Hotel Management System

For Wolf Inns, a popular Hotel Chain

CSC 540 Database Systems

Project Report #1

Srilatha Bekkem

1) Assumptions and Problem Statement

Assumptions

1. Customer id and bill id are generated by the database which identifies each customer in the customer entity and bill in the bill entity respectively.
2. Even if the same customer comes again to any of the hotel in the chain, still a new id is generated.
3. Room prices also vary on the hotel location, room type and occupancy.
4. If a hotel has presidential suite, then there is one catering and one service staff assigned to that room at the time of check in and removed at the time of check out.
5. The Manager, front desk representative, catering and service staff are the only staff a hotel has.
6. The nightly rate of each room is calculated by the adding all the cost factors of category of room, occupancy and location.
7. Managing Director is responsible to modify, enter the details of the cost factors of location, room, and occupancy and on the services.
8. Same room numbers can exist in different hotels, therefore the combination of hotel ID and the room ID will be the criteria to identify a room uniquely.
9. All the cities in different states have different name i.e. no two cities have same name.
10. Manager is a staff and we get the details of the manager of the hotel from staff table.
11. The basic details of the customer and the check in details are taken at the same time.
12. The front desk representative assigns the room requested by the customer (on behalf of) and that room is not made available until the customer checks out.
13. There are three modes of payment: credit i.e. hotel credit card/ card/ cash.
14. The hotel credit card is accepted if the card number and name of the customer match with the card number and name stored in database.
15. There is only zero or one manager for a hotel in the staff column
16. Deletion of room is possible only when it is available, deletion of customer is possible only after the customer checks out. Deletion of staff is allowed only when staff is not assigned to presidential room.
17. If customer uses a service then charges for it are applicable for the entire stay.

Problem Statement

The hotel management system database will be used for the *Wolf Inn, a popular hotel chain*. The database system is going to be used by the Manager of the hotel, the front desk representative and service staff in each hotel and the Managing director of the complete hotel chain. The database will maintain

information about the hotels, hotel staff, customers, rooms, services and billing. Each of the respective users of the database will have specific tasks and operations that they can perform.

When all the information of the customers and the rooms they booked, services used in the hotel and the finances is managed by the database, there are significant advantages.

- Concurrency Control can be achieved for example, optimised concurrency control can be implemented using the multiple granularity locking method present in database management systems and relational databases. For example : Updating billing costs for a customer by multiple hotel staffs would not generate inconsistencies in the system.

- In the database relations some fields will be made mandatory , so that while the hotel staffs are filling up information , be it details of the customers or providing details to generate bills , then they cannot leave it incomplete which will end up generating unexpected problems in the systems. Each granular operation either has to completed fully or has to fail altogether. There should not be anything called partially complete task.

- Using structured queries, we could find complex information from a database in an optimised manner. For example, it would be easy to find details about any customer who had once checked in to any of the hotels in the chain using simple, customer ID, hotel ID and location.

Using databases was the best approach to design all the functionalities for the given hotel chain because, had the information been stored in normal files, although we could have at least retrieved the files based on file names in normal programming, but finding out useful information by combining different files and parsing through them would not have given results in optimal time and also ,the file approach would mostly be sequential. But using relations and hence databases, such tasks can be performed in no time.

2) User Classes

We believe that the four tasks listed in the Project Narrative reflect perfectly the four classes of users that our system will need to support.

Managing Director is the user that controls the entire hotel chain and the one who analyzes reports on occupancy based on various aspects such as hotel, room type, date range and city.

Managers are the people who take care of the hotel they serve i.e. each hotel in the chain has one manager who maintains that particular hotel by analyzing reports on occupancy based on aspects room type, date range.

Front desk representatives are the ones who register customers, processes check in and check outs, assign available rooms based on customer request, bill customers, release rooms.

Service staff are the people who help the customers in using the services. They check if a service requested by the customer is offered or not. If not offered, then they enter the details into the database.

