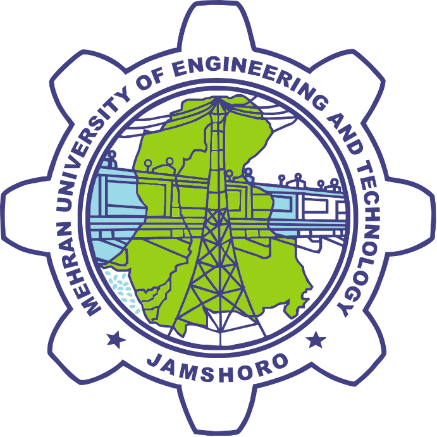
­­­

Dated: 28-03-2024



**MEHRAN UNIVERSITY OF ENGINEERING AND TECHNOLOGY JAMSORO**

A Database Project Report

on

Travel and

Tourism Management

System

Arbab Hussain Qureshi ( 22SW028 )

Ohm Parkash ( 22SW034 )

Submitted By

* **Purpose of The Database (Problem Definition):**

The travel and tourism industry faces different challenges in managing various aspects such as booking accommodations, transportation, activities, and user interactions efficiently. Traditional methods of managing these tasks involve paper-based records, and hard copied systems, leading to inefficiencies, anomalies, and difficulties in retrieving and maintaining data.

The problem definition for our DBMS revolves around optimizing the management of travel and tourism-related data to meet the needs of today's digital era. This includes addressing problems such as:

* Lack of centralized data management leading to data anomalies.
* Inefficient booking processes resulting in delays, errors, and poor customer satisfaction.
* Limited information reaches customers.
* Inadequate customer engagement tools.
* **Proposed Solution:**

To face all these challenges, there was a need for something that could centralize all the information into one place and ensure easy access, retrieval, and maintenance for which we have created the travel and tourism management system. It will enhance data accessibility, data analysis and provide seamless experience for both the travelers and tourists and service providers.

* **Aims and Objectives:**

The specific goals and targets of the Travel and Tourism Database System involves:

* **Develop a Centralized Data Management System:**

The database will centralize all travel and tourism related data, including destinations, accommodations, transportation, activities, bookings, user information, and reviews. This centralization will eliminate the need for separate systems and ensure easier access, storage, and retrieval of data.

* **Automating Processes:**

The database will automate various processes that travel agents do manually such as searching specific records, booking flights and tours, analyzing locations etc.

* **Modern User Experience:**

Users can find relevant information, make bookings, and give reviews, which will eventually lead to higher customer satisfaction.

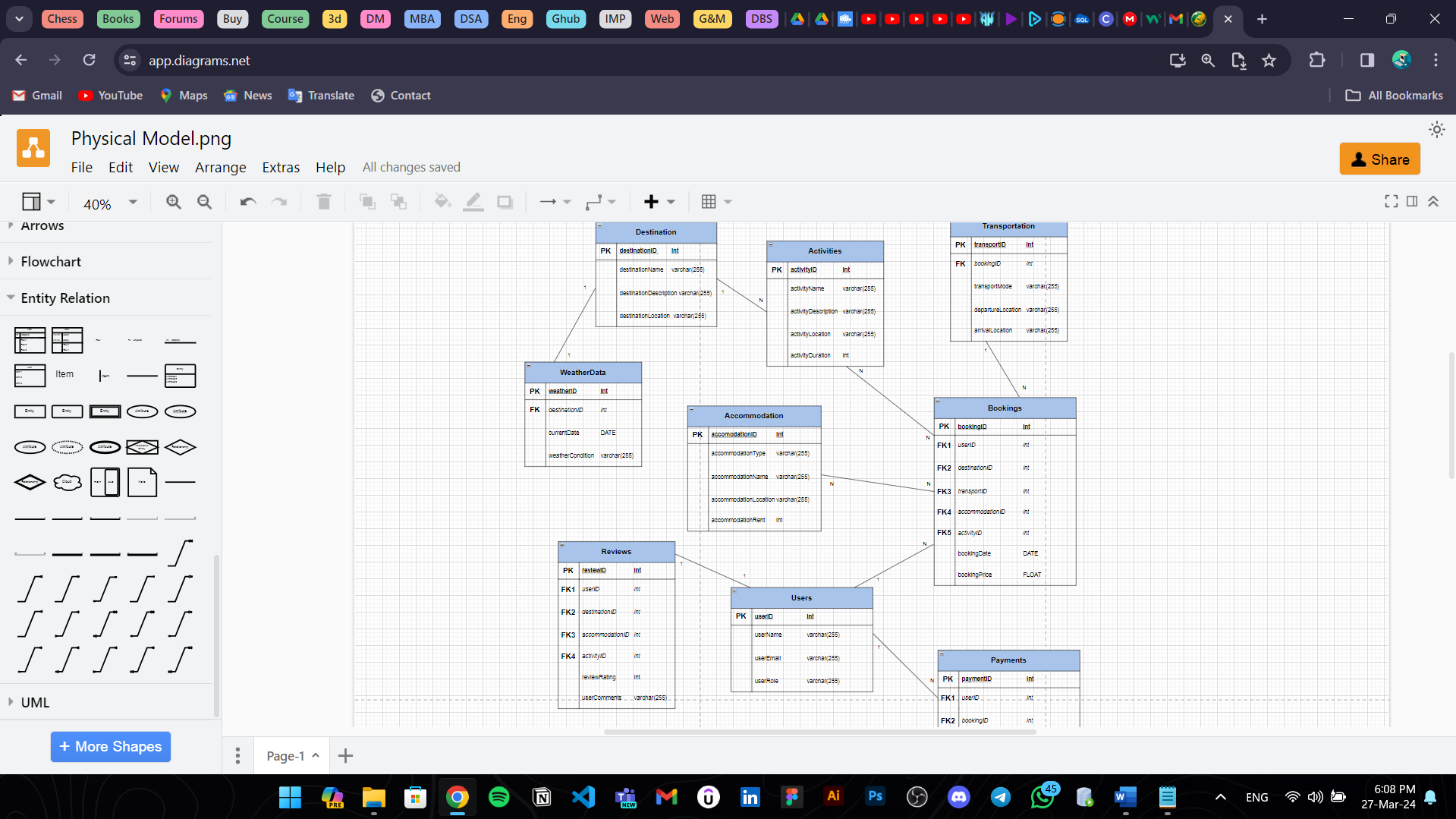
* **Enhanced Resource Allocation:**

The database system will ensure efficient resources allocation and inventory management for travelling service providers. They can optimize their services by minimizing resources and maximizing efficiency.

