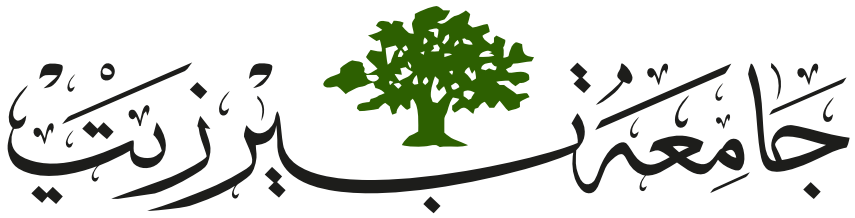

Department of Computer Science
COMP333 - Database Systems

Course Project Phase #2

Prepared by:

Mohamad Milhem 1211053
Ameer Rabie 1211226

Instructor:

Dr. Ahmed Abusnaina

Poject Idea

Hotelelo is a hotel reservation system that assists customers in finding the best-matching room based on their specifications. This project can be enhanced as a web application, mobile application, or possibly an internal system for a chain of hotels. Our main goal is to build an optimized and efficient system that can be relied upon by significant customers and can conveniently serve the users.

Contents

1 Entities

1.1 City

The city entity represents a city with all its information. The attributes of a city entities are:

1. `City_id`

   A database-generated field of type *INTEGER* that distinguishes each city. This field will be used as a **primary key (PK)** for this entity.

2. `City_name`

   A database field of type *VARCHAR* which will store the name of the city.

3. `Description`

   A database field of type *VARCHAR* which will describe the city briefly.

4. `Photo_url`

   A database field of type *VARCHAR* which will store a URL for a photo that represents this city.

1.2 Hotel

The hotel entity represents a hotel with all its information. The attributes of a hotel entities are:

1. `Hotel_id`

   A database-generated field of type *INTEGER* that distinguishes each hotel. This field will be used as a **primary key (PK)** for this entity.

2. `Hotel_name`

   A database field of type *VARCHAR* which will store the name of the hotel.

3. `Description`

   A database field of type *VARCHAR* which will describe the hotel briefly.

4. `Star_rating`

   A database field of type *INTEGER* with a value constrained to be from 1 to 5 and represents the star rating of the hotel.

5. `Longitude`

   A database field of type *REAL* which will determine the longitude coordinates of the hotel.

6. `Latitude`

   A database field of type *REAL* which will determine the latitude coordinates of the hotel.

1.3 Room

The room entity represents a room with all its information. The attributes of a room entities are:

1. `Room_id`

   A database-generated field of type *INTEGER* that distinguishes each room among all rooms in the system. This field will be used as a **primary key (PK)** for this entity.

2. `Room_number`

   A database field of type *VARCHAR* which will store the room number within a specific hotel. This number usually helps customers find their rooms (the first digit is the floor number, the second digit is the section number, etc...).

3. `Description`

   A database field of type *VARCHAR* which will describe the room briefly.

4. `Children_count`

   A database field of type *INTEGER* that represents the number of children this room can have.

5. `Adult_count`

   A database field of type *INTEGER*that represents the number of adults this room can have.

6. `Cost`

   A database field of type *REAL* that stores the cost of this room per night in US dollars.

1.4 Reservation

The reservation entity represents a reservation with all its information. The attributes of a reservation entities are:

1. `Reservation_id`

   A database-generated field of type *INTEGER* that distinguishes each reservation among all reservations in the system. This field will be used as a **primary key (PK)** for this entity.

2. `Start_date`

   A database field of type *DATE* which will store the start date of the reservation of a specified room.

3. `End_date`

   A database field of type *DATE* which will store the end date of the reservation of a specified room.

4. `Total_cost`

   A database field of type *REAL* that represents the total cost of the reservation of this room for the given time starting from the start day and ending with the end date.

5. `Payment_method`

   A database field of type *VARCHAR* with two choices "CARD" or "CASH".

1.5 Customer

The customer entity represents a customer with all its information. The attributes of a customer entities are:

1. `Cusomter_id`

   A database-generated field of type *INTEGER* that distinguishes each customer among all customers in the system. This field will be used as a **primary key (PK)** for this entity.

2. `Customer_name`

   A database field of type *VARCHAR* which will store full name of the customer.

3. `Email`

   A database field of type *VARCHAR* which will store email of the customer.

4. `Phone`

   A multivalue database field of type *VARCHAR* that represents the phone numbers of the customer.

5. `Remarks`

   A database field of type *VARCHAR* that stores the remarks about specific customer.

1.6 Photo

The photo entity represents a photo with all its information. The attributes of a photo entities are:

1. `Photo_id`

   A database-generated field of type *INTEGER* that distinguishes each photo among all photos in the system. This field will be used as a **primary key (PK)** for this entity.

2. `Photo_url`

   A database field of type *VARCHAR* which will store the URL of the photo.

1.7 Amenity

The amenity entity represents an amenity with all its information. The attributes of an amenity entities are:

1. `Amentiy_id`

   A database-generated field of type *INTEGER* that distinguishes each Amenity among all amenities in the system. This field will be used as a **primary key (PK)** for this entity.

2. `name`

   A database field of type *VARCHAR* which will store the name of the amenity.

2 Relationships

2.1 City-Hotel (one-to-many relationship)

A hotel must be located in a city, the city may have multiple hotels in it this relationship is meant to improve the customer experience by helping to find nearby hotels.
This relation will add City_id as a **Foreign Key (FK)** in the Hotel table referencing City_id in the City table.