3) The Main Entities

The six entities that are the most relevant to the tasks and operations required by our clients are the following.

Hotel – hotel_id,name, phone_no,hotel_address;

location- location_id,city,state,cost_factor.

Hotel Staff – staff_id, name, age, phone_no, email_id. This entity is subdivided based on the title such as Manager, catering staff, service staff.

Room – It is a weak entity set which is connected to the hotel via supporting relationship. It consists of room_number, nightly_rate, availability. It is subdivided based on category i.e. presidential. The room type and the occupancy is obtained from **Occupancy** and **category** tables.

Customer – customer_id,name, DOB, phone_no,email_id,card;

Billing – bill_no, total amount, final amount;

Services – service_id, type, charges,time;

Card-- Cardno,name, validity, limit, CVV, balance.

4) A Usage Situation

Mr. Sam, who lives in California, has come to New York to visit the place. For 3 days stay, he comes to the Wolf Inns hotel with his family (total of 4 members), he approaches the front desk representative, Ms. Kathy and asks for a deluxe room for 4 members and 3 nights stay. Ms. Kathy, checks for the requested room by entering the type and number of guests, the system showed there are two deluxe rooms for max 4 occupants along with the nightly rate as 300$, Ms. Kathy explains the details of the room and the services offered with this room type.

Mr. Sam, agreed to stay in the hotel for 3 nights, Ms. Kathy enters the details of the customer, details of the room, check in time and also hotel credit card details(if any). Keys are handed over to Mr.Sam. Gym and phone are the services that were offered to Deluxe room. On the second day of their stay, Jayson, son of Mr. Sam goes to the pool. As this service is not offered, this is recorded in the system.

At the time of check out, Ms. Kathy gives them itemized receipt which totaled to be 920$ that was generated by the system.
If he had a hotel credit card, then he would have obtained a discount of 5% on this $920 i.e, amount payable would have been $874 else, Mr Sam would have to pay $920 only.
Mr. Sam enjoyed the stay in the hotel, pays the bill and then leaves the place.

5) Application Program Interfaces

Information Processing

newLocation(city, state, cost factor)
returns confirmation.

updateLocationCost(city, state, costfactor)
return confirmation

newHotel(hotelID, hotelName, hotelPhoneNumber, hotelAddress,city,state)
return confirmation

updateHotelManager(hotelID, managerID)
return confirmation

deleteHotel(hotelID)
return confirmation

newStaff(staffID,sname, dateOfBirth, phone, email, title)
return staffID

updateStaffTitle(staffID, newtitle)
return confirmation

deleteStaff(staffID)
return confirmation

newCustomer(name, dateOfBirth, phoneNumber, emailID, cardnumber, startdate, enddate, checkin, checkout)
return confirmation

updateCustomer(customerID, name, dateOfBirth, phone, emailID, cardNumber()
return confirmation

deleteCustomer(customerID)
return confirmation

newRoom(roomNumber, hotelID, occupancy, type)
return confirmation

updateRoom(roomID, hotelID, occupancy)
return confirmation

deleteRoom(roomNumber, hotelID)
return confirmation

roomAvailable(hotelID)
- return rooms available
- if no rooms available return null

requestedRoomAvailble(hotelID, type, occupancy)
return confirmation

releaseRoom(roomID, hotelID)
return confirmation

Maintaining Service Records:

newServices(service id,name,charges,time)
returns confirmation.

updateServiceCharges(Service Id,Charges)
returns confirmation.

roomServiceOffered(Type,Service Id)
returns confirmation.

enterServiceUsed(cid,Ser_id,number_of_times,date)
returns confirmation.

Maintaining Billing Accounts:

generateBill(cid)
returns bill id.

printBill(cid,bid)
outputs cost of room for number of days of stay, cost of services used within the stay, discount if hotel credit card available, total bill to be paid.

