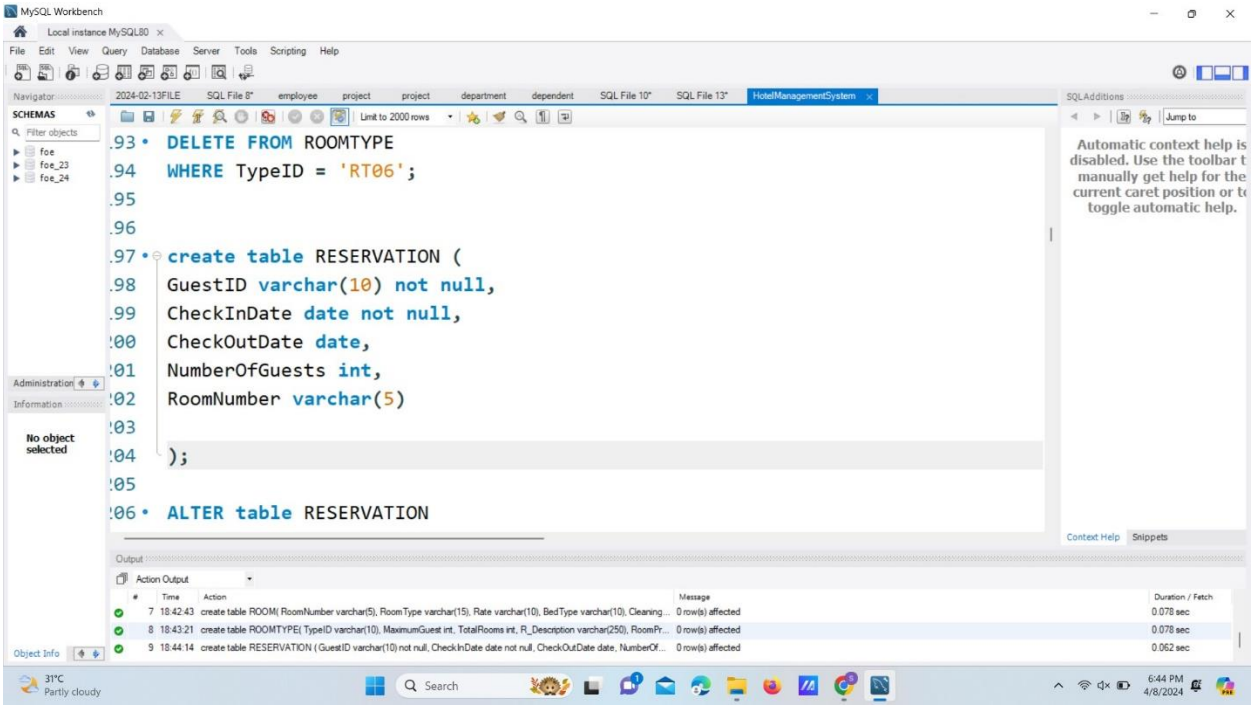
Room Type Table



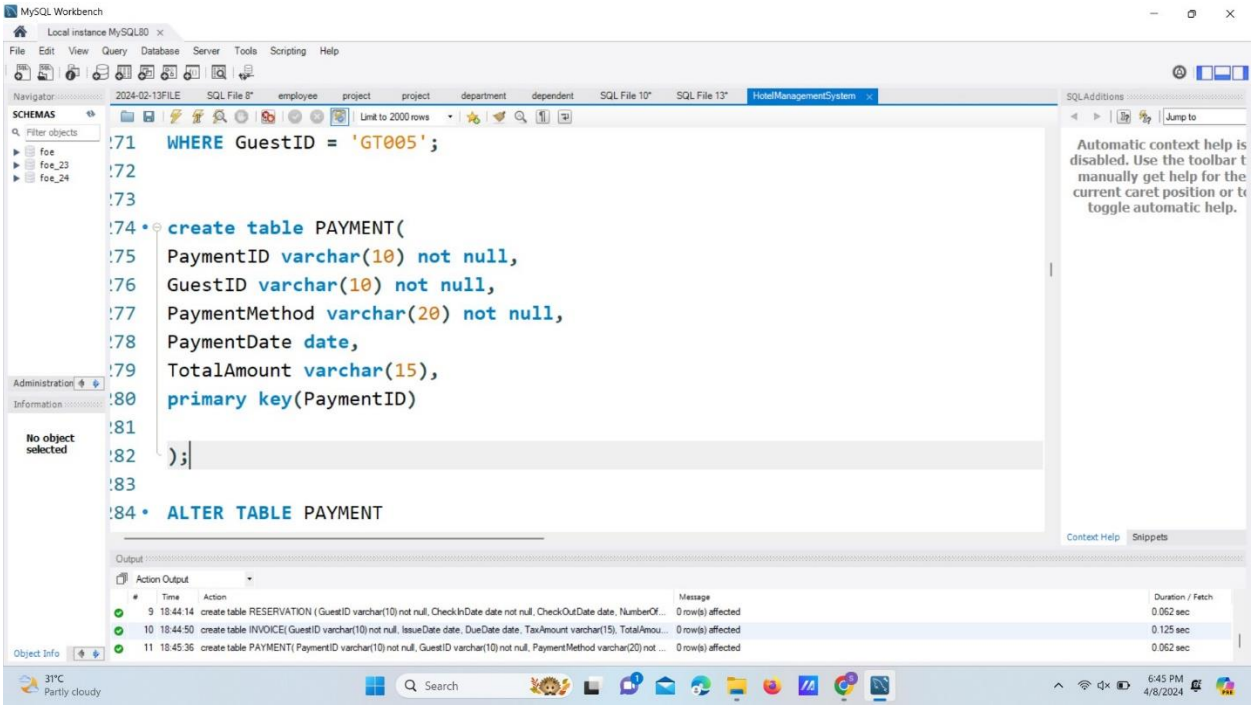
Room Table



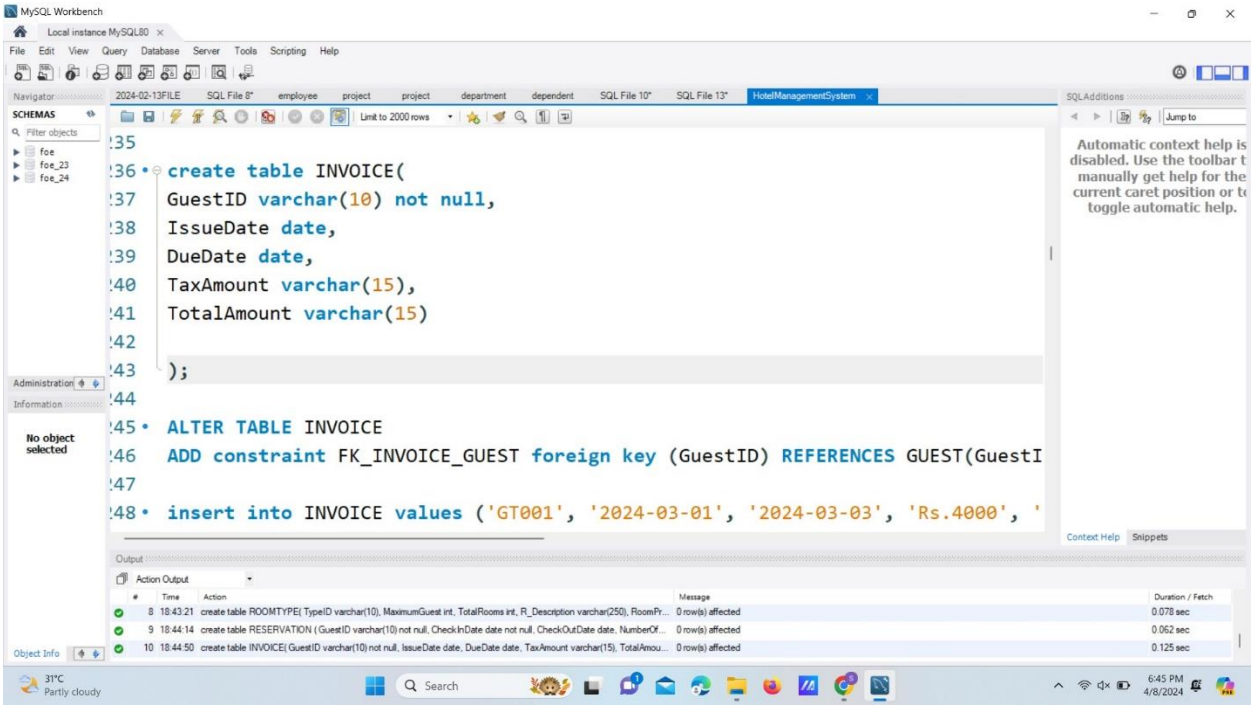
Reservation Table



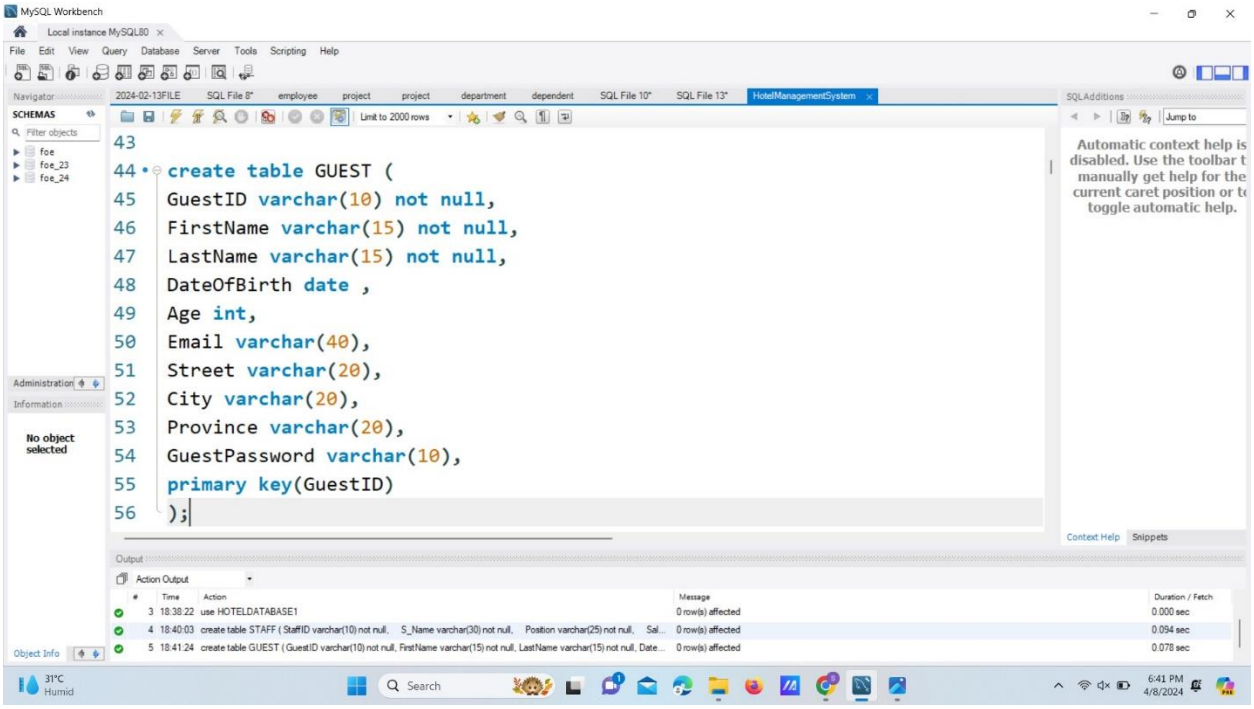
Payment Table



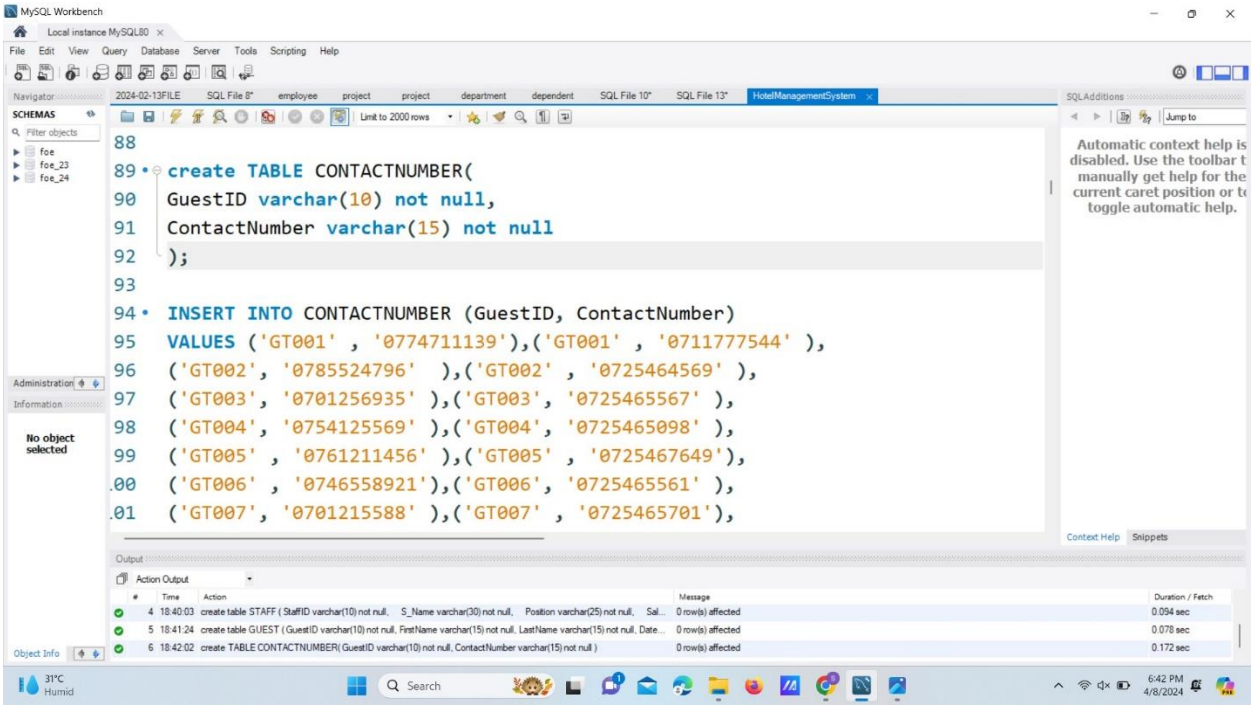
Invoice Table



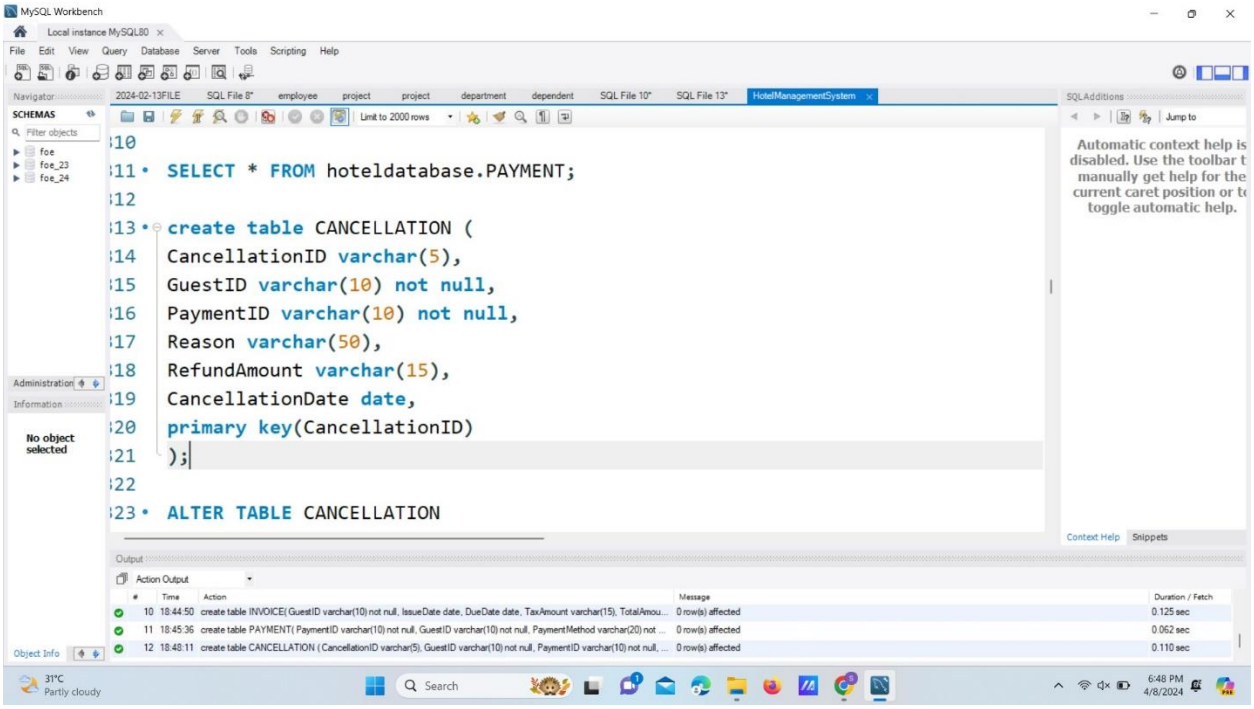
Guest Table



Contact Number Table

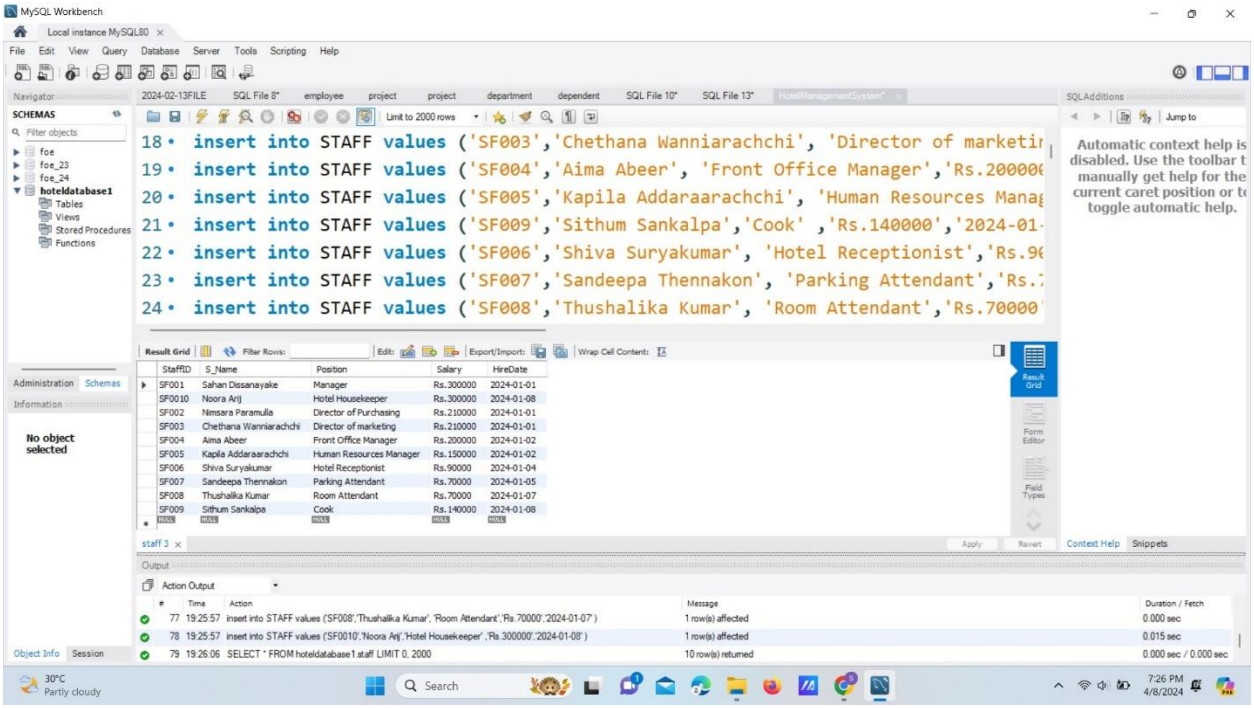


Cancellation Table

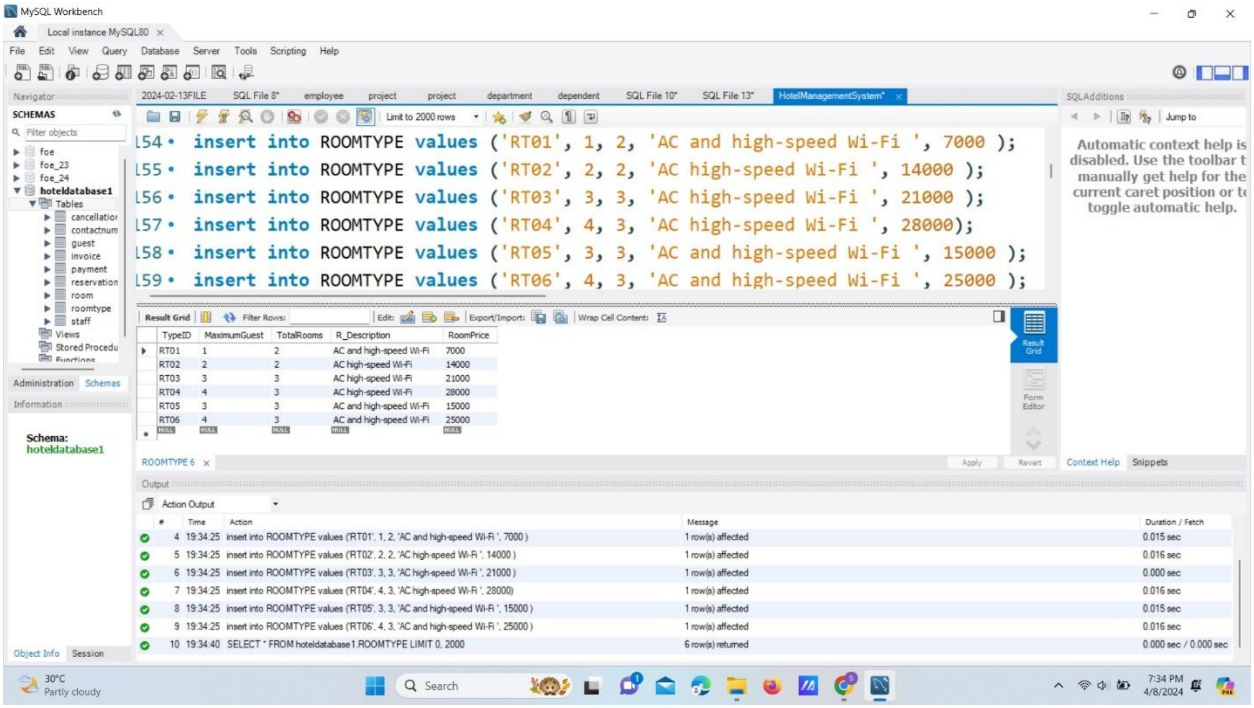


Inserting Data for Tables

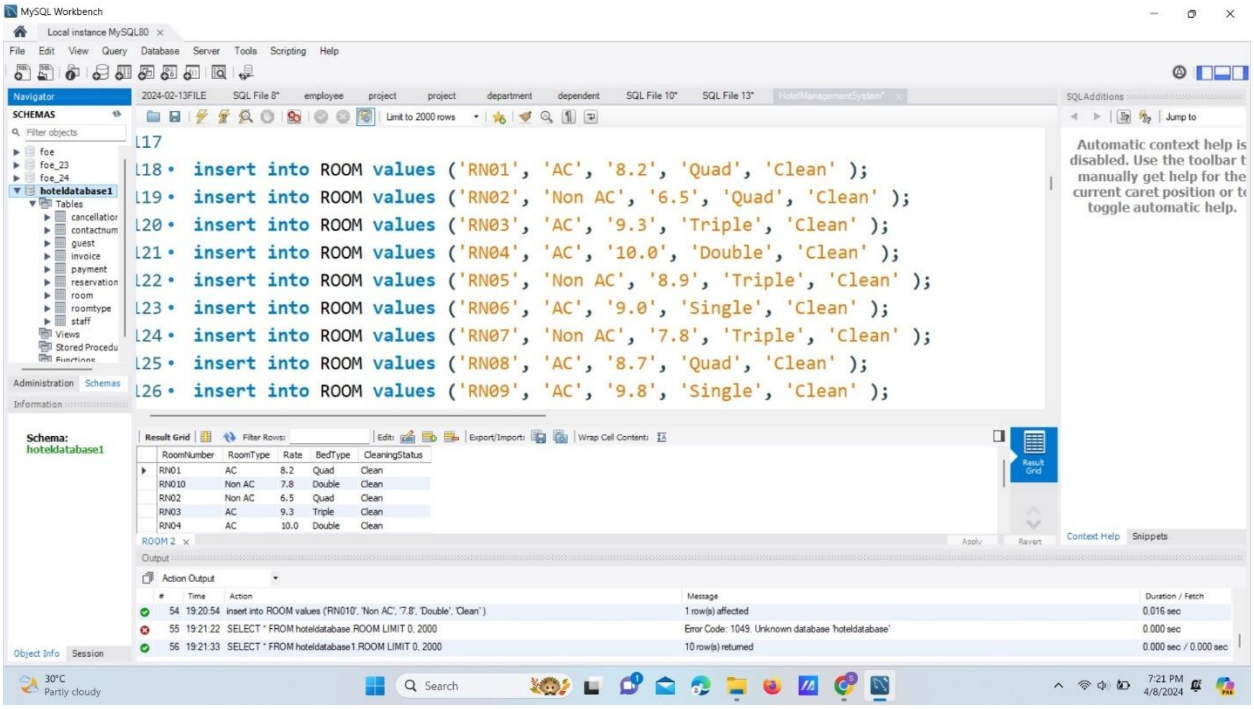
Staff Table



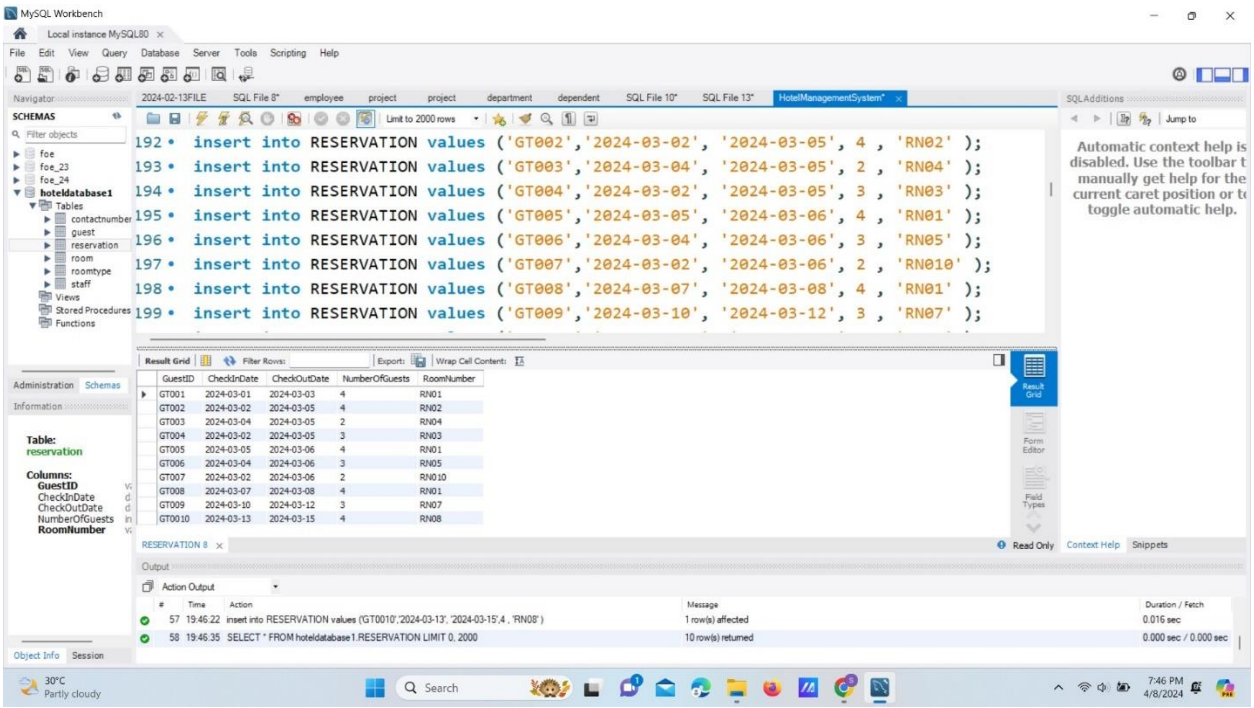
Room Type Table



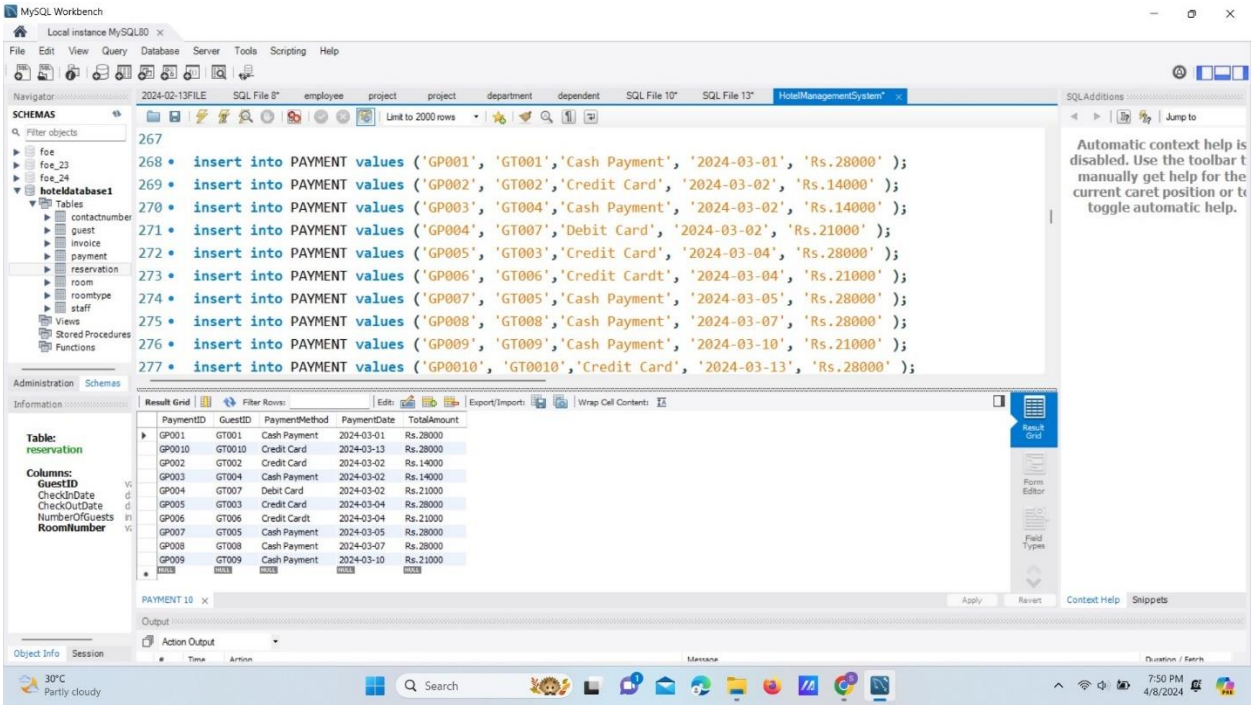
Room Table



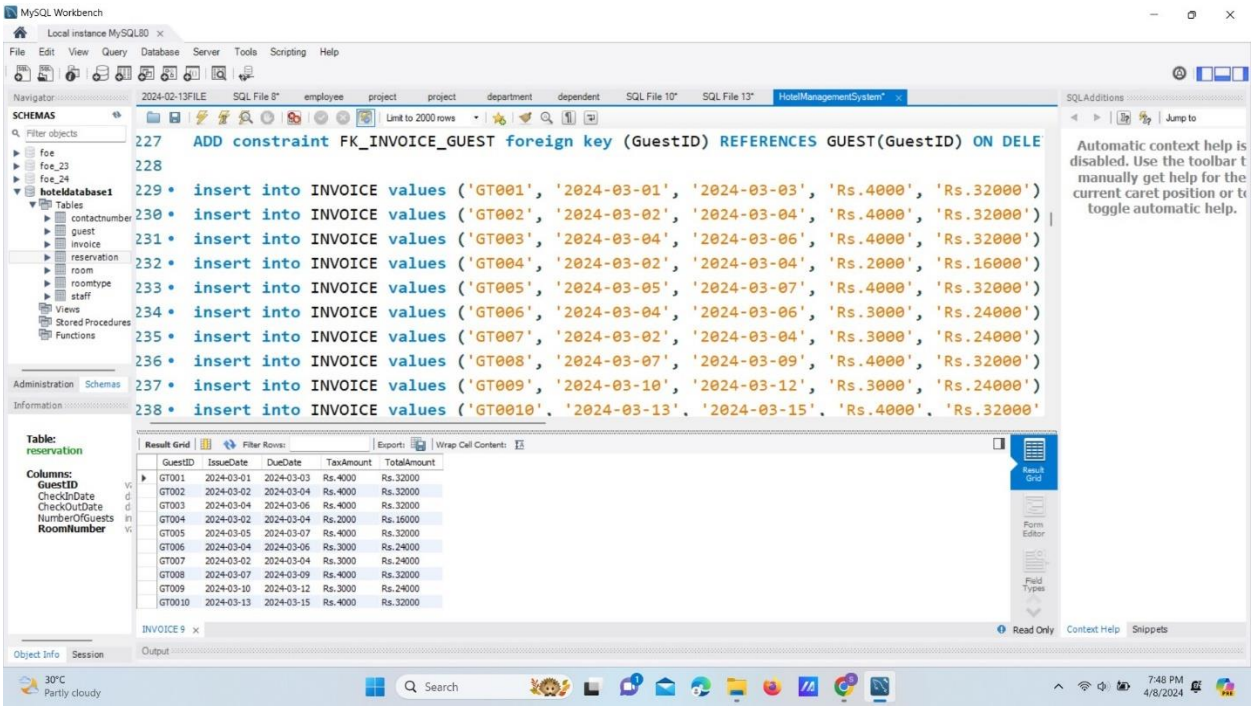
Reservation Table



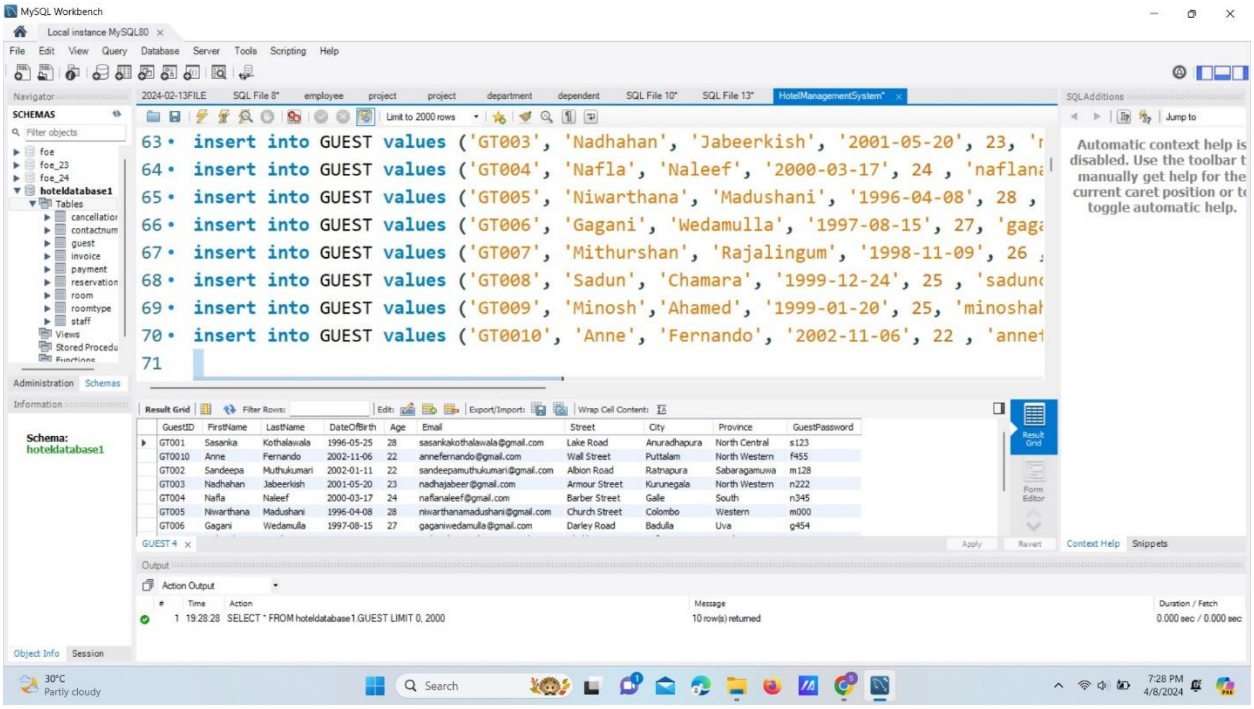
Payment Table



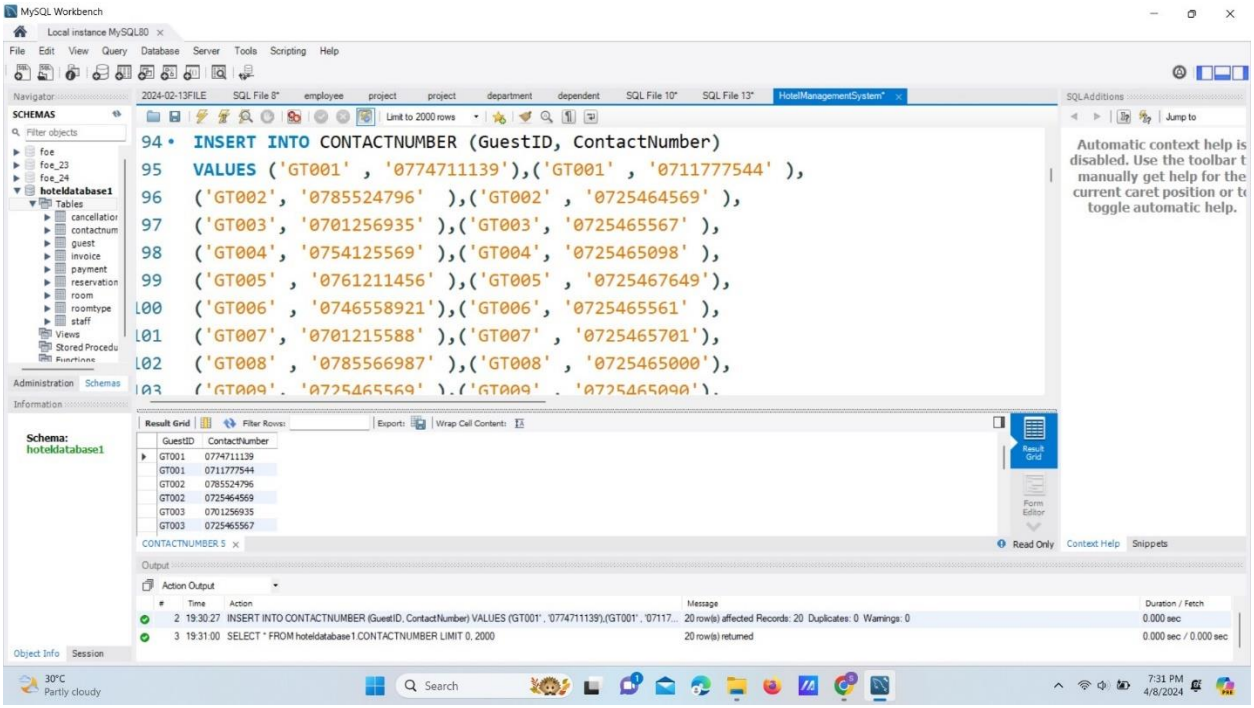
Invoice Table



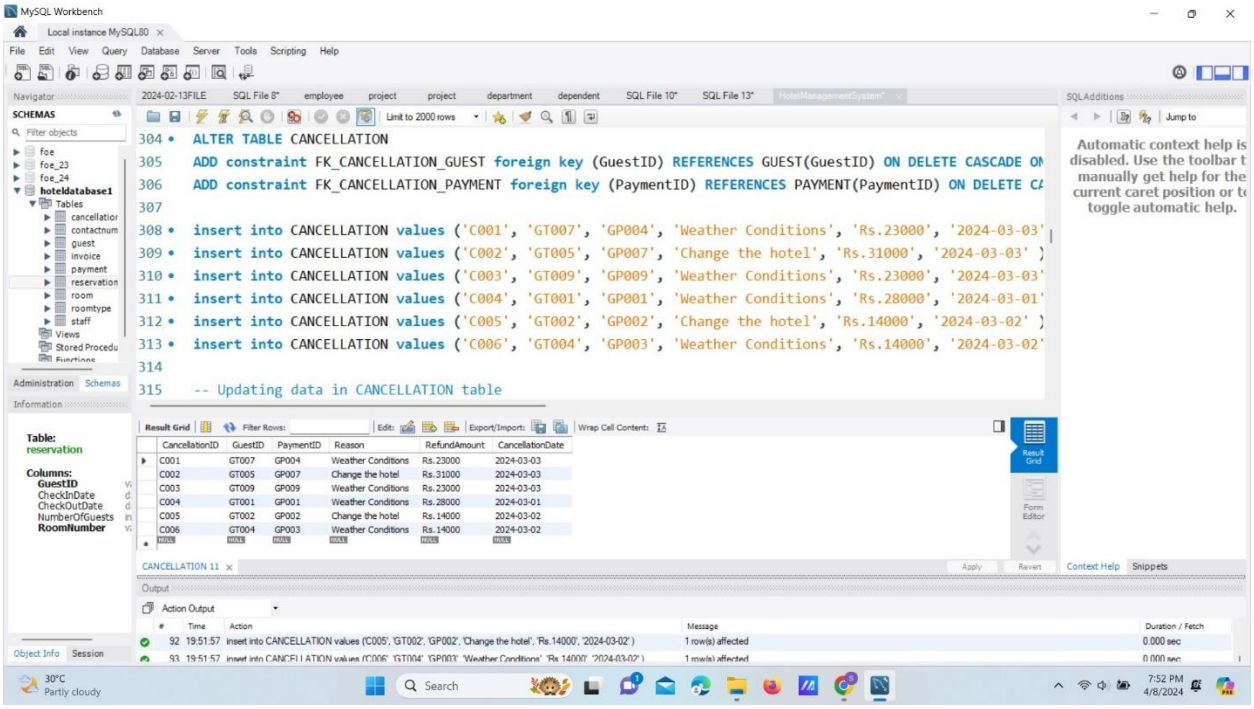
Guest Table



Contact Number Table

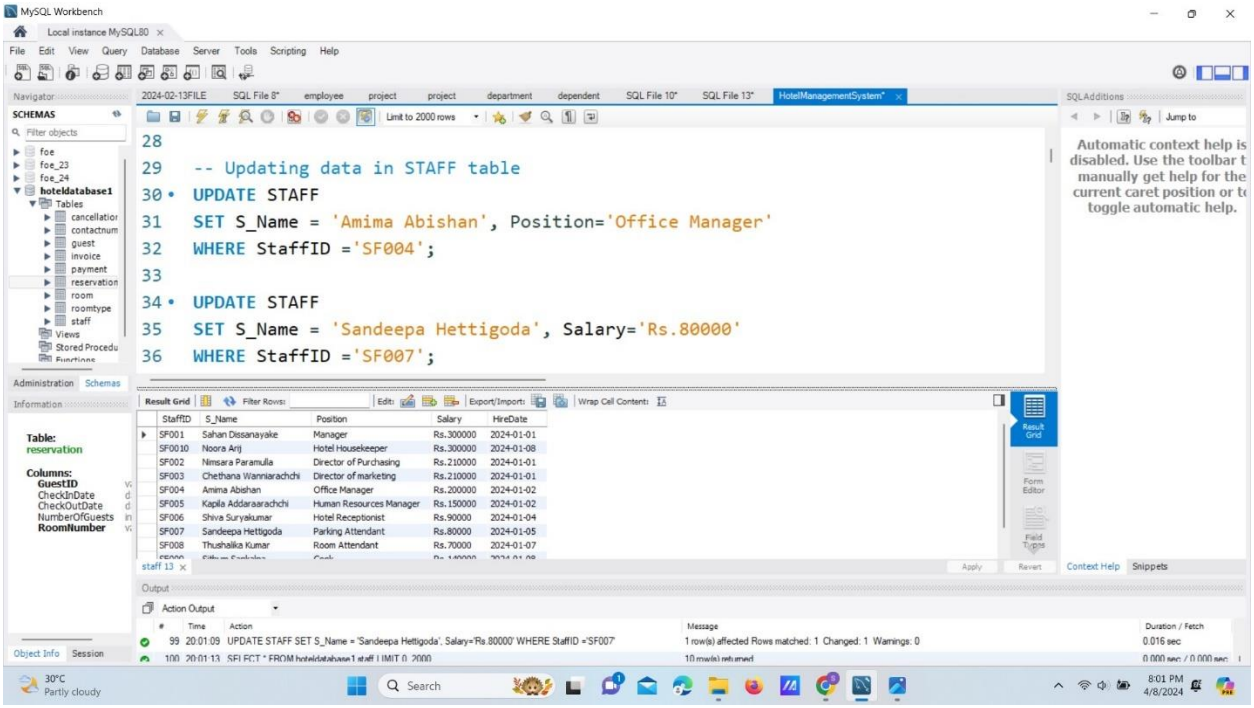


