MIS 381N
HW1 – Database Design Assignment

Team Members

Name	EID
Bhavana Reddy	bc35833
Safiuddin Mohammed	sm76833
Brandt Green	bwg537
Ramya Madhuri Desineedi	rd32895
Suchit Das	sd38448
Lucas Fernandez	LF23234

Objective:

Design a Database for Sour Apple Hotel

Background:

The Sour Apple Hotel, a South Austin boutique hotel is expanding its operations into a diverse set of properties. As a part of their expansion, they are looking forward to building a centralized reservation system to manage the ever-growing information that they are acquiring. The system should be able to handle all their key data points and support internal operations, so that it can be integrated into their website to support customers in setting up account, view room features and make reservations

Database Tables:

Customers – This is one of the most important entities in our database design which contains all the customers features. The attributes of this table are CustomerId, the primary key, and features such as names (First, Middle and Last), E-mail, Phone number, Date of Birth, current city, state, and zip code. For security purposes we are including payment information in another table.

CreditCards – All the payment related information such as the Credit Card number, card type, and other confidential details are stored in Payments entity. CardId is the primary key. Payments also contains additional information such as name (First, Middle and Last) and Billing address. It has a one-to-one relationship with Customers entity and contains CustomerId as the foreign key. This entity contains similar fields to Customers table to account for the fact that some customers may have different billing information.

Reservations - We log all the real time reservation information in this entity. Here ReservationId acts as the unique identifier along with all the supplemental information such as the reservation status, customer rating, CustomerId, Check-in, Checkout information, and confirmation details. Customers and DiscountCodes entities have one-to-many relationships with Reservations where CustomerId and DiscountCode act as foreign keys.

ReservationRooms – This entity bridges the many to many relationship between the Reservations table and the Rooms table. Here ReservationRoomId acts as the unique identifier with ReservationId and RoomId acting as foreign keys for sourcing respective information. Additionally, a field NumberOfGuests is included to capture expected guests.

CustomerCredits – This entity tracks all the credits earned and utilized for each of the customer with the customerId acting as the unique identifier. The remaining credits are determined by taking the difference of credits earned and credits utilized.

DiscountCodes – We log all our discount codes and associated amounts into this table which eventually feeds into other entities.

Rooms – This entity acts as the master table that contains information for all the rooms owned by Sour Apple Hotel containing RoomId as the unique identifier and information regarding room sizes, room types, capacity, and the location id of the room. We are also storing weekend and weekday rates for each of these rooms inside this entity.

Locations – We are planning to store all the attributes of entire locations onto this master table tracking information such as its name, address, phone, capacity, and URL. Here we use LocationId as the primary key.

Features – Features across all locations are entered into this central entity which acts as a source of information for Location entity. We associate FeatureId as a key to identify unique elements.

LocationFeatures – This entity acts as a bridging table between Features and Locations to avoid many-to-many mapping and has a mapping of location Ids with different feature Ids.

Conclusion:

We have separate entities to ensure that the system is scalable to account for any future expansion plans that the client might have. This database acts as a forward-looking data base on which information can be updated and the system can act as a central repository to support SSOT-MVOT architecture and aid in making better strategic decisions.

Entity Relationship Diagram