Reports

OccupancyByHotel()
Returns the percentage of rooms occupied grouped by hotel_id;

OccupancyByRoomType()
Returns the percentage of rooms occupied grouped by room_type;

OccupancyByDateRange()
Returns the percentage of rooms occupied grouped by startdate and enddate;

OccupancyByCity()
Returns the percentage of rooms occupied grouped by city;

reportRoomsOccupied()
returns total occupancy and percentage of rooms occupied.

staffRole()
returns the list of staff grouped by their role.

StaffAtCustomerService(CustomerID)
Returns a list of staff which served the customer

revenue(hotelID, startDate, endDate)
returns the revenue calculated.

6) Data Views

Managing director:

Managing director is the one who has the most number of privileges hence he will have access to all the data as compared to others. He is similar to what we generally say an admin. The specific data the managing director will view is:

- The hotel table with the name , ID, phone number , address and location of the hotel.
- The table of staff including ID, name. date of birth, phone number and title.
- The table of customer including ID no, name, date of birth, email address.
- The table of room which will include room no, availability and rate.
- The table of services which will include the name of the service, the charges associated with it and the time which is the time for which the person used.
- The category of the room which will be a table will also be available to the managing director.
- The managing director will have all the data of from the reports for the overall revenue.
- The table of cards with the following details of card number, validity, name, CVV, limit and balance.

Manager

This manager is specific to the hotel in particular location. Hence, all the views related to the hotel in one location only will be relevant to him.

- The hotel table with the name , ID, phone number , address
- The table of staff including ID, name. date of birth, phone number and title. The table will also include the details of customer which will be available to the manager. The details of customer are ID no, name, date of birth, email address.
- The data view of the room to the manager. He will be able to view all the details which will include room no, availability and rate.
- The table of services which will include the name of the service, the charges associated with it and the time which is the time for which the person used.
- The category of the room which will be a table will also be available to the manager.

Front desk representative

Front desk people should be able to :

- view all the data relevant to the customer, room and services.
- responsible for registering room for the customer ,check room availability and bill customers during check out.
- The tables of hotel, customer, rooms, billing, services and category with all the data of each table will be available to the front desk representatives.
- They will be able to view the card number of the customer from the card number.