Cancellation Table

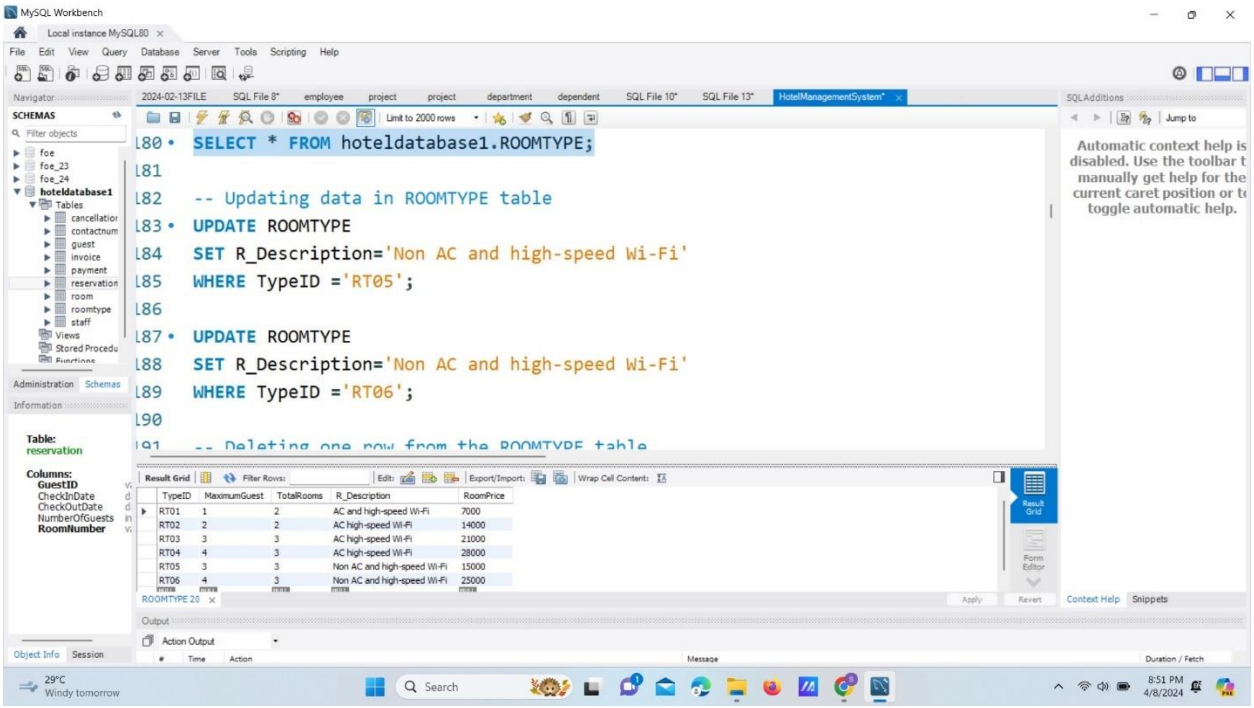


Update Operations

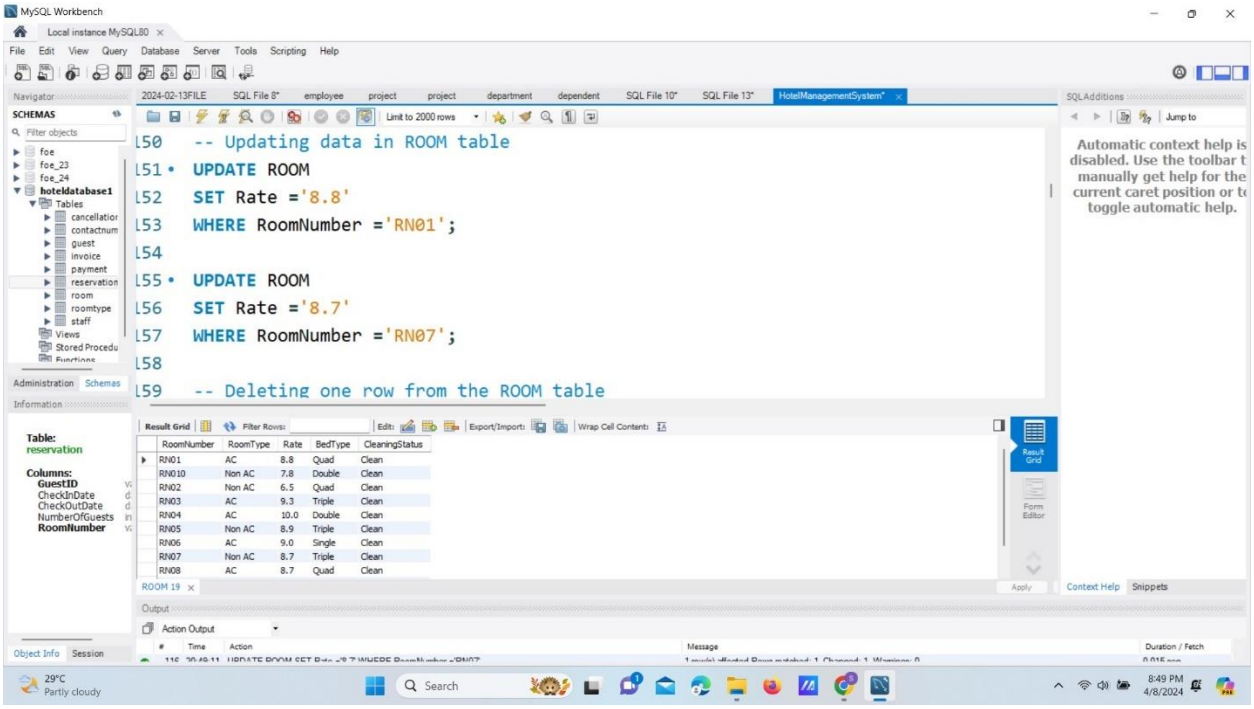
Staff Table



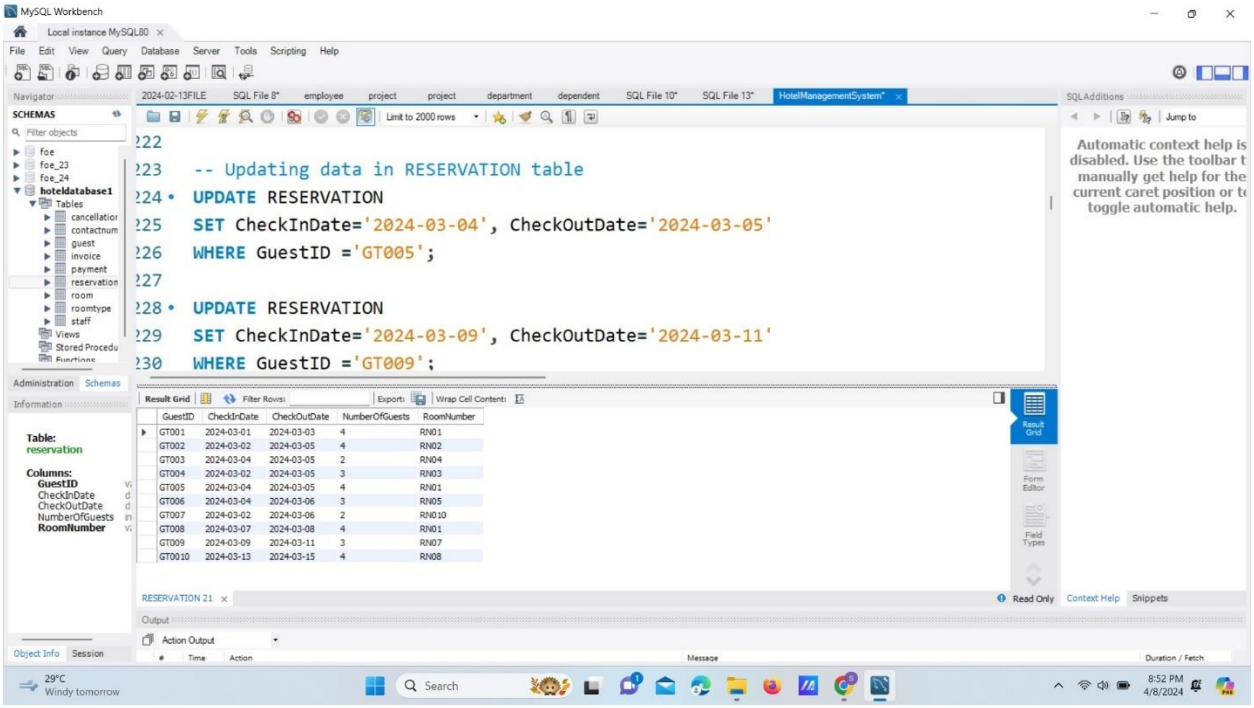
Room Type Table



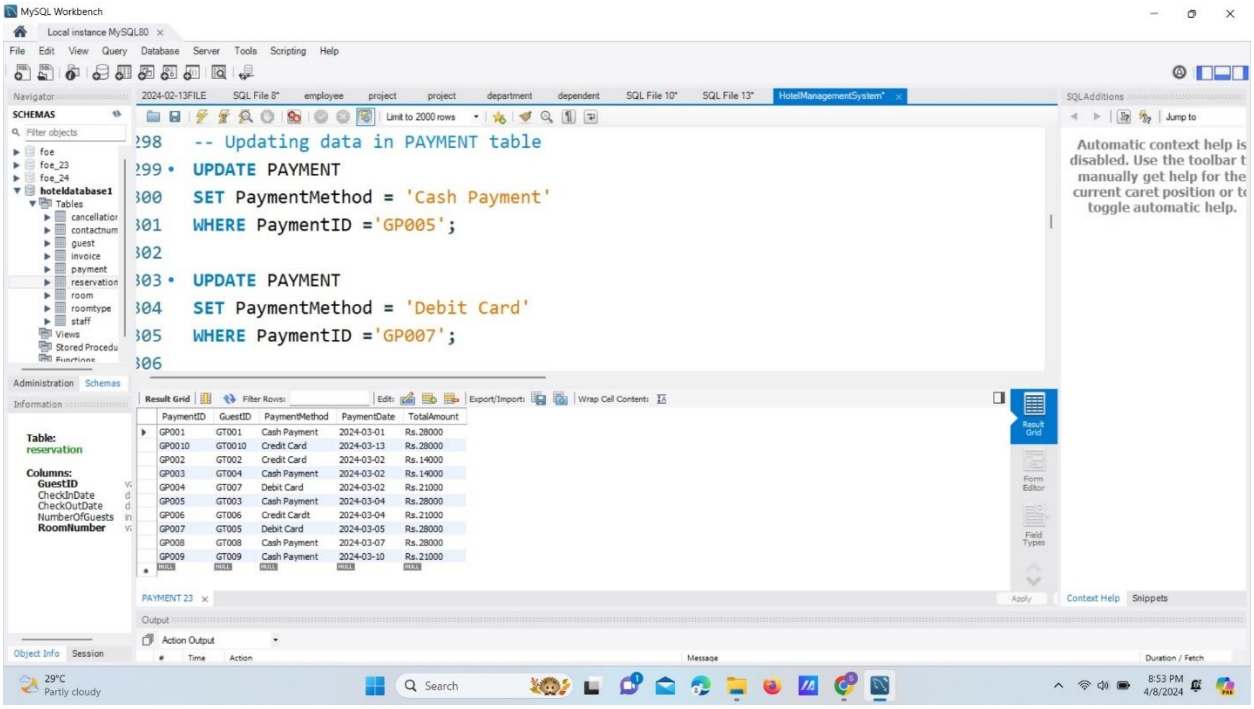
Room Table



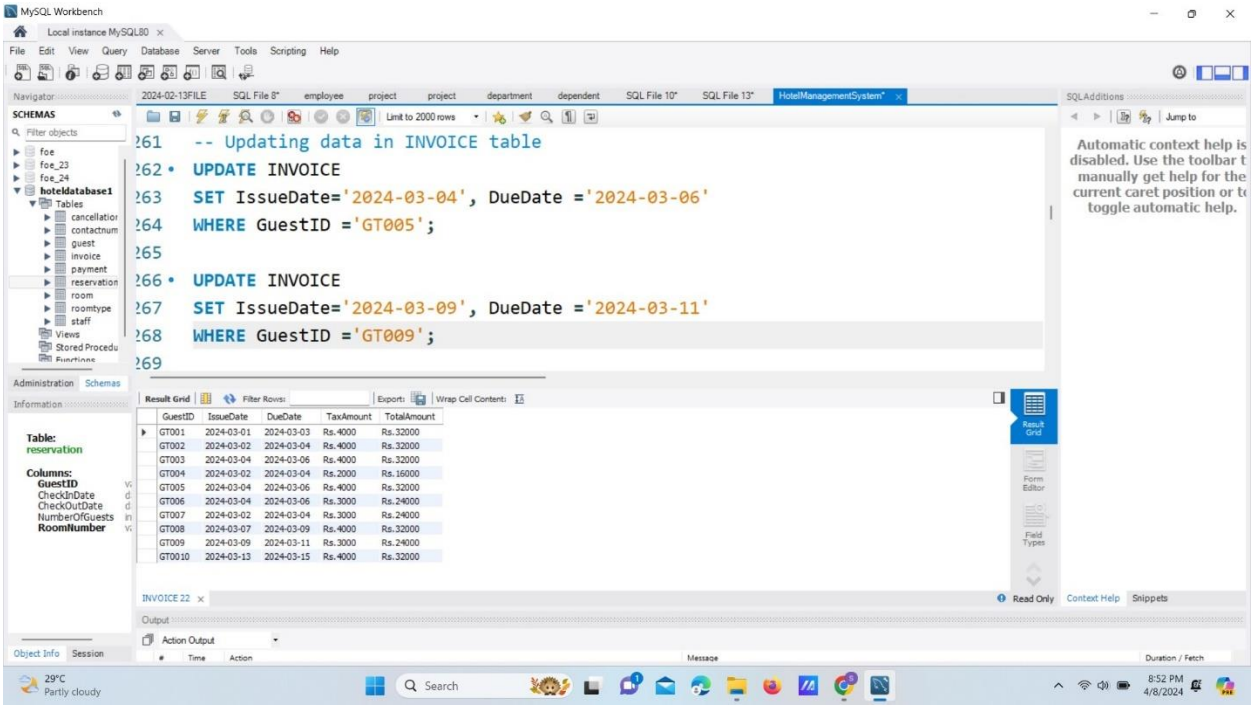
Reservation Table



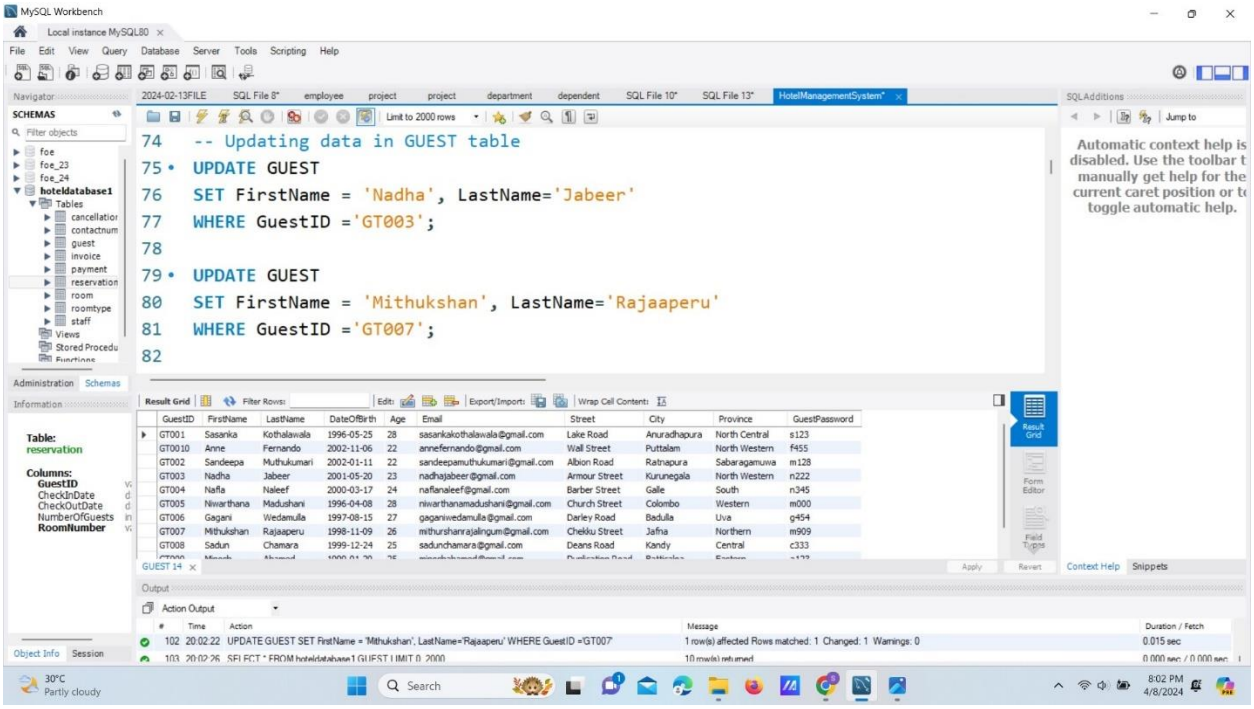
Payment Table



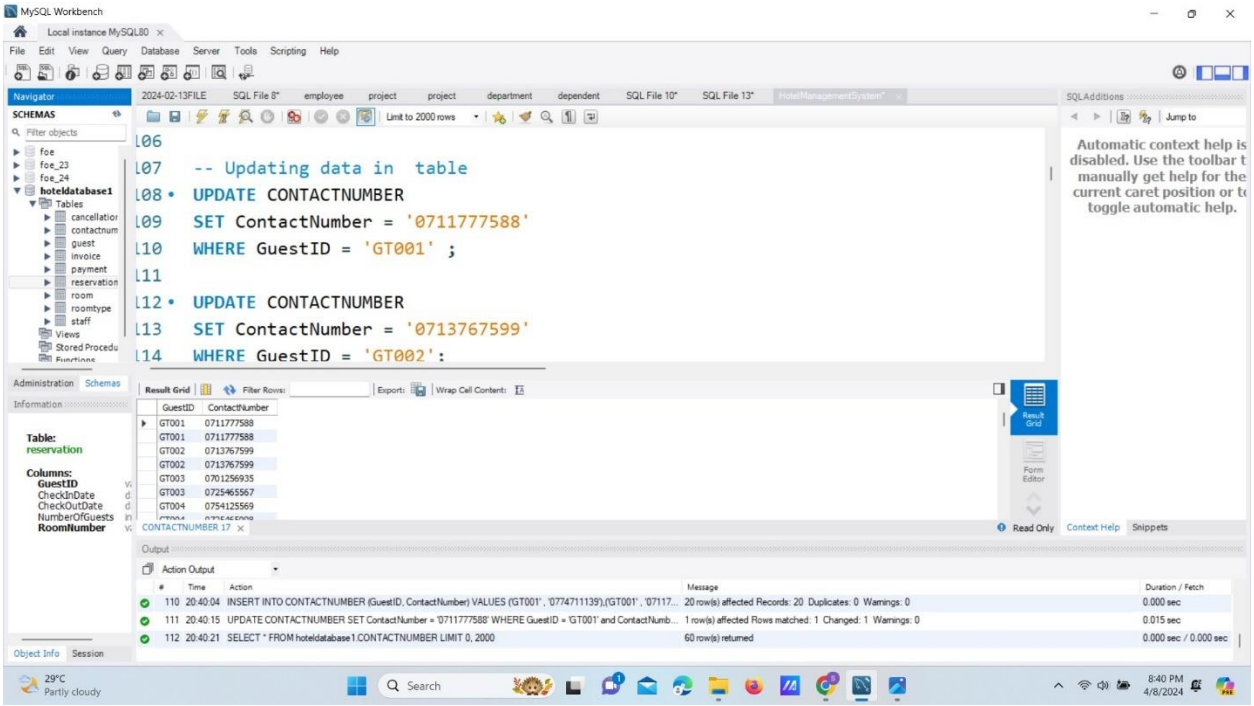
Invoice Table



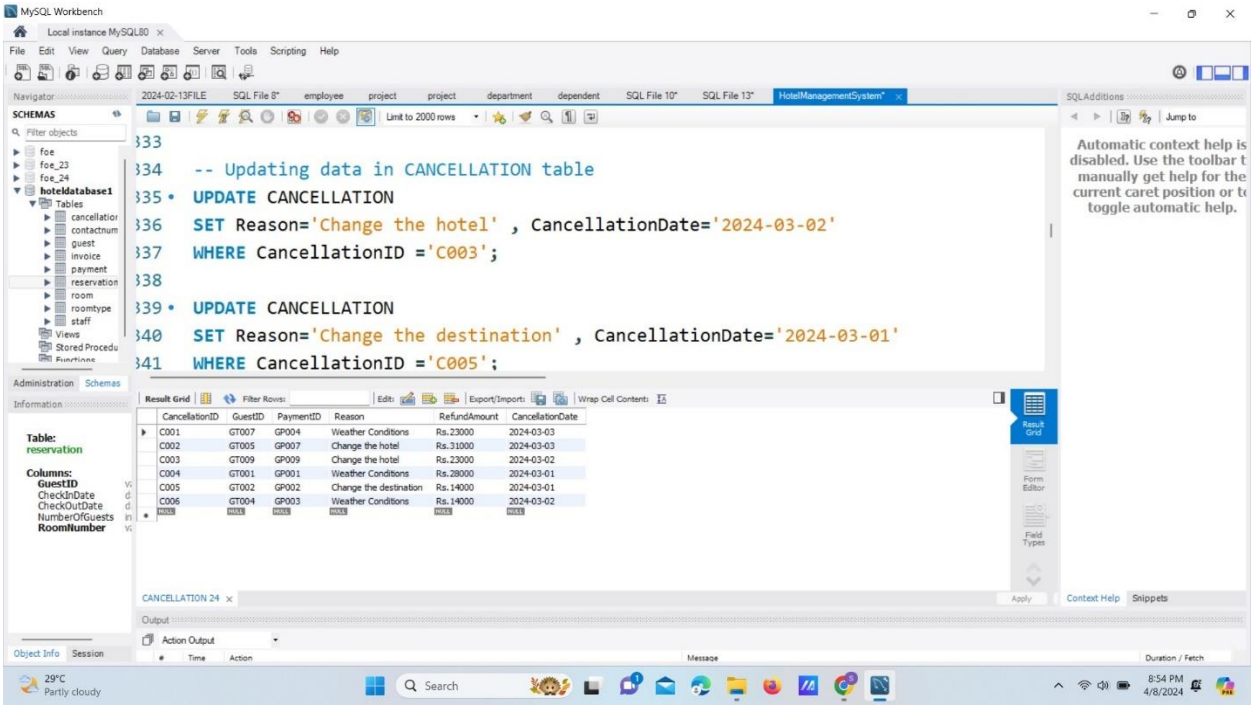
Guest Table



Contact Number Table

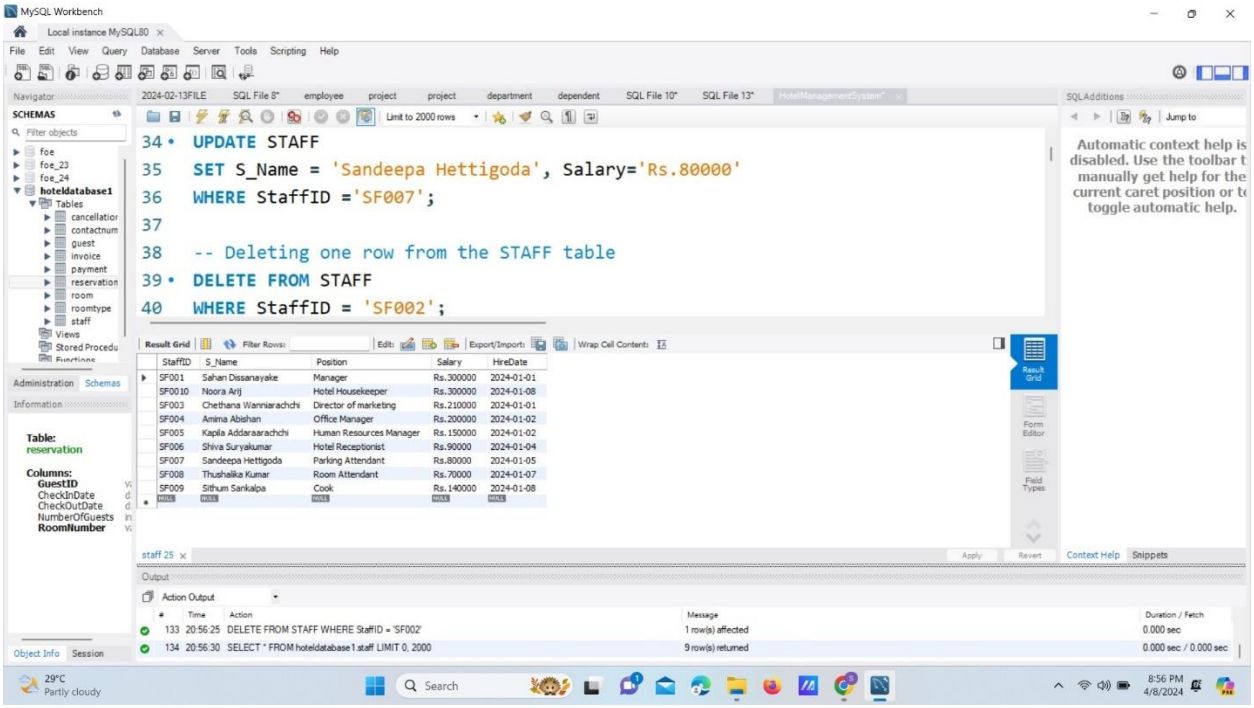


Cancellation Table

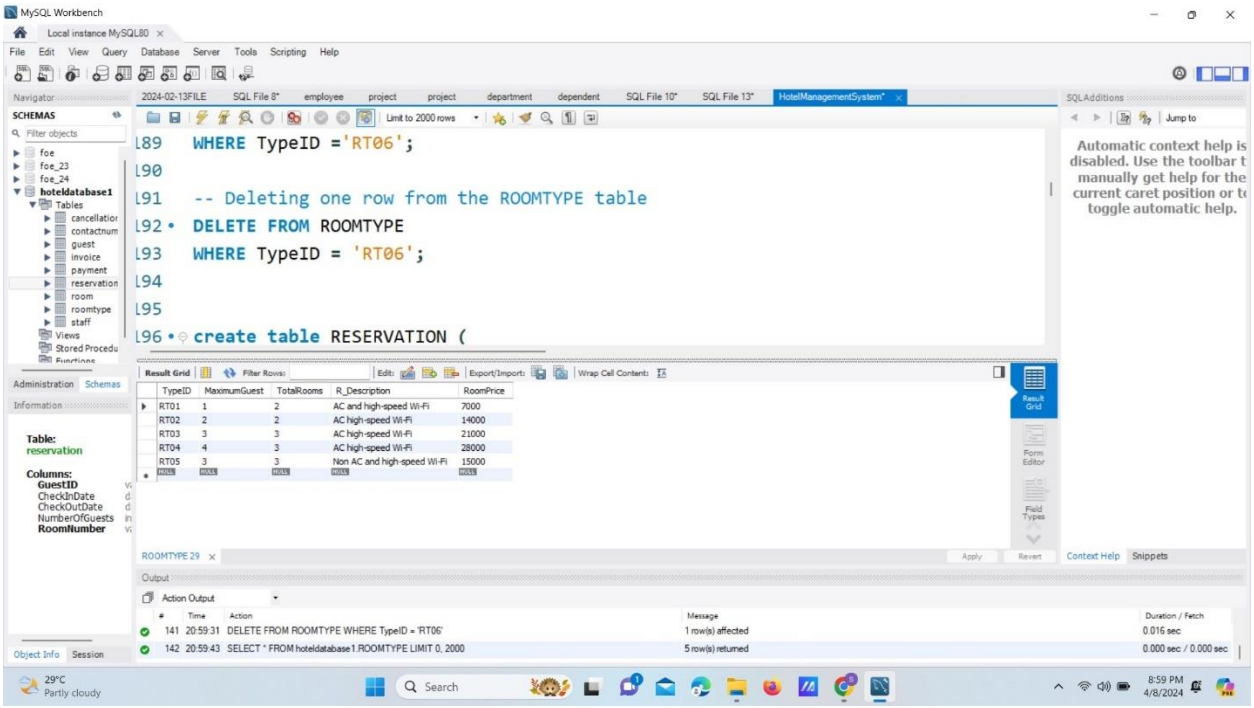


Delete Operations

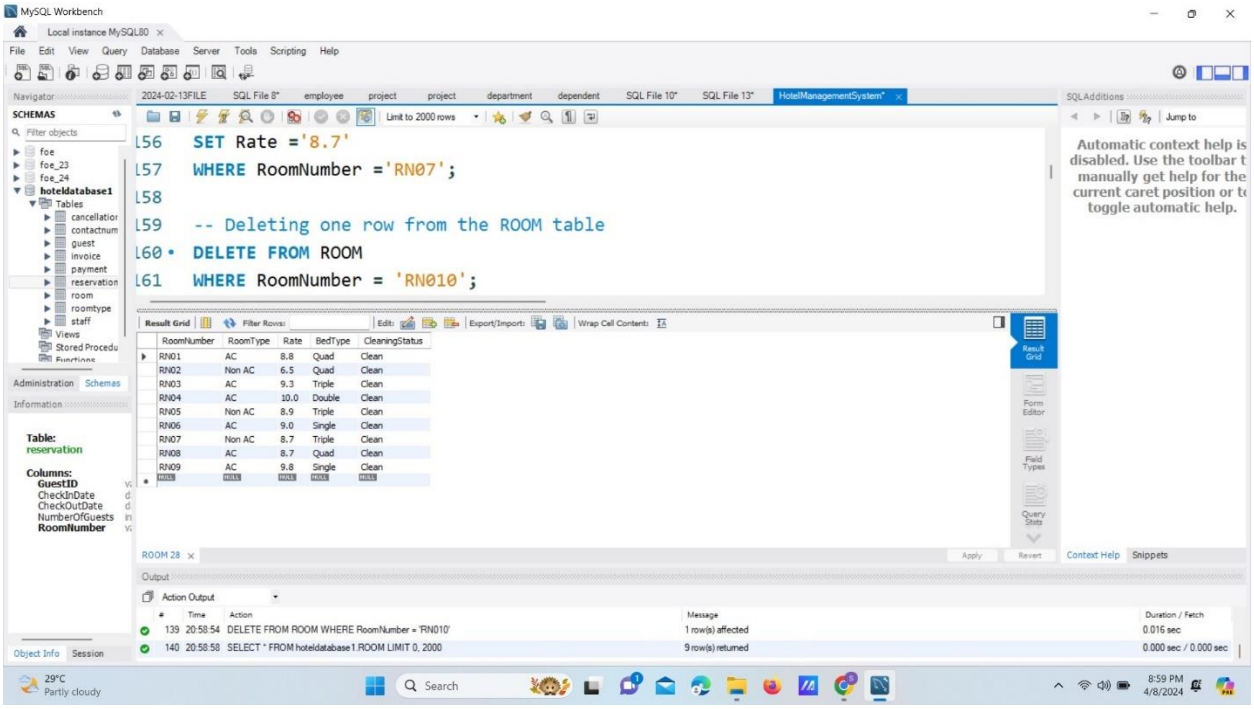
Staff Table



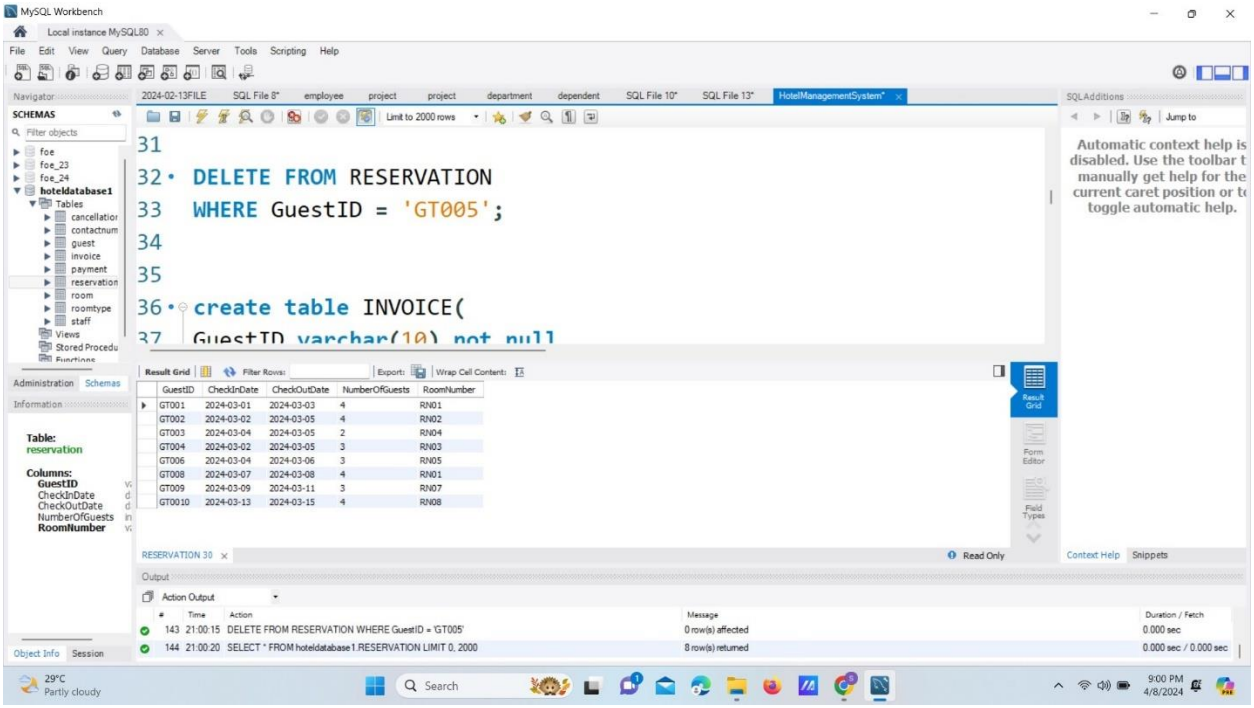
Room Type Table



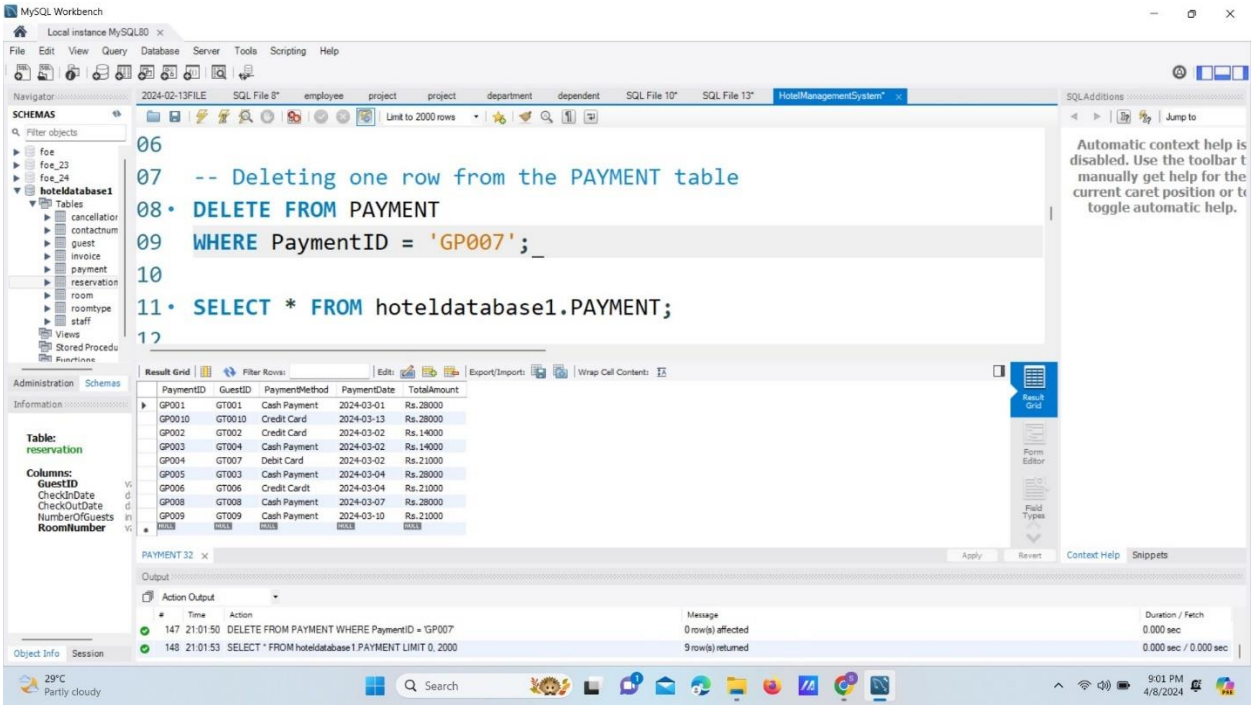
Room Table



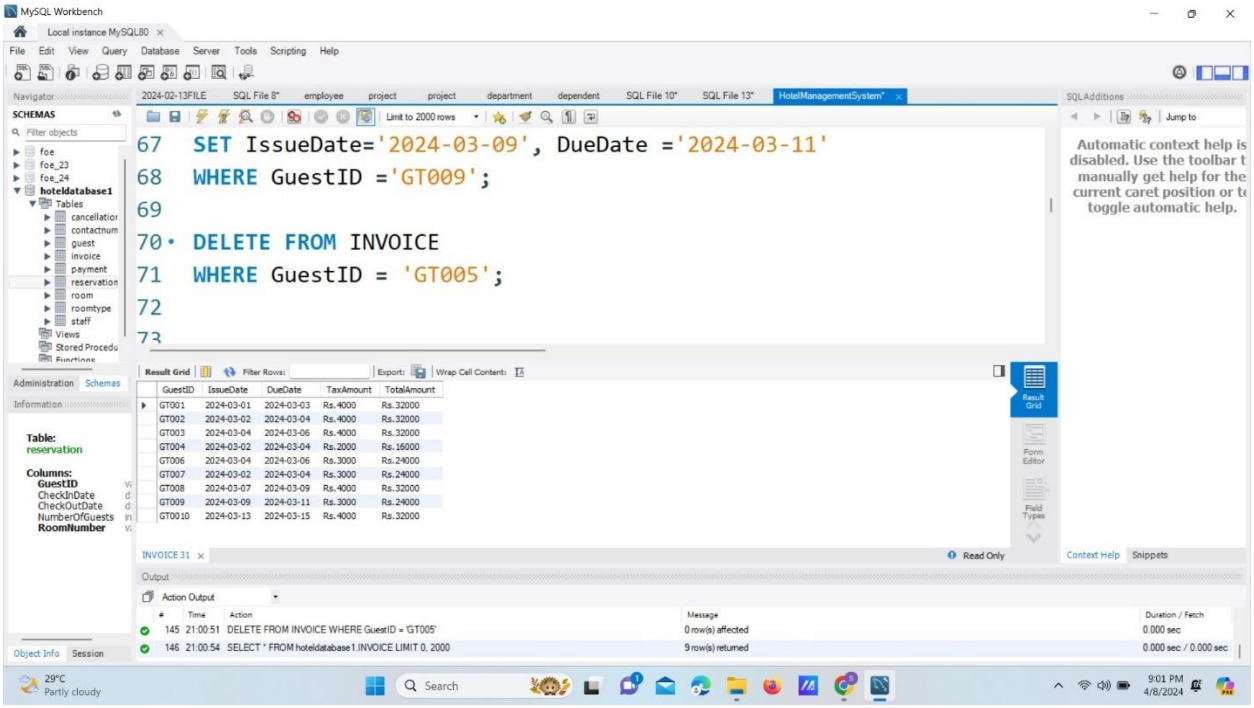
Reservation Table



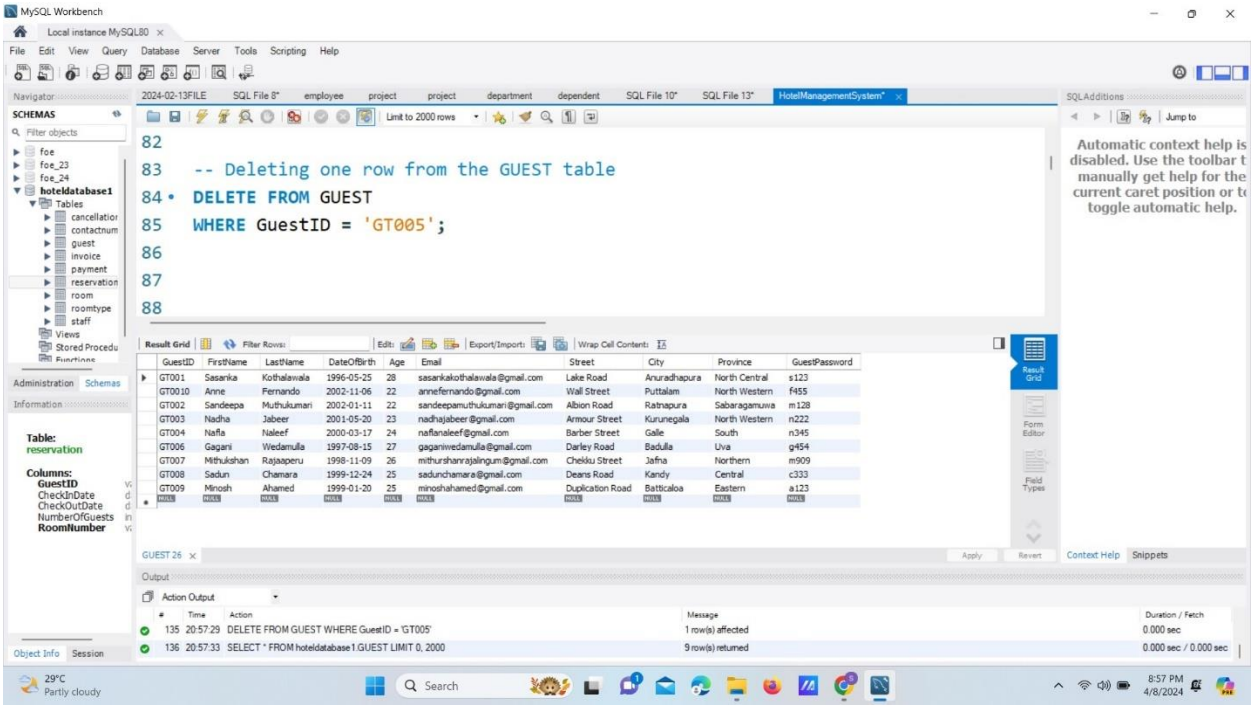
Payment Table



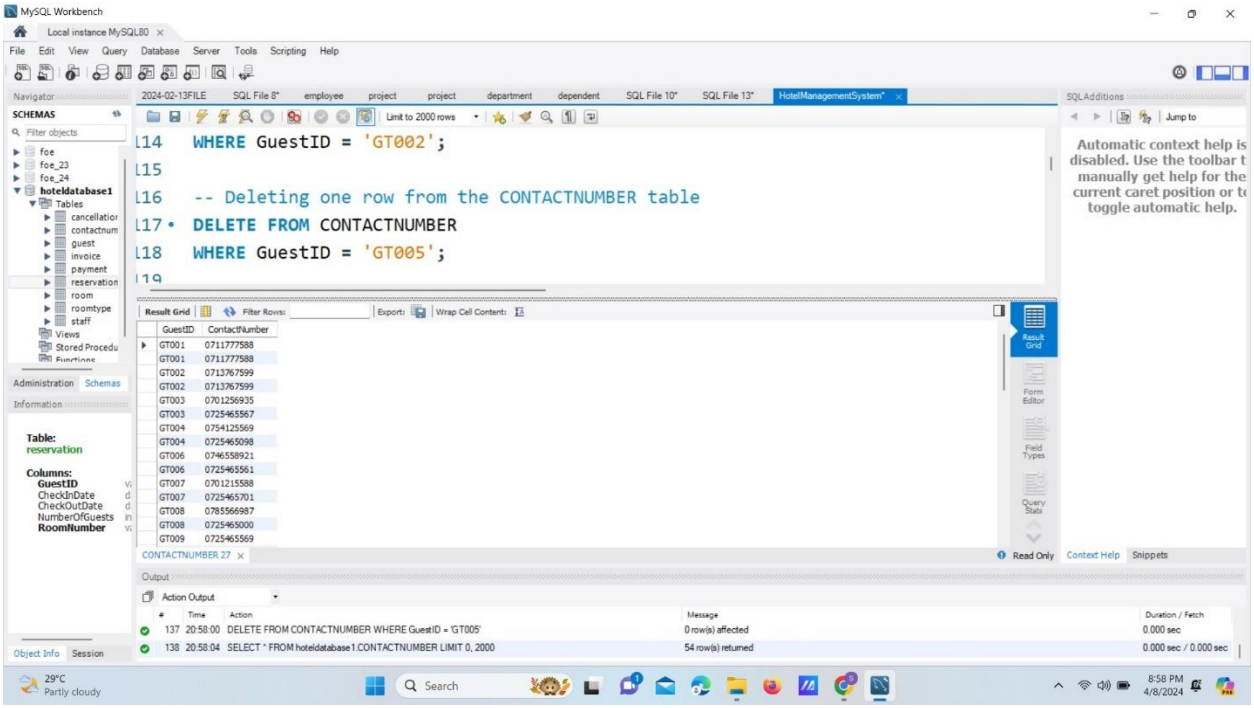
