Multi-City Hotel Chain Management System Database Design

Introduction

This document outlines the schema and sample data for a multi-city hotel chain management system. The system maintains data about hotels, rooms, guests, employees, bookings, and guest feedback. The schema ensures efficient data management and integrity for the hotel's operations.

Database Schema

Database Creation

The database is created and activated using the following SQL commands:

sql
CopyEdit

```
CREATE DATABASE HotelManagementSystem;
USE HotelManagementSystem;
```

---

1. Hotels Table

The `Hotel` table stores information about hotels in the chain.

Schema:

sql
CopyEdit

```
CREATE TABLE Hotel (
    hotel_code INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    city VARCHAR(255) NOT NULL,
    manager_id INT NOT NULL,
```

```
    num_rooms INT NOT NULL,
    star_rating INT NOT NULL
);
```

---

2. Rooms Table

The `Room` table stores information about individual rooms in each hotel.

Schema:

sql
CopyEdit

```
CREATE TABLE Room (
    room_number INT NOT NULL,
    hotel_code INT NOT NULL,
    type VARCHAR(50) NOT NULL,
    price_per_night DECIMAL(10, 2) NOT NULL,
    availability_status BOOLEAN NOT NULL,
    PRIMARY KEY (room_number, hotel_code),
    FOREIGN KEY (hotel_code) REFERENCES Hotel(hotel_code)
);
```

---

3. Guests Table

The `Guest` table stores information about guests, including loyalty level and booking history.

Schema:

sql
CopyEdit

```
CREATE TABLE Guest (
    guest_id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    loyalty_level VARCHAR(50) NOT NULL,
    booking_history TEXT
);
```

---

4. Employees Table

The `Employee` table stores details about employees assigned to each hotel.

Schema:

sql
CopyEdit

```
CREATE TABLE Employee (
    employee_id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    role VARCHAR(50) NOT NULL,
    hotel_code INT NOT NULL,
    shift_details TEXT,
    FOREIGN KEY (hotel_code) REFERENCES Hotel(hotel_code)
);
```

---

5. Bookings Table

The `Booking` table links guests with their room bookings.

Schema:

sql
CopyEdit

```
CREATE TABLE Booking (
    booking_id INT AUTO_INCREMENT PRIMARY KEY,
    guest_id INT NOT NULL,
    room_number INT NOT NULL,
    hotel_code INT NOT NULL,
    check_in_date DATE NOT NULL,
    check_out_date DATE NOT NULL,
    total_bill DECIMAL(15, 2) NOT NULL,
    FOREIGN KEY (guest_id) REFERENCES Guest(guest_id),
    FOREIGN KEY (room_number, hotel_code) REFERENCES Room(room_number,
hotel_code)
);
```

---

6. Feedback Table

The `Feedback` table stores guest feedback for each booking.

Schema:

sql
CopyEdit

```
CREATE TABLE Feedback (
    feedback_id INT AUTO_INCREMENT PRIMARY KEY,
    booking_id INT NOT NULL,
    feedback_text TEXT,
    FOREIGN KEY (booking_id) REFERENCES Booking(booking_id)
);
```

Sample Data Insertion

1. Insert Data into Hotels

sql
CopyEdit

```
INSERT INTO Hotel (name, city, manager_id, num_rooms, star_rating)
VALUES
('Grand Hyatt', 'New York', 1, 200, 5),
('Marriott', 'Los Angeles', 2, 150, 4);
```

2. Insert Data into Employees

sql
CopyEdit

```
INSERT INTO Employee (name, role, hotel_code, shift_details) VALUES
('John Doe', 'Manager', 1, 'Morning Shift'),
('Jane Smith', 'Receptionist', 1, 'Evening Shift'),
('Mark Lee', 'Manager', 2, 'Morning Shift');
```

3. Insert Data into Rooms

sql
CopyEdit

```
INSERT INTO Room (room_number, hotel_code, type, price_per_night,
availability_status) VALUES
(101, 1, 'Deluxe', 200.00, TRUE),
(102, 1, 'Suite', 300.00, TRUE),
(201, 2, 'Standard', 150.00, TRUE);
```

