**hotels.com system design**

**How to Design ER Diagrams for Hotel and Hospitality Management**

Last Updated : 27 Feb, 2024

Hotel and hospitality management enterprises provide various accommodation and leisure facilities to their guests and customers. They have a wide range of establishments including hotels, resorts, motels, inns, hostels, and other facilities.

This article will provide you with a comprehensive guide on designing**ER diagrams** for hotel and hospitality management by defining **relationships** and mapping cardinalities between its different internal modules.

**Designing ER Diagram for Hotel and Hospitality Management**

The representation of an ER diagram consists of different shapes and lines representing entities and their attributes. eg-rectangle represents entities, oval represents attributes of the entities, and a single line represents the relationship between 2 entities or the relation between an entity and its attributes.

In the ER diagram for Hotel and Hospitality management, the structure will contain details of the Customer, the different departments the information about the hotel rooms, etc. Let’s step-by-step design and create the ER diagram.

**Hotel and Hospitality Management features**

* **Understanding and making ER design** in the context of the Hotel and hospitality industry.
* **Entity Identification**: Identifying and defining the primary entities, from guests and reservations to staff roles and inventory items.
* **Attribute Definition**: Defining attributes for each entity to extract relevant data fields.
* **Relationship Mapping**: Establishing relationships between entities to find out the dependencies and interactions between one another.
* **SQL representation** of entities and their attributes.
* **Tips and Tricks** for Database Designing.

**Entities in Hotel Management**

While defining entities for Hospitality Management try thinking about how the hotel works and what are the most important components among them. First to think about are the customers who reserve the rooms, then the rooms themselves which are being reserved, then the different departments and the staff in the departments. List them all out.

1. **Hotel:** The main physical establishment providing lodging and hospitality services to guests.
2. **Rooms**: Space within the hotel premises for guest accommodation, typically equipped with furniture and amenities.
3. **Departments**: Segments within the hotel organization responsible for specific functions such as housekeeping, food and beverage, maintenance, and management. Some of the departments may be as:
   * **Finance Department**– the department handling all finances including payment salaries.
   * **Restaurant**– Department managing food and catering.
   * **Housekeeping**– Department associated with cleaning and maintaining the hotel and rooms.
   * **Front-desk**– The department deals with calls and guidance of the customer.
   * **Property management**– The Department managing the different properties and land.
4. **Guests**: Individuals or groups who rent the hotel or accommodation facility.
5. **Reservation**: The process of securing a room or accommodation for a specific period in advance.
6. **Staff**: Employees of the hotel who carry out various roles and responsibilities, including management, reception, housekeeping, and catering.

**Attributes in Hotel Management**

After determining the entities the attributes for each entity should be defined. Attributes are different characteristics such as the name of the entity, the id of the entity, etc., Also define a unique Key attribute that would later help in the **normalization** of the database and removing redundant data.

1. **Guest:**
   * **Guest\_ID**: Unique ID for each guest.
   * **Name**: Name of the guest.
   * **Contact Information**: Phone number, email, or address of the guest.
   * **Nationality**: Nationality or country of origin of the guest.
   * **Gender:**Sexuality of the guest.
   * **Reservation**\_**History**: Previous reservations made by the guest.
2. **Hotel**:
   * **Hotel\_ID**: Unique ID for each hotel.
   * **Name**: Name of the hotel.
   * **Location**: Physical location of the hotel.
   * **Number of Rooms**: Total count of rooms available in the hotel.
   * **Rating**: Overall rating or classification of the hotel.
   * **Contact Information**: Phone number or other contacts for contacting the hotel.
3. **Reservation:**
   * **Reservation\_ID**reservation: Unique reservation ID for each reservation.
   * **Check-in Date**The date: Date when the guest is scheduled to check in.
   * **Check-out Date**: Date when the guest is scheduled to check out.
4. **Department:**
   * **Department\_ID**: Unique ID for each department.
   * **D\_Head**: Identifier of the department manager.
   * **D\_Role**: Function of the department.
   * **Staff-Count**: Number of staff members assigned to the department.
   * **Contact Information**: Phone number, email, or address of the department.
5. **Staff:**
   * **Staff\_ID**: Unique ID for each staff member.
   * **Name**: Name of the staff member.
   * **Age**: Age of the employee.
   * **Contact Information**: Phone number, email, or address of the staff member.
   * **Salary**: Compensation or salary of the staff member.
6. **Room:**
   * **Room\_No.**: Unique number for each room.
   * **Category**: Type or category of the room (e.g., single, double, suite).
   * **Rent**: Price per night for the room.
   * **Status**: Current availability status of the room.