Invoice Table



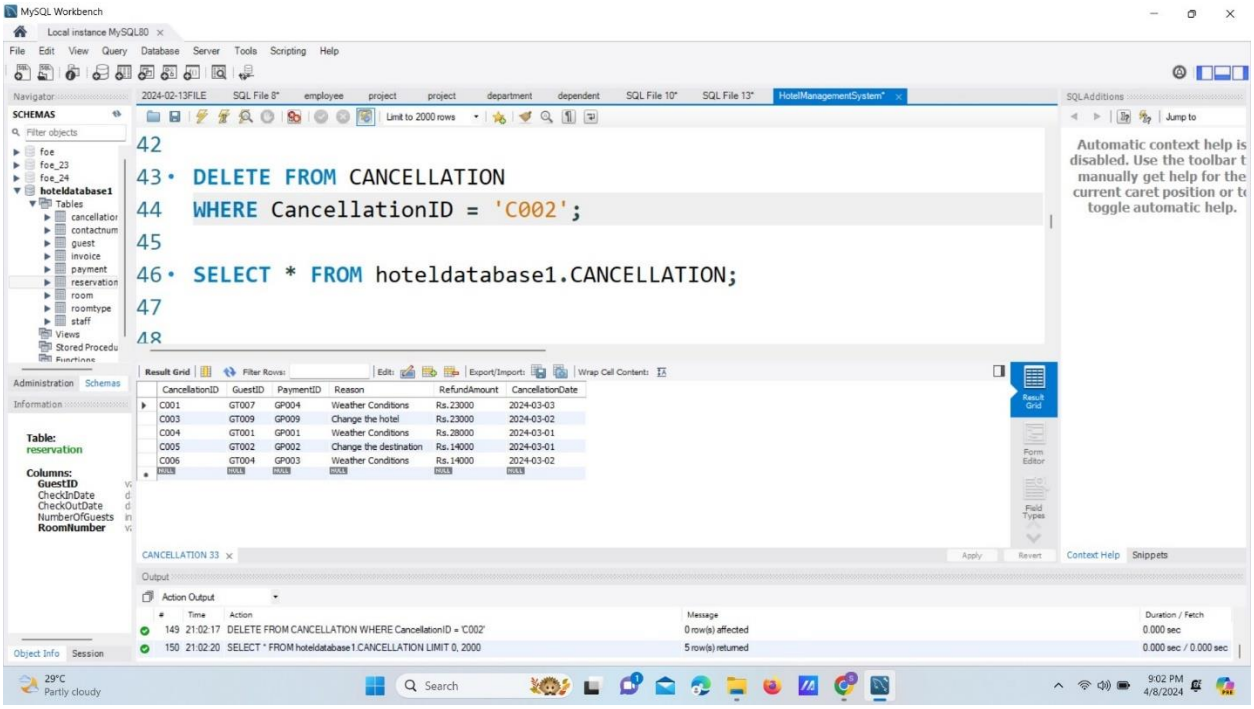
Guest Table



Contact Number Table



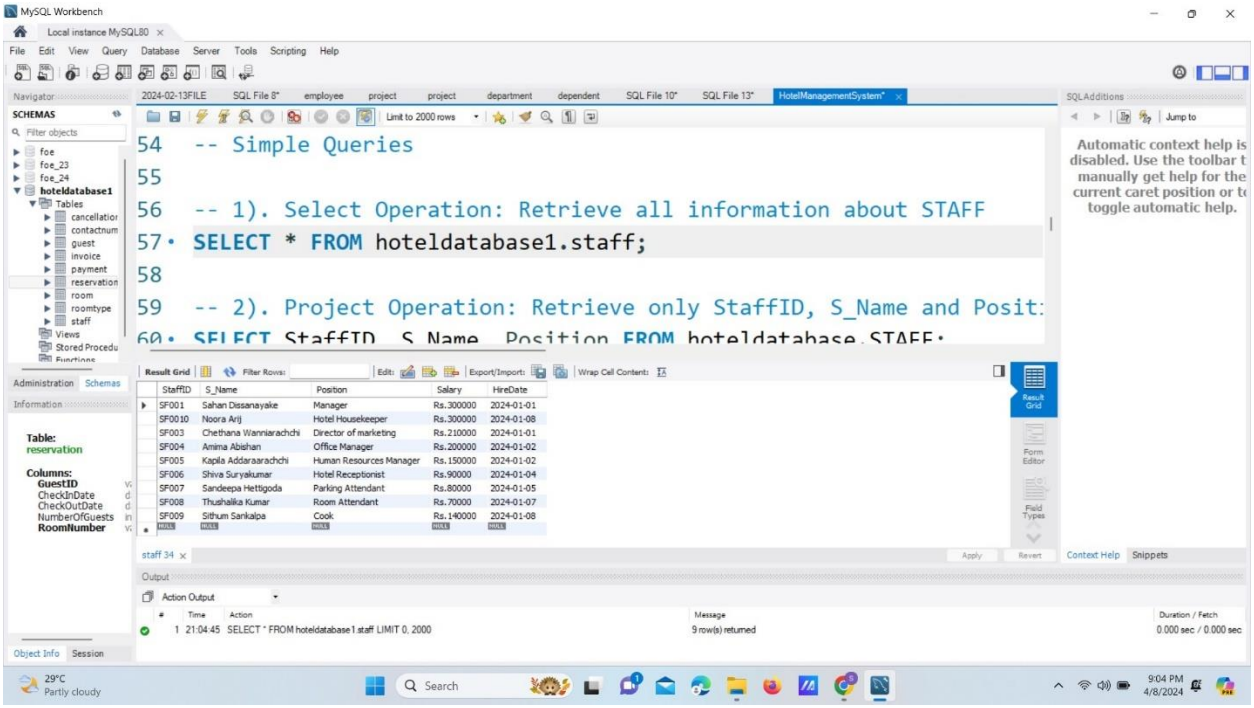
Cancellation Table



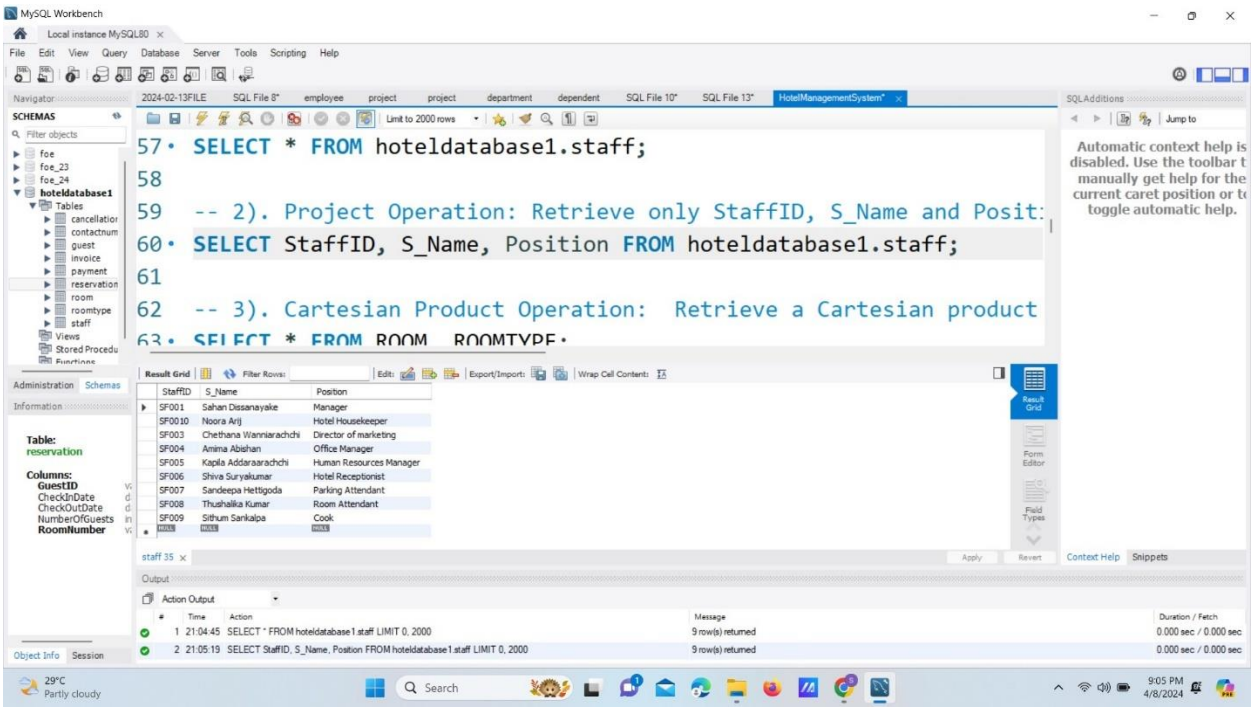
Chapter 4 – Transaction

Simple Queries

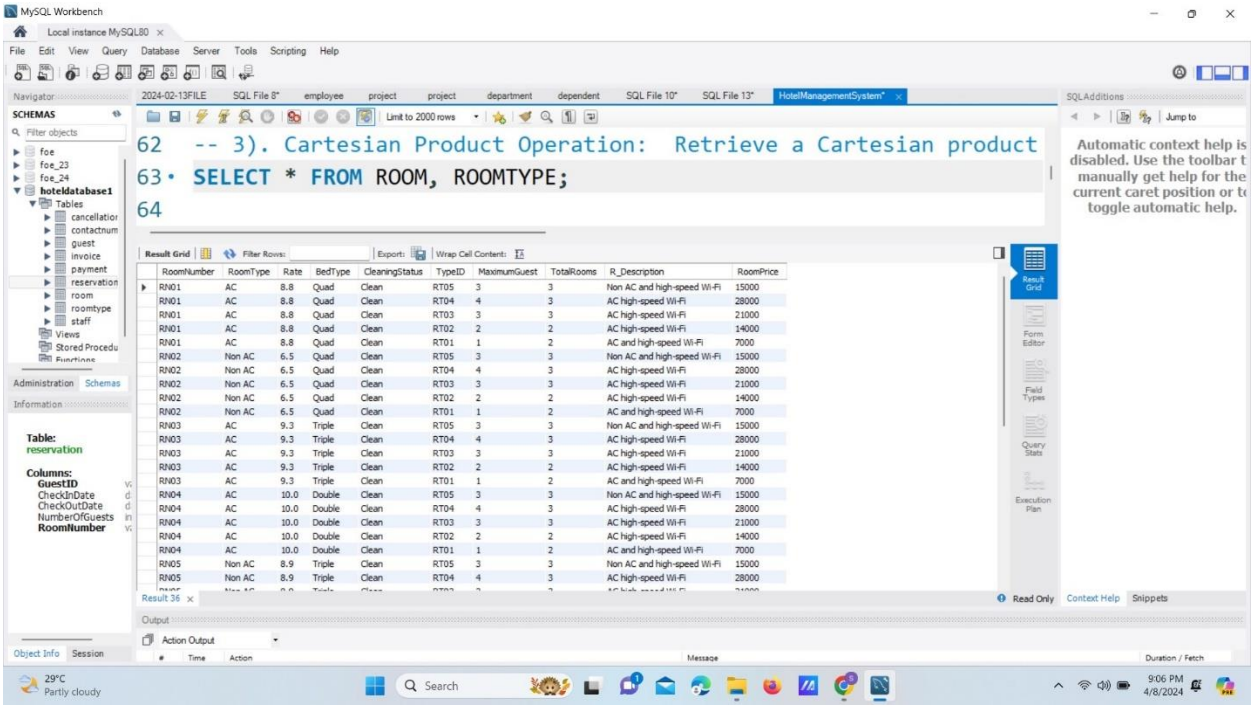
Select Operation



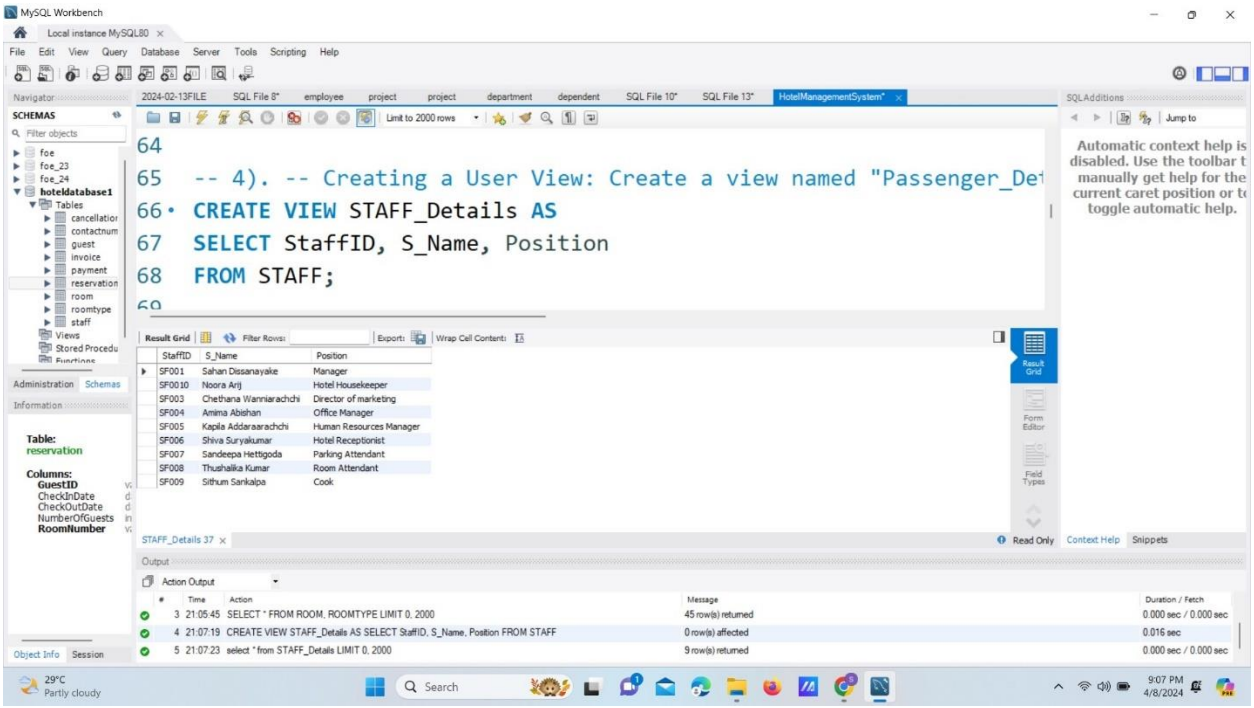
Project Operation



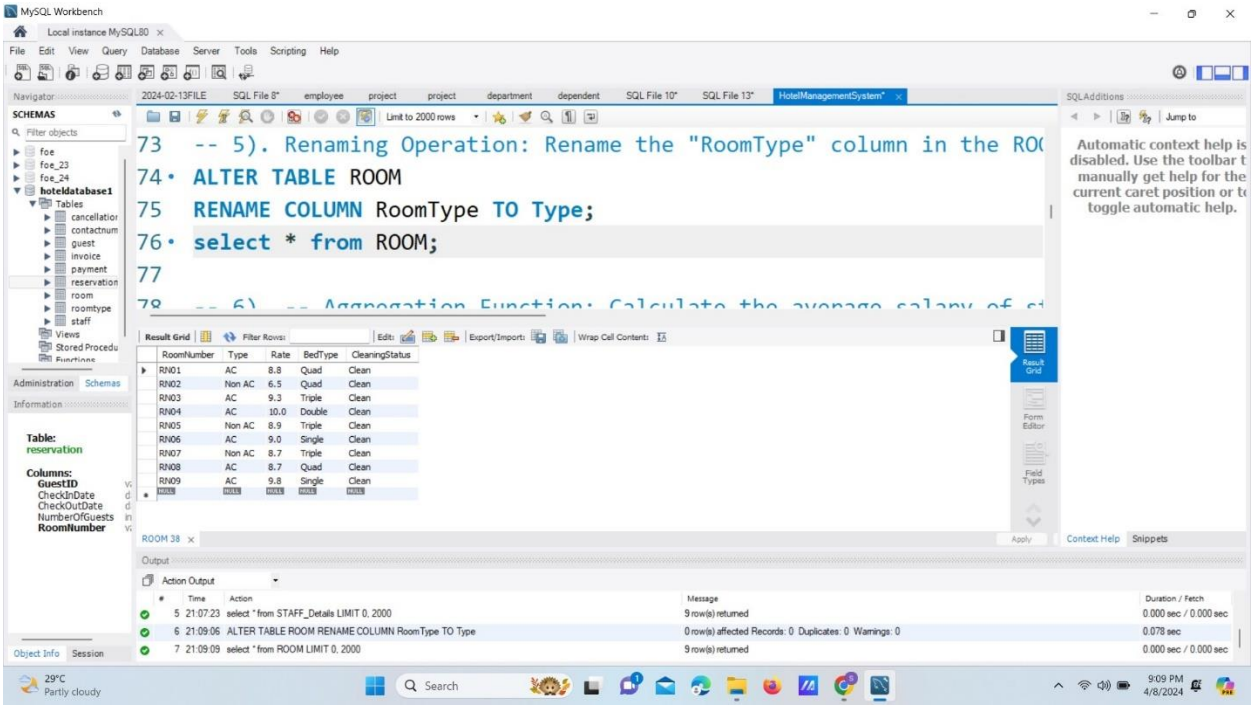
Product Operation



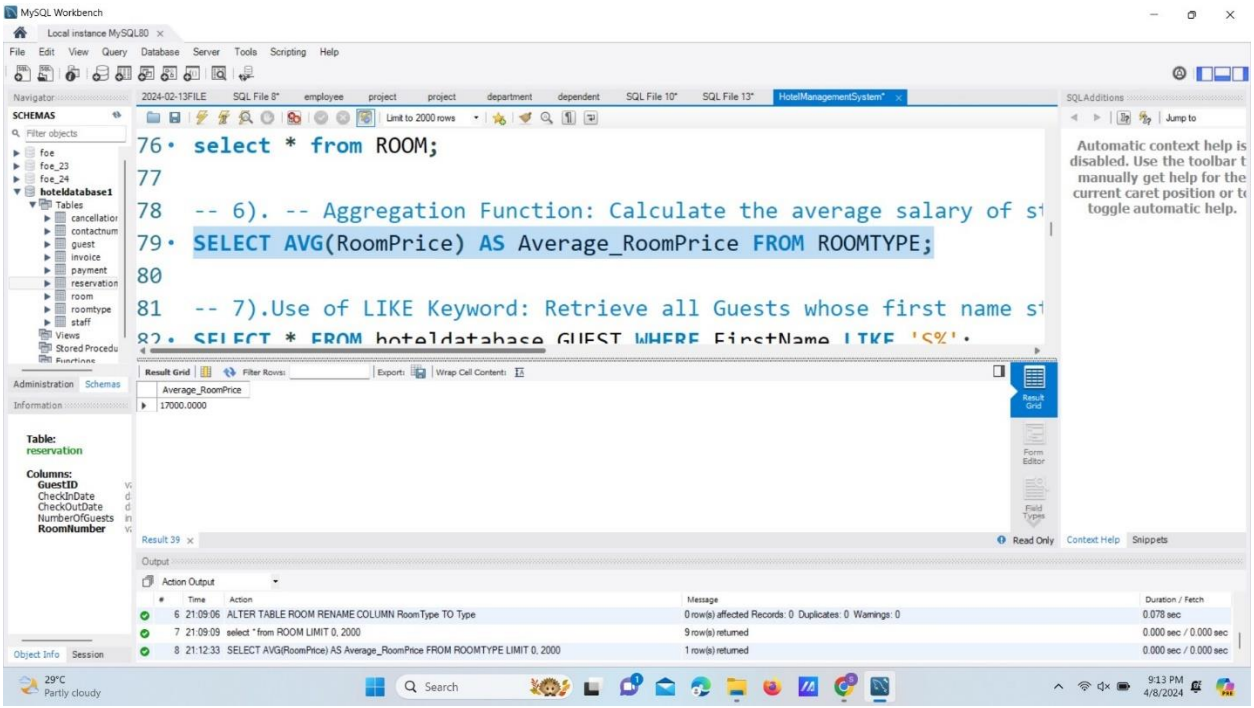
Create a User View



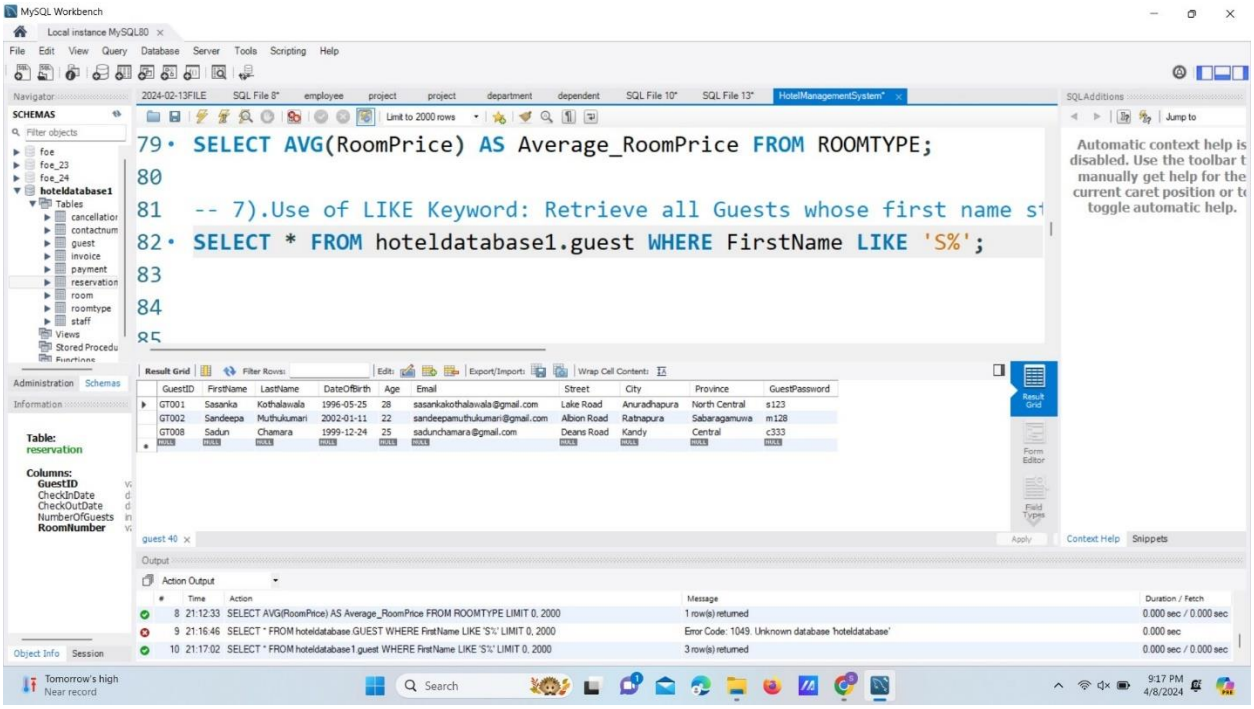
Renaming Operation



Aggregation Function

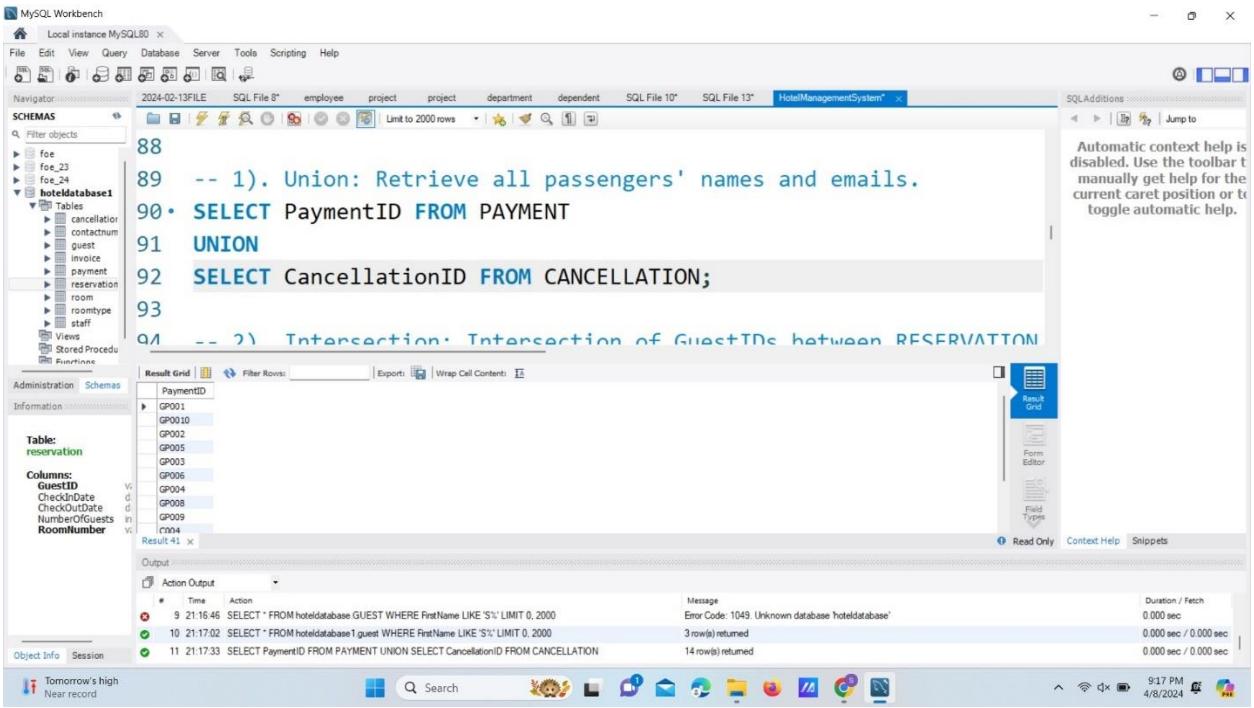


Use of LIKE Keyword

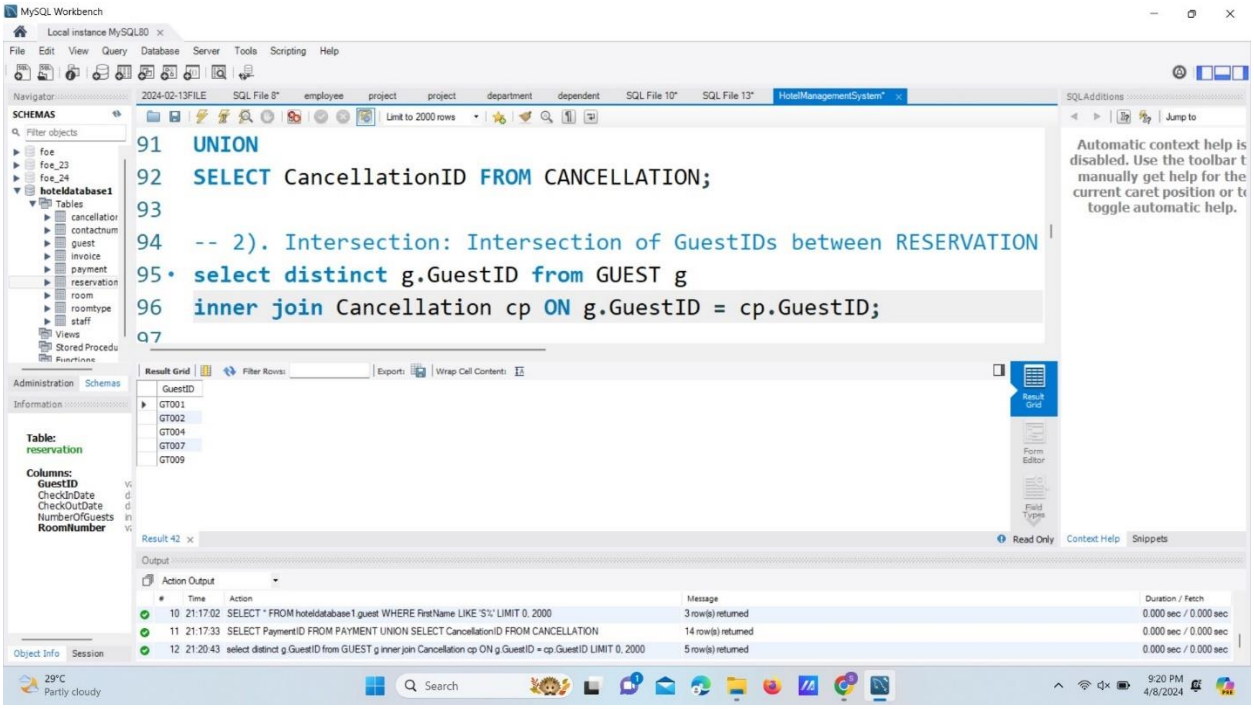


Complex Queries

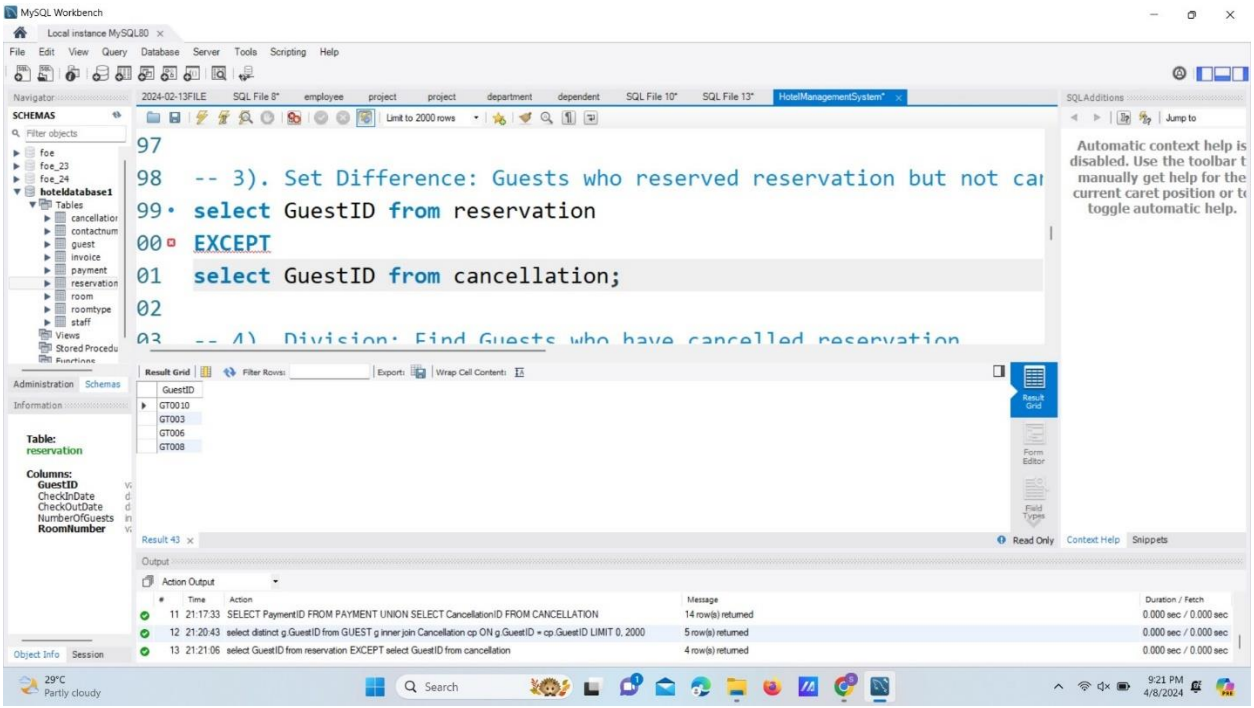
Union Operation



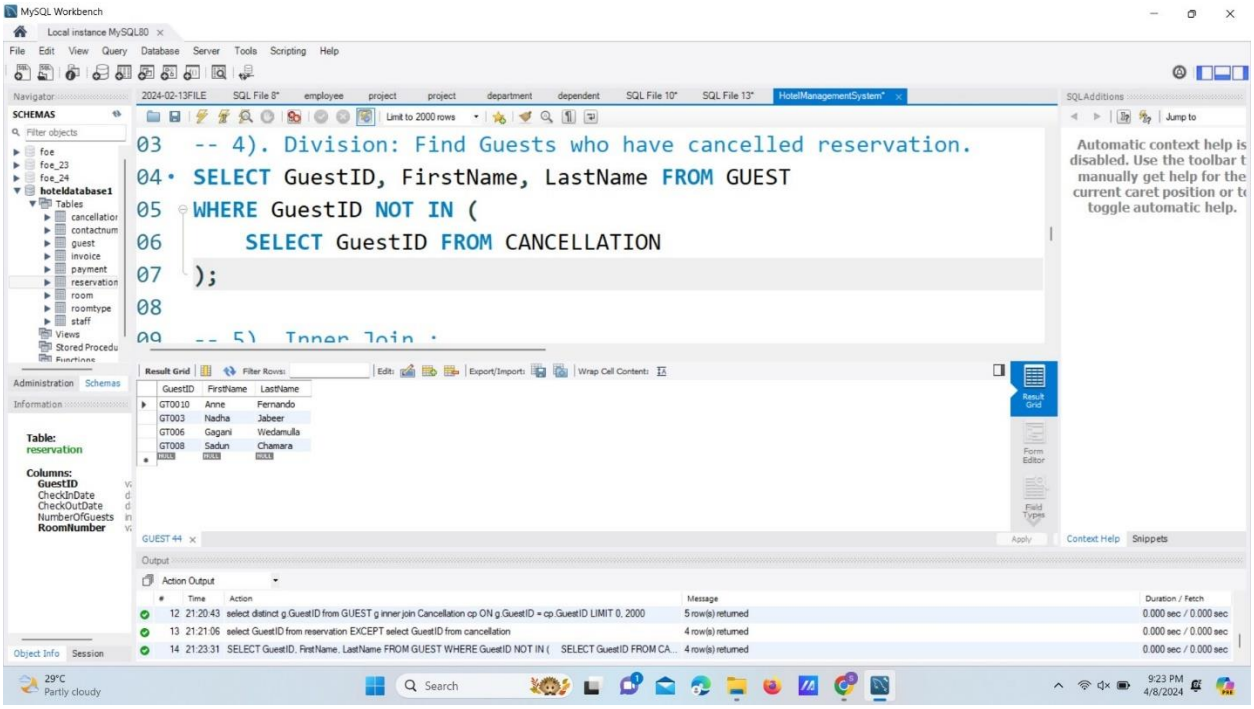
Intersection Operation



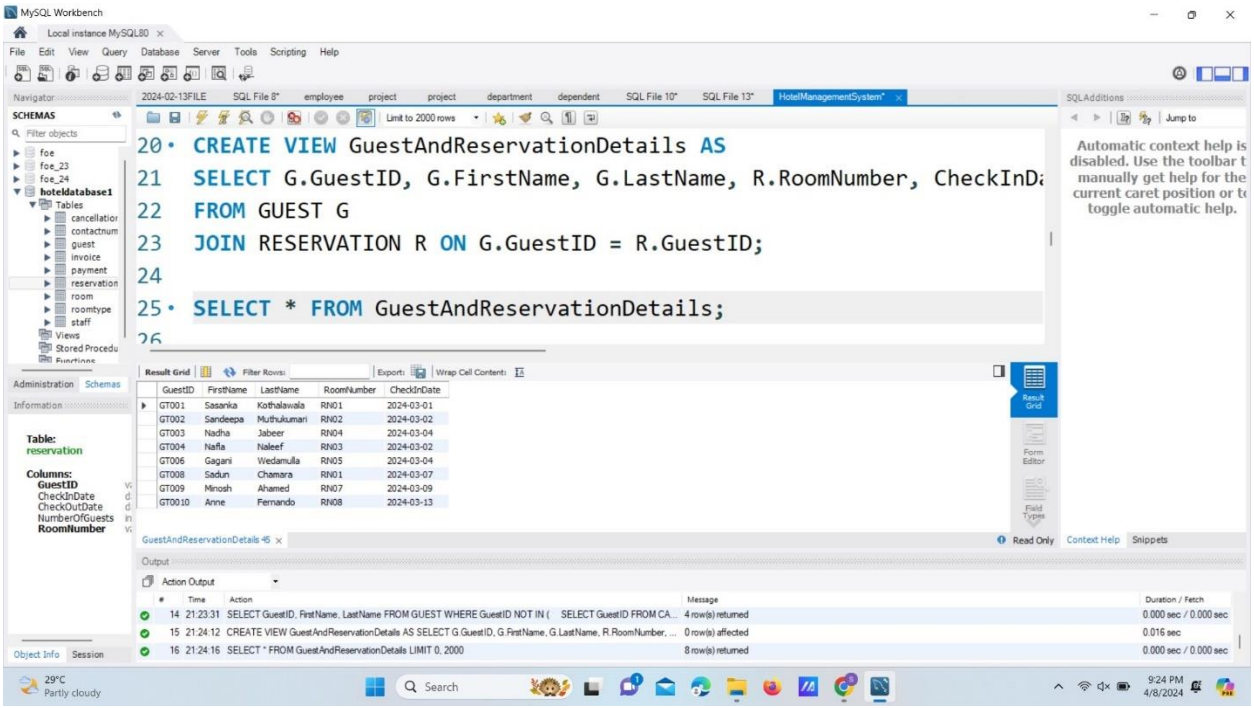
Set Difference Operation



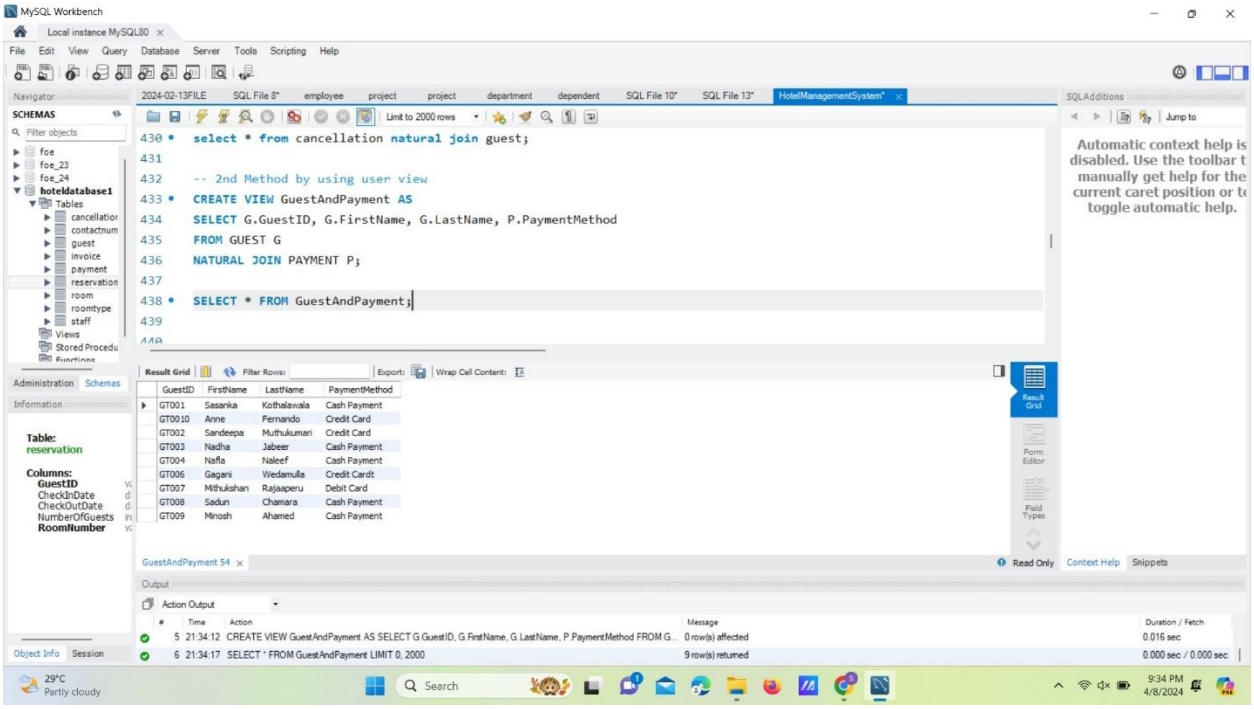
Division Operation