4. Insert Data into Guests

sql
CopyEdit

```
INSERT INTO Guest (name, loyalty_level, booking_history) VALUES
('Alice Brown', 'Gold', 'Booking 1, Booking 2'),
('Bob White', 'Silver', 'Booking 3');
```

5. Insert Data into Bookings

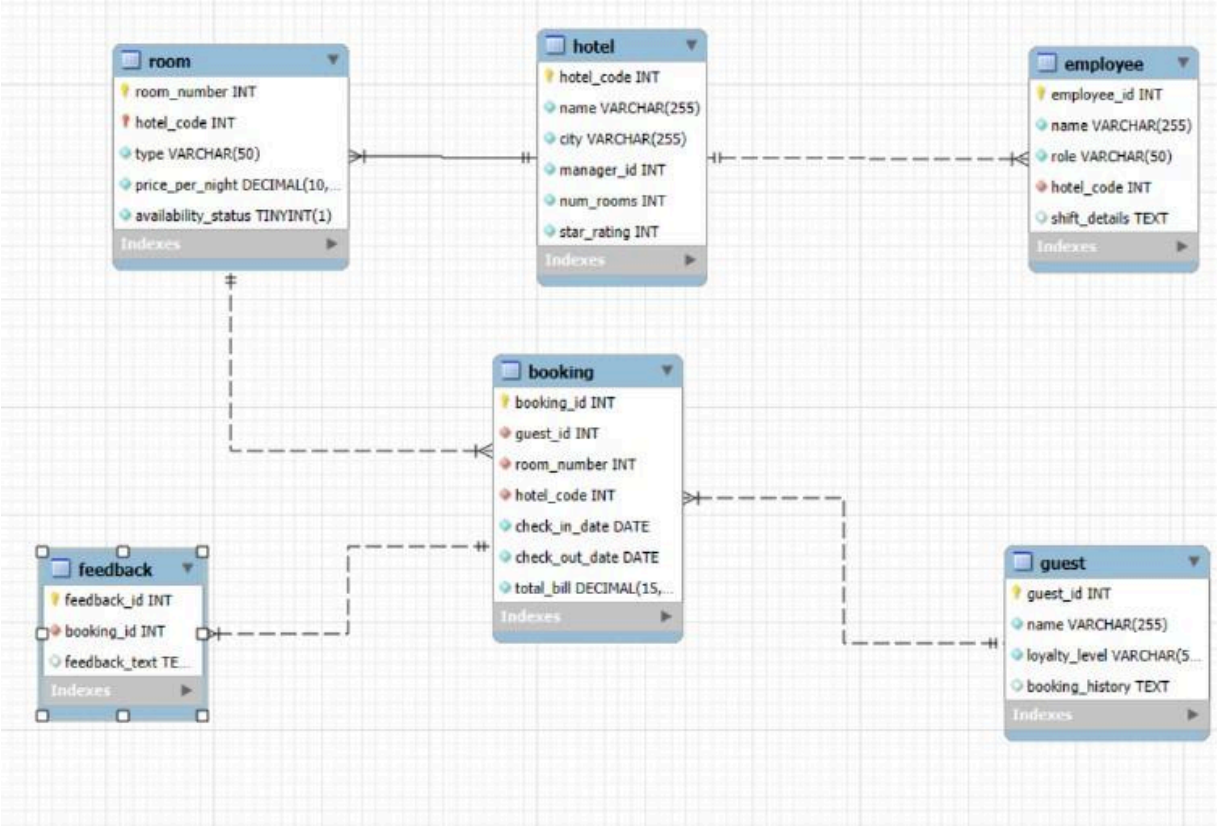
sql
CopyEdit

```
INSERT INTO Booking (guest_id, room_number, hotel_code, check_in_date,
check_out_date, total_bill) VALUES
(1, 101, 1, '2025-06-01', '2025-06-05', 800.00),
(2, 102, 1, '2025-06-03', '2025-06-07', 1200.00);
```

6. Insert Data into Feedback

sql
CopyEdit

```
INSERT INTO Feedback (booking_id, feedback_text) VALUES
(1, 'Excellent service and amenities!'),
(2, 'Very comfortable stay.');
```



Conclusion

This schema ensures efficient management of hotels, rooms, bookings, and employees. The detailed structure provides a strong foundation for scalable hotel chain operations, supporting features like room availability tracking and loyalty program management.