HOTEL BOOKING SYSTEM PROJECT

Project Summary:

This project involves the design and implementation of a comprehensive hotel booking system using SQL. The system manages core hotel operations, including guest reservations, room availability, service tracking, and payment processing. Key entities include Hotel, Room, Guest, Booking, Payment, and Service, each with clearly defined attributes and relationships. The database structure is normalized to ensure data integrity and efficient querying. SQL was used to create the schema, enforce constraints, and run complex queries to support realistic hotel management scenarios.

Entities:

- Hotel(Hotel_ID, Name, Address, Phone_Number, Email, Rating, Number_of_rooms)
- Room(Room_ID, Room_Number, Room_Type, Room_Capacity, Price, Availability)
- Service(Service_ID, Service_Date, Service_Category)
- Room_and_Services(Service_ID, Room_ID, Service_Date)
- Booking(Booking_ID, Guest_ID, Room_ID, Hotel_ID, CheckIn_Date, CheckOut_Date, Payment_Status, Payment_Date)
- Payment(Payment_ID, Amount, Payment_Date, Payment_Status, Booking_ID)
- Guest(Guest_ID, First_Name, Last_Name, Phone_Number, Email)

Step 2: Design the database using Chen ERD and UML for a conceptual ERM

MIS380-UML-ClassDiagram.vsdx

MIS380-ERD-ClassDiagram - Copy.vsdx

Step 3: Translate the conceptual design into its equivalent Relational Schema

Hotel(Hotel_ID, Name, Address, Phone_Number, Email, Rating, Number_of_rooms)
Booking(Booking_ID, Guest_ID, Room_ID, Hotel_ID, CheckIn_Date, CheckOut_Date, Payment_Status, Payment_Date)

- Foreign Key: Guest_ID references Guest
- Foreign Key: Room_ID references Room
- Foreign Key: Hotel_ID references Hotel

Guest(Guest_ID, First_Name, Last_Name, Phone_Number, Email, Age, Loyalty_Status)
Payment(Payment_ID, Amount, Payment_Date, Payment_Status, Booking_ID)

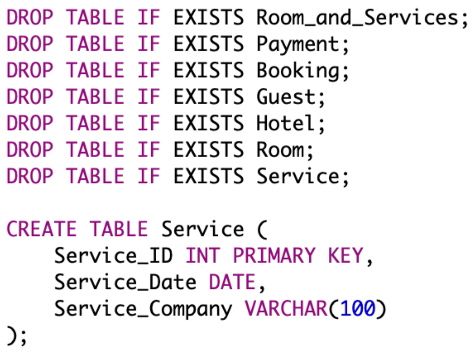
- Foreign Key: Booking_ID references Booking

Room(Room_ID, Room_Number, Room_Type, Room_Capacity, Price, Availability)
Service(Service_ID, Service_Date, Service_Company)
Room_and_Services(Service_ID, Room_ID, Service_Date)

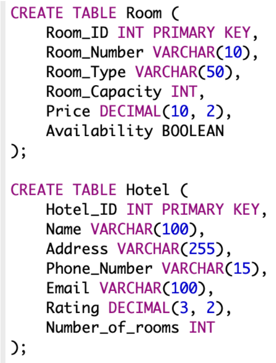
- Foreign Key: Service_ID references Service
- Foreign Key: Room_ID references Room

Step 4: Build the actual database with sufficient make up data and a few queries

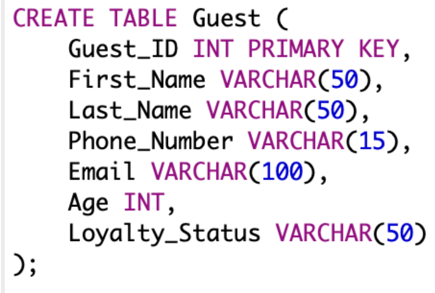
CODE:



