

Hotel Reservation System - Executive Summary

Group 7: Archit Patel (ajp4737), Charan Musunuru(cm59982), Divyansh Karki(dk27856), Sahil Arora (sa52643), Sukeerth Cheruvu (sc63232), Yashaswini Kalva(yk8348)

Assumptions

1. Coupons can be applied at Reservation level and are applicable for any location
2. Only one coupon can be applied per reservation and the coupon codes are non-generic i.e., specific to the customer

Entity Description

Reservations: This is the central table in the database that stores all the details corresponding to a reservation. It has *ReservationID* as its primary key, and *CustomerId* as a foreign key that can be used to map customer related information for each reservation. This entity also stores information about the points earned with each reservation and discount applied, if any.

Customers: Customer details are added to this table for the very first time a customer makes reservation with the hotel and a unique *CustomerId* is created in the system which acts as primary key. It primarily stores customer contact information including the birthdate of the customer which can be used by the admin team to send discount coupons. It also contains the total points available with the customer at a particular point of time.

CreditCardDetails: Since each patron can have one and only one Credit and, *CreditCardNumber* will act as a primary key for this table. It consists of credit card owner's details which are necessary to process the payments. Although the customer table has address dimensions, this table also has address dimensions since the mailing address could be different from the credit card billing address,

CustomerDiscounts: CustomerDiscounts is a repository of all the valid discount coupons applied or applicable to the customers while making reservations. DiscountCode acts as a primary key to this entity and CustomerId is a foreign key that can be used to map customer related information for each reservation. It also stores details such as CouponExpirationDate, DiscountAmount and Status of the coupon

RoomReservations: Reservations and Rooms entities share a many-to-many relationship and hence a bridge entity RoomReservations is created to avoid many-to-many relationships. ReservationId from Reservations entity and RoomsId from Rooms entity serve as a composite key for this entity

Rooms: The Rooms entity contains a record for each of the rooms at all locations. Location of room is tracked here using the foreign key to Locations table. Other details of the room such as capacity, size, weekend rate, weekday rate, and type are also tracked here.

Locations: This entity stores all the location details where the hotel has a branch, such as address, phone number, URL, rooms etc. Each location can have multiple rooms and hence the relationship between rooms and location is M:1.

LocationFeatures: Since Location and Features entities have a many-to-many relationship between them, a bridge entity called LocationFeatures was introduced to eliminate the illicit many-to-many relationship. The primary keys from Location and Features serve as the composite key for this table

Features: Features is an entity that has information about features of a location and each feature has a corresponding feature ID which forms the primary key of the table. It has a one-to-many relationship with the bridge entity LocationFeatures

Conclusion

The database model provides The Sour Apple Hotel with much needed data robustness as well as flexibility to take faster decisions. As the brand expands to more locations and fulfills more patrons, the database model will optimize data storage by reducing redundancy and help faster processing. This improves the turnaround time for decision making, gives deep insights into customer behavior and acts as the Single Source of Truth for the brand.

Entity Relationship Diagram (ERD)

