Mini Project:ER diagram for a hotel booking platform.

DEPARTMENT OF UNIVERSITY INSTITUTE OF COMPUTING CHANDIGARH UNIVERSITY



Subject Name/Code: DATABASE MANAGEMENT SYSTEM LAB (23CAP-252)

Subject-Database Management System

Submitted by-Guranshveer Singh Benipal (23BCA20044)

Submitted to: Mr. Jagwinder Singh

Designation: Professor

**📄 Project Introduction: Hotel Booking Management System**

The **Hotel Booking Management System** is a comprehensive platform designed to streamline the reservation and service processes of a hotel. This system enables efficient management of guests, rooms, bookings, check-ins and check-outs, room types, and additional services such as meals, laundry, and transportation. The primary goal is to provide an intuitive and centralized solution for hotel administrators to handle daily operations while enhancing the guest experience through seamless booking and service fulfillment.

At its core, the system maintains detailed records of guests and allows them to make bookings for one or more rooms, categorized by room types such as single, double, or suite. Each booking can include additional services, which are tracked and billed accordingly. The check-in and check-out functionality is tied to specific room bookings, ensuring accurate tracking of room availability and guest stays.

The data model is structured using an Entity-Relationship (ER) diagram, which identifies key entities such as **Guest**, **Room**, **Booking**, **RoomType**, **Service**, and their interrelations through entities like **BookingRoom**, **BookingService**, and **CheckInOut**. This structure ensures data integrity, scalability, and supports complex queries and reporting requirements essential for hotel operations.

Overall, the Hotel Booking Management System aims to improve operational efficiency, reduce manual errors, and offer a scalable solution adaptable to hotels of varying sizes.

### ****Core Functionalities of the Hotel Booking Management System****

#### 1. ****Guest Management****

* Register new guests with personal details (name, email, phone, address).
* View and update guest profiles.
* Search guests by name, ID, or contact information.

#### 2. ****Room Management****

* Add, update, or delete rooms with details such as room number, floor, status (available, booked, maintenance).
* Assign rooms to specific room types (e.g., single, double, suite).
* View real-time room availability.

#### 3. ****Room Type Management****

* Define different room categories (single, double, suite, etc.).
* Set and modify prices and descriptions for each room type.

#### 4. ****Booking Management****

* Create bookings for one or more rooms.
* Associate bookings with registered guests.
* Set booking dates, check-in/check-out dates, and calculate total cost.
* Update or cancel existing bookings.
* View booking history and status (confirmed, completed, cancelled).

#### 5. ****Check-In / Check-Out Management****

* Record actual check-in and check-out times per booked room.
* Update room status automatically during check-in/out.
* Generate logs for occupancy tracking.

#### 6. ****Service Management****

* Manage a catalog of additional services (meals, laundry, transport).
* Set service prices and descriptions.
* Track service availability.

#### 7. ****Service Booking and Billing****

* Allow services to be added to guest bookings.
* Record service usage per booking with quantity and date.
* Calculate and display additional charges.
* Integrate service charges with the final billing summary.

#### 8. ****Search and Reporting****

* Search functionality for guests, rooms, and bookings.
* Generate reports on room occupancy, revenue, guest history, and service usage.
* Filter data by date range, room type, or service category.

#### 9. ****Admin Dashboard****

* Overview of hotel operations: current bookings, room availability, and active guests.
* Alerts for check-ins due, check-outs pending, or services scheduled.

#### 10. ****Security and User Roles (Optional Advanced Feature)****

* Role-based access for hotel staff (admin, receptionist, service manager).
* Basic authentication and data access control.

### ER DIAGRAM :

### C:\Users\Lenovo\Downloads\23BCA20044 ER DIAGRAM.png

### Database Schema and SQL Code :

#### 1. ****Guests Table****

CREATE TABLE Guests (

GuestID INT PRIMARY KEY AUTO\_INCREMENT,

Name VARCHAR(100) NOT NULL,

Email VARCHAR(100) UNIQUE NOT NULL,

Phone VARCHAR(20),

Address TEXT

);

#### 2. ****RoomTypes Table****

CREATE TABLE RoomTypes (

RoomTypeID INT PRIMARY KEY AUTO\_INCREMENT,

TypeName VARCHAR(50) NOT NULL,

Description TEXT,

PricePerNight DECIMAL(10, 2) NOT NULL

);

#### 3. ****Rooms Table****

CREATE TABLE Rooms (

RoomID INT PRIMARY KEY AUTO\_INCREMENT,

RoomNumber VARCHAR(10) UNIQUE NOT NULL,

Floor INT,

Status ENUM('Available', 'Booked', 'Maintenance') DEFAULT 'Available',

RoomTypeID INT,

FOREIGN KEY (RoomTypeID) REFERENCES RoomTypes(RoomTypeID)