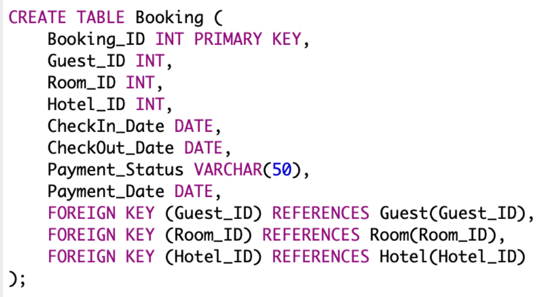
```
DROP TABLE IF EXISTS Room_and_Services;
DROP TABLE IF EXISTS Payment;
DROP TABLE IF EXISTS Booking;
DROP TABLE IF EXISTS Guest;
DROP TABLE IF EXISTS Hotel;
DROP TABLE IF EXISTS Room;
DROP TABLE IF EXISTS Service;

CREATE TABLE Service (
    Service_ID INT PRIMARY KEY,
    Service_Date DATE,
    Service_Company VARCHAR(100)
);
```



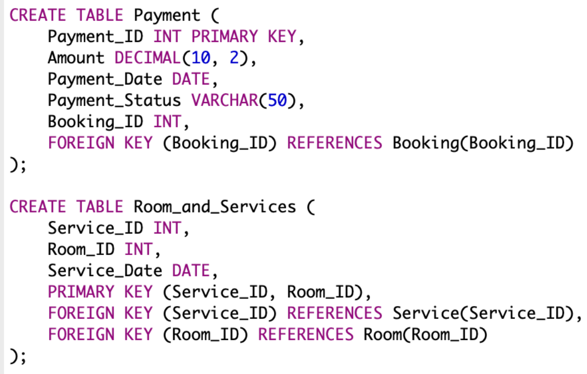
```
CREATE TABLE Room (
    Room_ID INT PRIMARY KEY,
    Room_Number VARCHAR(10),
    Room_Type VARCHAR(50),
    Room_Capacity INT,
    Price DECIMAL(10, 2),
    Availability BOOLEAN
);

CREATE TABLE Hotel (
    Hotel_ID INT PRIMARY KEY,
    Name VARCHAR(100),
    Address VARCHAR(255),
    Phone_Number VARCHAR(15),
    Email VARCHAR(100),
    Rating DECIMAL(3, 2),
    Number_of_rooms INT
);
```



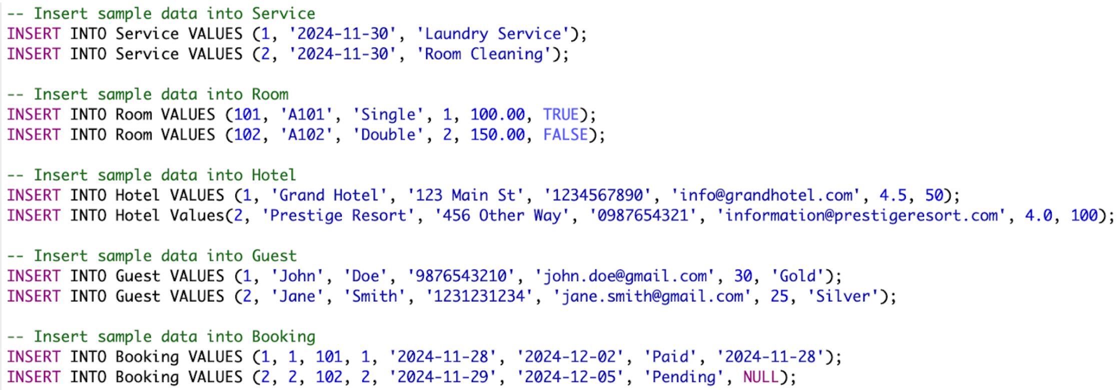
```
CREATE TABLE Guest (
    Guest_ID INT PRIMARY KEY,
    First_Name VARCHAR(50),
    Last_Name VARCHAR(50),
    Phone_Number VARCHAR(15),
    Email VARCHAR(100),
    Age INT,
    Loyalty_Status VARCHAR(50)
);
```