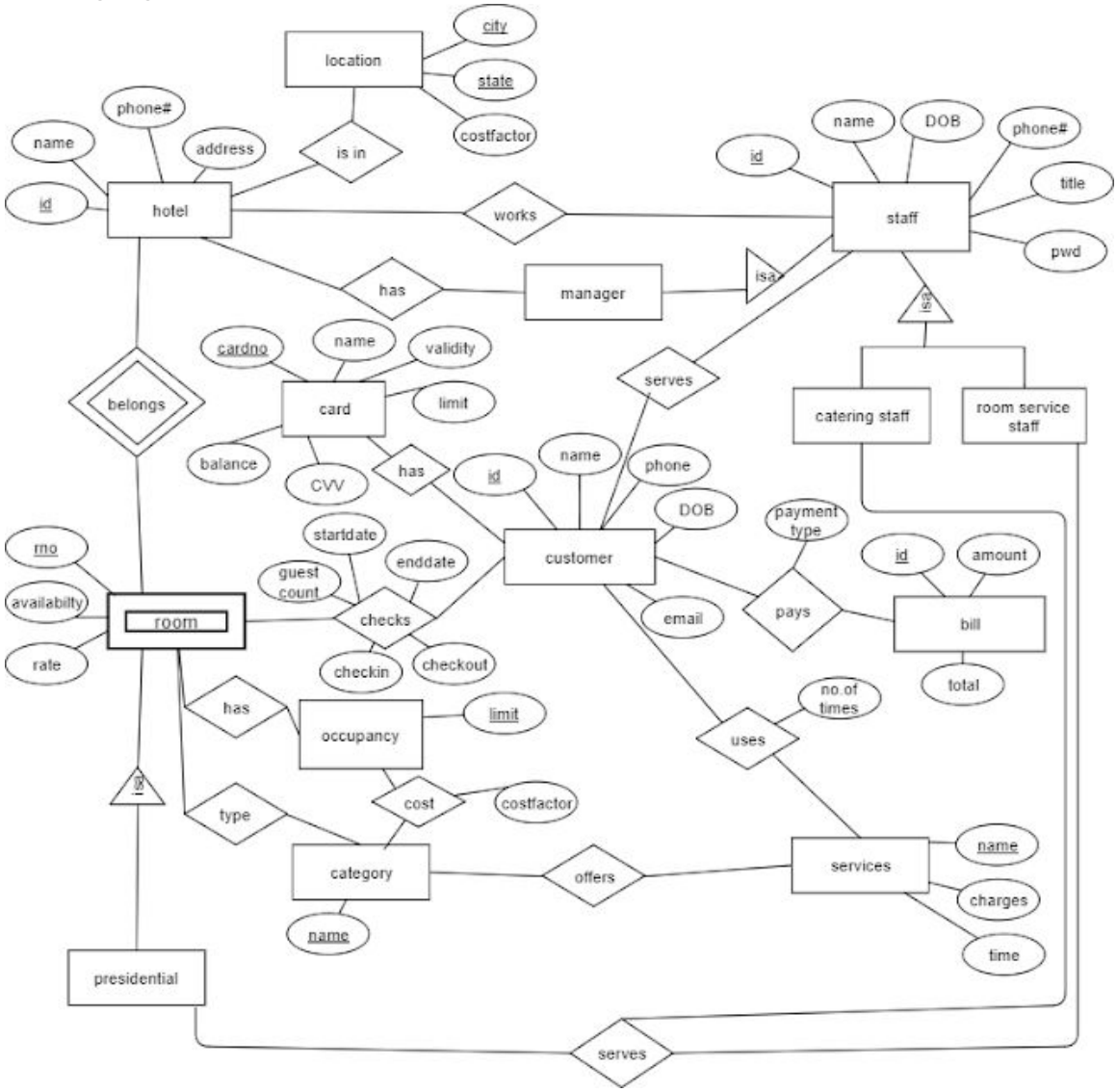
Service staff

Service staff should view all the data related to the room, customer and the services provided.