* **Description of Tools:**

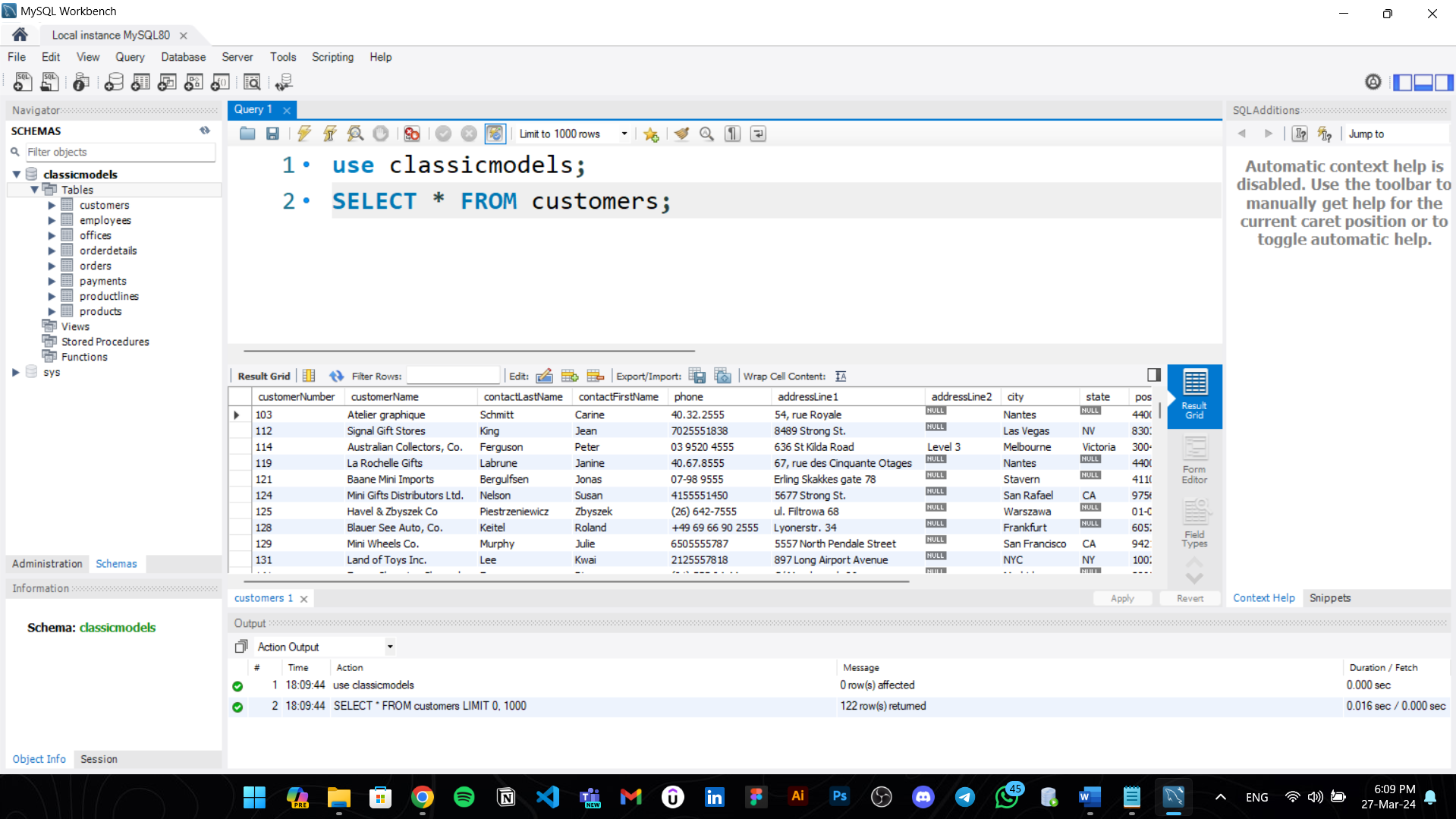
For the designing and implementation of the database, we are going to use the following tools.

**Design Tool: Draw.io**

****

We are going to use draw.io, a famous and flexible online diagramming tool, to construct database diagrams. Draw.io provides an easy-to-use interface with multiple features that facilitate the creation of lots of diagrams, such as ER Diagrams, UML Diagrams, Data Flow Diagrams etc. Users may simply add and organize shapes, connectors, and text to visualize their database architecture thanks to its drag-and-drop feature. Draw.io also makes it easy to include diagrams in project reports and presentations by offering different choices for exporting them in various formats.

**Implementation Tool: MySQL**

****

We will use MySQL, an open-source relational database management system (RDBMS), to implement the database structure and handle database operations. Because of its scalability, performance, and dependability, MySQL is a popular option for database applications of all sizes. It has extensive capabilities for handling user permissions and security settings in addition to building, editing, and querying databases, tables, and data. Furthermore, MySQL provides compatibility with a wide range of frameworks and programming languages, making it easy to integrate with web and application development projects.

* **Database Specifications:**

**Entities**

There are a total of nine entities in the database. Those are

**1. Destinations:**

Each destination entity represents a specific location that tourists visit, such as cities, landmarks, or natural attractions.

**Attributes:**

destinationID

destinationName

destinationDescription

destinationLocation

**2. Transportation:**

Each transportation entity represents a mode of transportation available for travel, such as flights, trains, buses, or rental cars

**Attributes:**

transportID

bookingID

transportMode

departureLocation

arrivalLocation

**3. Accommodations:**

Each accommodation entity represents a lodging option available for tourists, such as hotels, hostels, resorts, or vacation rentals.

**Attributes:**

accomodationID

accomodationType

accomodationName

accomodationLocation

accomodationRent

**4. Activities:**

Each activity entity represents something tourists can do at destinations, such as sky diving, swimming, events etc.

**Attributes:**

activityID

activityName

activityDescription

activityLocation

activityDuration

**5. Users:**

Each user entity represents an individual interacting with the database system, including tourists, travel agents, and administrators.

**Attributes:**

userID

userName

userEmail

userRole

**6. Bookings:**

Each booking entity represents a reservation made by a user for transportation, accommodations, or activities.

**Attributes:**

bookingID

userID

destinationID

transportID

accomodationID

activityID

bookingDate

bookingPrice

**7. Reviews:**

Each review entity represents feedback provided by users about destinations, accommodations, activities, etc.

**Attributes:**

reviewID

userID

destinationID

accomodationID

activityID

reviewRating

userComments

**8. Payments:**

Each payment entity represents a financial transaction related to bookings.

**Attributes:**

paymentID

userID

bookingID

amount

paymentDate

**9. WeatherData:**

Each weather data entity represents information about weather conditions at various destinations.