**Mapping Relationships and Cardinalities**

Defining meaningful relationships is a very important step in the**r**estaurantcreation of an ER diagram. Relationships between entities of ER diagram are the role/association of one entity to another, For eg. the relationship between hotel and rooms. Some of the relationships between the entities for the Hotel and Hospitality management sector would be as follows:

**Hotel to Room (1:N):**

* One hotel can have many rooms, but each room belongs to only one hotel.
* The relationship between the hotel and its rooms is a one-to-many relationship.

**Guest to Reservation (1:N):**

* A guest can make multiple reservations, but each reservation is made by only one guest.
* There is a one-to-many relationship between the guest and reservations.

**Reservation to Room (1:1):**

* A reservation is for one room, but each of the rooms can be reserved multiple times.
* This is a one-to-one relationship between reservation and room.

**Staff to Department (N:1):**

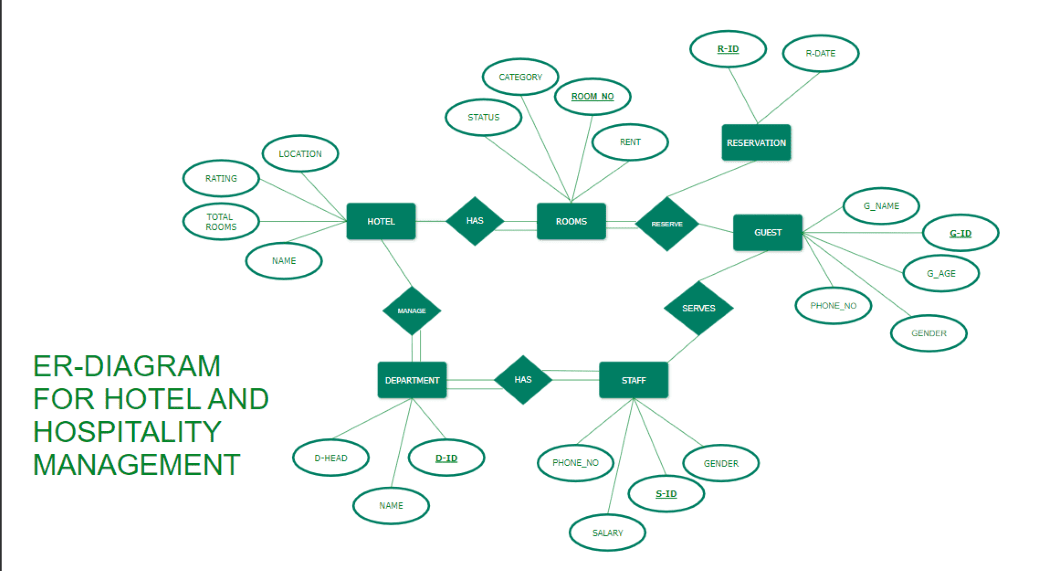
* Each staff member works in one department, but each department can have multiple staff members.
* For eg. housekeeping staff can also work with restaurant staff or vice versa
* There is a many-to-one relationship, as many staff members can belong to one department, but each staff member is associated with only one department.

**Hotel and Department (1:N):**

* Each hotel has multiple departments, such as Front Desk, Housekeeping, Food and Beverage, Maintenance, and Management.
* Each department operates within a specific hotel.
* This is a one-to-many relationship, as one hotel can have multiple departments, but each department belongs to only one hotel.

**ER Diagram for Hotel and Hospitality Management**

The following depicts the [ER diagram](https://www.geeksforgeeks.org/introduction-of-er-model/) for a Hotel and Hospitality management enterprise

Hotel and hospitality management ER diagram

The rectangles represent**,** **entities**, multiple lines represent **cardinalities**, circles represent respective **attributes** of the entity and the diamond shape represents the **relationship** between 2 or more entities.

Also, the highlighted attributes are **key attributes** which can be used to uniquely identify improved**Database Design** entity and can be used as primary key in the database schema.