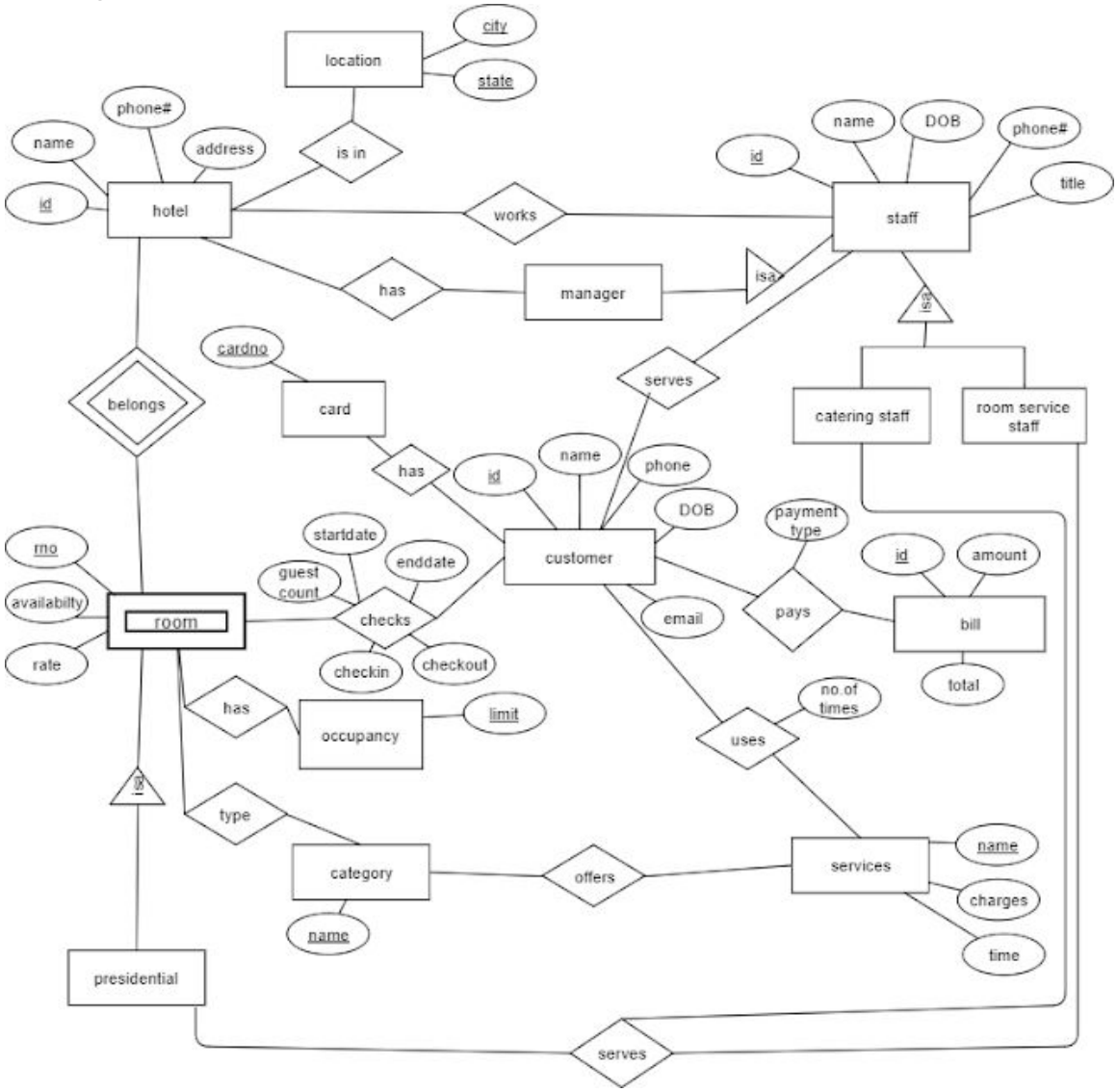
- The table of the customer with the details of ID, name, phone number, date of birth, email and card details.
- The details of service which the customer will be using. The name, charges and time of the service.
- The details and category of the room.