**Attributes:**

weatherID

destinationID

currentDate

weatherCondition

**Relationships**

**One – One Relationships:**

1. Destination –> WeatherData

2. User –> Review

**One – Many Relationships:**

1. User -> Bookings

2. User -> Payments

3. Bookings -> Transportation

4. Destination -> Activities

**Many – Many Relationships:**

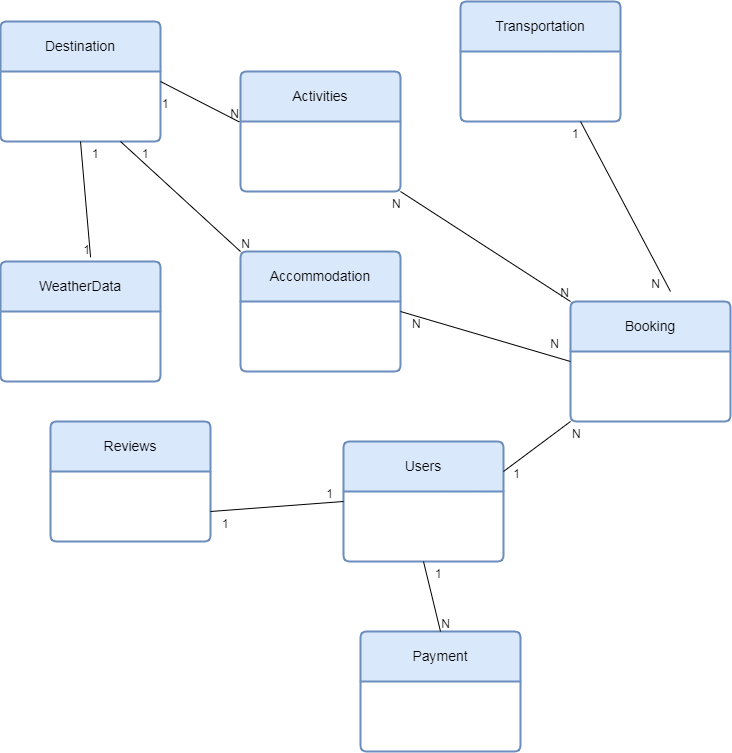
1. Activities -> Bookings

2. Accommodation -> Bookings

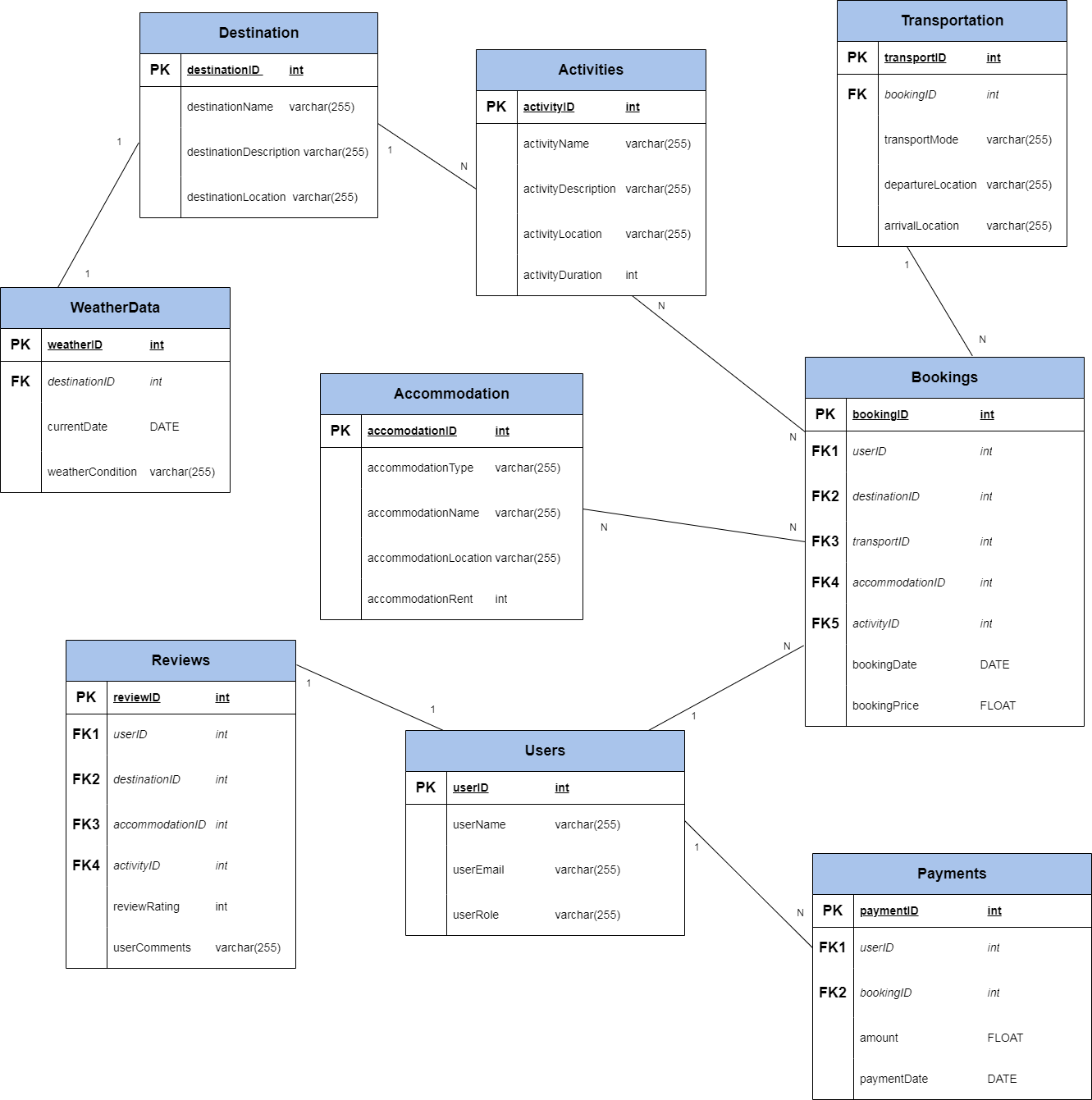
These are the identified relationships. Using these database specifications, we have developed the database design. i.e., Conceptual, Logical and Physical Database Models which are illustrated below.