**Entity Structure in SQL Format**

CREATE TABLE Guest (  
 Guest\_ID INT PRIMARY KEY,  
 Name VARCHAR(100),  
 Contact\_Info VARCHAR(255),  
 Nationality VARCHAR(50),  
 Gender VARCHAR(10),  
 Reservation\_History VARCHAR(255)  
);  
  
  
CREATE TABLE Hotel (  
 Hotel\_ID INT PRIMARY KEY,  
 Name VARCHAR(100),  
 Location VARCHAR(255),  
 Num\_Rooms INT,  
 Rating DECIMAL(3, 1),  
 Contact\_Info VARCHAR(255)  
);  
  
  
CREATE TABLE Reservation (  
 Reservation\_ID INT PRIMARY KEY,  
 Guest\_ID INT,  
 Hotel\_ID INT,  
 Check\_in\_Date DATE,  
 FOREIGN KEY (Guest\_ID) REFERENCES Guest(Guest\_ID),  
 FOREIGN KEY (Hotel\_ID) REFERENCES Hotel(Hotel\_ID)  
);  
  
  
CREATE TABLE Department (  
 Department\_ID INT PRIMARY KEY,  
 D\_Head VARCHAR(100),  
 D\_Role VARCHAR(100),  
 Staff\_Count INT,  
 Contact\_Info VARCHAR(255)  
);  
  
  
CREATE TABLE Staff (  
 Staff\_ID INT PRIMARY KEY,  
 Name VARCHAR(100),  
 Age INT,  
 Contact\_Info VARCHAR(255),  
 Salary DECIMAL(10, 2),  
 Department\_ID INT,  
 FOREIGN KEY (Department\_ID) REFERENCES Department(Department\_ID)  
);  
  
  
CREATE TABLE Room (  
 Room\_No INT PRIMARY KEY,  
 Category VARCHAR(50),  
 Rent DECIMAL(10, 2),  
 Status VARCHAR(20)  
);

**Tips and Tricks to Improve Database Design**

1. **Identify and Define Entities Clearly**: Ensure that all relevant entities in the system are identified and defined clearly. Use descriptive and concise names for the entities.
2. **Establish Proper Relationships:** Define relationships between entities accurately, considering cardinality (one-to-one, one-to-many, many-to-many) and participation constraints.
3. **Normalize Data:** Apply normalization techniques (such as First Normal Form, Second Normal Form, etc.) to reduce redundancy and improve data integrity.
4. **Avoid Complications:** Don’t overcomplicate the diagram with unnecessary entities or relationships. Keep it simple and focused on representing the core business requirements.
5. **Use Consistent Naming Conventions:** Use consistent naming conventions for entities, attributes, and relationships to maintain clarity across the diagram.
6. **Include Attribute Details:** Specify attributes for each entity with relevant data types and constraints. Ensure attributes capture all necessary information without redundancy.
7. **Consider Performance Optimization:** Design the database structure with performance in mind. This includes proper indexing, partitioning, and optimizing query performance.

**Conclusion**

In Conclusion, to design a reliable entity-relationship design for hotel and hospitality management enterprises, one must first understand the working of the internal management of the company and the relationship between each of its components. the visual representation of database design provides a clear and concise understanding of administrative tasks, enhances room allocation system, enhance customer satisfaction and improves operational efficiency.

**Use Case Diagram**

Here are the main Actors in our system:

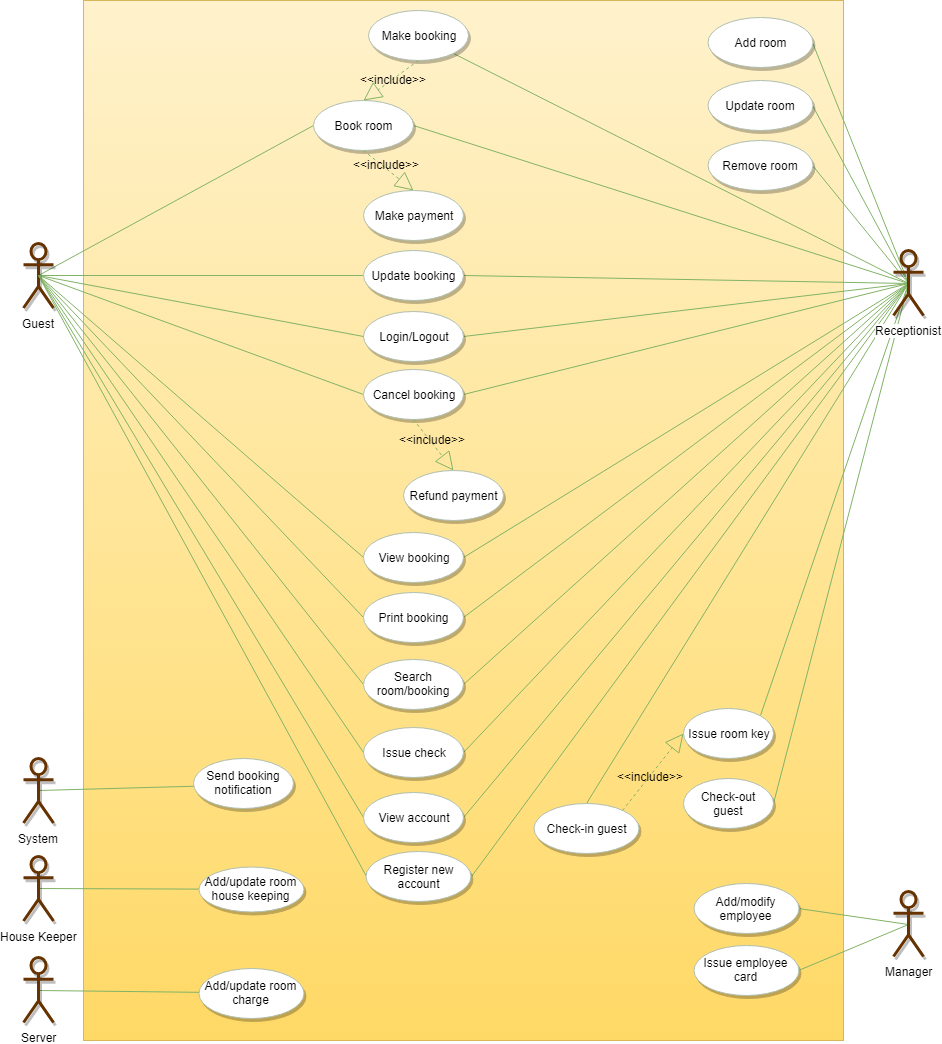
* **Guest:** All guests can search the available rooms, as well as make a booking.
* **Receptionist:** Mainly responsible for adding and modifying rooms, creating room bookings, check-in, and check-out customers.
* **System:** Mainly responsible for sending notifications for room booking, cancellation, etc.
* **Manager:** Mainly responsible for adding new workers.
* **Housekeeper:** To add/modify housekeeping record of rooms.
* **Server:** To add/modify room service record of rooms.

Here are the top use cases of the Hotel Management System:

* **Add/Remove/Edit room:** To add, remove, or modify a room in the system.
* **Search room:** To search for rooms by type and availability.
* **Register or cancel an account:** To add a new member or cancel the membership of an existing member.
* **Book room:** To book a room.
* **Check-in:** To let the guest check-in for their booking.
* **Check-out:** To track the end of the booking and the return of the room keys.
* **Add room charge:** To add a room service charge to the customer’s bill.
* **Update housekeeping log:** To add or update the housekeeping entry of a room.

Here is the use case diagram of our Hotel Management System:

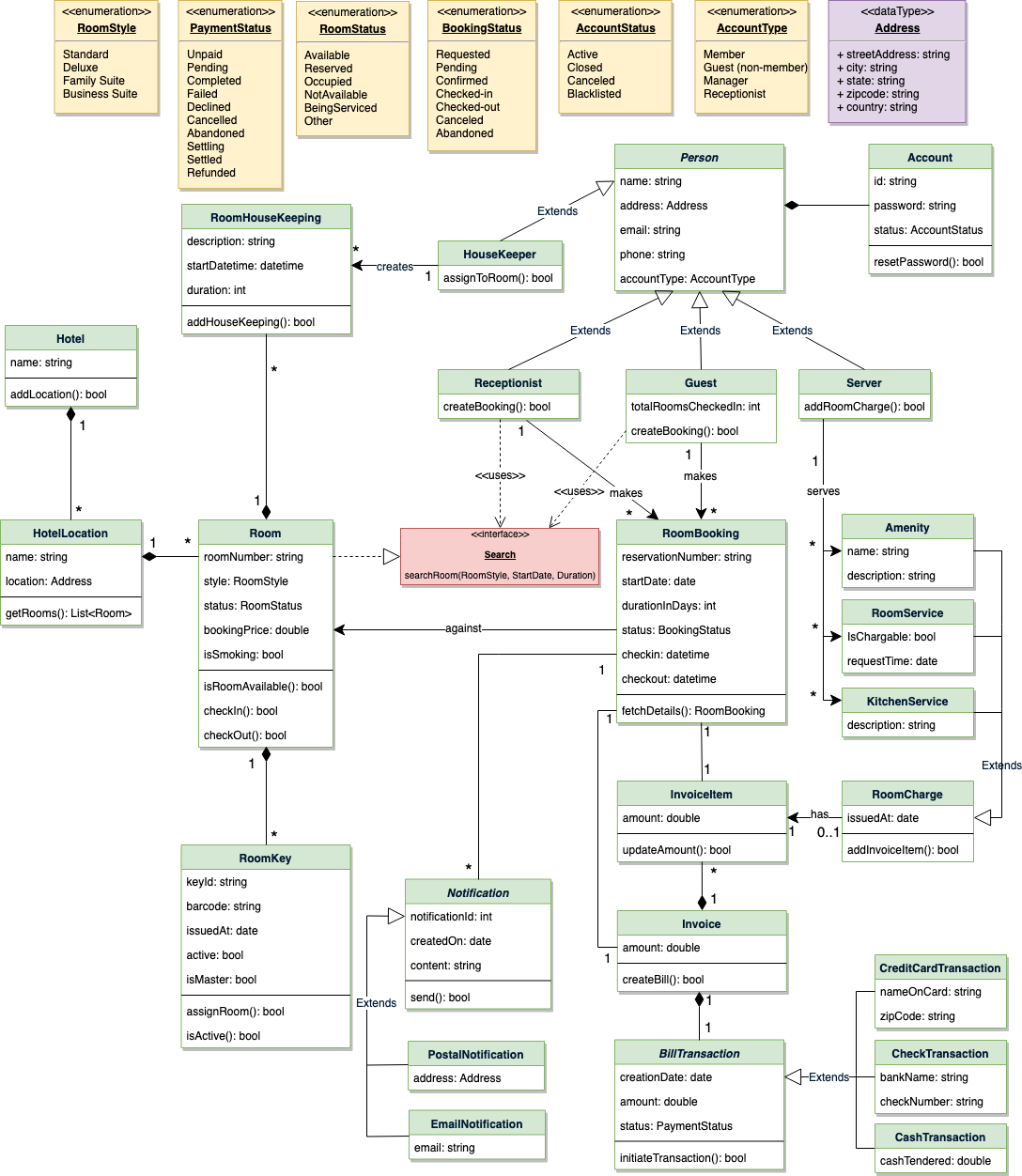
Use case Diagram: -



**Class Diagram**

Here are the main classes of our Hotel Management System:

* **Hotel and HotelLocation:** Our system will support multiple locations of a hotel.
* **Room:** The basic building block of the system. Every room will be uniquely identified by the room number. Each Room will have attributes like Room Style, Booking Price, etc.
* **Account:** We will have different types of accounts in the system: one will be a guest to search and book rooms, another will be a receptionist. Housekeeping will keep track of the housekeeping records of a room, and a Server will handle room service.
* **RoomBooking:** This class will be responsible for managing bookings for a room.
* **Notification:** Will take care of sending notifications to guests.
* **RoomHouseKeeping:** To keep track of all housekeeping records for rooms.
* **RoomCharge:** Encapsulates the details about different types of room services that guests have requested.
* **Invoice:** Contains different invoice-items for every charge against the room.
* **RoomKey:** Each room can be assigned an electronic key card. Keys will have a barcode and will be uniquely identified by a key-ID.

[](https://github.com/tssovi/grokking-the-object-oriented-design-interview/blob/master/media-files/hms-class-diagram.png)  
Class Diagram for Hotel Management System

UML for Hotel Management System