ECM2419

047635&060669

Split 50/50

Data requirements

Hotels

The requirement for Hotels is that there must be an ability to book any hotel. The individual hotels will have individual names, so the requirement will also include the name of the hotel as well as the ID. It will likely need the address of the hotel including the post code. The phone number for the individual hotels. The rating for each which will be a rating out of 10. The facilities of the hotel for instance whether the hotel has a gym or can include breakfast. The hotel prices will also be required, these include all the rooms and the grace period for the discount.

Room

Rooms will need to be identified by their name which is likely a number, this will be the same as their ID if it's a number. The hotel that the room belongs to via the Hotel ID, as well as the room type.

Room Types

There will be different types of rooms; Single, Double and Family rooms. These will also require separate discounts and their individual prices.

Reserved Rooms

All reserved rooms need to be identified in queries and so therefore they will need to be identified. Information about each room will be needed like the number of adults and children, whether breakfast is included or not and the check in and check out dates.

Reservations

Each reservation will need a reference or ID. The dates of the reservation arrival and the leave dates will also be required. It is required to know whether the guest wants breakfast and how many breakfasts they want. The date and time the reservation is made is required for any pricing issues. The price that they booked it for may also be needed. The payment options of prepaid or at the checkout and cash or card. The customer ID will be needed as well to link to the customer.

Cancellation

The original reservation ID will be needed and the current date of the cancelation. It will be helpful for the hotel to have a reason of cancellation.

Customers

When it comes to the customers we will require their names, the credit card information and their address. We also require a phone number and an email to contact them.

Transaction requirements

Data Queries	
1	Identify the hotel within a city
2	Identify rooms that are available
3	Identify hotels higher or lower than a certain rating
4	Identify the booker of any of the rooms
5	Identify the hotels with free parking
6	Find customer booking information
7	Filter and list rooms by a certain price
8	List the number of rooms
9	List the number of guests
10	Show the availability of a room
11	Identify the number of breakfasts on a certain day

Data Manipulation	
1	Update a public Rating of a specific hotel
2	Update a booking changing either the dates, the number of rooms and the payment method
3	Create a cancellation and updating the bookings table to remove the cancelled booking
4	Update the discount that is given based on the day of booking
5	Delete a reservation (Cancelation)

Data Insertion	
1	Insert a new customer
2	Insert reservations between two dates as refurbishment is taking place

Assumptions

- A customer can make a reservation to any hotel and more than just one reservation.
- A customer can reserve more than one room in the same hotel.
- The Discount is applied based on the booking date and the arrival date.
- There is at least one hotel on the database else it wouldn't exist.
- Any guest that hasn't prepaid pays when they check out of the hotel.
- Single room has 1 person in, Double room as 2 people in and Family room has 4 people in

Design of Derived Data

The reason we Derive the data in the queries is because it is advantageous as it means updating the table will require less changes. To calculate the number of nights stayed it will derive the output from the entered dates. The available rooms will be derived in the queries, from the reservation list. The average room price will be calculated in the queries to prevent having to update the table whenever a new hotel is added, or one hotel has changed their price. Counting of the rooms will be done in the queries as well as the number of adult guests and breakfasts. The prices of the rooms discounts will be derived in the queries as well as the final price.

Logical Model Design

RoomType(roomType,hotelID,roomPrice)

Composite Primary key: hotelID, roomType

Foreign Key :hotelID references Hotels(hotelID)

Room(roomID,roomType,hotelID)

Composite Primary Key: roomID, hotelID

Foreign Key : hotelID references Hotels(hotelID)

ReservedRoom(roomId,bookingReference,numberOfChildren,numberOfAdults,breakfast)

Composite Primary Key: roomID, bookingReference

Foreign Key: (hotelID,roomID) references Room(hotelID,roomID)

Foreign Key: bookingReference references Reservation(bookingReference)

Hotels(hotelID,hotelName,hotelAddress,hotelPostcode,hotelFacilities,phoneNumber,publicRating,breakfastPrice,singlePrice,doublePrice)

Primary Key: hotelID

Reservation(bookingReference,customerID,hotelID,bookingDate,arrivalDate,leavingDate,totalAmount,paymentOption,cardOrCash,specialInstructions)

Primary Key: bookingReference

Foreign Key: hotelID references Hotels(hotelID)

Foreign Key: customerID references Customer(CustomerID)