* **Database Design:**

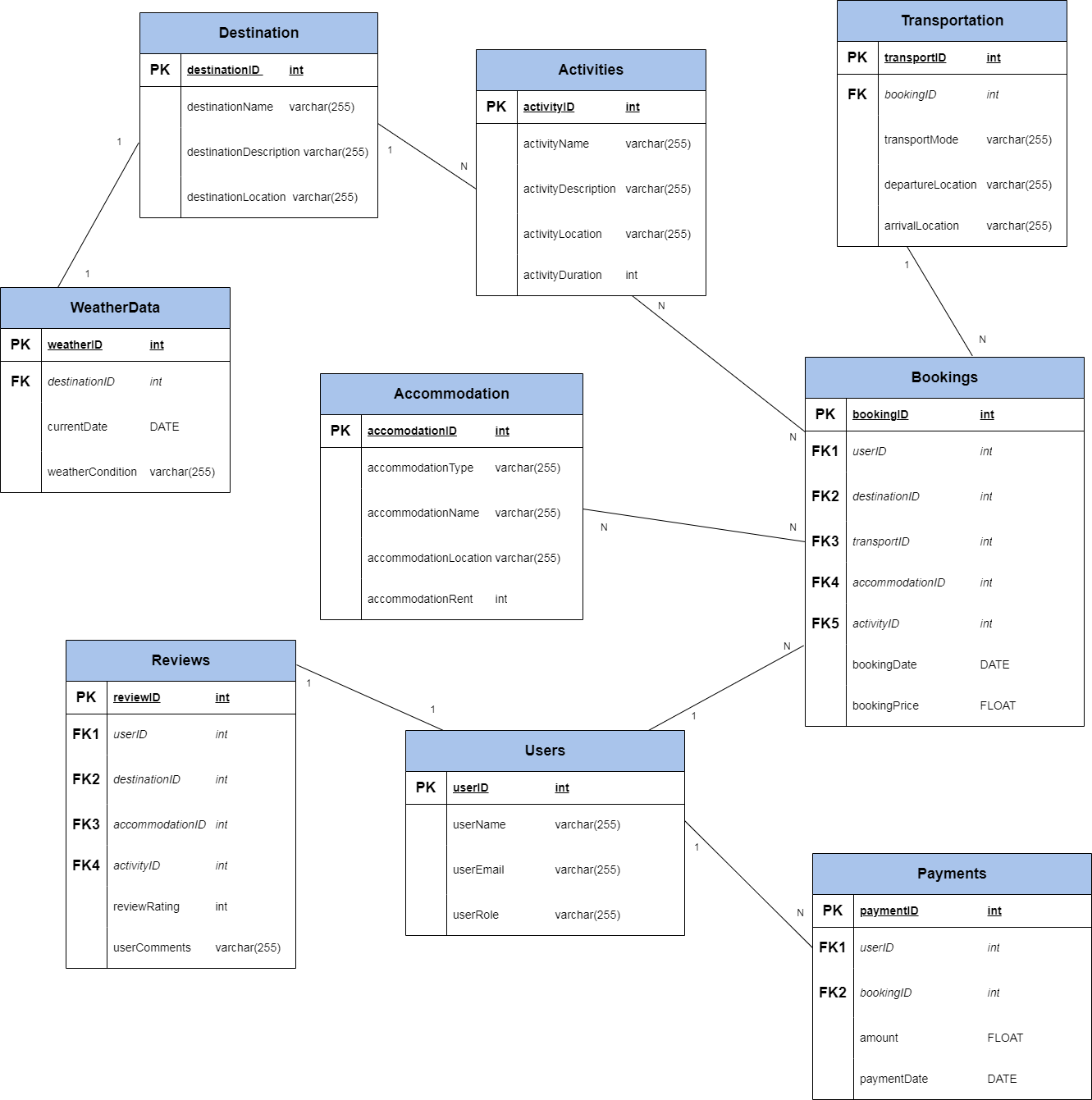
**Conceptual Data Model**

****

**Logical Data Model**



**Physical Data Model**



**DATABASE SYSTEM DEVELOPMENT**

**DDL Commands:**

Following are the DDL (Data Definition Language) commands used to create the database, tables and define the overall structure of the system.

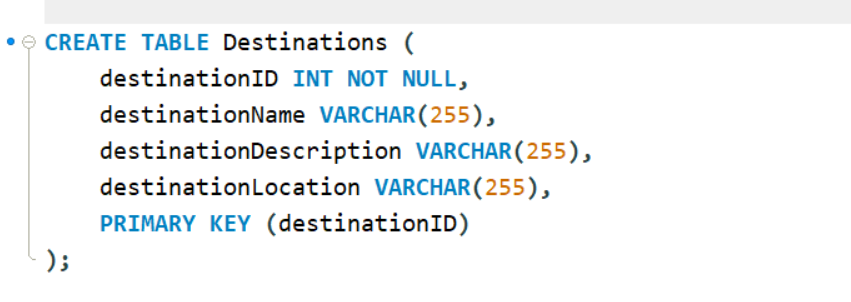
The queries and their corresponding outputs are given below.