7) Local E/R Diagrams

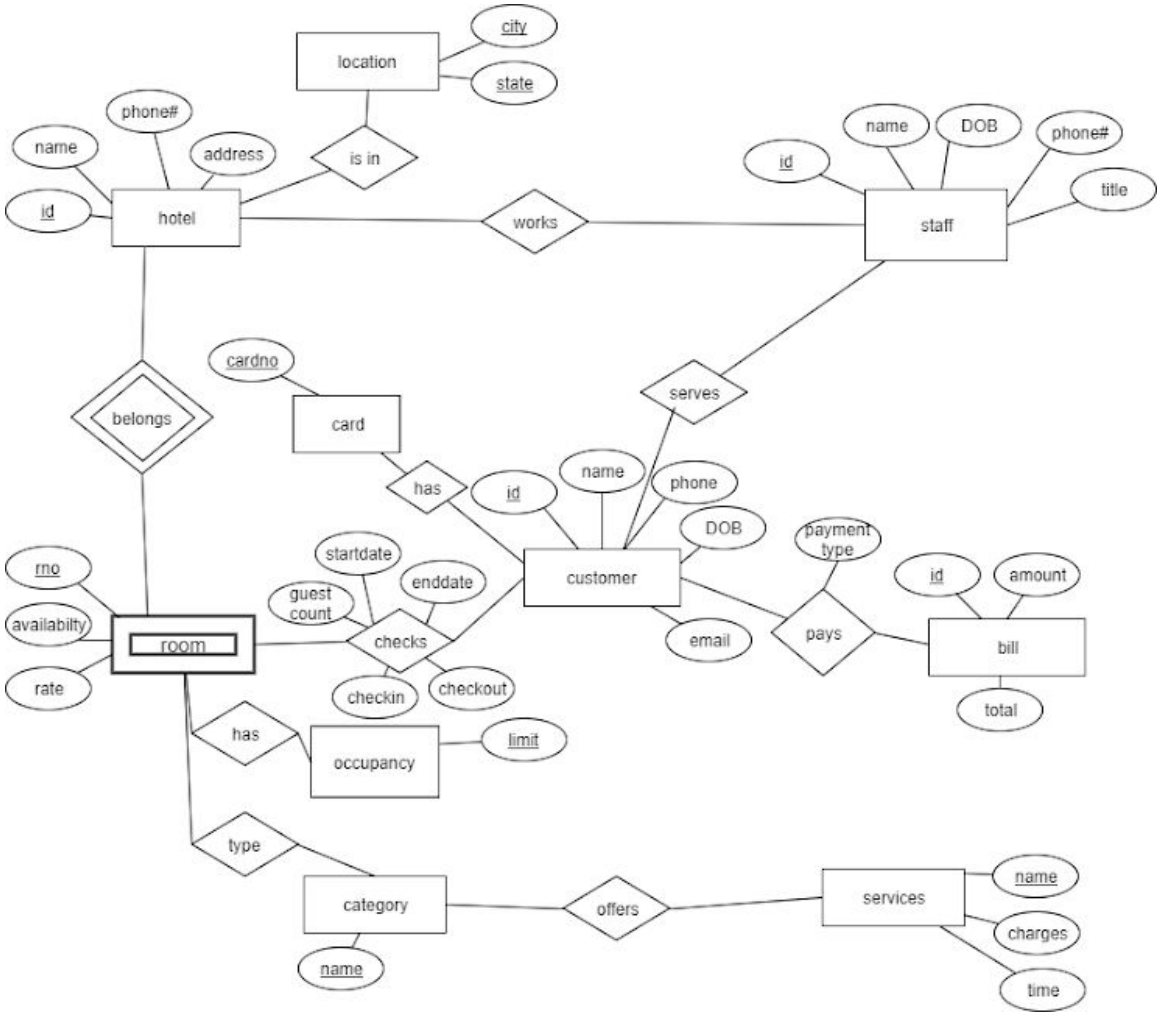
Managing Director



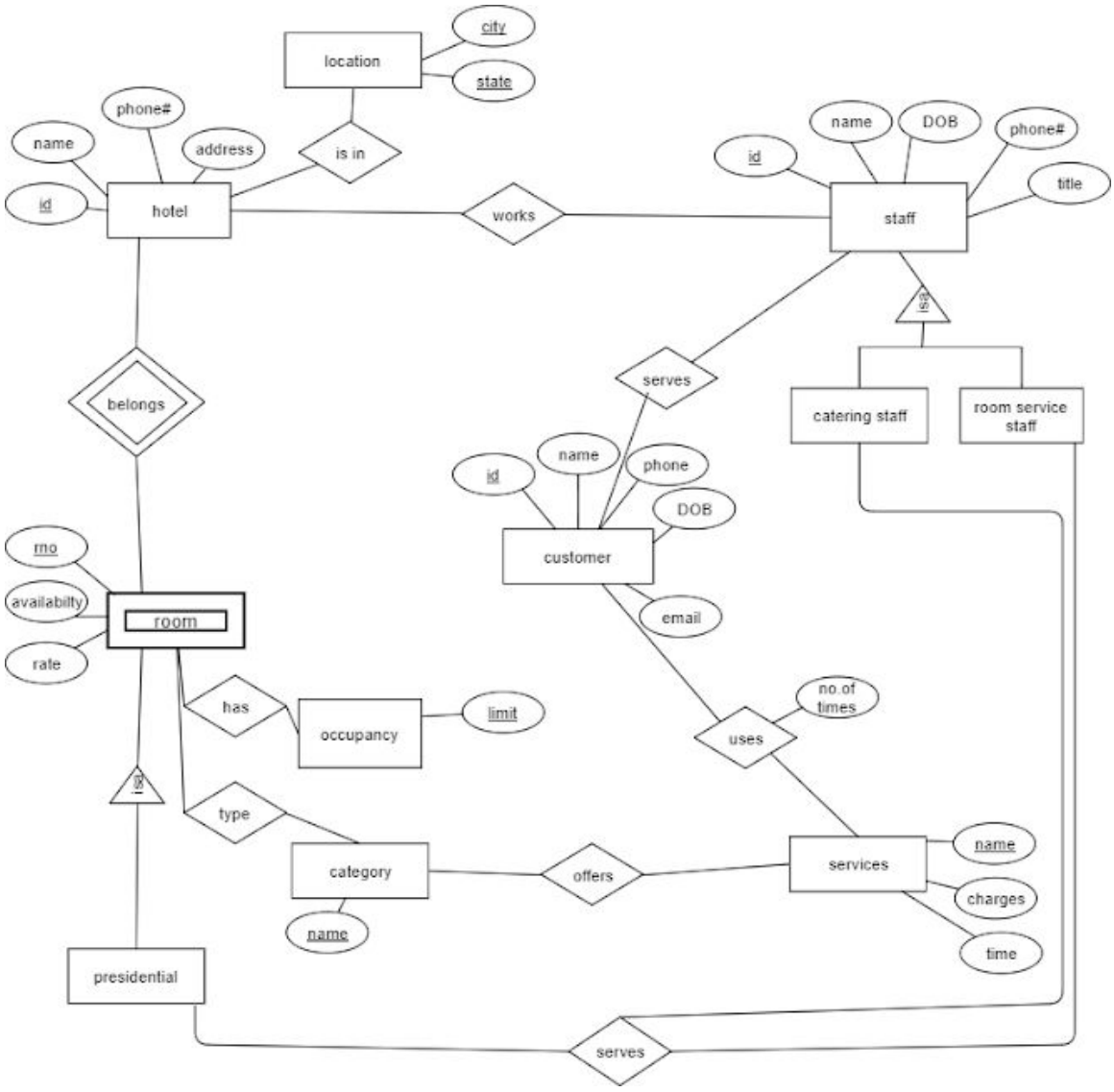
Manager


location
city
state
is in
hotel
name
phone#
address
id
works
staff
id
name
DOB
phone#
title
has
manager
isa
isa
belongs
cardno
card
has
serves
catering staff
room service staff
id
name
phone
DOB
customer
payment type
email
pays
id
amount
bill
total
rno
availabilty
rate
room
startdate
enddate
guest count
checks
checkin
checkout
has
occupancy
limit
no.of times
uses
type
category
name
offers
services
name
charges
time
presidential
serves

Front desk representative



Service Staff



8) Local E/R Documentation

1. People working for the hotel can be managers, front desk representatives and service staff.

2. We have given unique IDs to the entities for the following reasons:

    - *hotels*: This allows multiple hotels with same name to be distinct with phone number, address and ID.
    - *staff*: This allows multiple people with the same name, date of birth and title
    - *customer*: allows multiple people with the same name and details to be distinct.
    - *bills*: there could be multiple bills with the same amount and total.

3. Occupancy, card, services and category of the room were made entity sets rather than attributes to avoid redundancy that is multiple entries and would make searching for a specific entry easier.

4. We have made use of the following relationships:

    - *Works*: Multiple staff would work for a single hotel and a single hotel would be associated with a specific hotel.
    - *Belongs*: Many rooms can be associated with a hotel and one room will be in one hotel only.
    - *Serves*: Each staff will serve a particular room and a particular customer.
    - *Has*: Each hotel will have many managers and each manager will be associated with one hotel.
    - *Pays*: Each customer will pay for one or more bills and there could be many bills associated with one customer.
    - *Offers*:
    - *Cost*: Each type of occupancy will have a cost associated with it.
    - *Has*: Each customer will have one or many cards(credit/ debit cards) and each card will be associated with one customer.
    - *Uses*: Each customer will use on or more type of services.
    - *Checks:* Each customer checks in a room.