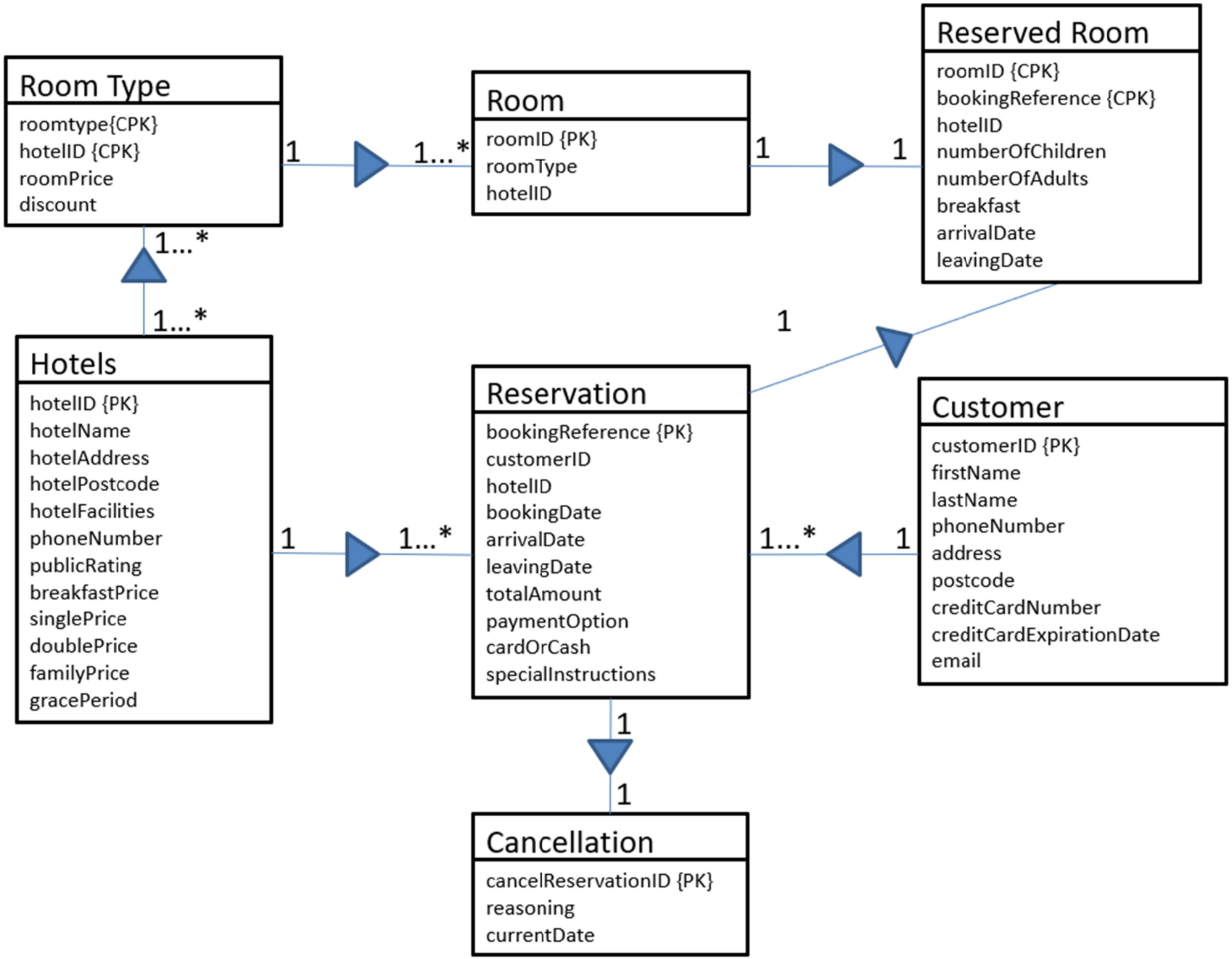
Customer(customerID,firstName,lastName,phoneNumber,address,postcode,creditCardNumber,creditCardExpirationDate,email)

Primary Key:customerID

Cancelation(cancelReservationID,reasoning,currentDate,bookingReference)

Primary Key: cancelReservationID

Foreign Key: cancelReservationID references Reservation(bookingReference)


Room Type
roomtype{CPK}
hotelID {CPK}
roomPrice
discount
Room
roomID {PK}
roomType
hotelID
Reserved Room
roomID {CPK}
bookingReference {CPK}
hotelID
numberOfChildren
numberOfAdults
breakfast
arrivalDate
leavingDate
Hotels
hotelID {PK}
hotelName
hotelAddress
hotelPostcode
hotelFacilities
phoneNumber
publicRating
breakfastPrice
singlePrice
doublePrice
familyPrice
gracePeriod
Reservation
bookingReference {PK}
customerID
hotelID
bookingDate
arrivalDate
leavingDate
totalAmount
paymentOption
cardOrCash
specialInstructions
Customer
customerID {PK}
firstName
lastName
phoneNumber
address
postcode
creditCardNumber
creditCardExpirationDate
email
Cancellation
cancelReservationID {PK}
reasoning
currentDate
1
1...*

Entity Choice Reasoning

Customer

The customer entity is needed to store all the information about them. The Primary key is the customerID to prevent two customers with the same name getting mixed up. Another option that we chose not to use is the credit card number as the primary key, we chose not to use this because people change their credit cards over time and you shouldn't have a non-static primary key.