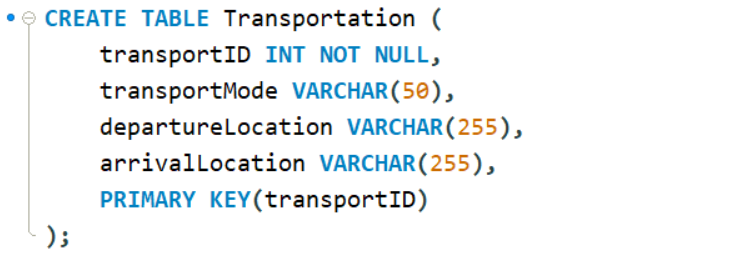
**=>Creation of Database:**

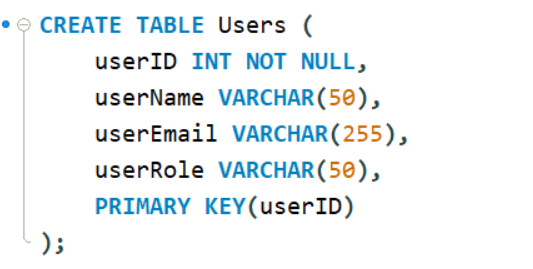
**A close-up of a sign

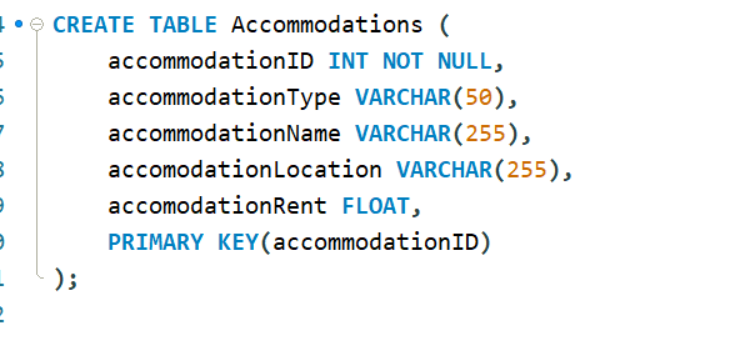
Description automatically generated**

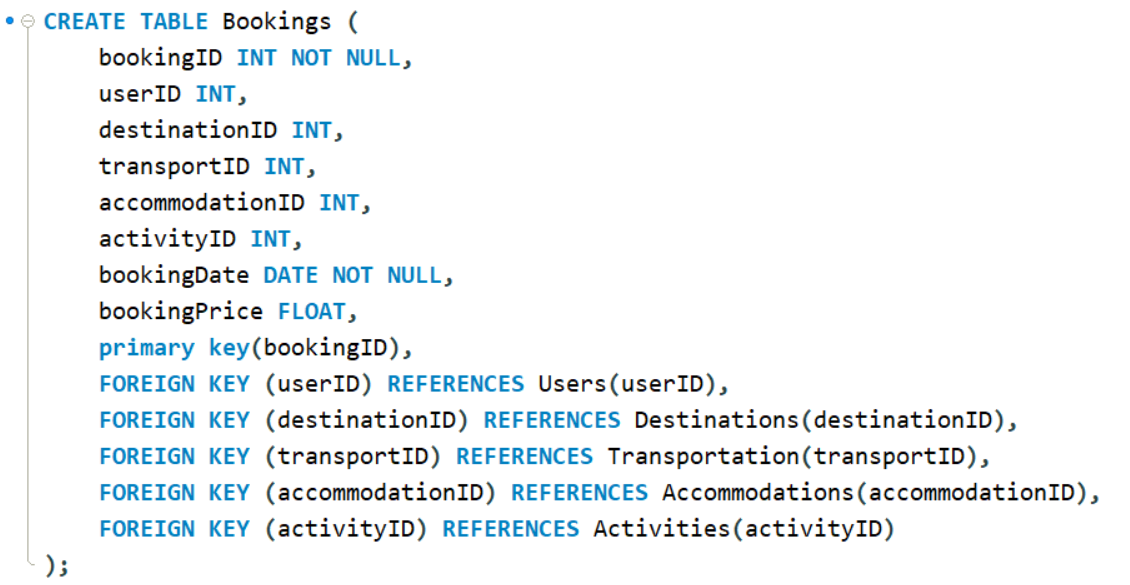
This will create the database and ensure all further commands to be applied on travel\_tourism\_db.

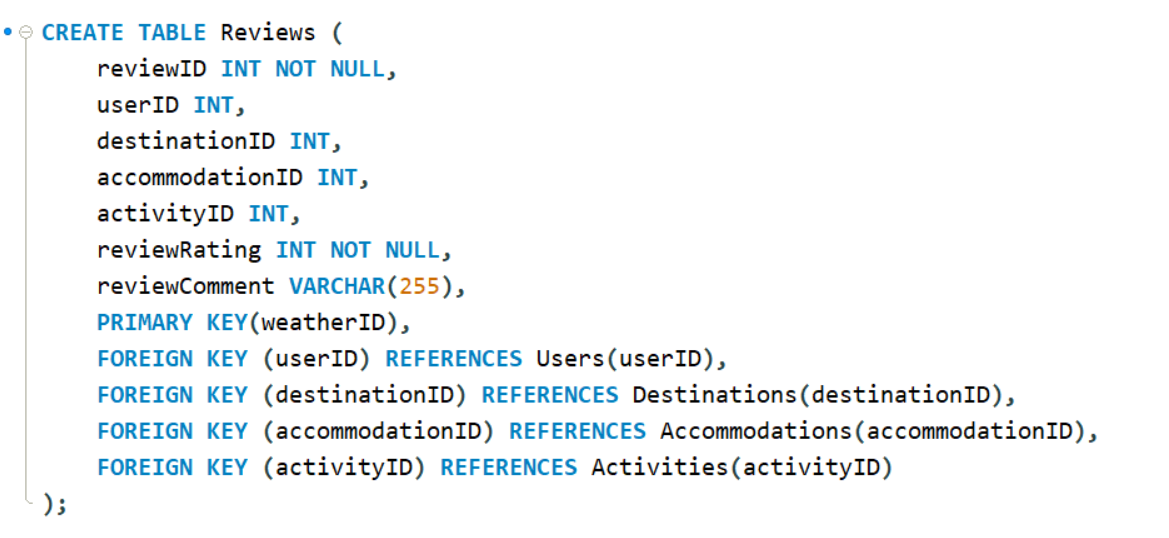
**=>Creation of Tables:**

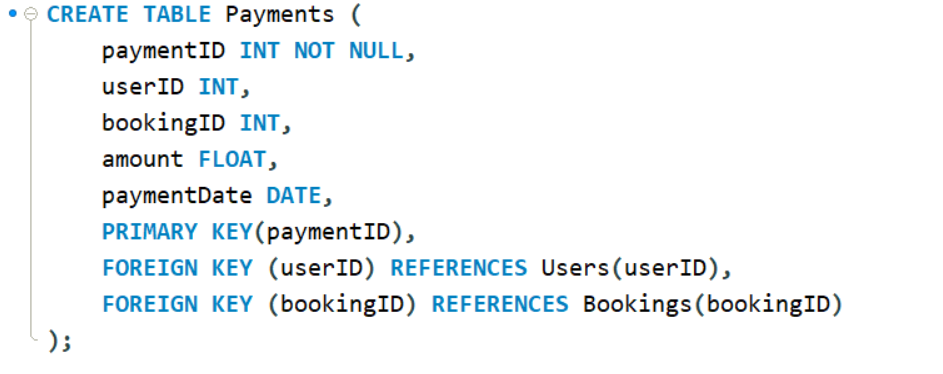


A close up of text

Description automatically generated







A computer code with text

Description automatically generated

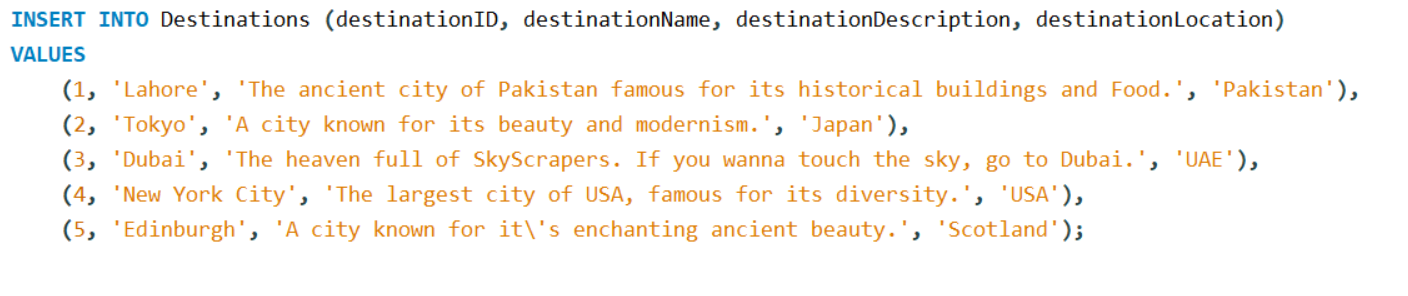
After executing all these queries. The structure of the database will look like this