);

#### 4. ****Bookings Table****

CREATE TABLE Bookings (

BookingID INT PRIMARY KEY AUTO\_INCREMENT,

GuestID INT,

BookingDate DATE NOT NULL,

CheckInDate DATE NOT NULL,

CheckOutDate DATE NOT NULL,

TotalAmount DECIMAL(10, 2),

Status ENUM('Confirmed', 'Cancelled', 'Completed') DEFAULT 'Confirmed',

FOREIGN KEY (GuestID) REFERENCES Guests(GuestID)

);

#### 5. ****BookingRoom Table**** (for many-to-many between Bookings and Rooms)

CREATE TABLE BookingRoom (

BookingRoomID INT PRIMARY KEY AUTO\_INCREMENT,

BookingID INT,

RoomID INT,

FOREIGN KEY (BookingID) REFERENCES Bookings(BookingID),

FOREIGN KEY (RoomID) REFERENCES Rooms(RoomID)

);

#### 6. ****CheckInOut Table****

CREATE TABLE CheckInOut (

CheckInOutID INT PRIMARY KEY AUTO\_INCREMENT,

BookingRoomID INT,

ActualCheckIn DATETIME,

ActualCheckOut DATETIME,

FOREIGN KEY (BookingRoomID) REFERENCES BookingRoom(BookingRoomID)

);

#### 7. ****Services Table****

CREATE TABLE Services (

ServiceID INT PRIMARY KEY AUTO\_INCREMENT,

ServiceName VARCHAR(50) NOT NULL,

Description TEXT,

Price DECIMAL(10, 2) NOT NULL

);

#### 8. ****BookingService Table**** (many-to-many between Bookings and Services)

CREATE TABLE BookingService (

BookingServiceID INT PRIMARY KEY AUTO\_INCREMENT,

BookingID INT,

ServiceID INT,

Quantity INT DEFAULT 1,

ServiceDate DATE NOT NULL,

Amount DECIMAL(10, 2),

FOREIGN KEY (BookingID) REFERENCES Bookings(BookingID),

FOREIGN KEY (ServiceID) REFERENCES Services(ServiceID)

);

## Schema Relationships Summary

* **Guests** ⟶ can make multiple **Bookings**.
* **Bookings** ⟶ can include multiple **Rooms**
* **Rooms** ⟶ are categorized by **RoomTypes**.
* **BookingRoom** ⟶ has a one-to-one relationship with **CheckInOut** for tracking.
* **Bookings** ⟶ can include multiple **Services** .

## RELATIONSHIPS IN THE ER DIAGRAM

### 1. ****Guest → Booking****

* **Type**: One-to-Many
* **Explanation**: A guest can make multiple bookings, but each booking is made by one guest.
* **FK**: GuestID in the Bookings table references GuestID in Guests.

### 2. ****RoomType → Room****

* **Type**: One-to-Many
* **Explanation**: A room type (e.g., single, double, suite) can have many rooms. Each room belongs to one room type.
* **FK**: RoomTypeID in Rooms references RoomTypeID in RoomTypes.

### 3. ****Booking → BookingRoom → Room****

* **Type**: Many-to-Many (via BookingRoom)
* **Explanation**: A booking can include multiple rooms, and a room can be booked in multiple bookings (on different dates).
* **FKs**:
  + BookingID in BookingRoom → Bookings(BookingID)
  + RoomID in BookingRoom → Rooms(RoomID)

### 4. ****BookingRoom → CheckInOut****

* **Type**: One-to-One
* **Explanation**: Each room in a booking has one check-in/check-out record.
* **FK**: BookingRoomID in CheckInOut → BookingRoom(BookingRoomID)

### 5. ****Booking → BookingService → Service****

* **Type**: Many-to-Many (via BookingService)
* **Explanation**: A booking can include multiple additional services (e.g., meals, laundry), and each service can be used in multiple bookings.
* **FKs**:
  + BookingID in BookingService → Bookings(BookingID)
  + ServiceID in BookingService → Services(ServiceID)

### 🧭 Summary Table

| **Relationship** | **Type** | **Bridge Table** |
| --- | --- | --- |
| Guest → Booking | One-to-Many | — |
| RoomType → Room | One-to-Many | — |
| Booking ↔ Room | Many-to-Many | BookingRoom |
| BookingRoom → CheckInOut | One-to-One | — |
| Booking ↔ Service | Many-to-Many | BookingService |