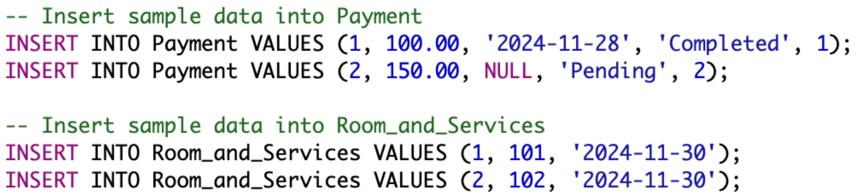
```
CREATE TABLE Booking (
    Booking_ID INT PRIMARY KEY,
    Guest_ID INT,
    Room_ID INT,
    Hotel_ID INT,
    CheckIn_Date DATE,
    CheckOut_Date DATE,
    Payment_Status VARCHAR(50),
    Payment_Date DATE,
    FOREIGN KEY (Guest_ID) REFERENCES Guest(Guest_ID),
    FOREIGN KEY (Room_ID) REFERENCES Room(Room_ID),
    FOREIGN KEY (Hotel_ID) REFERENCES Hotel(Hotel_ID)
);
```



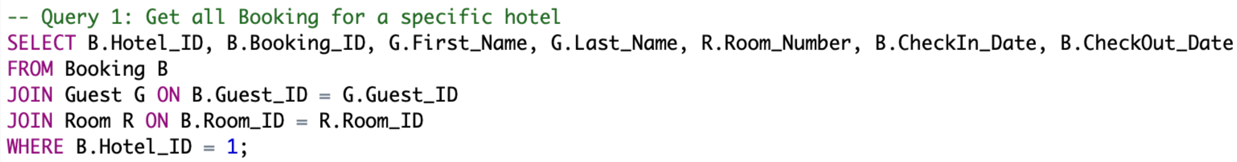
```
CREATE TABLE Payment (
    Payment_ID INT PRIMARY KEY,
    Amount DECIMAL(10, 2),
    Payment_Date DATE,
    Payment_Status VARCHAR(50),
    Booking_ID INT,
    FOREIGN KEY (Booking_ID) REFERENCES Booking(Booking_ID)
);

CREATE TABLE Room_and_Services (
    Service_ID INT,
    Room_ID INT,
    Service_Date DATE,
    PRIMARY KEY (Service_ID, Room_ID),
    FOREIGN KEY (Service_ID) REFERENCES Service(Service_ID),
    FOREIGN KEY (Room_ID) REFERENCES Room(Room_ID)
);
```



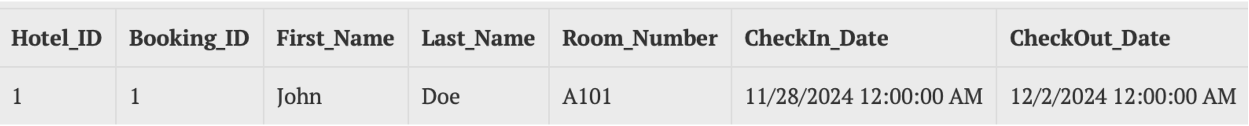
```
-- Insert sample data into Service
INSERT INTO Service VALUES (1, '2024-11-30', 'Laundry Service');
INSERT INTO Service VALUES (2, '2024-11-30', 'Room Cleaning');

-- Insert sample data into Room
INSERT INTO Room VALUES (101, 'A101', 'Single', 1, 100.00, TRUE);
INSERT INTO Room VALUES (102, 'A102', 'Double', 2, 150.00, FALSE);

-- Insert sample data into Hotel
INSERT INTO Hotel VALUES (1, 'Grand Hotel', '123 Main St', '1234567890', 'info@grandhotel.com', 4.5, 50);
INSERT INTO Hotel Values(2, 'Prestige Resort', '456 Other Way', '0987654321', 'information@prestigeresort.com', 4.0, 100);

-- Insert sample data into Guest
INSERT INTO Guest VALUES (1, 'John', 'Doe', '9876543210', 'john.doe@gmail.com', 30, 'Gold');
INSERT INTO Guest VALUES (2, 'Jane', 'Smith', '1231231234', 'jane.smith@gmail.com', 25, 'Silver');

-- Insert sample data into Booking
INSERT INTO Booking VALUES (1, 1, 101, 1, '2024-11-28', '2024-12-02', 'Paid', '2024-11-28');
INSERT INTO Booking VALUES (2, 2, 102, 2, '2024-11-29', '2024-12-05', 'Pending', NULL);
```



