CSC 440 Course Project

WolfVillas Hotel Database

Grayson Jones - gcjones

Rodolphe Njiwa Sandjong - rnjiwas

Julie Persinger - jcpersin

Tsu-Hsin Yeh - tyeh3

CSC 440

Professor Chirkova

Fall 2016

1) Assumptions and Problem Statement

Assumptions

1. The customer can only process check-in through a front desk representative (no online purchases, only over the phone or in person with a front desk representative).
2. When the customer is checked in, the start/end date, check-in/check-out time should be specified (customer cannot change their mind for staying longer or leave earlier).
3. Given room occupancy and room type, there is only one nightly rate associated with room of these type combination.
4. If a group of customers check in, then only one customer needs to be associated with check-in information and billing account. Meanwhile, only one customer is responsible for the payment.
5. Total expenses during a stay = phone bill + laundry bill + restaurant bill + (nightly rate * number of days staying)
6. Front desk staff will assign catering staff and service staff to a specific set of check in information if customers request to stay in a presidential room.
7. Service records are associated with a specific check-in instead of the customer they are serving (If person A, B and C check in as a group, the services availed for A or B or C will generate service records for check-in which represents the entire group).

Problem Statement

The hotel chain *WolfVillas* needs a database to manage their hotels in various cities across the country. The database system is mainly used by managers, front desk staff, billing staff, catering staff and service staff. The database shall maintain the information about hotels, staff, rooms, customers, billing accounts, service records and check-in information. Every user has different tasks and operations to perform on the database system.

There are many advantages to construct database system to manage the complex relationships for the domain.

- Database guarantees the integrity of data. The database designer can enforce constraints on the data based on the application domain so that data provided by user must be valid before it can store inside the system.
- Database abstracts the implementation of the underlying database structure from users. Furthermore, the complex query issued by the user will be optimized by database engine to achieve shortest query response time.
- Database support transaction and concurrent. Concurrency enables multiple users access and update the data without corruption. Transaction ensures that either atomic group of operations must be execute successfully together or none of the operations within the group gets execute. The feature cannot be satisfied by using traditional database.

2) User Classes

We believe that there are five user classes according to the narrative relating perform essential tasks and operations.

Managers are users who access and update the information for hotels, rooms, staff. Manager can generate reports on various perspective of the hotels. Manager can also view the information for the customers.

Front desk staff are users who register customer's basic information, process check-in/check-out and assign available room to customers. Front desk staff also bill the customer according to service records information.

Billing staff are users who generate and maintain billing accounts for every customer.

Catering staff are users who serve food to a group of customers who stay in the hotel. The catering staff can generate and update restaurants bills as a part of service records.

Service staff are users who serve a group of customers who stay in the hotel. The service staff can generate and update phone bills and laundry service bills as a part of service records.

3) The Main Entities

The five main entities that are most important to keep are listed in the following.

Hotel – hotel id, hotel name, hotel address, hotel phone number, hotel manager id.

Staff – staff id, SSN, name, age, gender, job title (Manager, Front desk representative, Room service staff, Billing staff, Catering staff), department, contact information (phone and address), hotel currently serving.

Room – room number, room category (Economy, Deluxe, Executive Suite, Presidential Suite), max occupancy, nightly rate, availability.

Customer – customer ID, name, gender, contact information (phone and address), email address.

Billing – customer ID, SSN of person responsible for the payment, billing address, payment information (payment method, card number).

4) A Usage Situation

1. **Information Processing**
   The manager has hired a new person for the room service staff. They will enter

the new person's staff ID, social security number, name, age, gender, job title (Room Service Staff), department, phone number, address, and hotel into the system. The new hire will then be assigned by the front desk staff to either a presidential suite or assigned to multiple rooms to service.

2. **Reports**
   The manager wants to know how well their hotel is doing. They get the number of occupants and the percent of the hotel that is occupied. They also check how many staff they have to ensure that each department has the staff it needs.