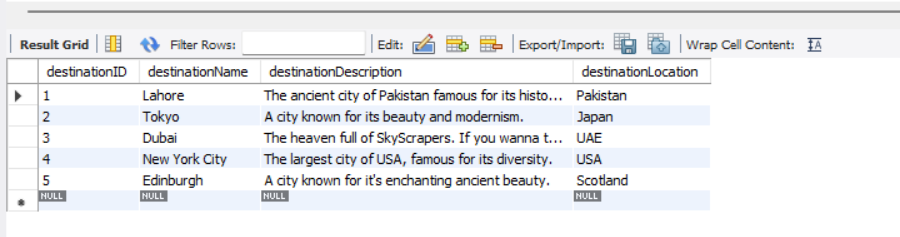
A screenshot of a computer

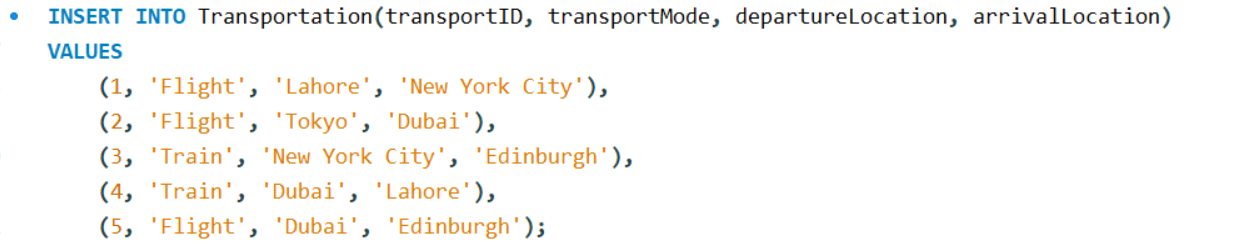
Description automatically generated

**=>Populating The Database: DML Commands**

After creating the database, we used DML (Data Manipulation Language) Commands specifically INSERT INTO command to insert the data inside relevant tables in the database. The DML Commands Queries and the populated table results are given below.