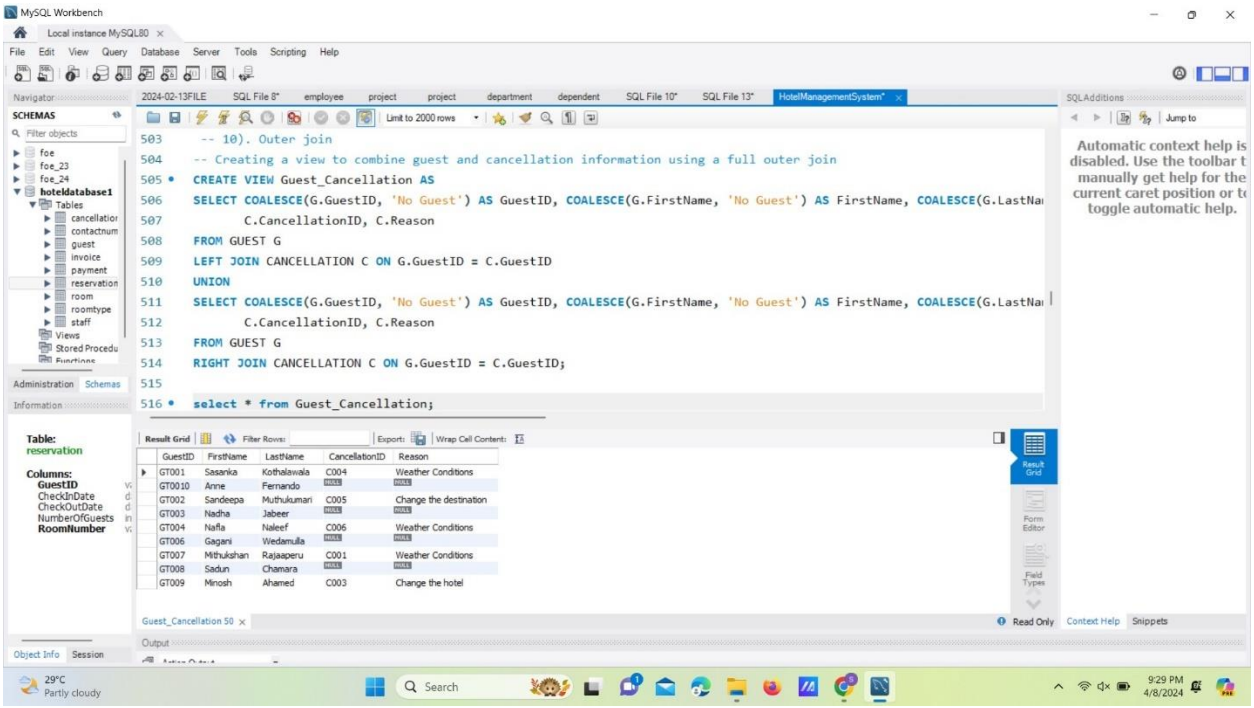
Inner Join Operation



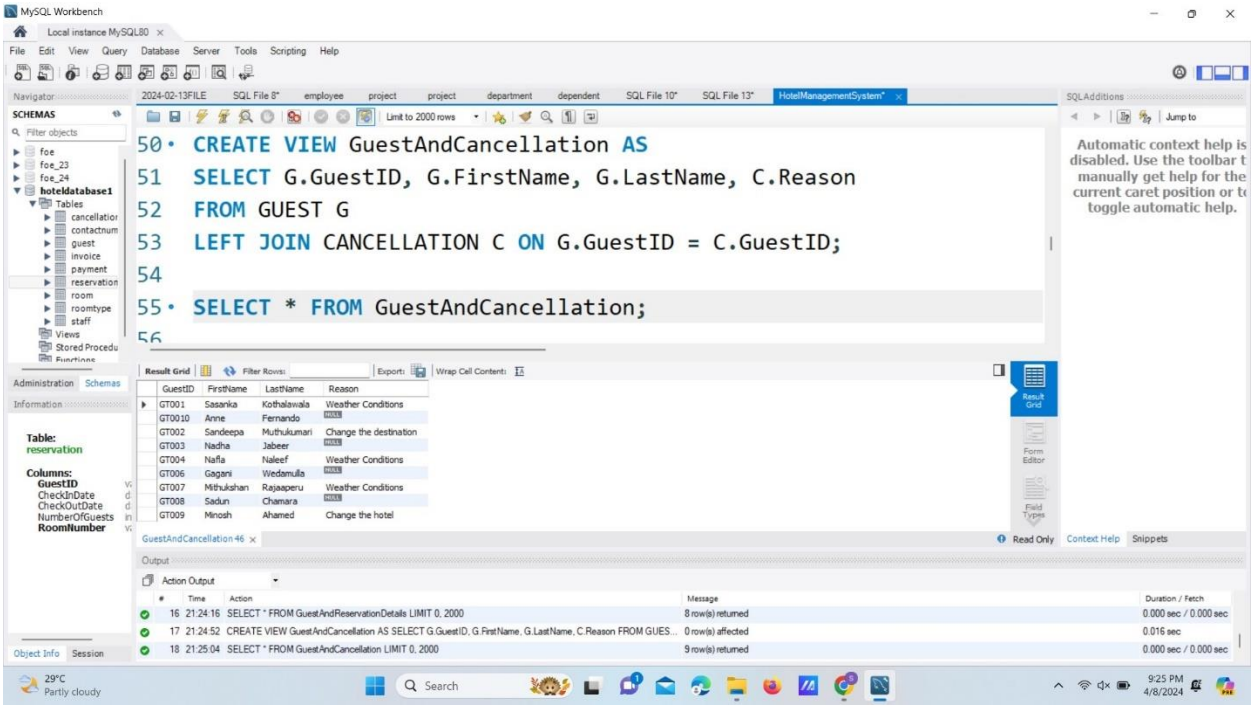
Natural Join Operation



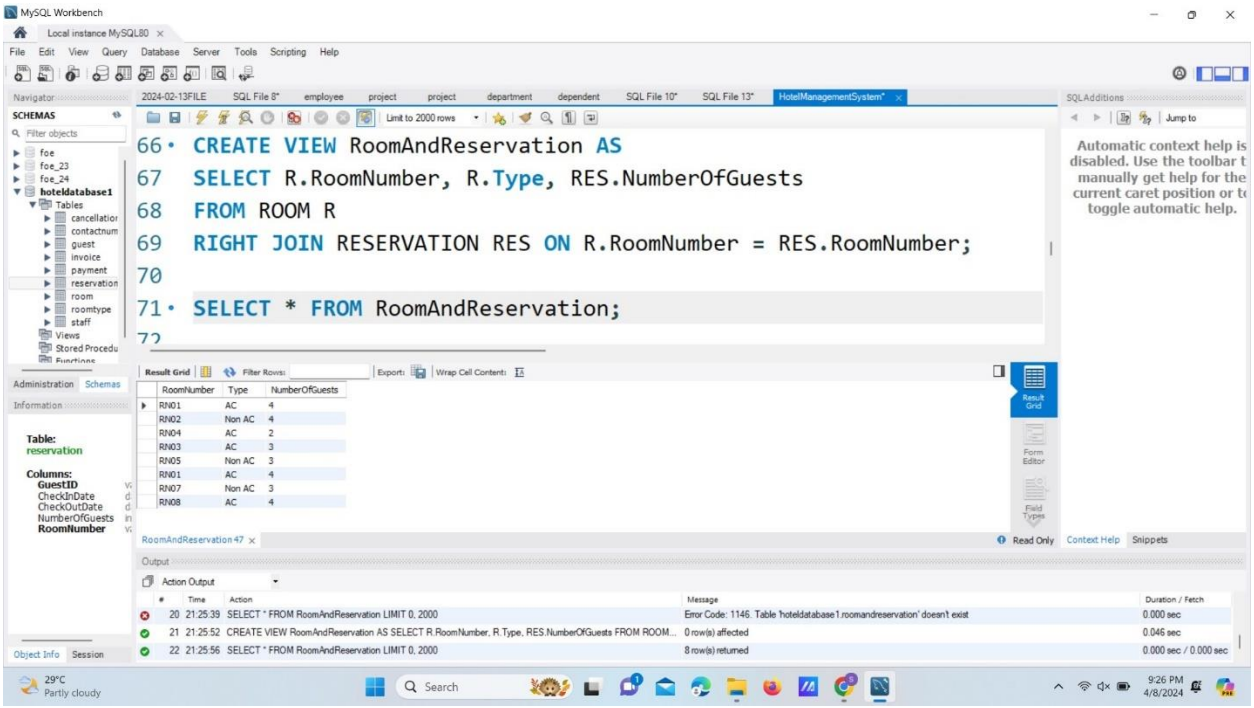
Outer Join Operation



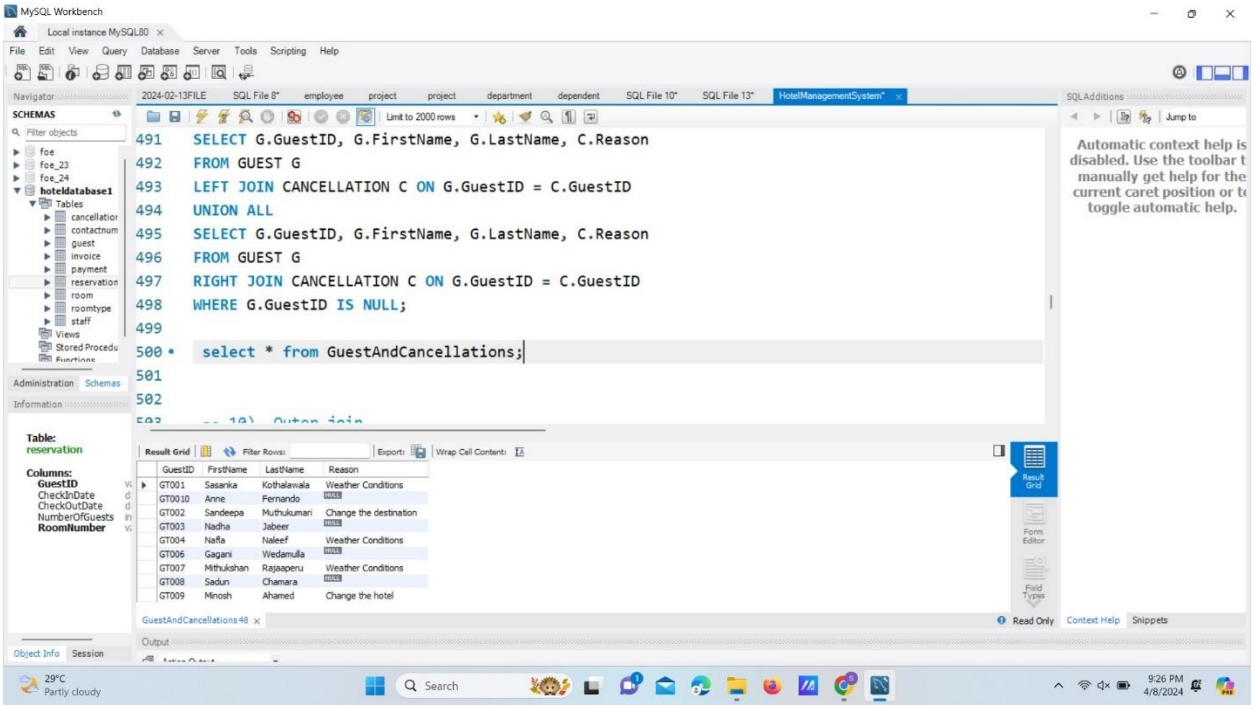
Left Outer Join Operation



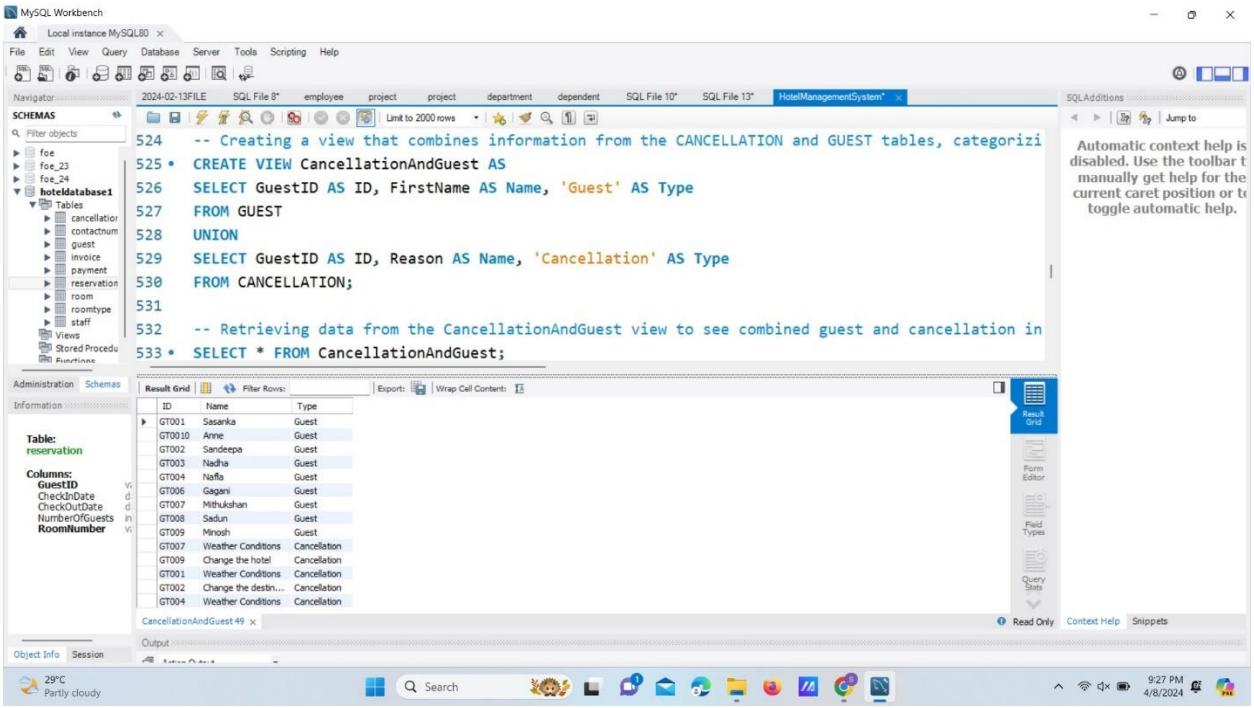
Right Outer Join Operation



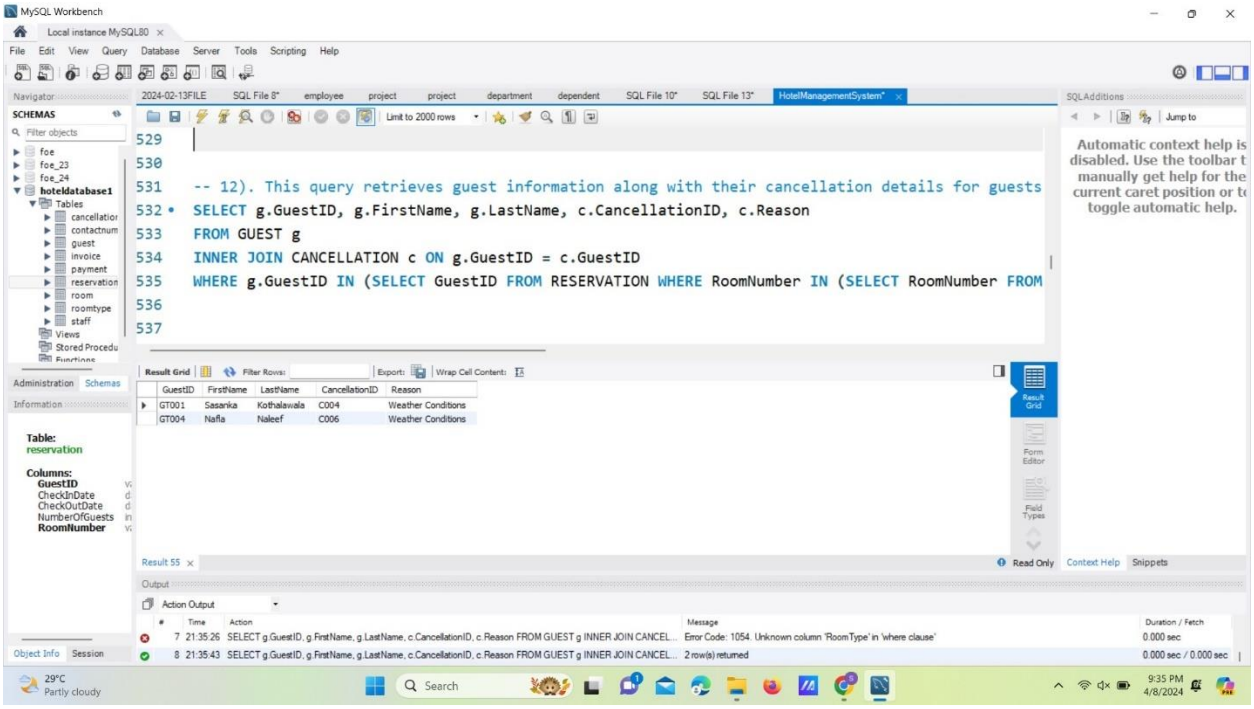
Full Outer Join Operation



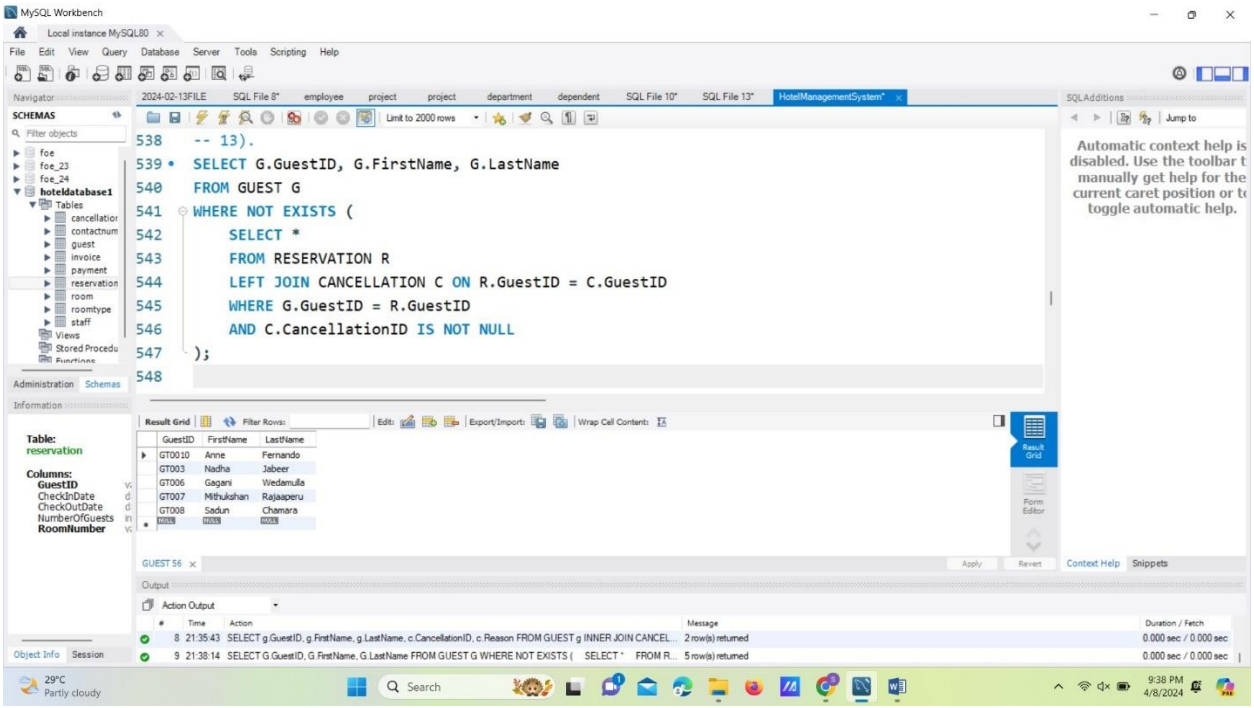
Outer Union Operation



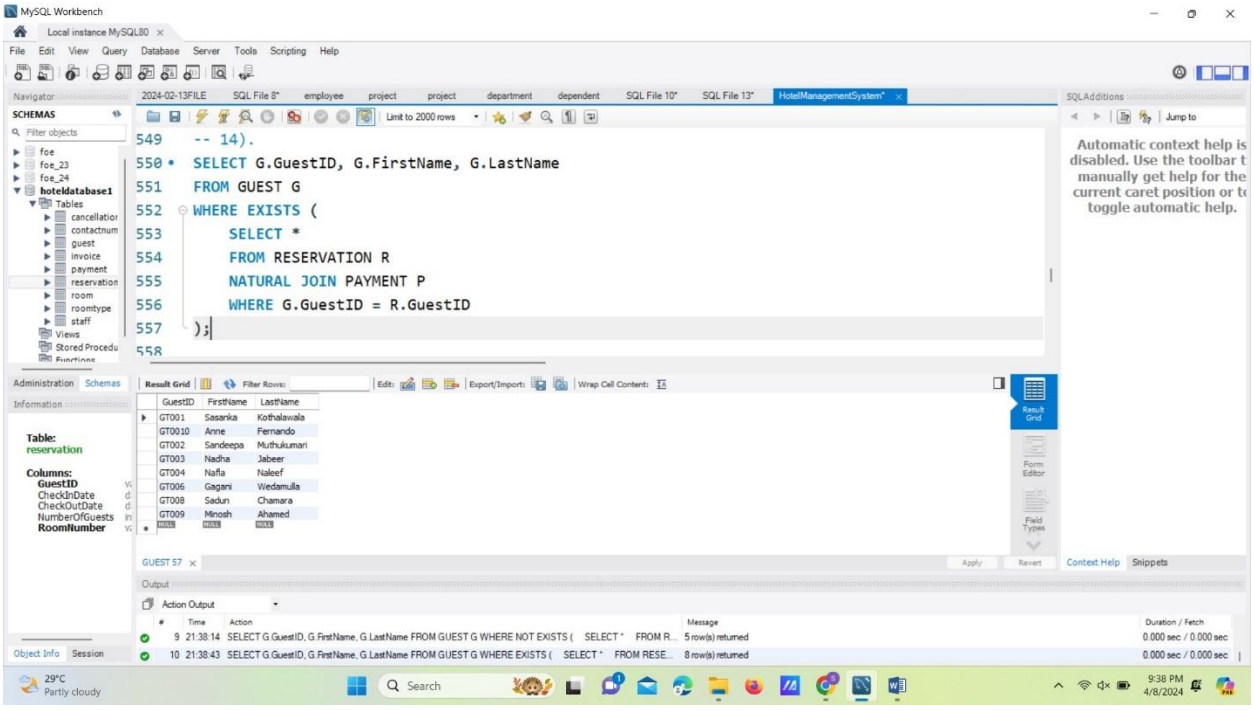
Nested Query with Intersection Operation



Nested Query with Set Operation



Nested Query with Division Operation



Chapter 5 – Tuning

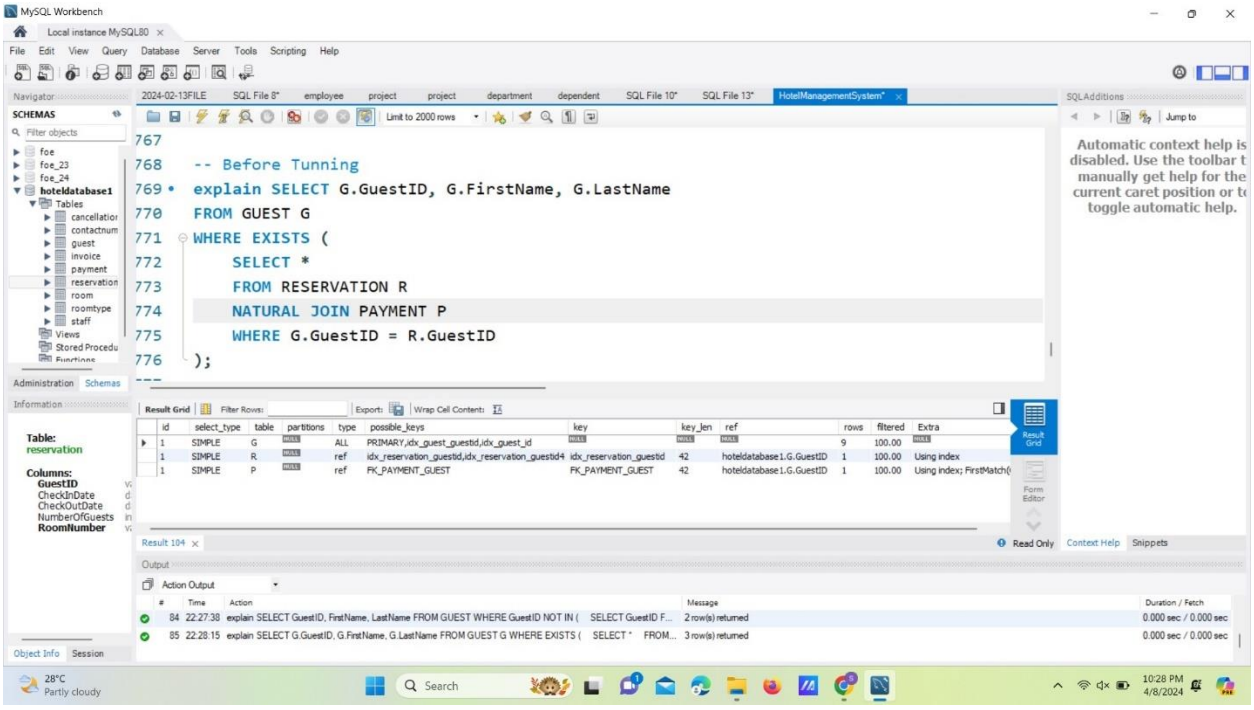
Chapter 5 describes how indexing strategies are used to improve the efficiency of complex queries. For every complex query, the following steps are applied during query tuning in order to make the process easier to understand:

1. Removing Existing External Indexes: Start by getting rid of any old indexes that weren't needed for query optimization.
2. Displaying Table Indexes Before Index Creation (Using SHOW INDEX Command): Evaluate the current state of indexes on the relevant tables before making any modifications.
3. Determining the Number of Accessed Rows before Index Creation (Using EXPLAIN Command)
4. . Creating a Suitable Index: Based on the analysis, establish an appropriate index to optimize the query.
5. . Displaying Table Indexes after Index Creation (Using SHOW INDEX Command).
6. Determining the Number of Accessed Rows after Index Creation (Using EXPLAIN Command)

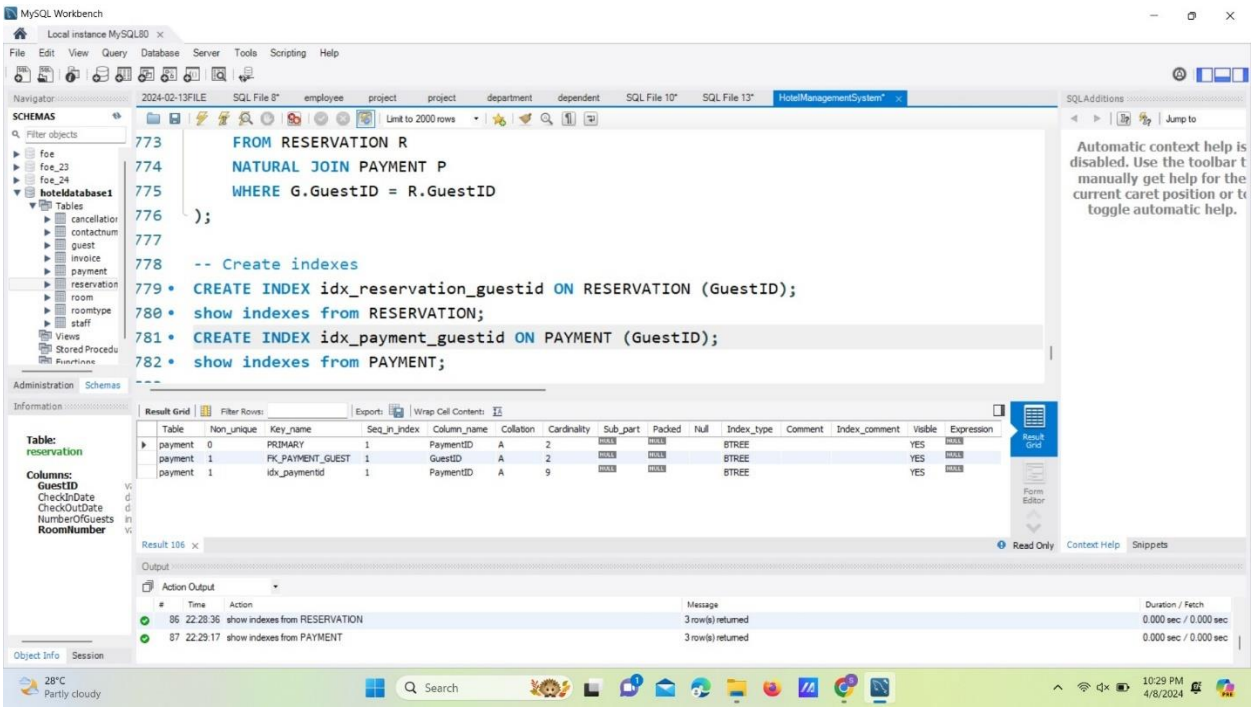
A comparison of the number of rows accessed in the query execution plan prior to and during the development of the appropriate index serves as an example of the query tuning procedure. After adding the index, if there is a decrease in the number of accessed rows, the query has been successfully optimized.