2.2 Hotel-Room (one-to-many relationship)

Hotels in the table will have rooms by this relation with the room table. The room should be linked to a hotel in order to exist so if the hotel is deleted its associated rooms will also be deleted (note that the relationship between the hotel and room will have the option ON DELETE: CASCADE. But the room is not a weak entity since it is a crucial part of the system).
This relation will add Hotel_id as a **Foreign Key (FK)** in the Room table referencing Hotel_id in the Hotel table.

2.3 Room-Reservation (one-to-many relationship)

A reservation should have a room associated with it, the room can have multiple reservations on it, note that there a constraints on the reservations the added reservation should not have any reservation that overlaps with its time at a certain room.
This relation will add Room_id as a **Foreign Key (FK)** in the Reservation table referencing Room_id in the Room table.

2.4 Customer-Reservation (one-to-many relationship)

A reservation should be made by a customer, the customer may make multiple reservations.
This relation will add Customer_id as a **Foreign Key (FK)** in the Reservation table referencing Customer_id in the Customer table.

2.5 Room-Amenity (many-to-many relationship)

Rooms have some amenities and one amenity can be in multiple rooms, the amenities will help to improve the customer experience in terms of matching his preferences.
This will add a new table RoomAmenity to break the many-to-many relation with a composite key of both Room_id and Amentiy_id.

2.6 Hotel-Photo (one-to-many relationship)

Hotels in the table may have a photo or more which identifies it, by this relation with the photo table. (the photos will give an impression to the customer about the hotel).[1]
This relation will add Hotel_id as a **Foreign Key (FK)** in the Photo table referencing Hotel_id in the Hotel table.

2.7 Room-Photo (one-to-many relationship)

Rooms in the table may have a photo or more which identifies, it by this relation with the photo table. (the photos will give an impression to the customer about the room).
This relation will add Room_id as a **Foreign Key (FK)** in the Photo table referencing Room_id in the Room table.

[1] Note that the relationships with the photos table are all ON DELETE: CASCADE since every photo is associated with an entity to help describe it.

3 ERD

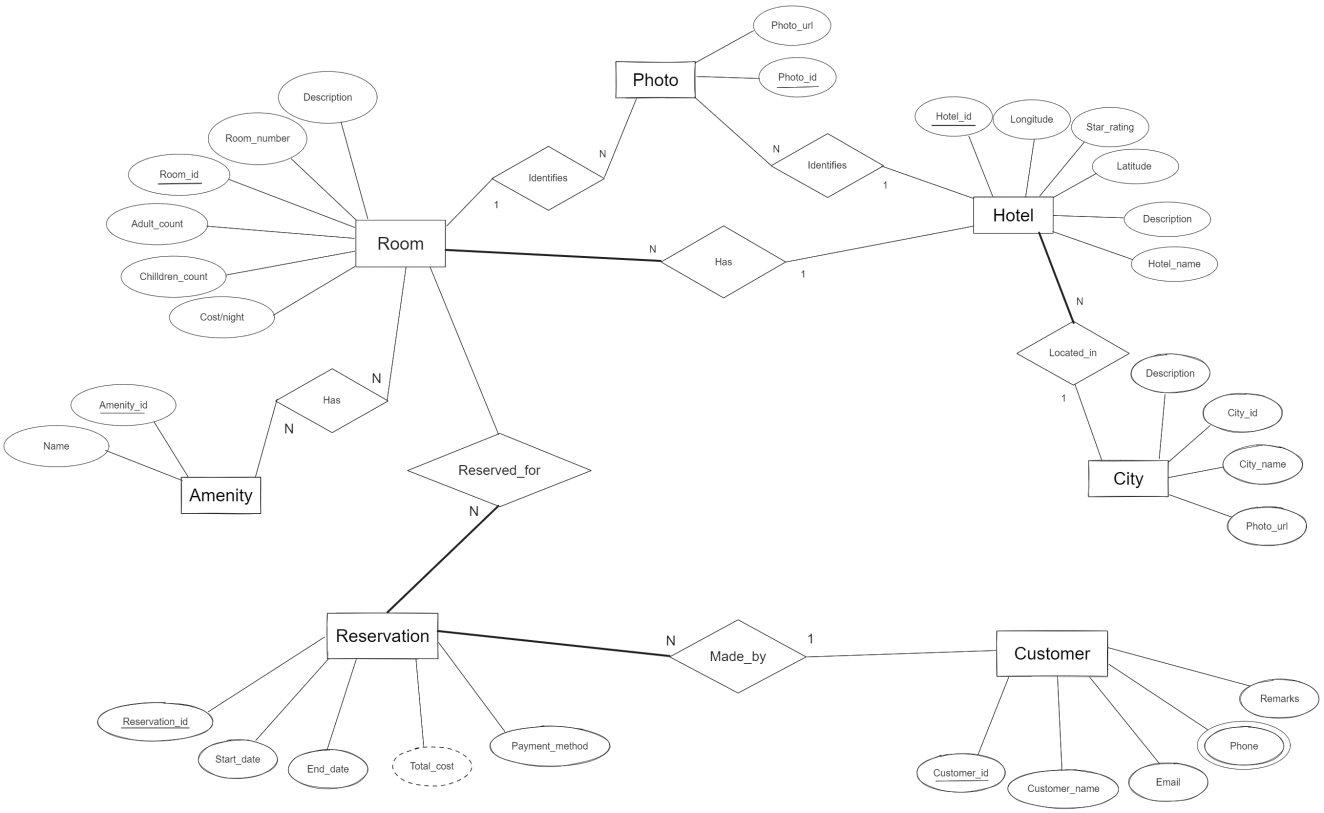

Figure 3.1: ER-Diagram

List of Figures