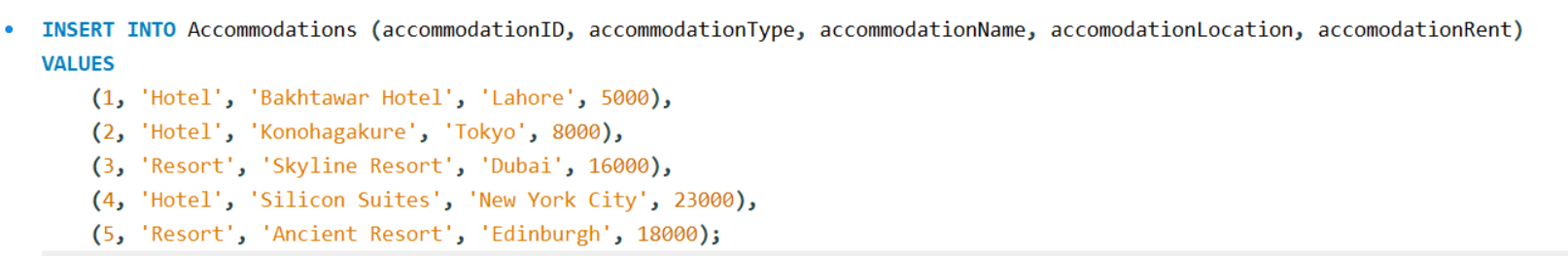


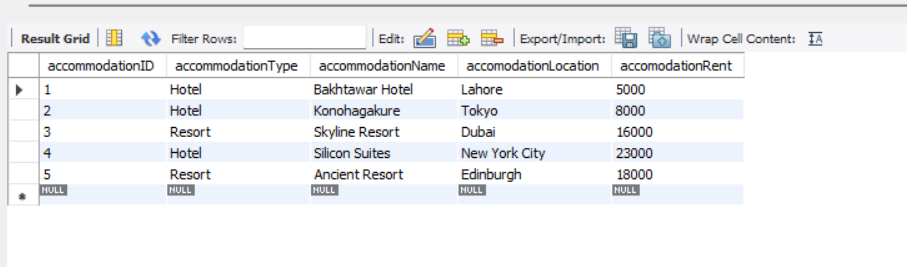


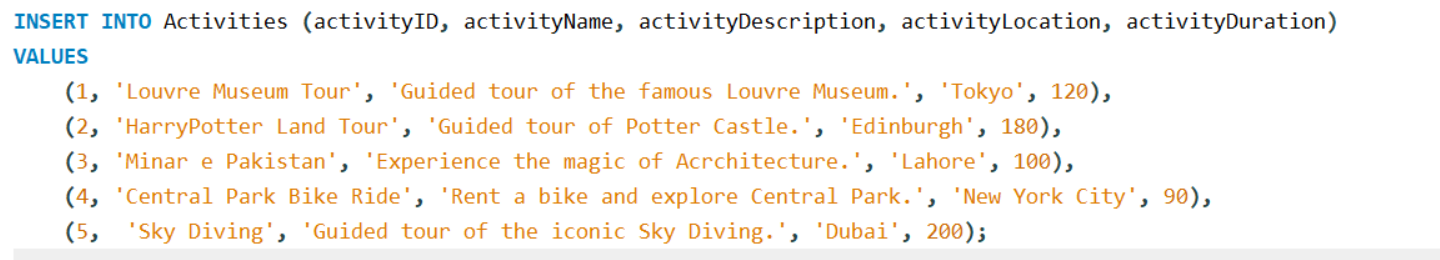


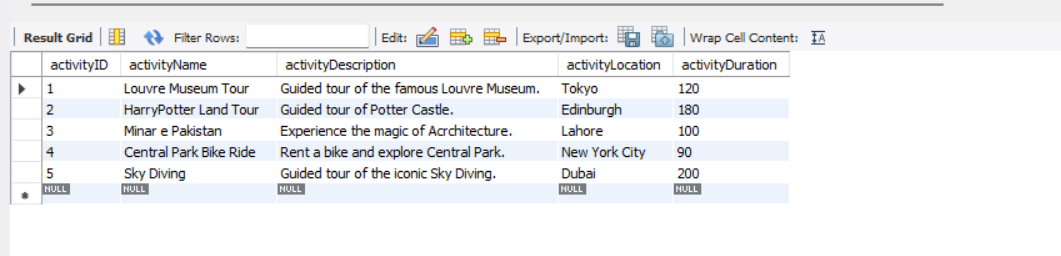
A screenshot of a computer

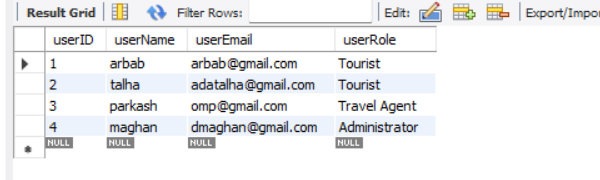
Description automatically generated

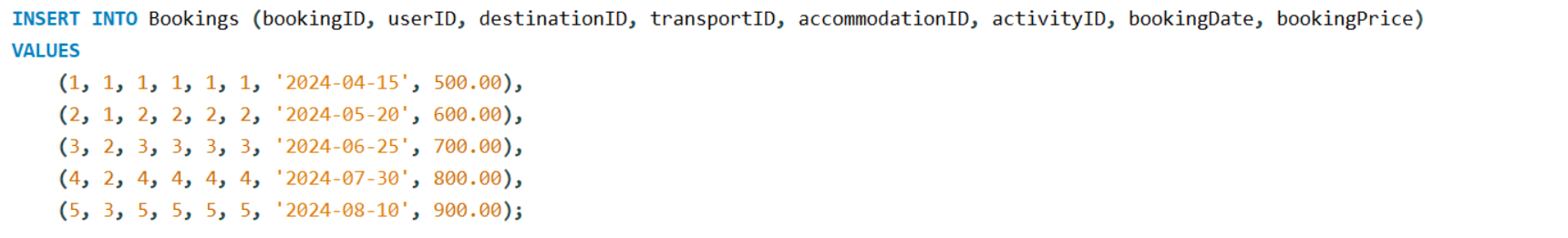


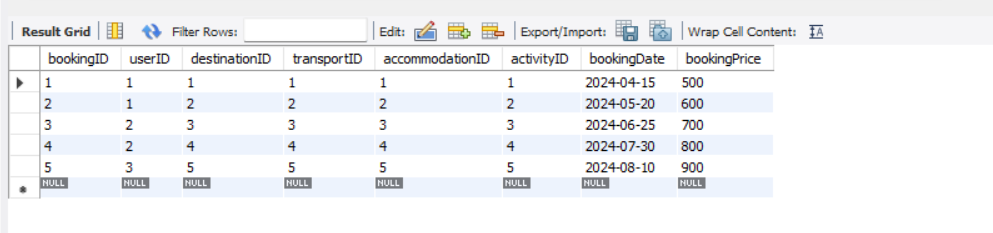


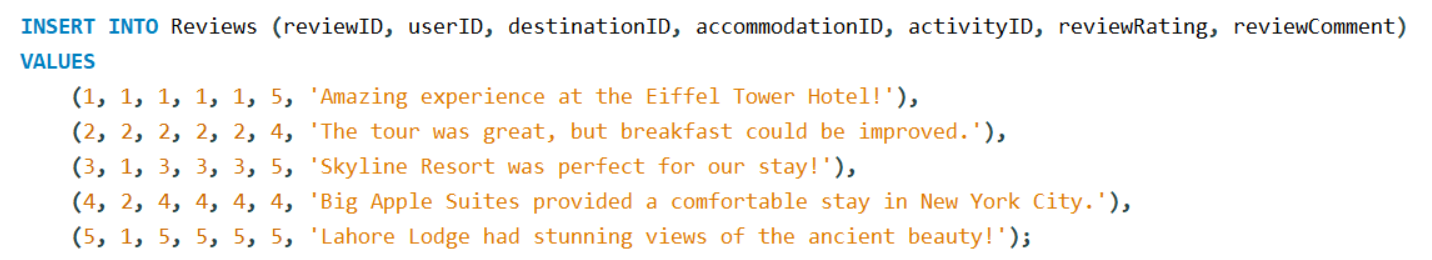


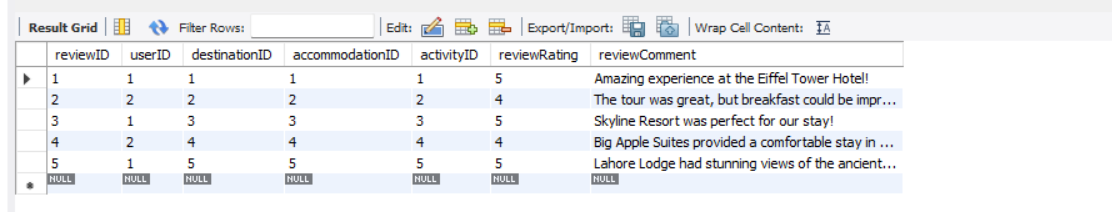


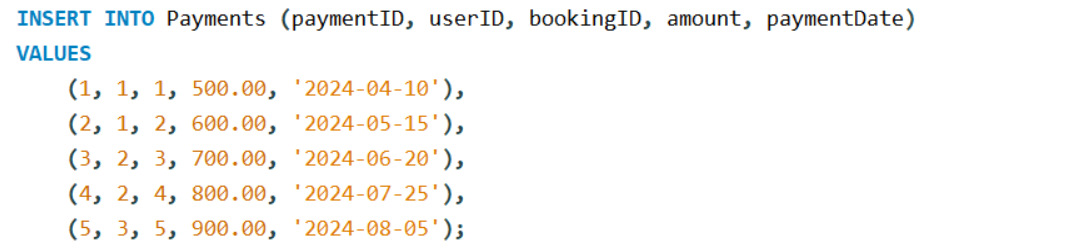
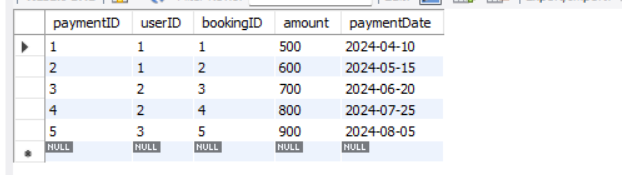