1)Tuning Nested Complex Query

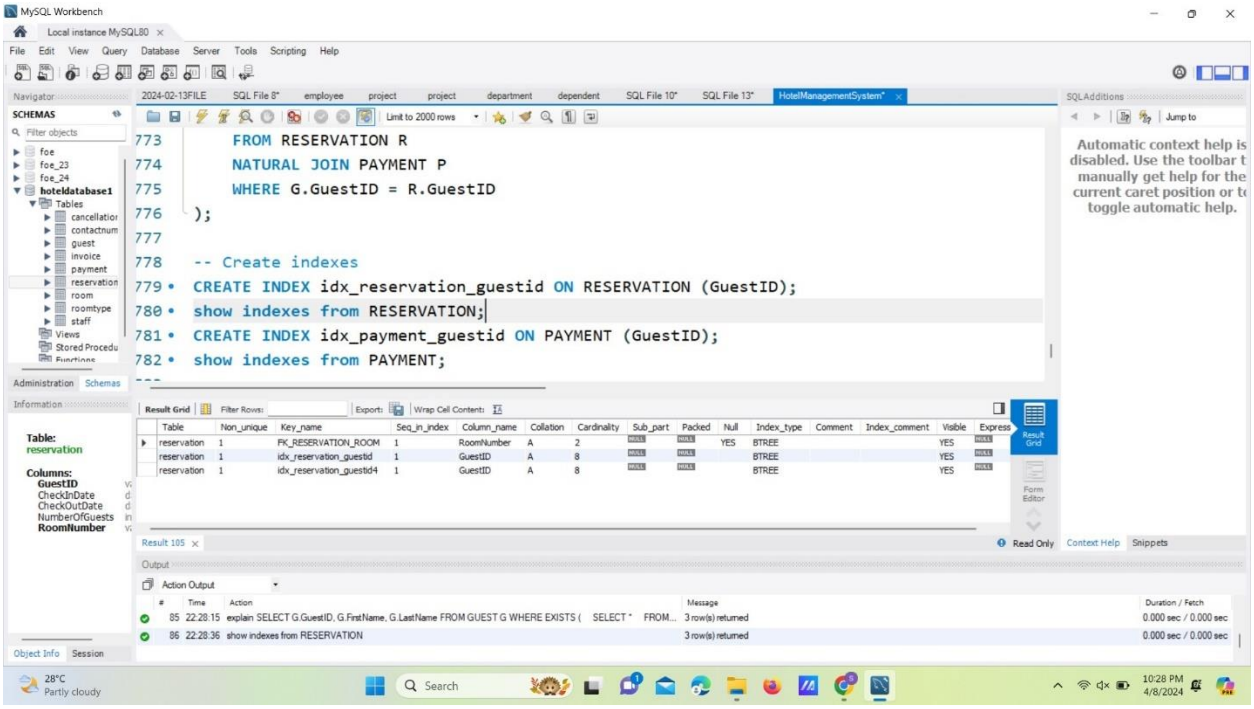
Before Tuning



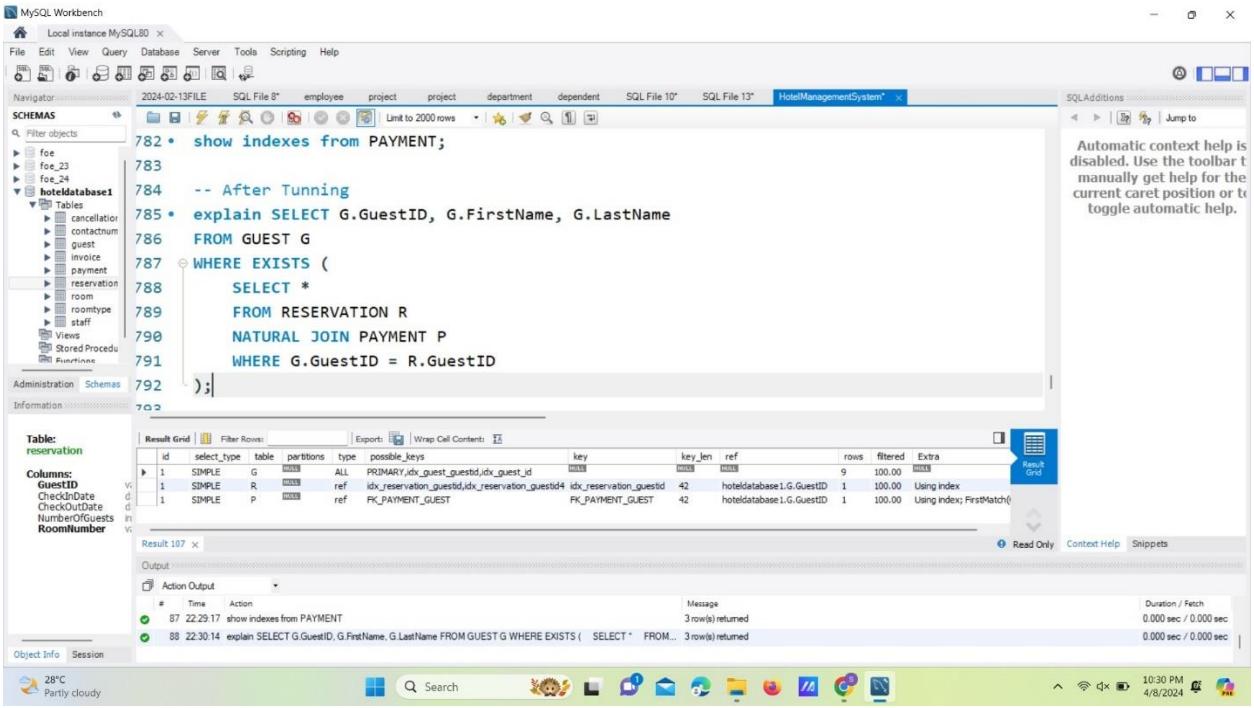
Creating Index (Payment)



Creating Index (Reservation)

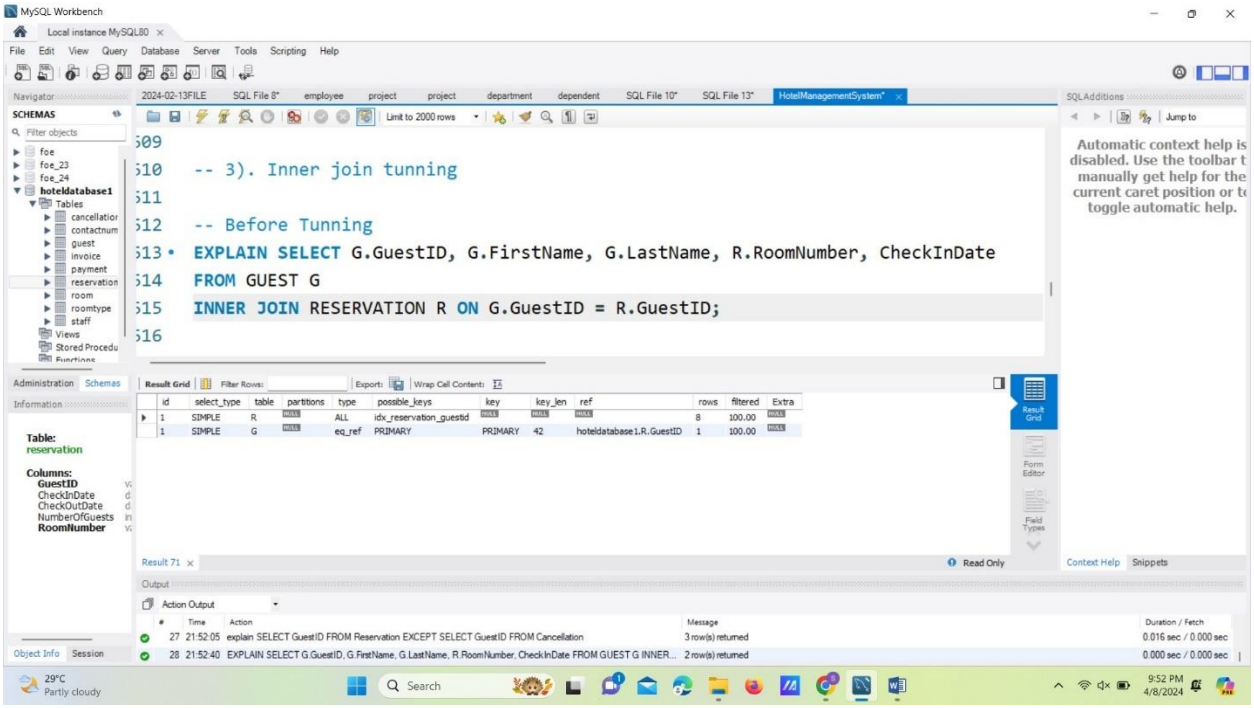


After Tuning

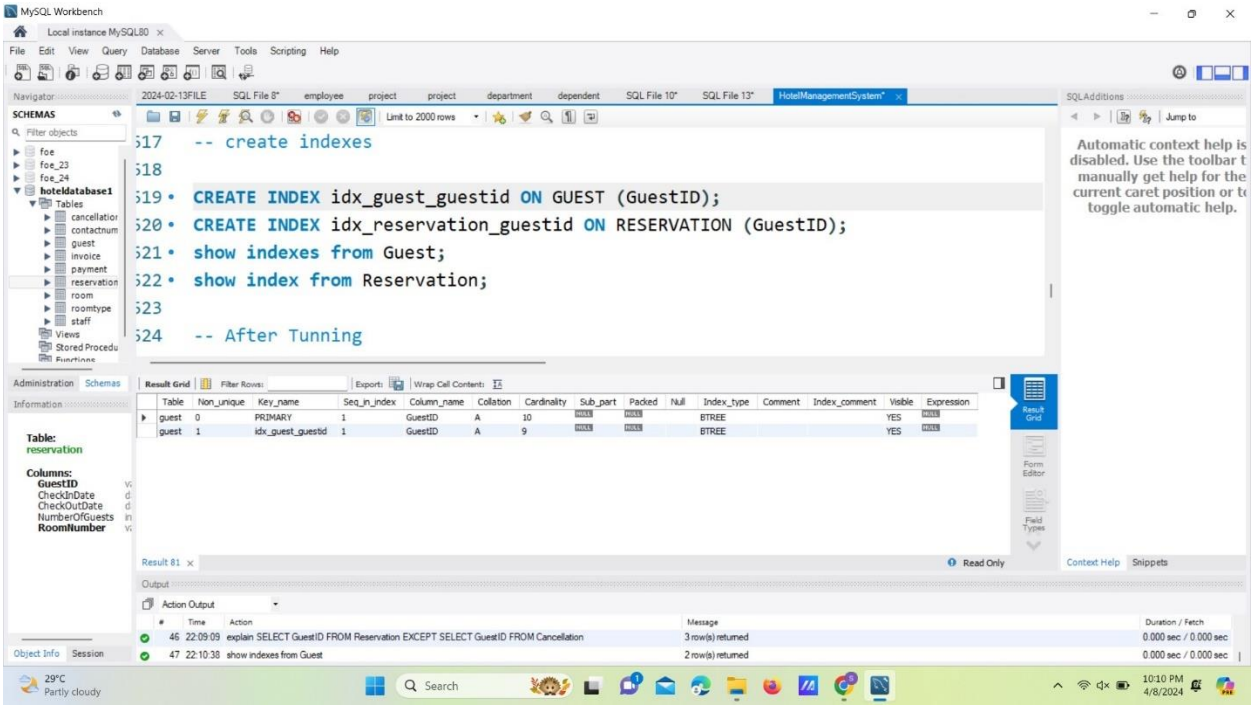


2)Tuning Inner Join Complex Query

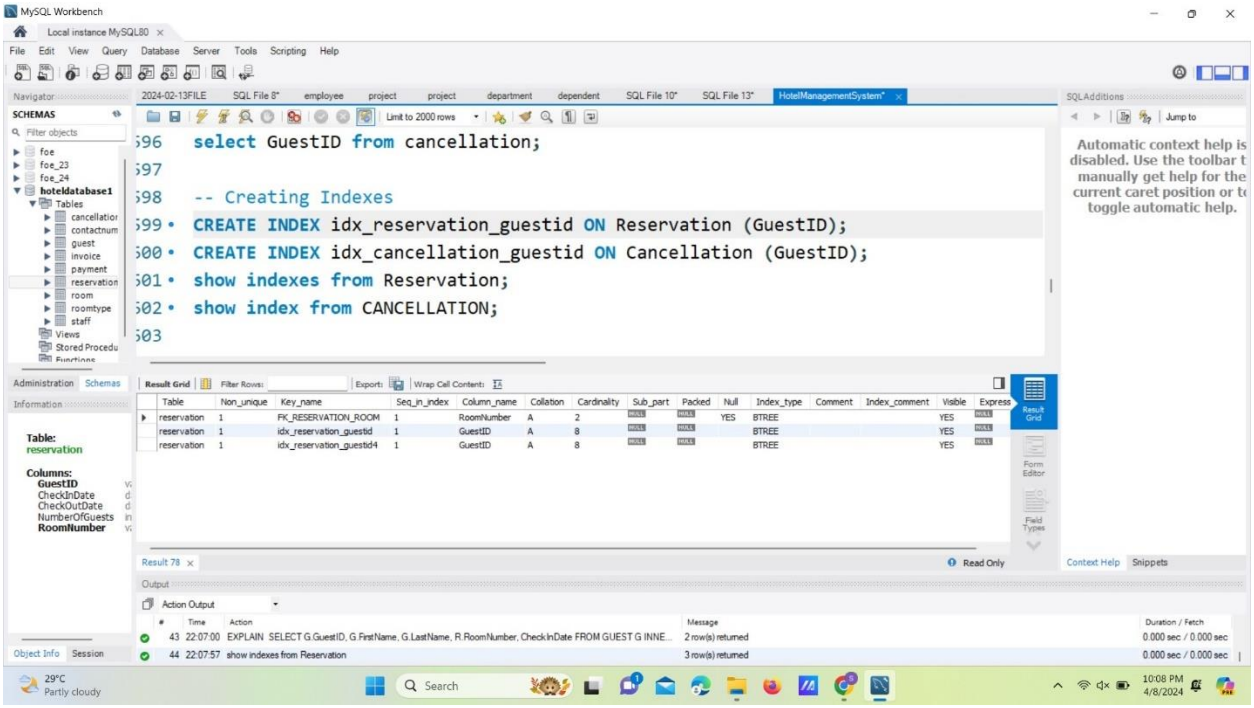
Before Tuning



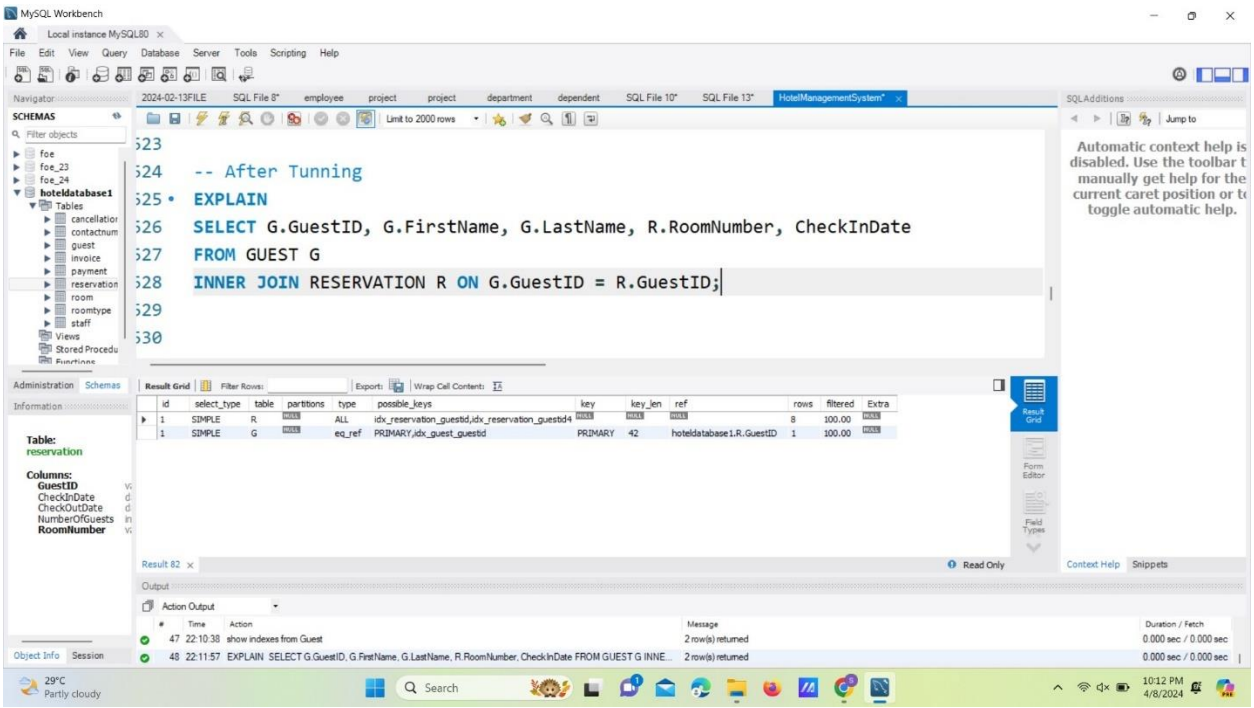
Creating Index (Guest)



Creating Index (Reservation)

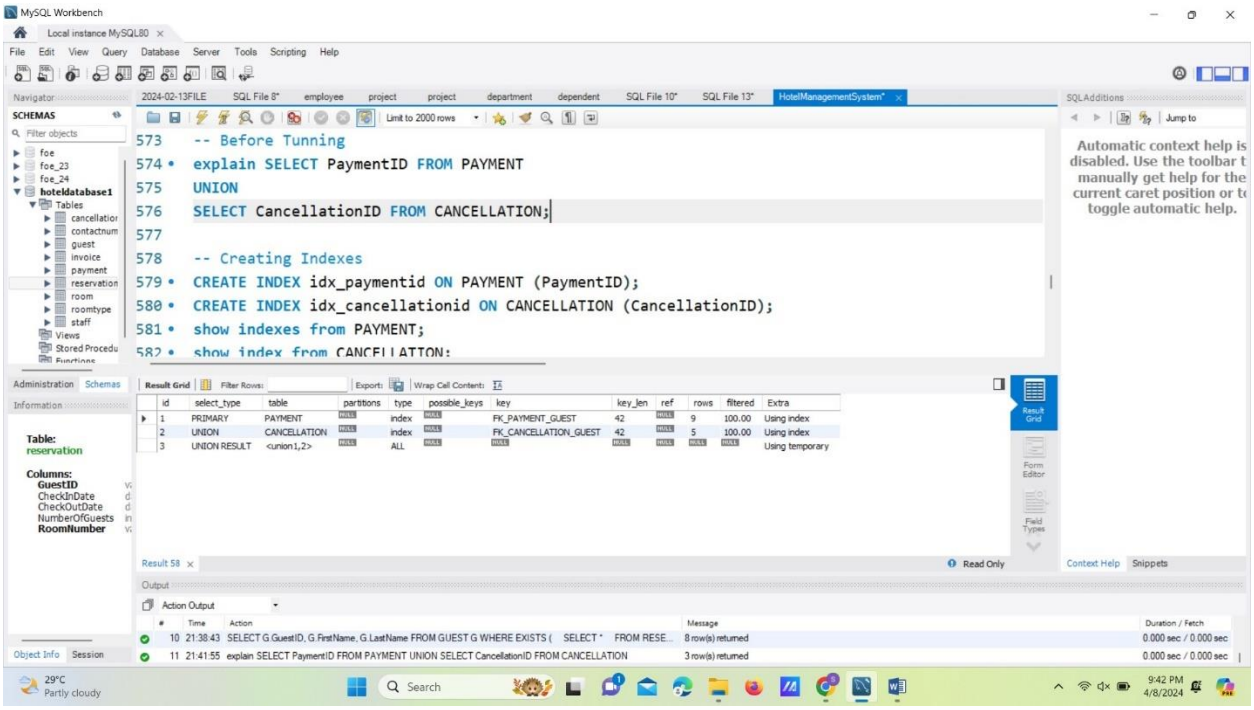


After Tuning

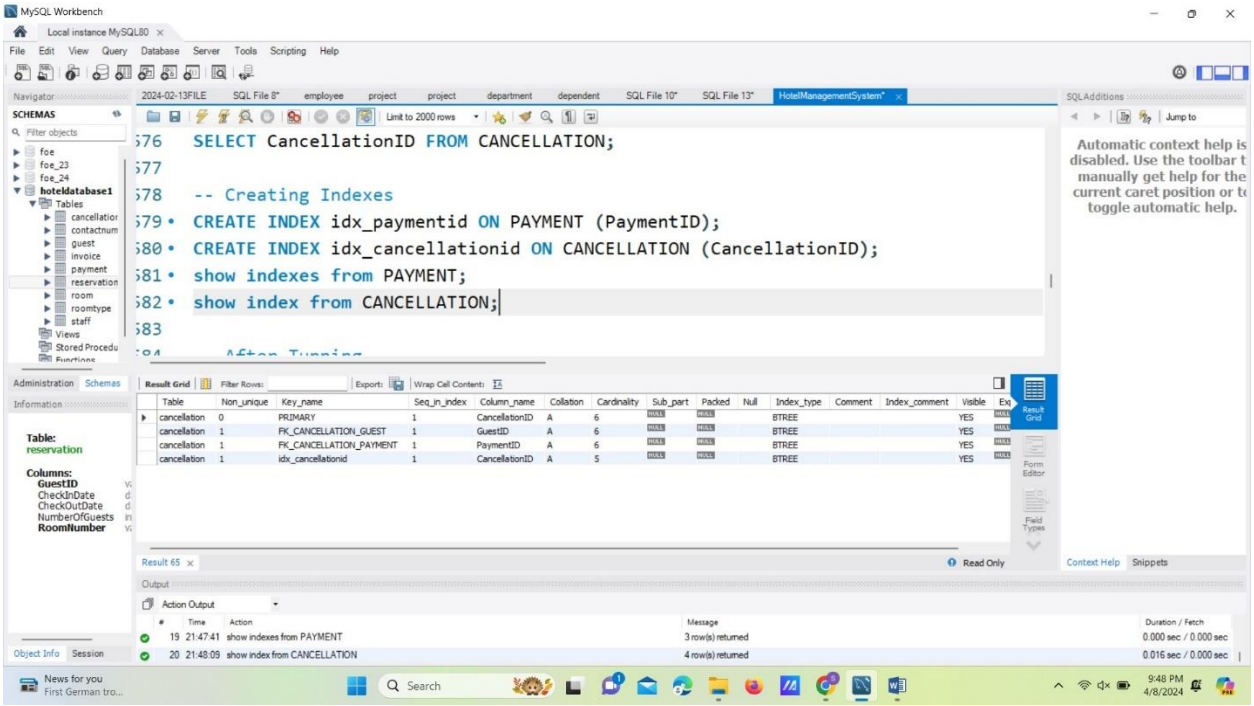


3)Tuning Union Complex Query

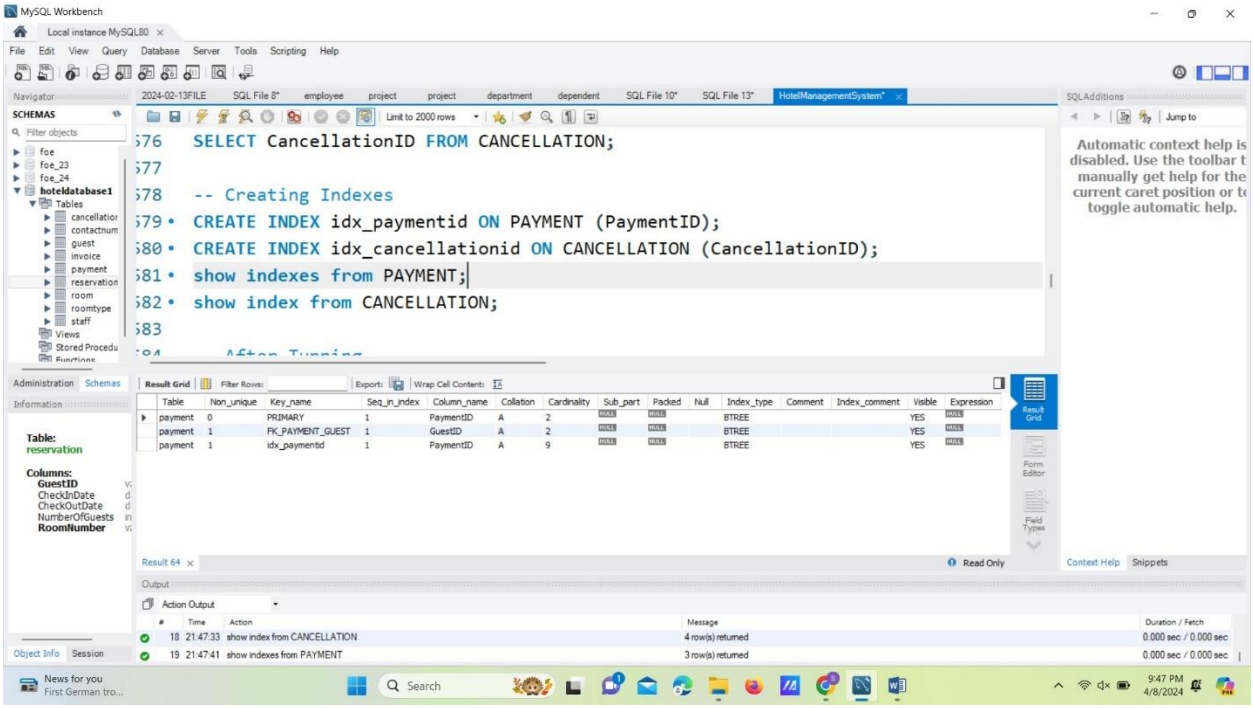
Before Tuning



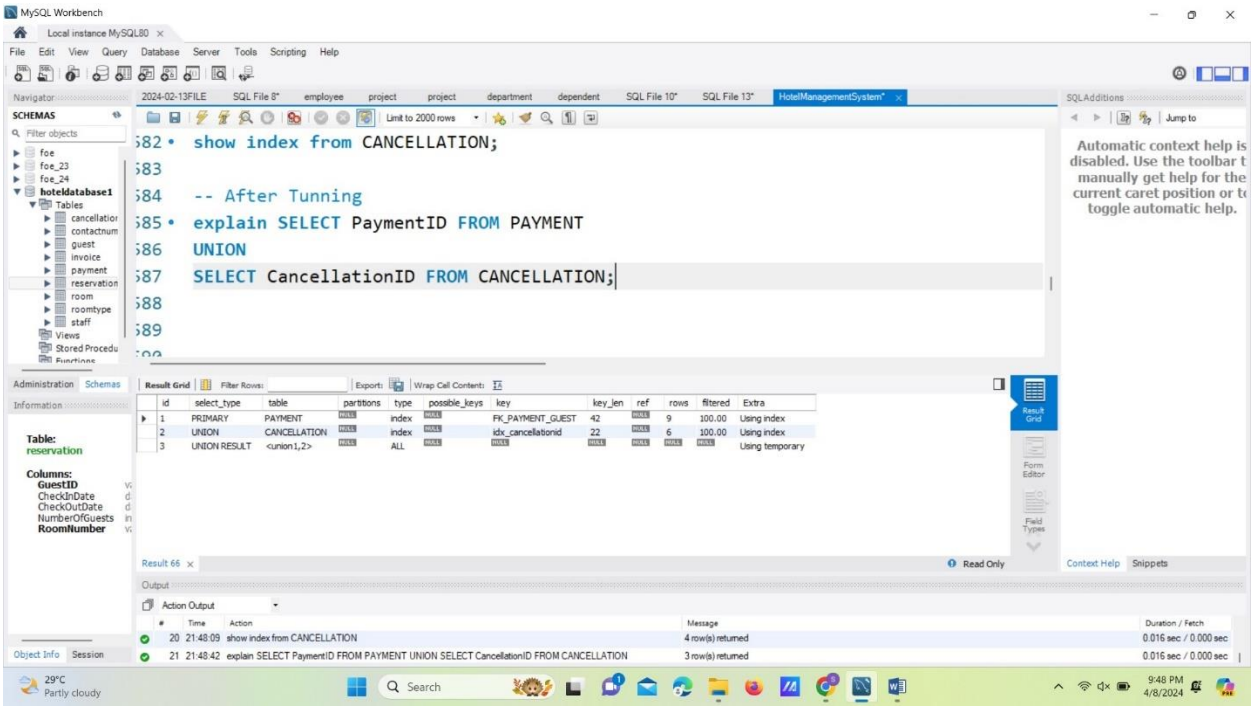
Creating Index (Cancellation)



Creating Index (Payment)

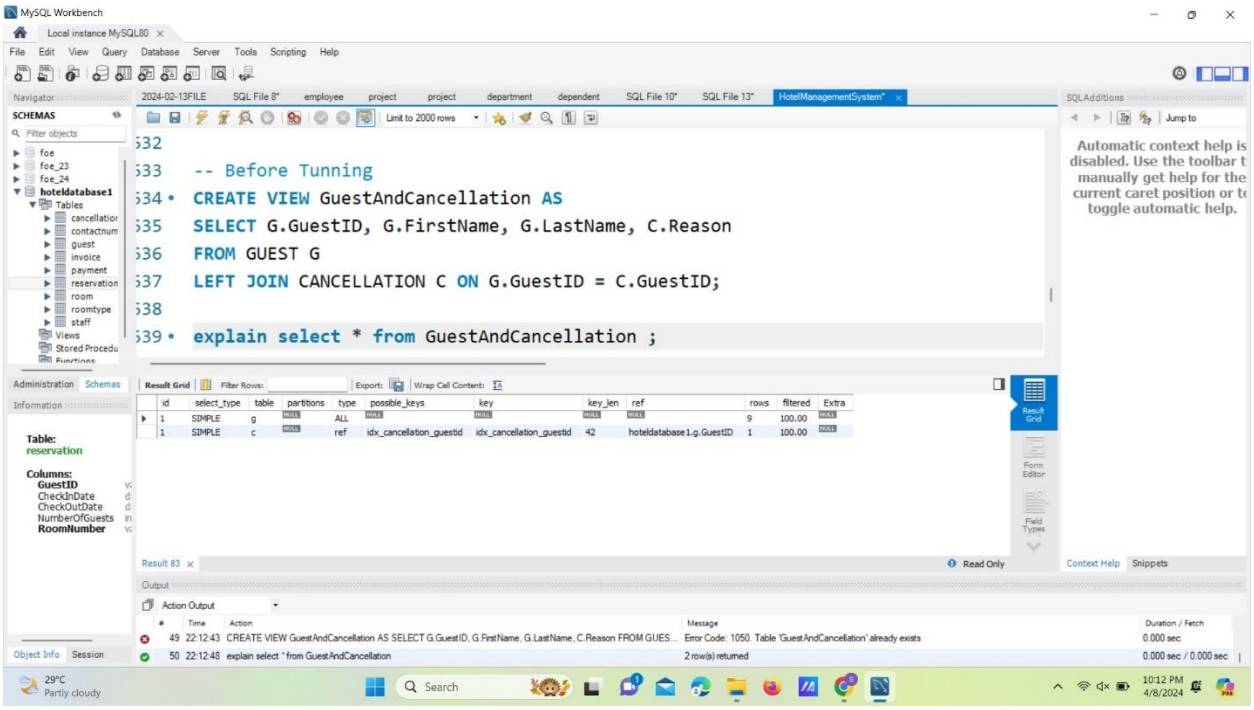


After Tuning

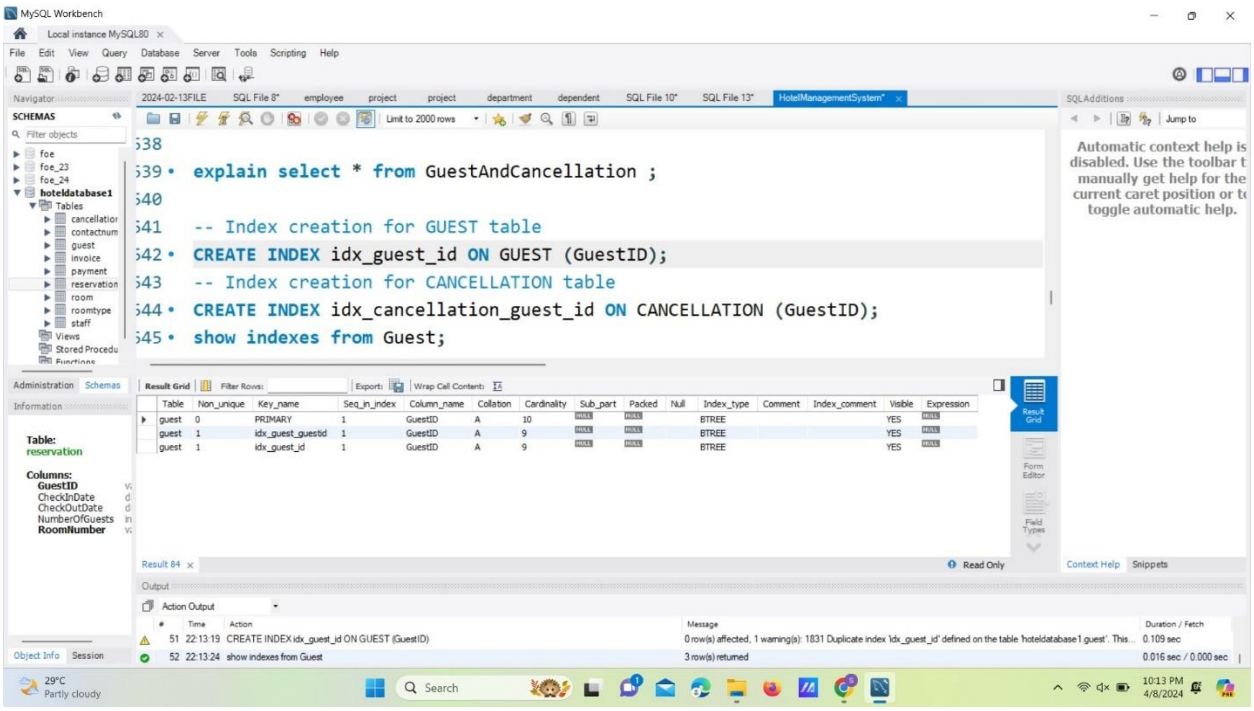


4) Tuning Left Outer Join

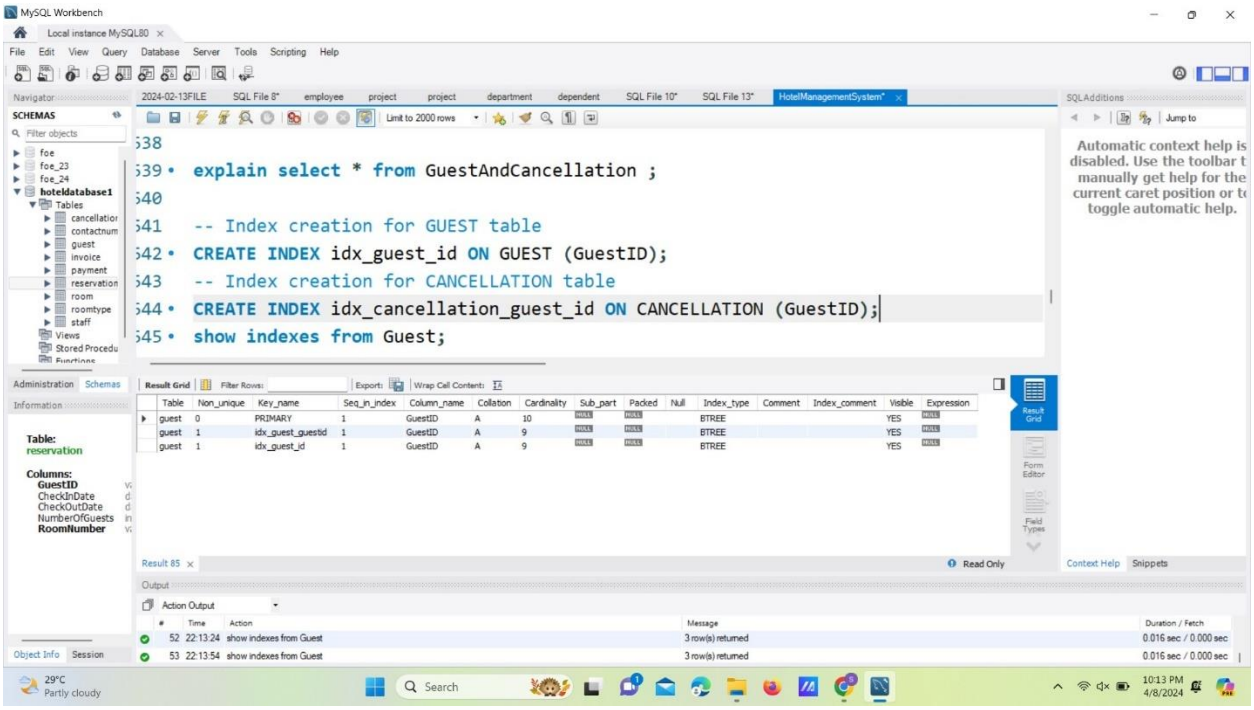
Before Tuning



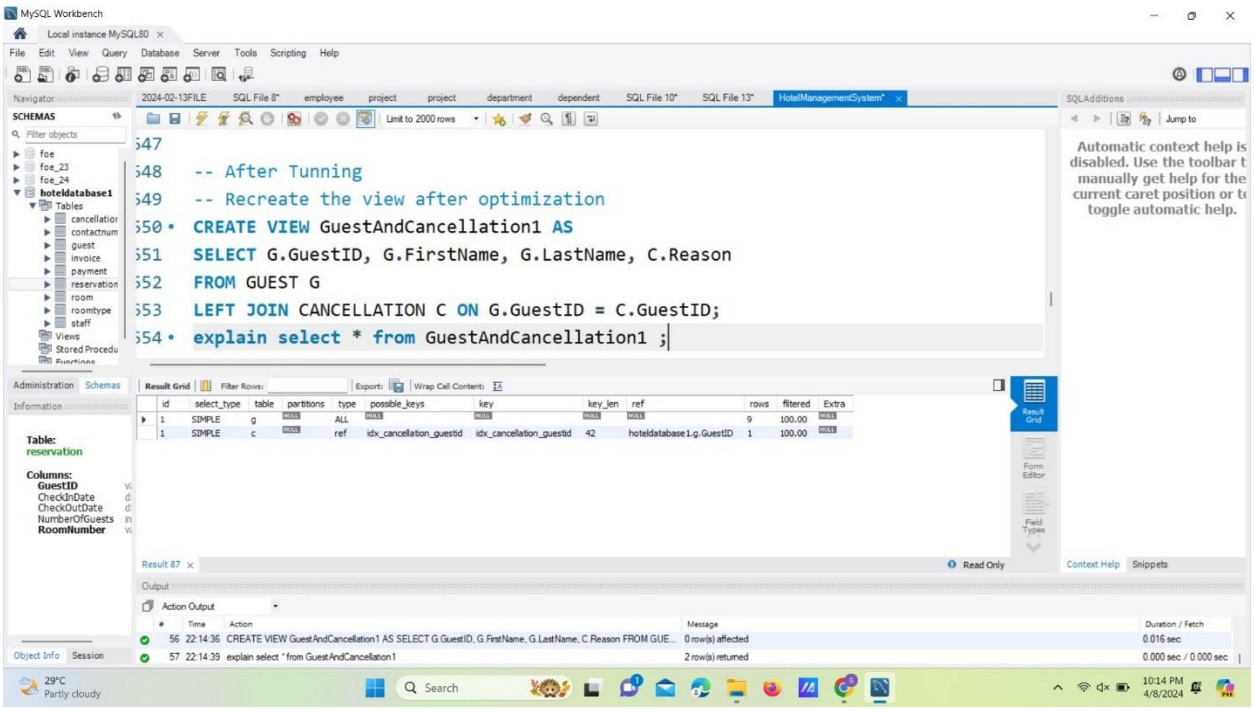
Creating Index (Guest)