5. We have made use of hierarchical relations. The hierarchy relation between the rooms and presidential room is present because there are certain privileges given to the customers staying at the presidential room. The second hierarchy present in the er diagram is between the staff and managers, catering staff and

room service staff. Basically, staff is divided into different categories as different staff may have different functionalities and have different data views.

9) Local Relational Schemas

Managing director

- location(city,state,costfactor)
- hotel(id,name,phone#,address,city,state,manager_id)
- staff(id,name,phone#,title,pwd,DOB,hid)
- manager(id)
- catering staff(id)
- room service staff(id)
- occupancy(limit)
- category(name)
- room(rno,hid,availability,rate,limit,category_name)
- presidential(rno,hid,catering_id,service_id)
- customer(id,name,DOB,phone#,email,SSN,card_no,r_no,hid,guest_count, check_in,check_out,start_date, end_date)
- card(cardno,validity,name,limit,balance,cvv)
- bill(bid,cid,amount,total,payment_type)
- services(name,charges,time)
- serves(staff_id,customer_id)
- cost(limit,name,costfactor)
- offers(category_name,service_name)
- uses(cid,service_name,no.of.times)

Manager

- location(city,state)
- hotel(id,name,phone#,address,city,state,manager_id)
- staff(id,name,phone#,title,DOB,hid)
- manager(id)
- catering staff(id)
- room service staff(id)
- occupancy(limit)
- category(name)
- room(rno,hid,availability,rate,limit,category_name)
- presidential(rno,id,catering_id,service_id)
- customer(id,name,DOB,phone#,email,SSN,card_no,r_no,hid,guest_count, check_in,check_out,start_date, end_date)
- card(cardno)
- bill(bid,cid,amount,total,payment_type)
- services(name,charges,time)
- serves(staff_id,customer_id)

- offers(category_name,service_name)
- uses(cid,service_name,no.of.times)

Service Staff

- location(city,state)
- hotel(id,name,phone#,address,city,state)
- staff(id,name,phone#,title,DOB,hid)
- catering staff(id)
- room service staff(id)
- occupancy(limit)
- category(name)
- room(rno,hid,availability,rate,limit,category_name)
- presidential(rno,id,catering_id,service_id)
- customer(id,name,DOB,phone#,email,SSN)
- services(name,charges,time)
- serves(staff_id,customer_id)
- offers(category_name,service_name)
- uses(cid,service_name,no.of.times)

Front Desk

- location(city,state)
- hotel(id,name,phone#,address,city,state)
- staff(id,name,phone#,title,DOB,hid)
- occupancy(limit)
- category(name)
- room(rno,hid,availability,rate,limit,category_name)
- customer(id,name,DOB,phone#,email,SSN,card_no,r_no,hidguest_count, check_in,check_out,start_date, end_date)
- card(cardno)
- bill(bid,cid,amount,total,payment_type)
- services(name,charges,time)
- serves(staff_id,customer_id)
- offers(category_name,service_name)

10) Local Schema Documentation

Initially we decided upon the entities whether strong or weak , manually from the project narrative and later on went on to mechanically create the corresponding relations for each of the entities having the same number of attributes.

Our design highlights would include the below :-

a) Making Room a weak entity set as it has to take the key attribute from the Hotel entity set for the unique identification .
b) Hierarchical relationship with the entity staff for the catering and the room service staff so that they can be easily linked to the Presidential Suite.

We have followed the E/R approach to convert entities into relations schema. As a result, we have created separate relation schemas for each of the hierarchies. For example : Presidential which is in hierarchy with Room has been made as a separate relation schema.

Also, room service staff which is in hierarchy with staff entity also has been made as a relation schema.