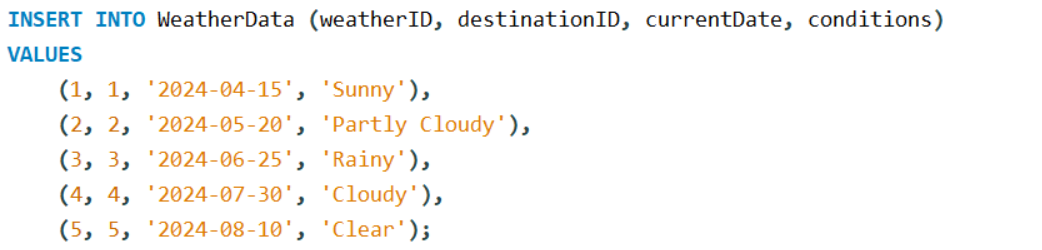
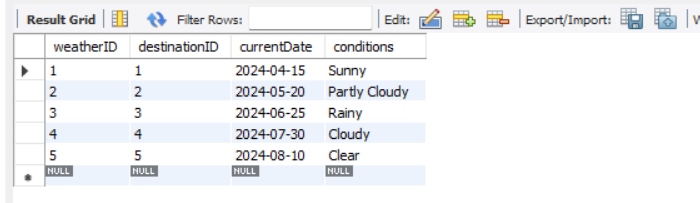






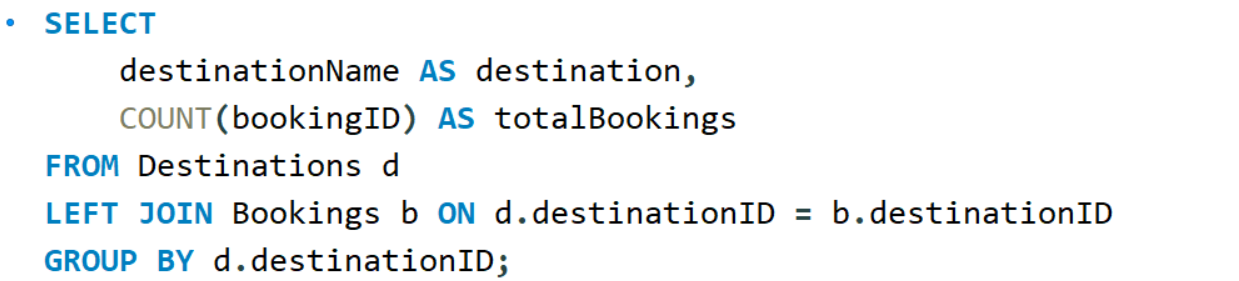


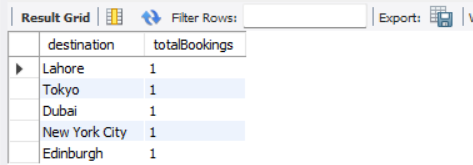




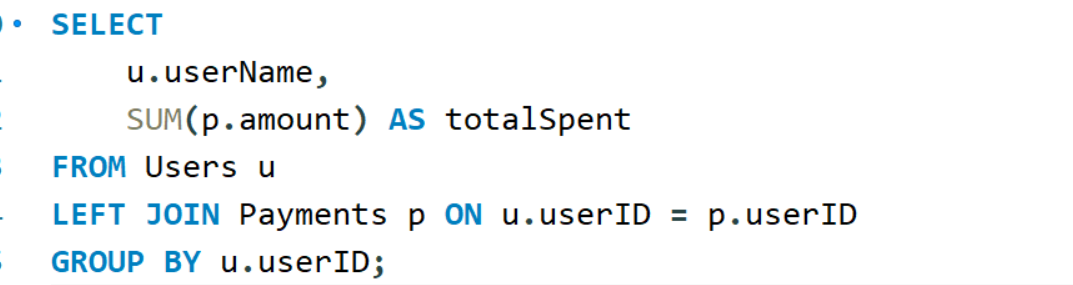
After populating the database, we ran some DQL statements to ensure that access to data in database was efficient. The queries and respective output screenshots are as under.

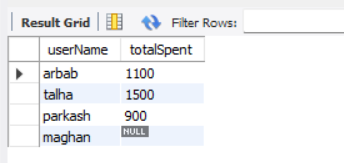
**1. Get total number of bookings for each destination:**



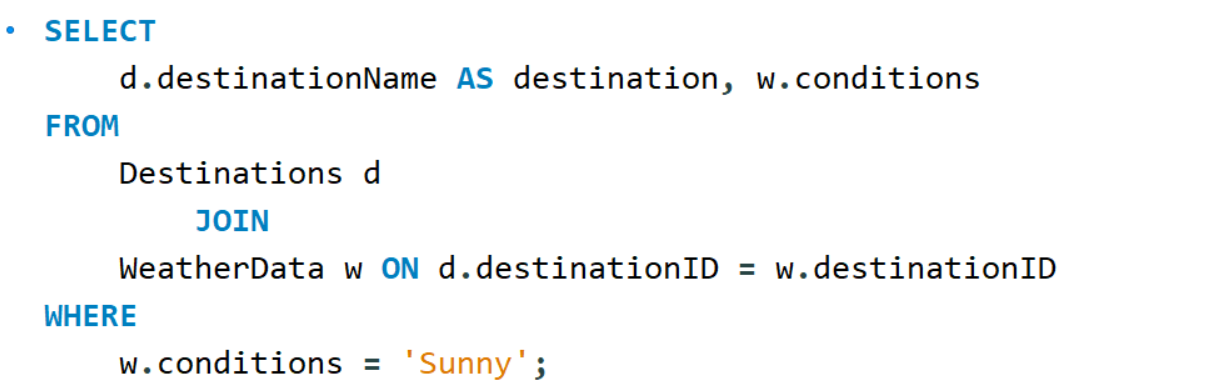


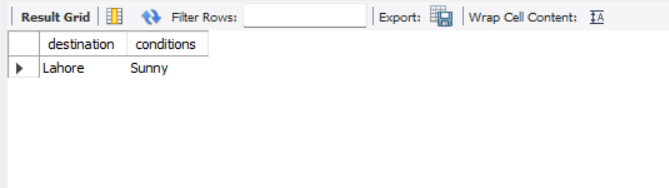
**2. Get the total amount spent by User on all his/her Bookings:**





**3. Show all Destination Names where weather is sunny.**

****

****

Hence the database is working completely fine and travelers, tourists and travel agents can easily access and fetch data from the database.

The above mentioned SQL script can also be downloaded from my GitHub id.