Creating Index (Cancellation)

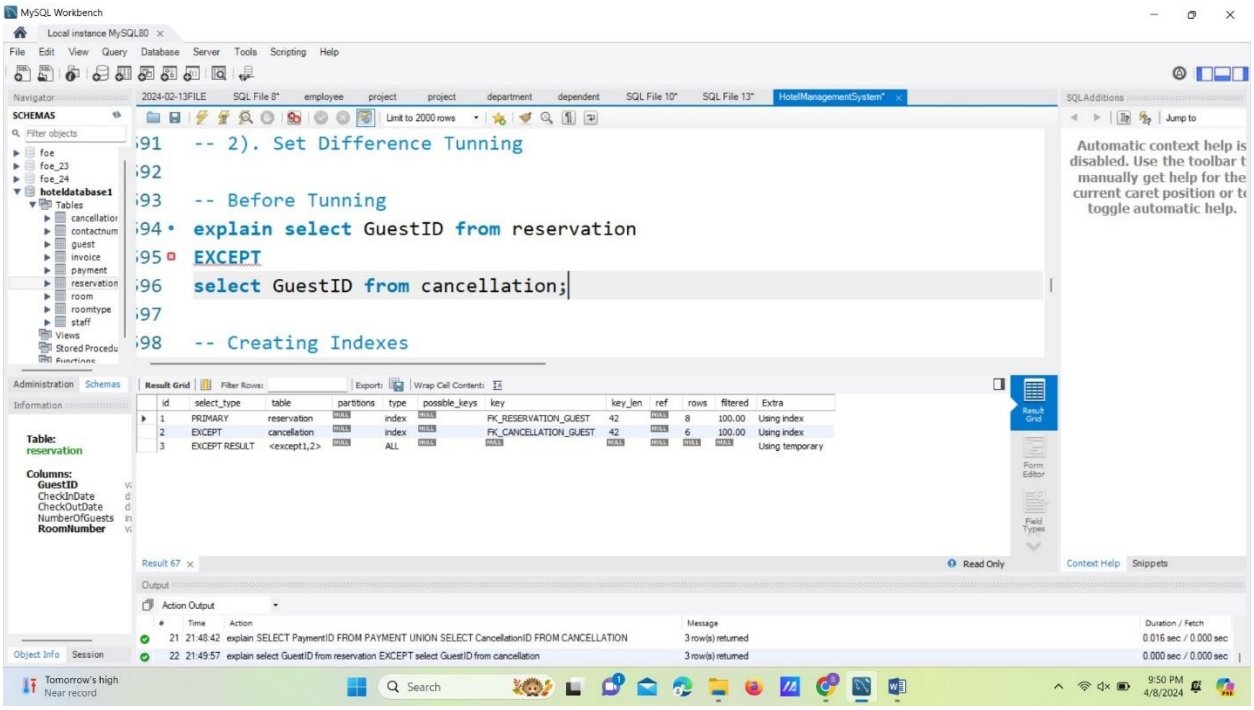


After Tuning

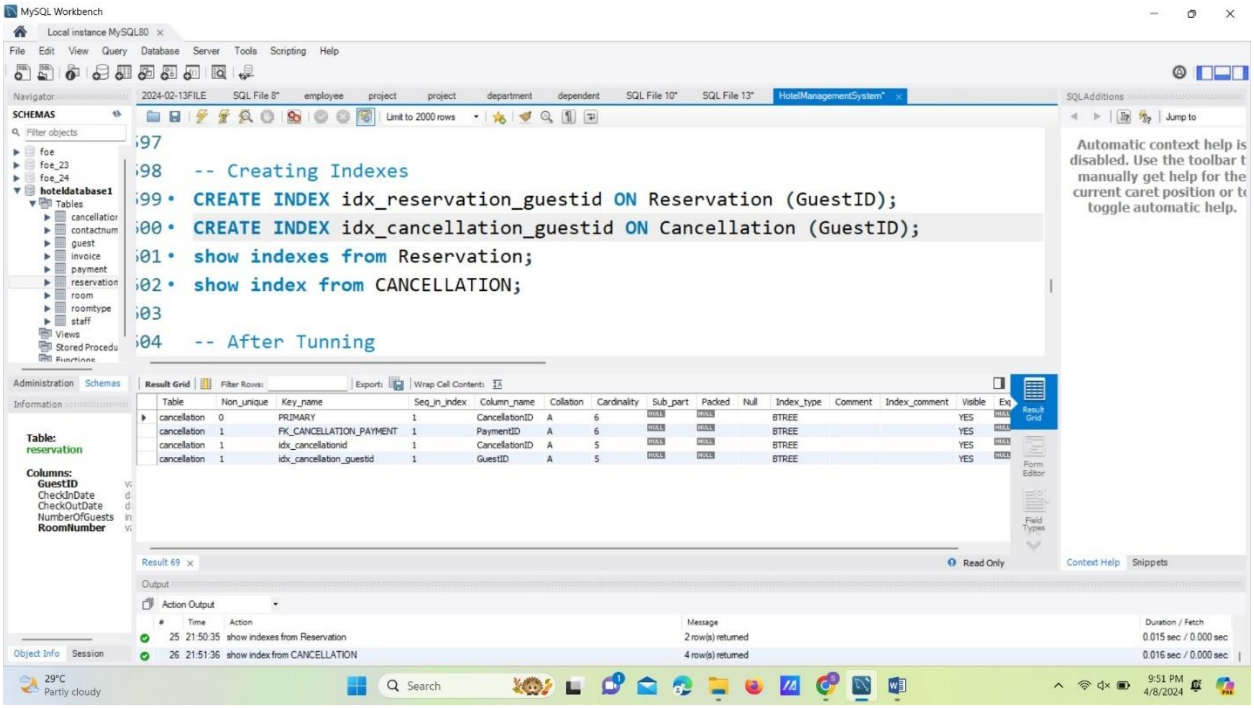


5)Tuning Difference

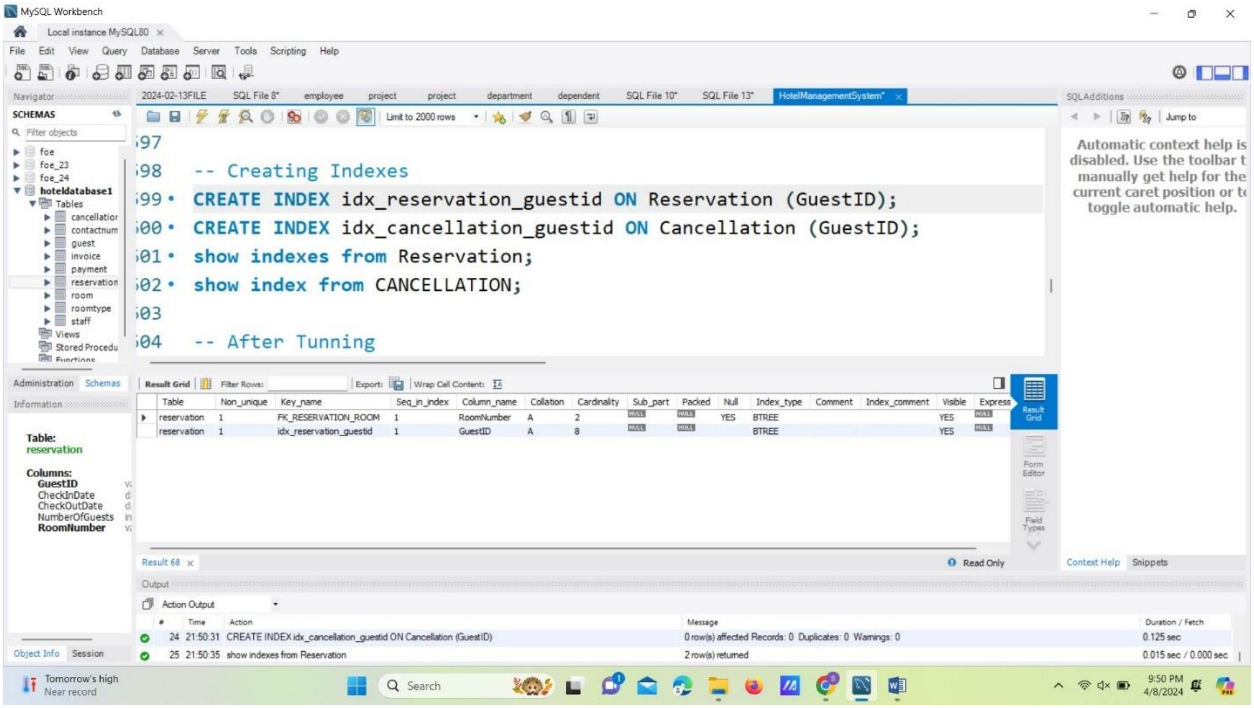
Before Tuning



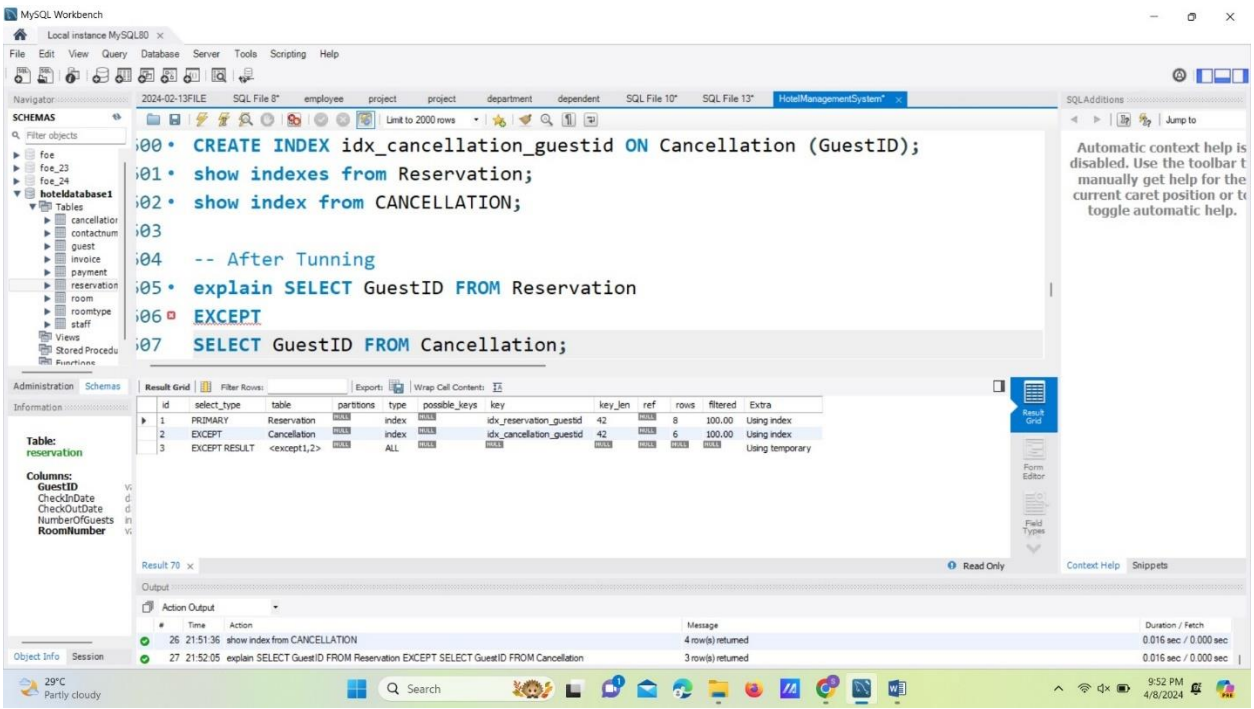
Creating Index (Cancellation)



Creating Index (Reservation)

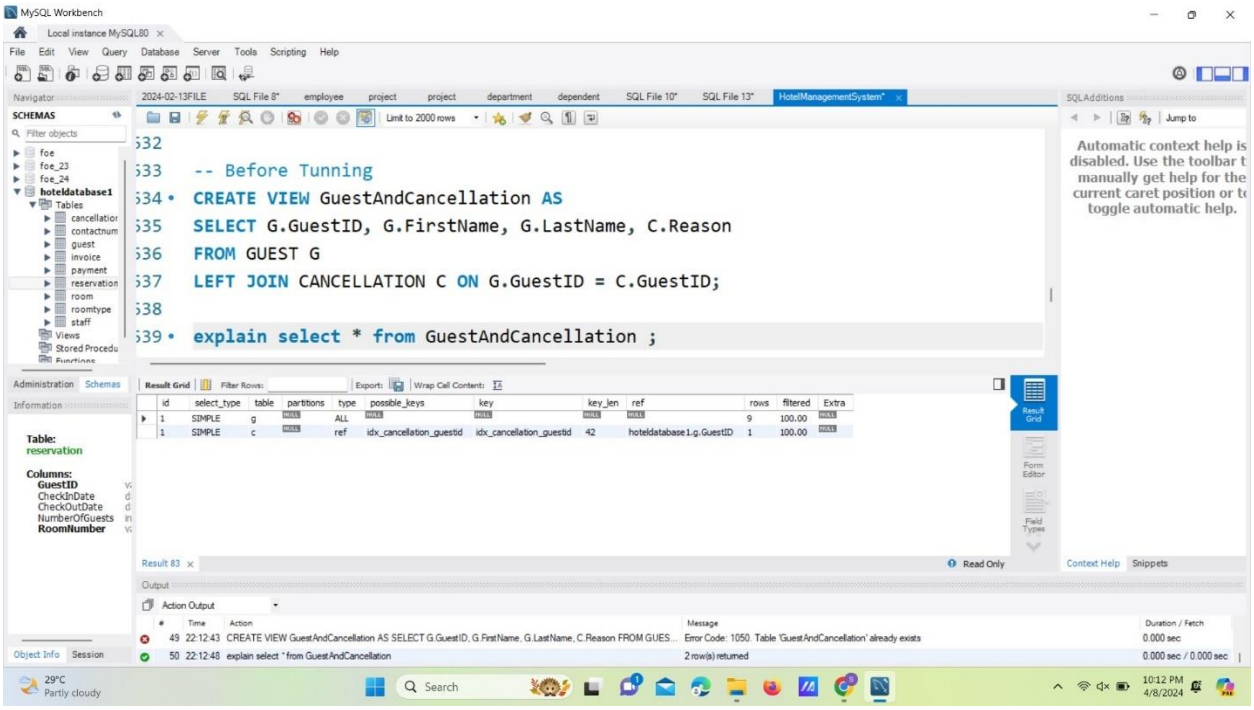


After Tuning

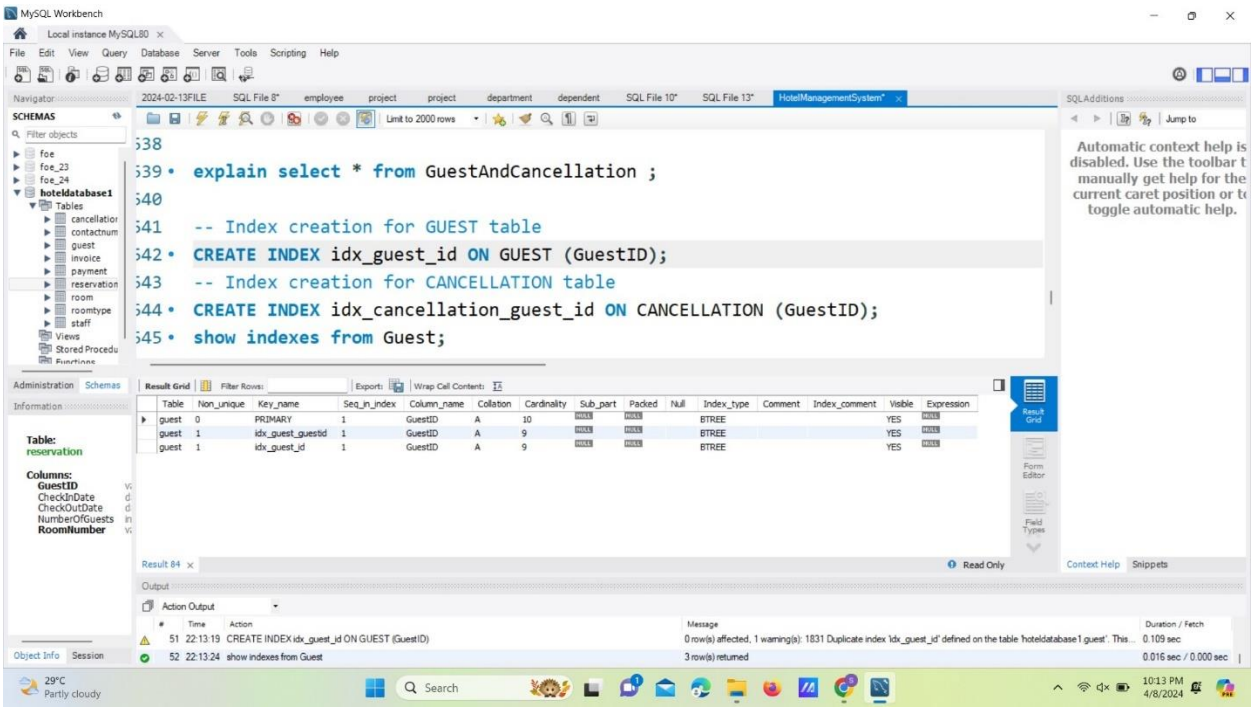


6)Tuning Left Outer Join

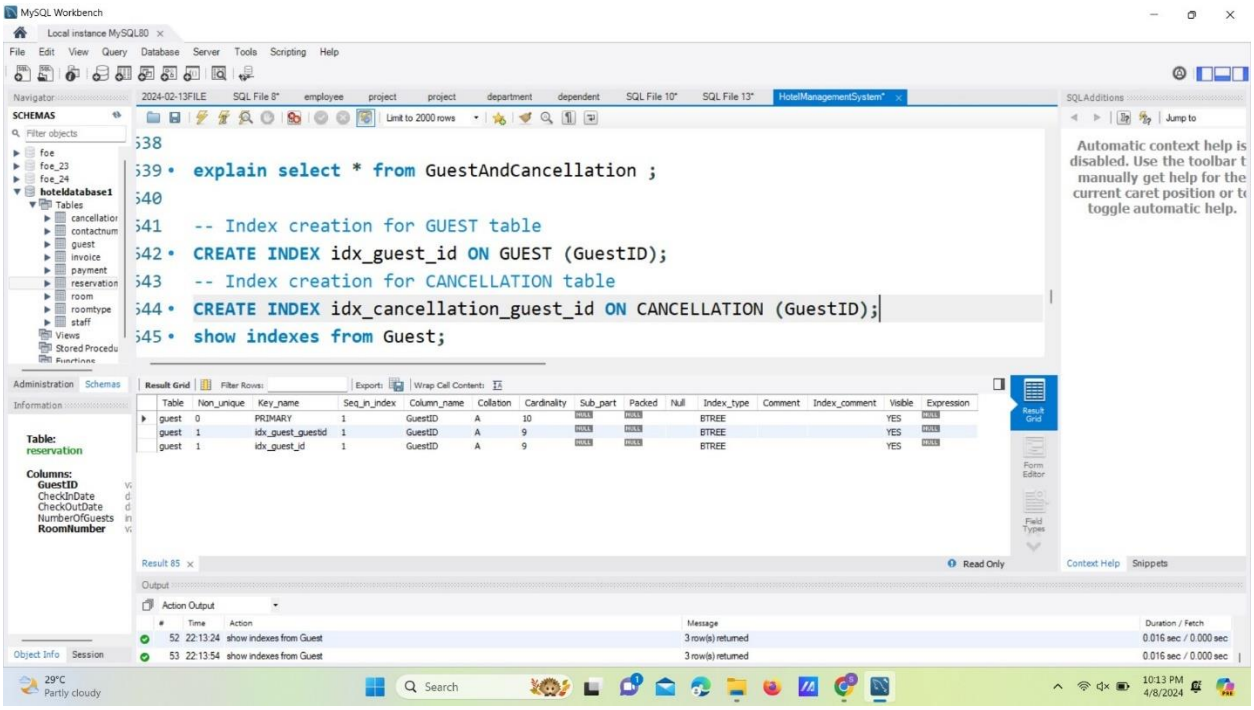
Before Tuning



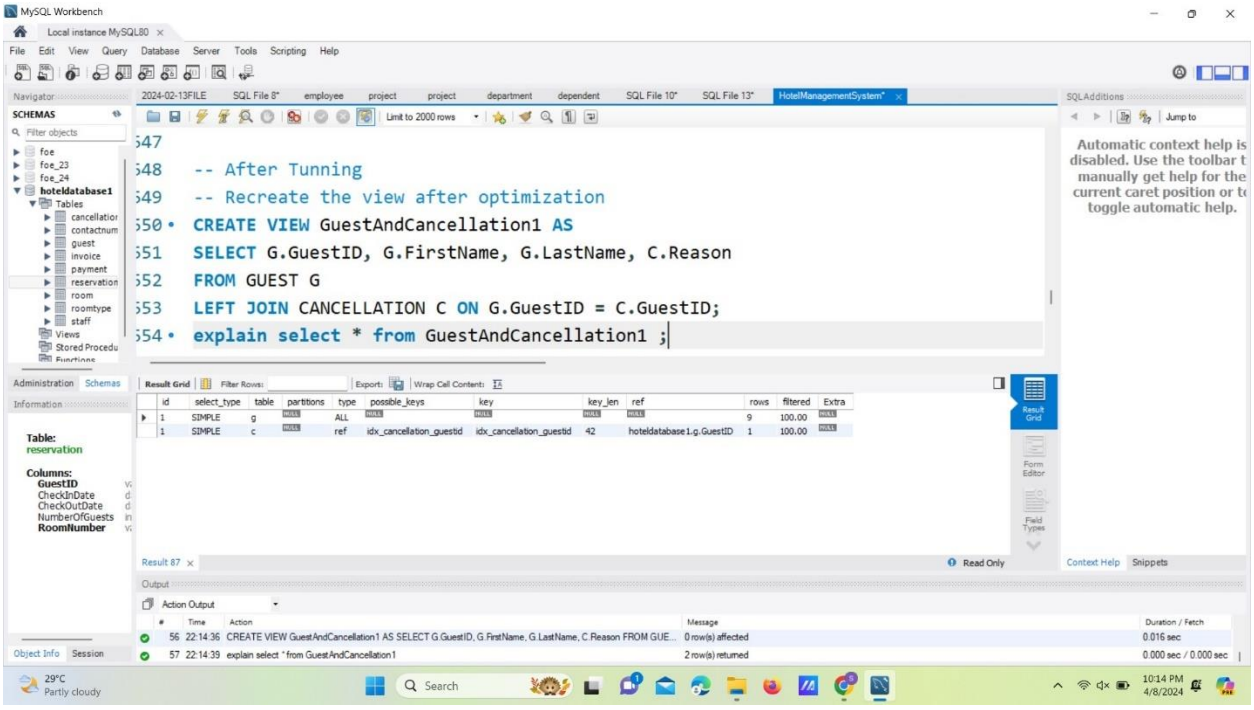
Creating Index (Guest)



Creating Index (Cancellation)

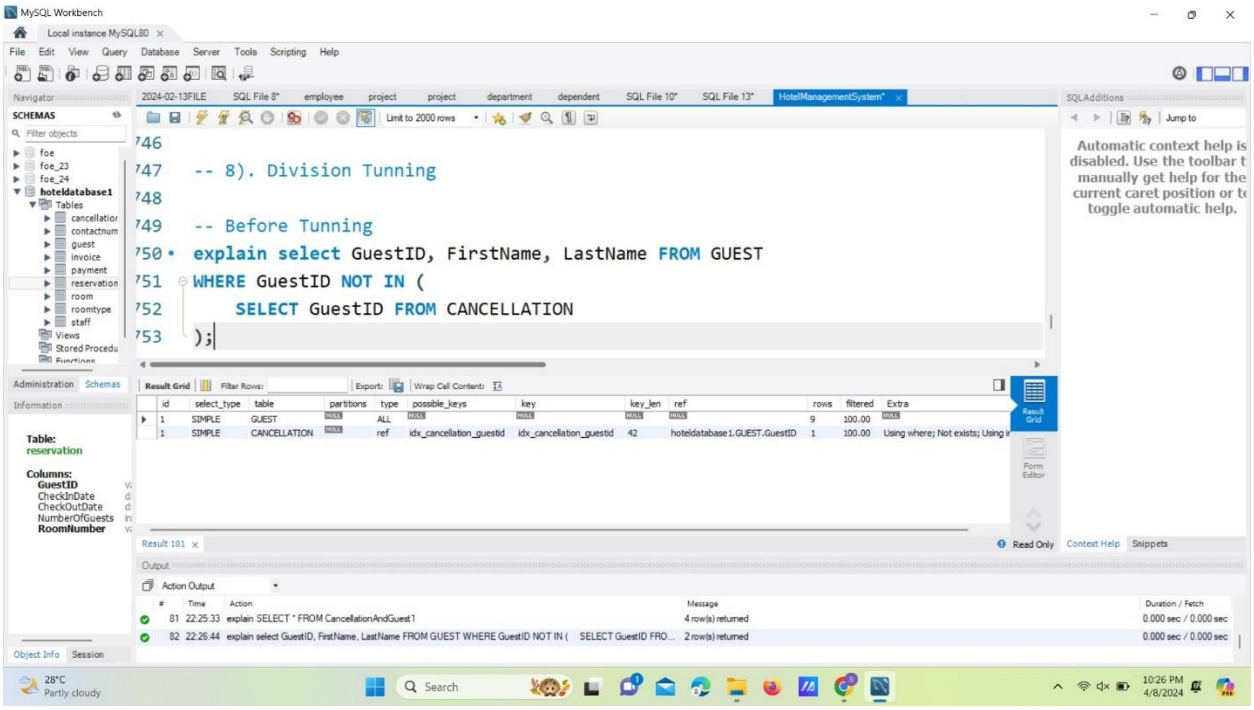


After Tuning

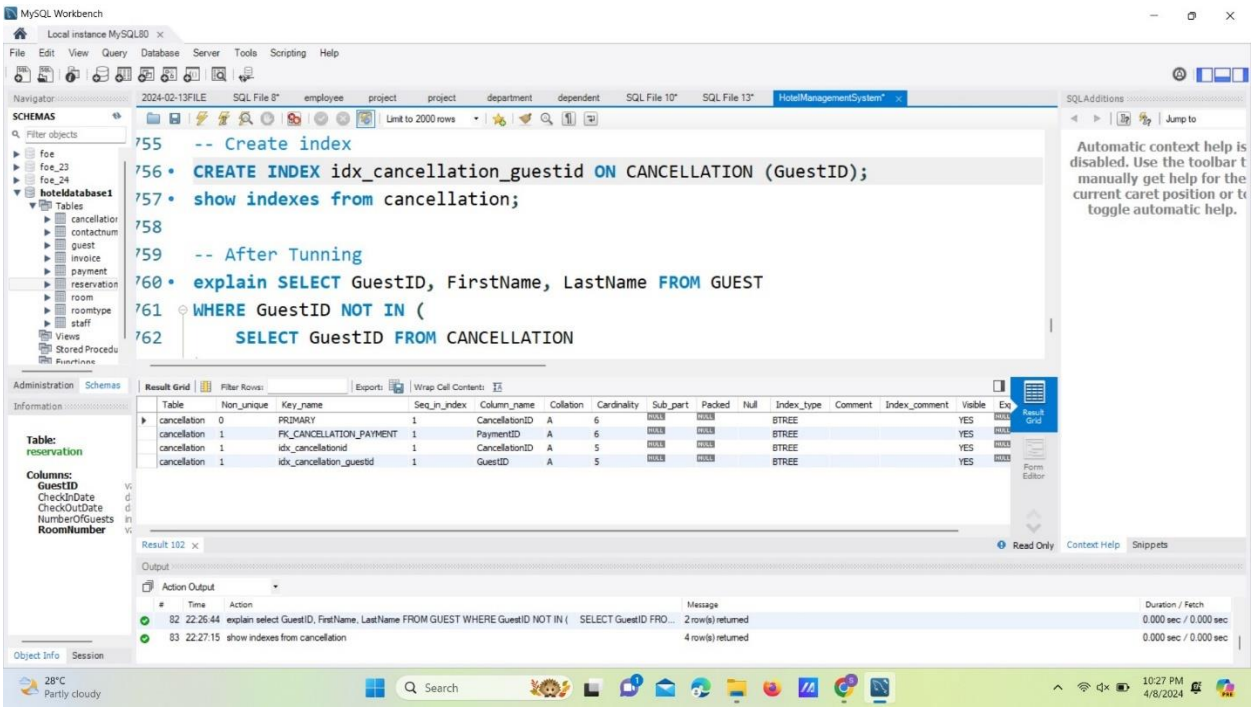


7)Tuning Nested Query with Division

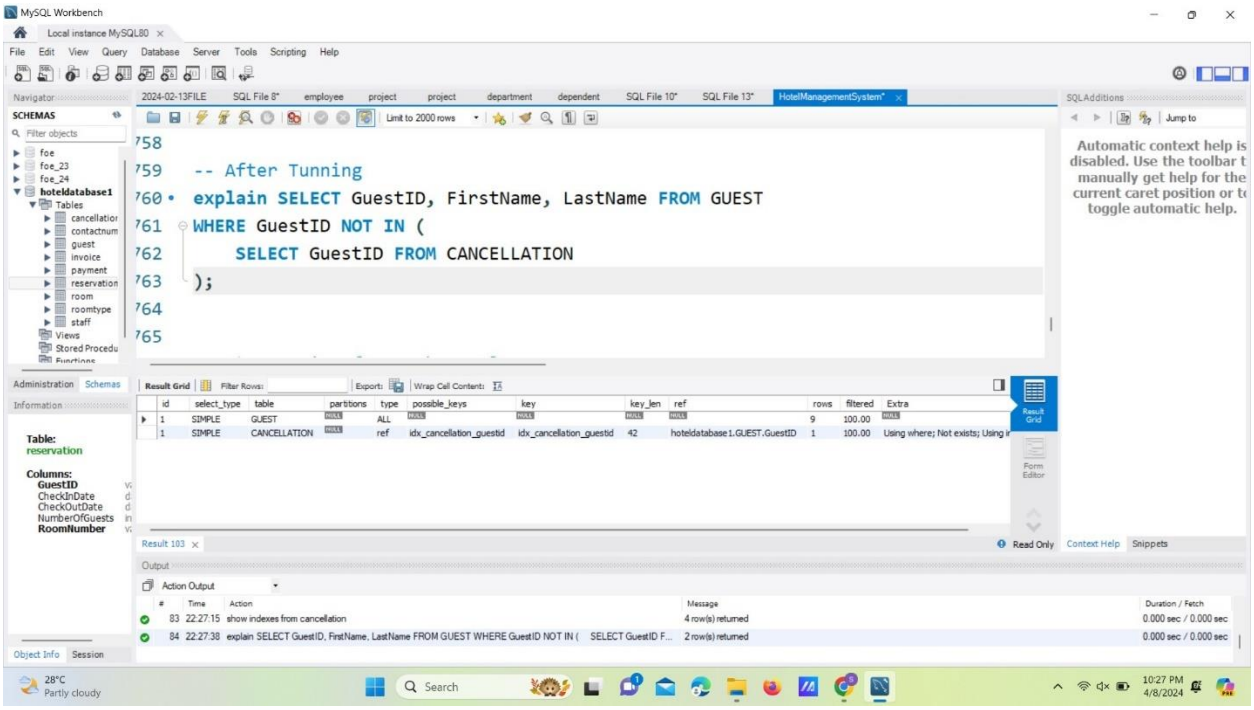
Before Tuning



Creating Index (Cancellation)