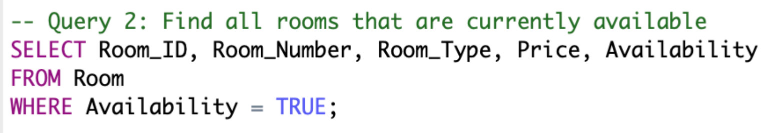
```
-- Insert sample data into Payment
INSERT INTO Payment VALUES (1, 100.00, '2024-11-28', 'Completed', 1);
INSERT INTO Payment VALUES (2, 150.00, NULL, 'Pending', 2);

-- Insert sample data into Room_and_Services
INSERT INTO Room_and_Services VALUES (1, 101, '2024-11-30');
INSERT INTO Room_and_Services VALUES (2, 102, '2024-11-30');
```



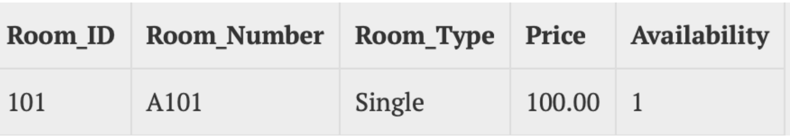
```
-- Query 1: Get all Booking for a specific hotel
SELECT B.Hotel_ID, B.Booking_ID, G.First_Name, G.Last_Name, R.Room_Number, B.CheckIn_Date, B.CheckOut_Date
FROM Booking B
JOIN Guest G ON B.Guest_ID = G.Guest_ID
JOIN Room R ON B.Room_ID = R.Room_ID
WHERE B.Hotel_ID = 1;
```



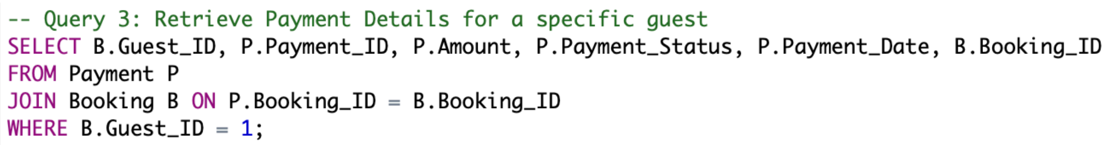
Hotel_ID	Booking_ID	First_Name	Last_Name	Room_Number	CheckIn_Date	CheckOut_Date
1	1	John	Doe	A101	11/28/2024 12:00:00 AM	12/2/2024 12:00:00 AM



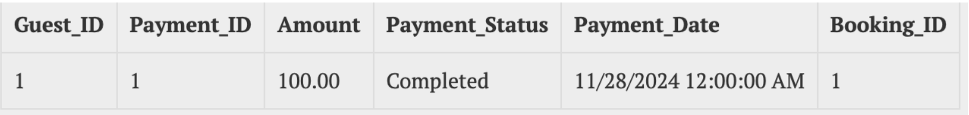
```
-- Query 2: Find all rooms that are currently available
SELECT Room_ID, Room_Number, Room_Type, Price, Availability
FROM Room
WHERE Availability = TRUE;
```



Room_ID	Room_Number	Room_Type	Price	Availability
101	A101	Single	100.00	1



```
-- Query 3: Retrieve Payment Details for a specific guest
SELECT B.Guest_ID, P.Payment_ID, P.Amount, P.Payment_Status, P.Payment_Date, B.Booking_ID
FROM Payment P
JOIN Booking B ON P.Booking_ID = B.Booking_ID
WHERE B.Guest_ID = 1;
```



Guest_ID	Payment_ID	Amount	Payment_Status	Payment_Date	Booking_ID
1	1	100.00	Completed	11/28/2024 12:00:00 AM	1

```sql
-- Query 4: List all servies availed by a specific room
SELECT RAS.Room_ID, RAS.Service_ID, S.Service_Company, RAS.Service_Date
FROM Room_and_Services RAS
JOIN Service S ON RAS.Service_ID = S.Service_ID
WHERE RAS.Room_ID = 101;
```

Room_ID	Service_ID	Service_Company	Service_Date
101	1	Laundry Service	11/30/2024 12:00:00 AM