Reservation

I determined that 'Reservation' is needed as an entity to keep track of all reservations that customers make, the primary key is the booking reference ID. The booking reference is used as an ID so that if someone makes more than one reservation it won't clash like it would if we used their name as a primary key. There is a one to one relationship with Hotels because per reservation there can only be one hotel. There is also a one to many relationships with the Room entity, this is because each reservation can have multiple rooms.

Room

Room is a strong entity; the primary key is the composite of roomID and hotelId since there is nothing about the individual room's ID that is unique to distinguish them from other hotels. It has a one to one relationship with Reserved Room, this is because per reserved room there is only one room type.

Hotels

Hotels is a Strong entity of all the different hotels, the primary key for each is hotelID. There is a possibility for a hotel to change name or even have the same name that is why we decided to create an ID instead of using the name as the primary key.

Cancelation

Cancel Reservation ID is the primary key for this strong entity. It has a one to one relationship with reservation, this is because for each reservation there can only be one cancelation.

Room Type

We made this entity with the primary key as a composite of room type and Hotel ID. There is a 1 to one to many relationships with Room, this is because each Room can have just one type. It also has a relationship with Hotels of *one to many* to *one to many* this is because more than one hotel can have more than one room type

Reserved Room

This entity is needed in case there are multiple rooms on one reservation therefore the primary key is a composite of roomID and bookingReference. It is related to room on a one to one relationship because for each reserved room there is only one room.

Attribute Reasoning

The attributes for Room Type are roomType, hotelID and roomPrice. To identify the room price, we need the other two attributes, for different room types there are different prices and so we have the attribute roomType, the prices per roomType vary based on the hotel therefore we need the hotelID to find the price. Discount is another attribute that is the percentage discount off the price that will be applied if the booking is made far enough before the check in date.

The attributes for Room are needed to identify the specific room therefore the roomID gives it an identification, the hotelID is required to make sure a room with same ID from another hotel isn't confused and the roomType is needed to identify whether the room is a single double or family.

Reserved Room needs the bookingReference to identify what reservation the room is part of, the roomID is required to know what room is booked. The numberOfChildren and numberOfAdults are needed so we know how many are in each room. The hotelID is needed so that we know what hotel the room is reserved in and breakfast is needed so we know whether the adults and children want breakfast included. The arrivalDate and leavingDate is needed in order to work out when the room is reserved and when it's not.

Hotels have a hotelID to distinguish between the different hotels, the hotelName is needed to display when the customer searches, the hotelAddress and hotelPostcode are needed to geographically locate the hotel. hotelFacilities is needed for the hotel to list its facilities, this is written by the hotel as opposed to chosen from a limited selection of facilities. The phoneNumber is needed for the customer to contact the hotel. The publicRating is used for filtering hotels apart from good to bad. The breakfastPrice, singlePrice, doublePrice and familyPrice are all needed in the queries for working out the pricing for each reservation. The gracePeriod is the time before the booking is made that a discount is applied.

Reservation has bookingReference as its primary key, this is unique for each reservation, so they can all be identified from this number. The customerID and hotelID need to be on the Reservation to identify the guest and which hotel the reservation is for. The bookingDate is required to work out the discount that needs to be applied to the price. The arrivalDate and leavingDate are needed for the hotel staff to know when the guests will check in and check out. The totalAmount is the final cost of the reservation. For paymentOption and cardOrCash both are needed and that is because the cardOrCash isn't specified in the paymentOption, the paymentOption is whether they prepaid or pay when they check out.

Lastly specialInstructions is needed to specify any requirements or arrival times that the customer may want to add.

Customer has a customerID in for an event in which two people with the same name make a reservation it's possible to tell them apart. the firstName and lastName are needed for invoice information and for the staff to call the customer by. The phoneNumber, address, and email are all ways the hotel can contact the customer about any issues, the address and email are needed for booking confirmations and billing information. The creditCardNumber and creditCardExpirationDate is needed to charge the customer, take note this is needed even if the customer wants to pay on checkout to prevent people not paying.

Cancellation needs a cancelReservationID as a way of referring to a cancelation made. The reasoning is helpful as feedback to the hotel for instance if reasoning is that the hotel is too expensive and there are cheaper hotels in the area they may want to consider changing the price. The currentDate is required to make sure no late cancelation fines are needed.