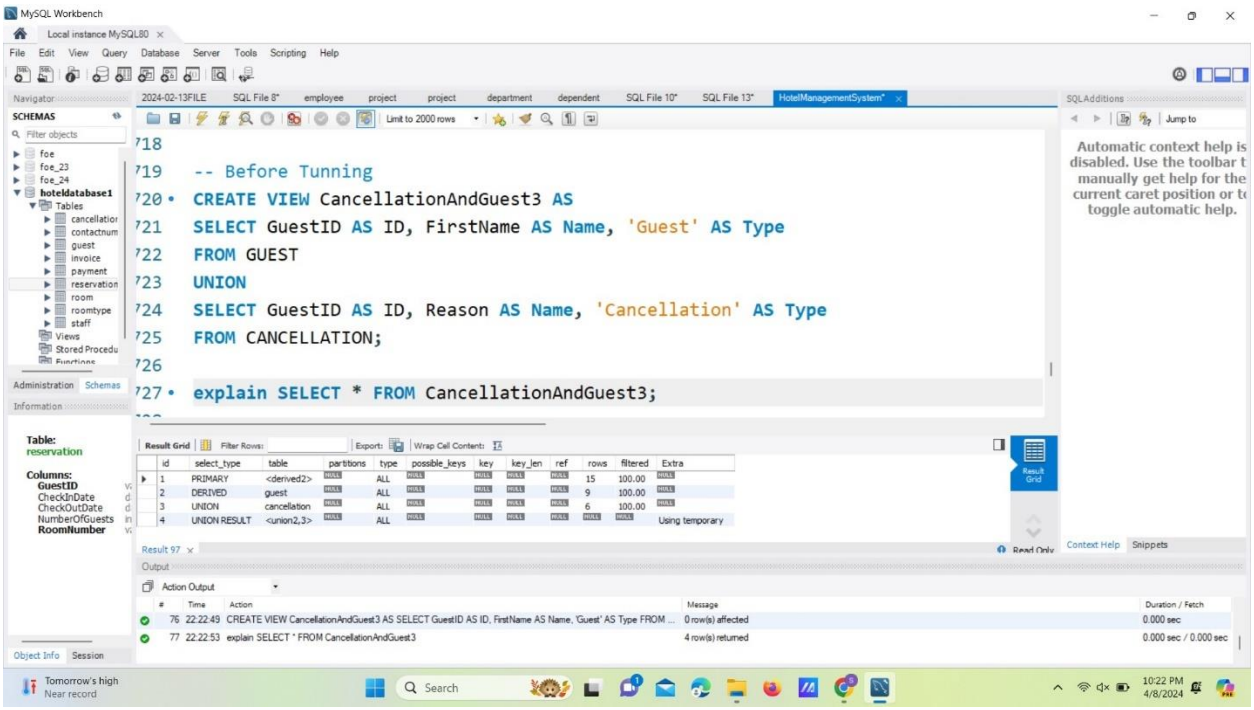


After Tuning

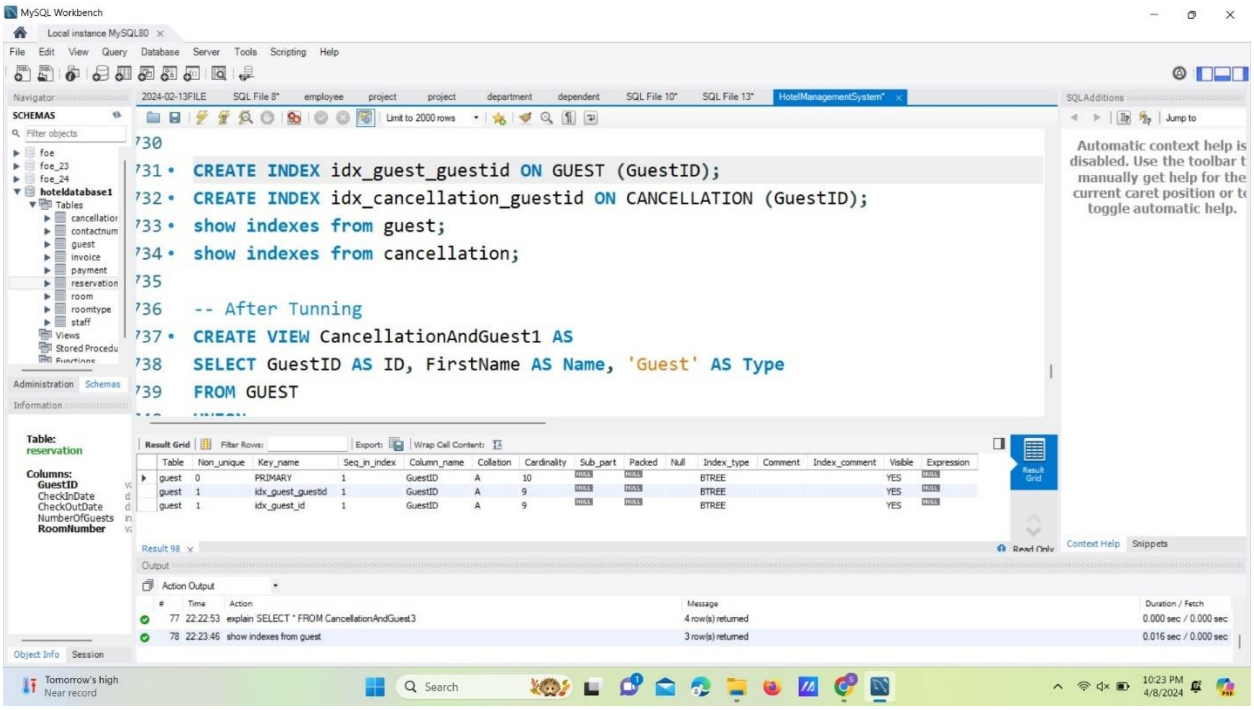


8)Tuning Outer Union

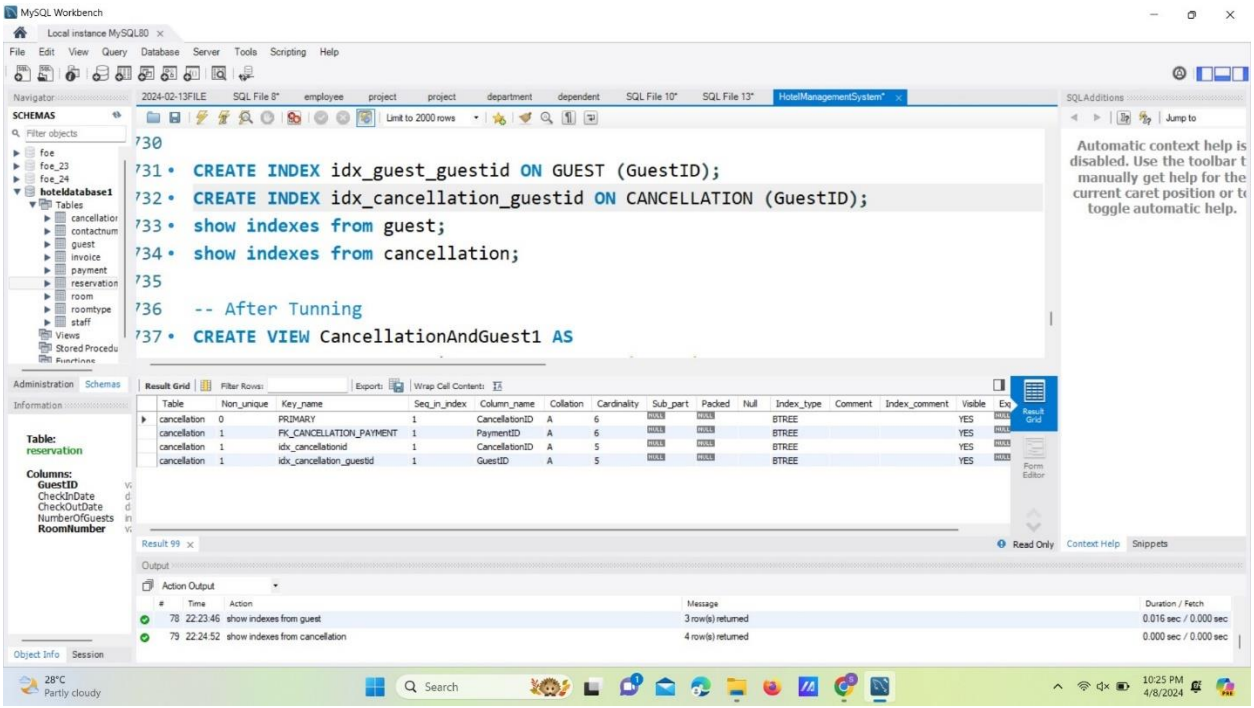
Before Tuning



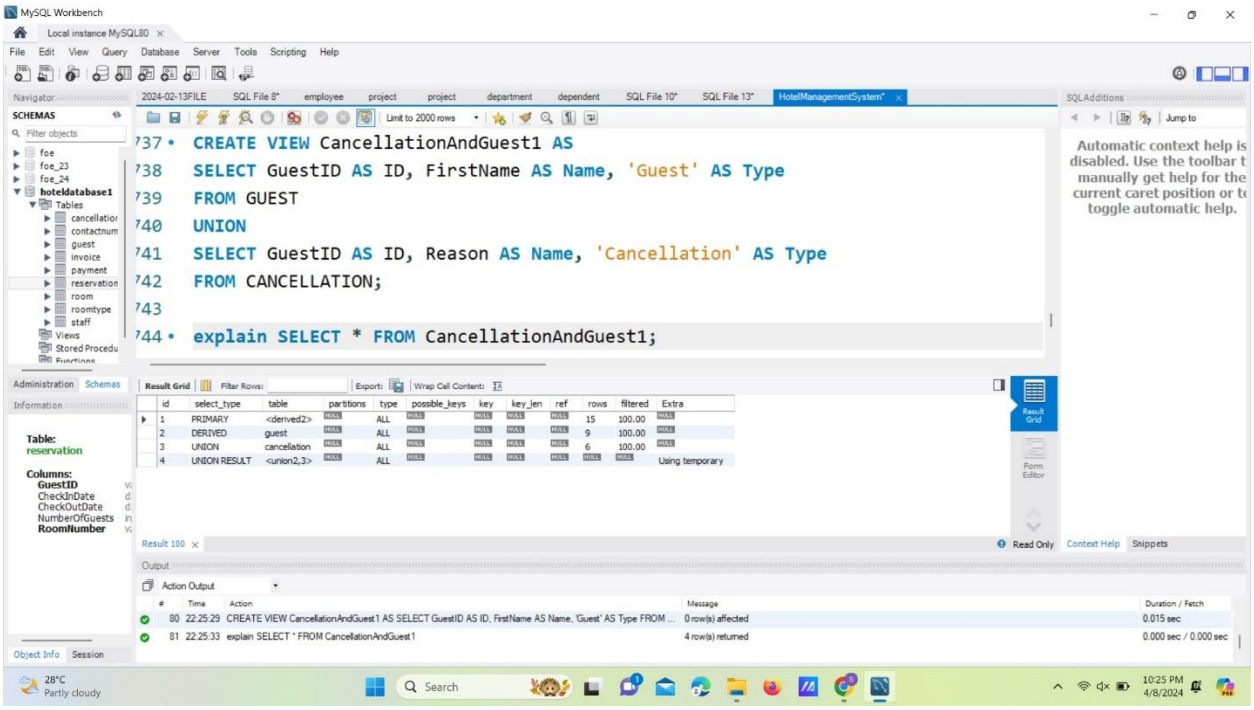
Creating Index (Guest)



Creating Index (Cancellation)

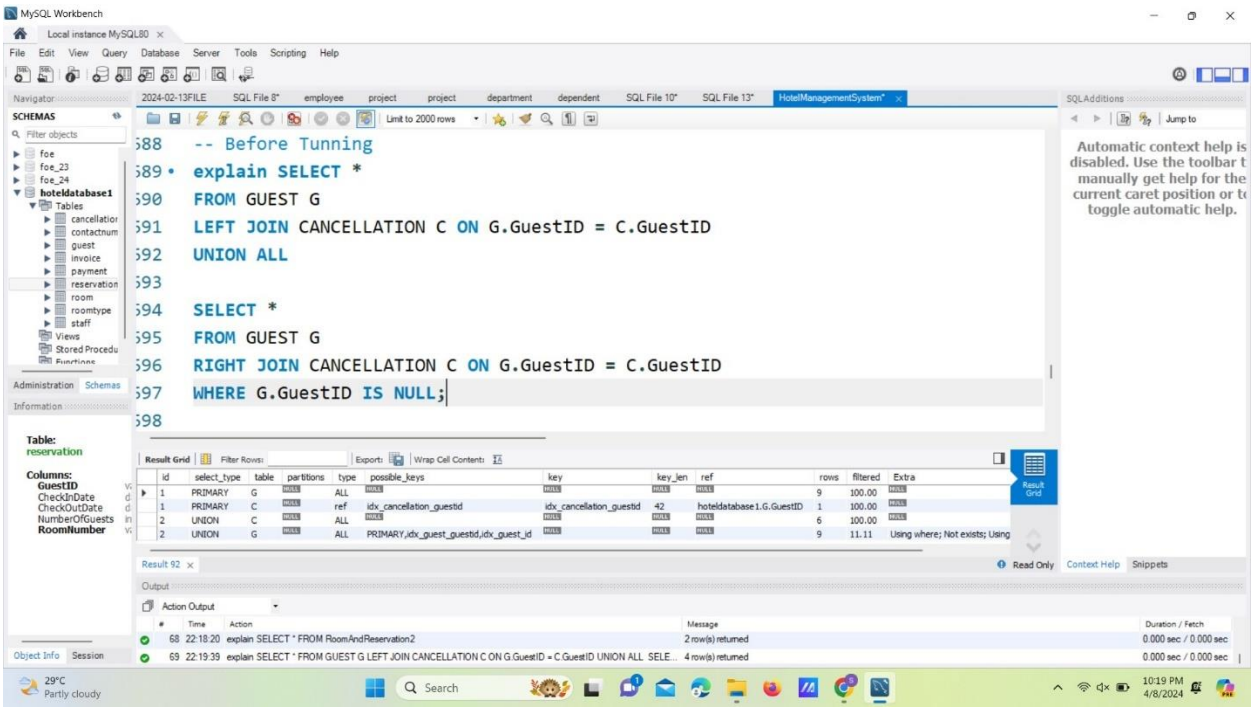


After Tuning

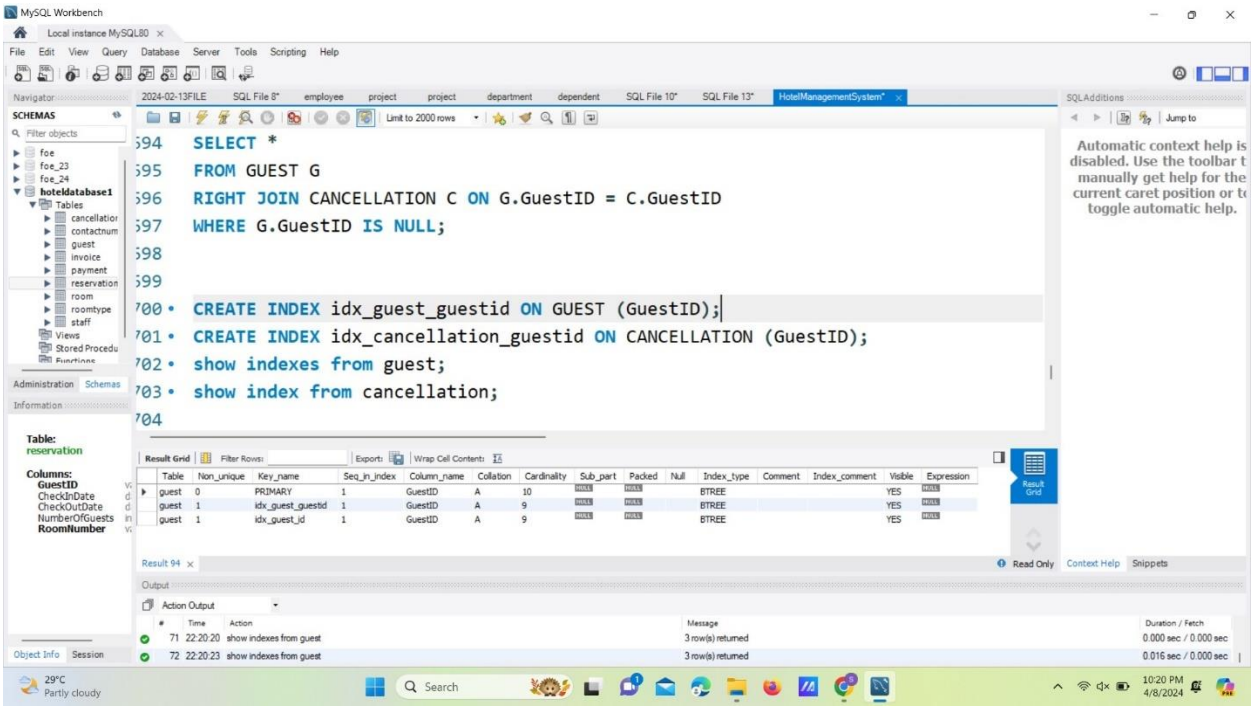


9)Tuning Full Outer Join

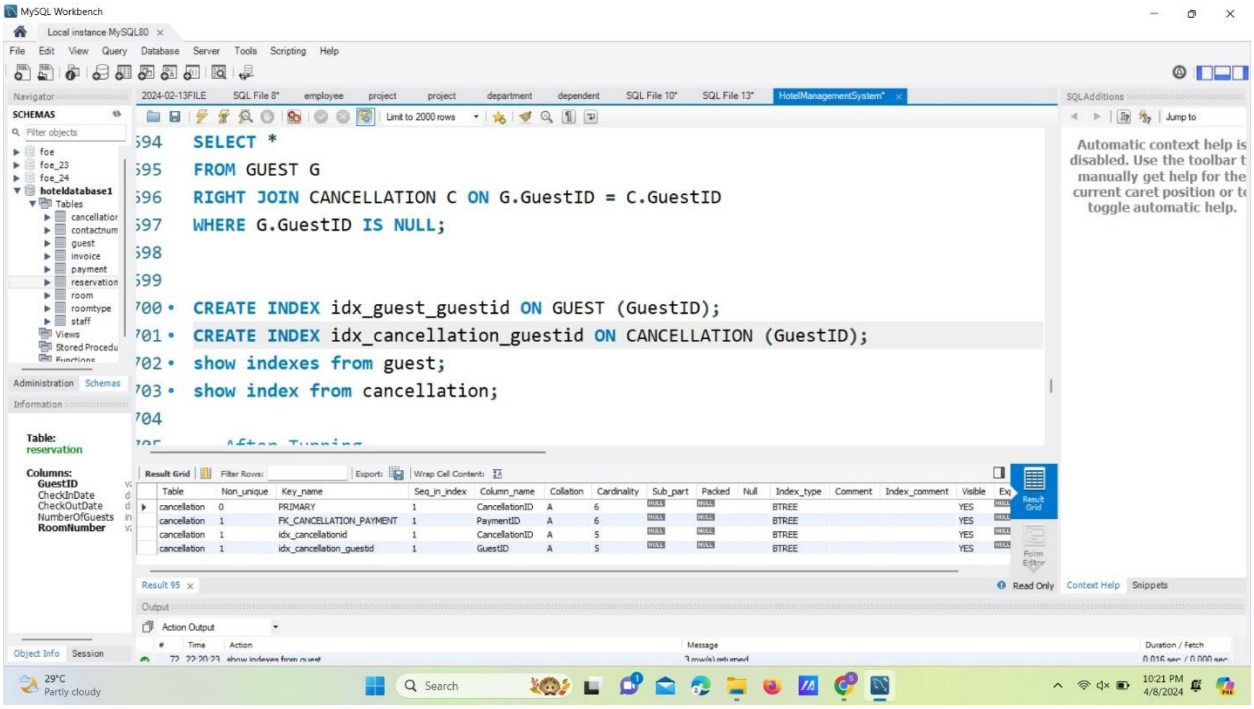
Before Tuning



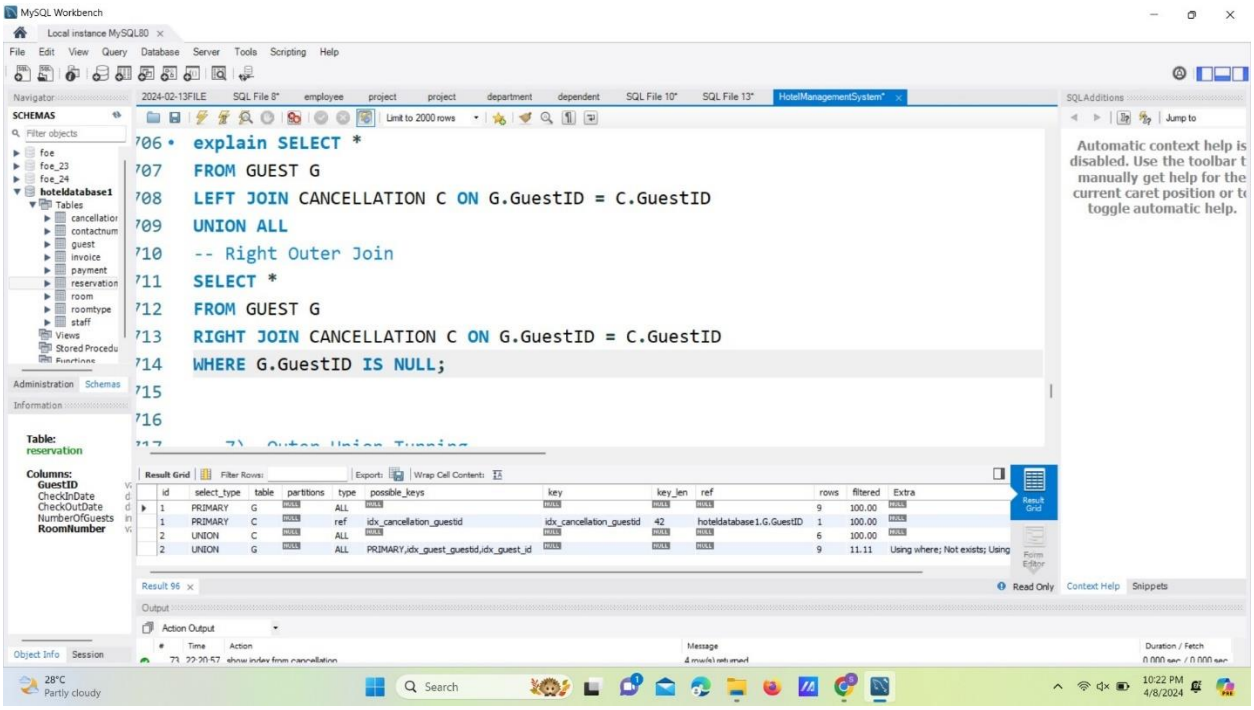
Creating Index (Guest)



Creating Index (Cancellation)

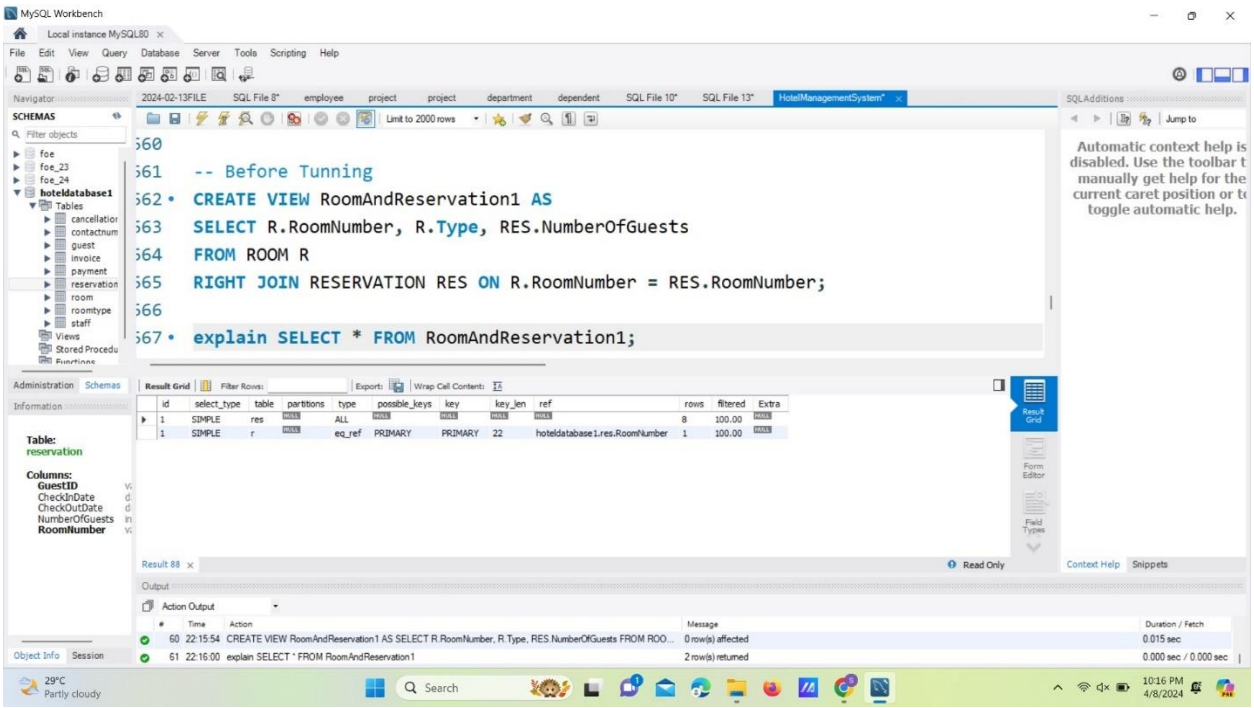


After Tuning

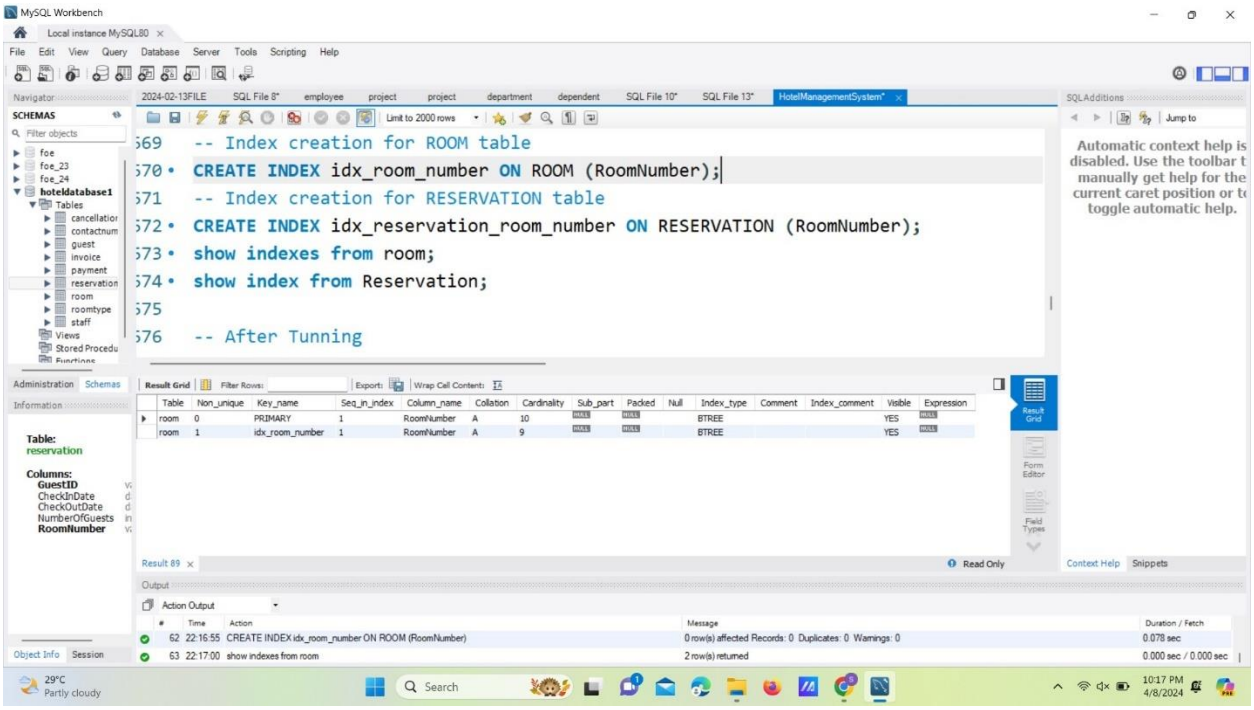


10)Tuning Right Outer Join

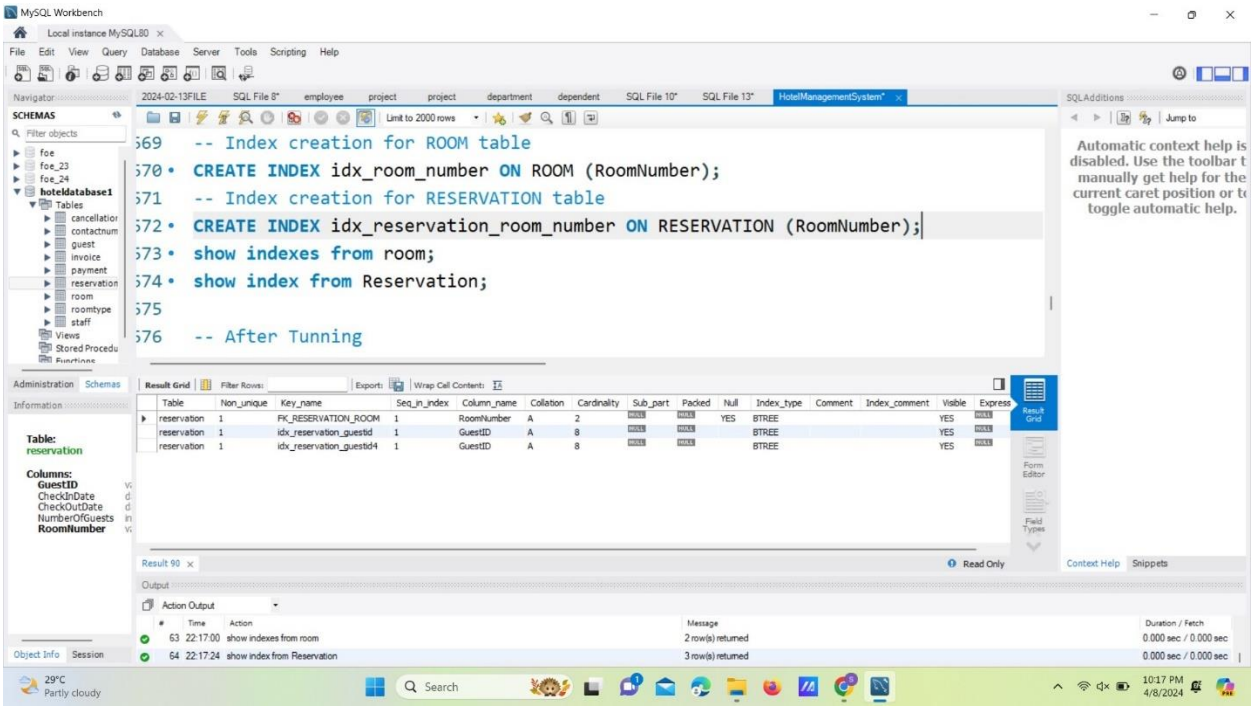
Before Tuning



Creating Index (Room)



Creating Index (Reservation)



After Tuning