

Introduction to Data Management – HW1: Database Design Assignment

Team Number: 20

Team Members:

Abhinav Sharma (ass2575), Allie Touchstone (awt529), Aritra Chowdhury (ac79277),
Avery Shepherd (ams9694), Harsh Mehta (hdm564), Vishal Gupta (vg22846)

Problem Statement

The Sour Apple Hotel, a South Austin boutique hotel, wants to build a single centralized hotel reservation system that will manage all its diverse locations, customers, and reservations. Before the actual system is built, an Entity Relationship Diagram needs to be created.

Proposed Solution

The solution makes use of the primary entities – Customers, Credit Cards (named *CreditCards* in the system), Reservations, Locations, Features, Rooms, and Room Types (named *RoomTypes* in the system). There are also two junction tables to remove many-to-many relationships, *ReservRoom* (between Reservations and Rooms) and *LocFeat* (between Locations and Features).

The *Customers* entity holds all the information about customers, including the breakup of their names and address. This entity is linked to the *CreditCards* entity using *CustomerID* as the join field. The *CreditCards* entity securely contains all the information about credit cards the customers use. The *Customers* entity is also linked to the *Reservations* entity using the *CustomerID* as the join field. The *Reservations* entity contains information about each individual reservation made at any location from any customer and includes provisions for the Customer Rating and any additional notes as well. This *Reservations* entity is then linked to the *Rooms* entity using a linking table named *ReservRoom*, which makes use of the *ReservationID* field in the *Reservations* entity and the *RoomID* field in the *Rooms* entity as the join fields. This is because multiple reservations can be made for the same room and the same reservation can be made for multiple rooms, creating a non-optimal many-to-many relationship. This issue is dealt with by the linking table. The *Rooms* entity contains details about each room in every location and is also joined to the *Locations* entity (more about this later) through the *LocationID* field and the *RoomTypes* entity through the *RoomTypeInit* field. The *RoomTypes* entity contains information on all the types of rooms available in the hotels. Both *Rooms* and *Reservations* are connected to the *Locations* entity using the *LocationID* field. The *Locations* entity informs on all current locations the hotel chain is situated in and has details of the corresponding hotels. This entity is then connected to the *Features* entity using the linking table, *LocFeat*, due to the same reason as in the case of *ReservRoom*.

Assumptions

- We use *CustomerID* as the Primary and Foreign Key in the *CreditCards* entity to keep the system as secure as possible, as the field is already unique.
- Although *ConfirmationNumber* is unique and can be used as the Primary Key in *Reservations*, we use a separate *ReservationID* as the Primary Key to prevent any mishaps in data warehousing from causing a system failure.
- We create a separate entity for *RoomTypes* in case of changes made to Room Initials and corresponding descriptions later.

Conclusion and Next Steps

To conclude, the system described above and through the ERD should perform well and provide a scalable, efficient, and reliable system for all data warehousing and reporting purposes. The next step is to build a database using the database architecture.