5) Application Program Interfaces

Information processing

newStaff(ssn, name, age, gender, title, department, phone, address)

return staffID for success or NULL for error

updateStaff(ssn, name, age, gender, title, department, phone, address, staffID)

return confirmation

- If NULL value for any of the attributes, the corresponding ones will not be updated

deleteStaff(staffID)

return confirmation

newRoom(hotelID, roomNumber, category, occupancy, nightly-rate, availability)

return confirmation

updateRoom(category, occupancy, nightly-rate, availability, hotelID, roomNumber)

return confirmation

- If NULL value for any of the attributes, the corresponding ones will not be updated

deleteRoom(hotelID, roomNumber)

return confirmation

newCustomer(ssn, name, gender, phone, address, email)

return customerID or NULL for error

updateCustomer(ssn, name, gender, phone, address, email, customerID)

return confirmation

- If NULL value for any of the attributes, the corresponding ones will not be updated

deleteCustomer(customerID)

return confirmation

checkAvailability(hotelID)

return a list of available rooms for specified hotel

*assignRoom(*hotelID, roomID, customerID, caterStaffID, serviceStaffID, currentOccupancy, start-date, start-time, end-date, end-time)

return checkinID or null for error

- If the customer is not assigned a presidential suite then caterStaffID and serviceStaffID can be left as NULL.

releaseRoom(checkinID)

return confirmation

Maintaining Service Availed Records

newPhoneBill(amount, customerID)

return billID or NULL for error

updatePhoneBill(billID, amount)

return confirmation

newLaundryBill(amount, customerID)

return billID or NULL for error

updateLaundryBill(billID, amount)

return confirmation

newRestaurantBill(billID, amount)

return billID or NULL for error

updateRestaurantBill(billID, amount)

return confirmation

Maintaining billing accounts

generateBillingAccount(billingAddress, cardNumber, paymentMethod, CustomerID)

return accountID or NULL for error

updateBillingAccount(billingAddress, cardNumber, paymentMethod, accountID)

return confirmation

- If NULL value for any of the attributes, the corresponding ones will not be updated

Reports

getOccupancy(roomtype, start-date, end-date, hotelID)

return number of occupancy specified by the condition

- If NULL for any attribute, then it will not be a part of constrain

getOccupants(hotelID)

return number of current occupant for specified hotel

getPercentOccupied(hotelID)

return percentage of room occupied for specific hotel

getStaffInfo(hotelID)

return a list staff information sorted by staff role in a given hotel

getCustomerAssigned(StaffID)

return a list of customer assigned to specified staff

- If staff ID does not represent catering/service staff, then return NULL

6) Data Views

Manager

The managers have the most power to control the hotel chain information thus can view most of the parts in the database. The data viewed by the manager is listed as follows:

- The table of staff with SSN, name, age, gender, title, department, phone and address.
- The table of hotels with name, address and phone.
- The table of room with room number and availability.
- The table of room type with category, occupancy and nightly-rate.
- The table of check-in information with current-occupancy, start-date, end-date, check-in time, check-out time.

The managers do not need to see billing account information because it is delegated to billing staff. Likewise, the managers do not see service records because it is delegated to catering staff and service staff.

Front desk staff

The front desk staff shall view the information about check-in/check-out information, catering staff, service staff, hotel customer and hotel room. The data viewed by the front desk staff is listed as follow:

- The table of room with hotel id, room number and availability.
- The table of room type with category, occupancy and nightly-rate.

- The table of check-in information with current-occupancy, start-date, end-date, check-in time, and check-out time.
- The table of staff with name, title, and department.

Billing staff

The billing staff shall view the information about billing staff, billing account, hotel customer, service records, check-in/check-out information and hotel room. The data viewed by the billing staff is listed as follows:

- The table of staff with name, title,and department.
- The table of billing with billing address, payment method, and card number.
- The table of customers with name, gender, phone, address, email and SSN.
- The table of check-in information with current occupancy, start date, end date, check-in time, and check-out time.
- The table of rooms with hotel id, room number, and availability.
- The table of room types with category, occupancy, and nightly-rate.
- The table of service records with type and amount.

Catering staff

The catering staff shall view the information about catering staff, service records, check-in/check-out information, and hotel room. The data viewed by the catering staff is listed as following:

- The table of staff with name, title, and department.
- The table of service records with type and amount.
- The table of customers with name and gender.
- The table of rooms with hotel id, room number, and availability.
- The table of room types with category, occupancy, and nightly-rate.
- The table of check-in information with current occupancy, start date, end date, check-in time, and check-out time.

Service staff

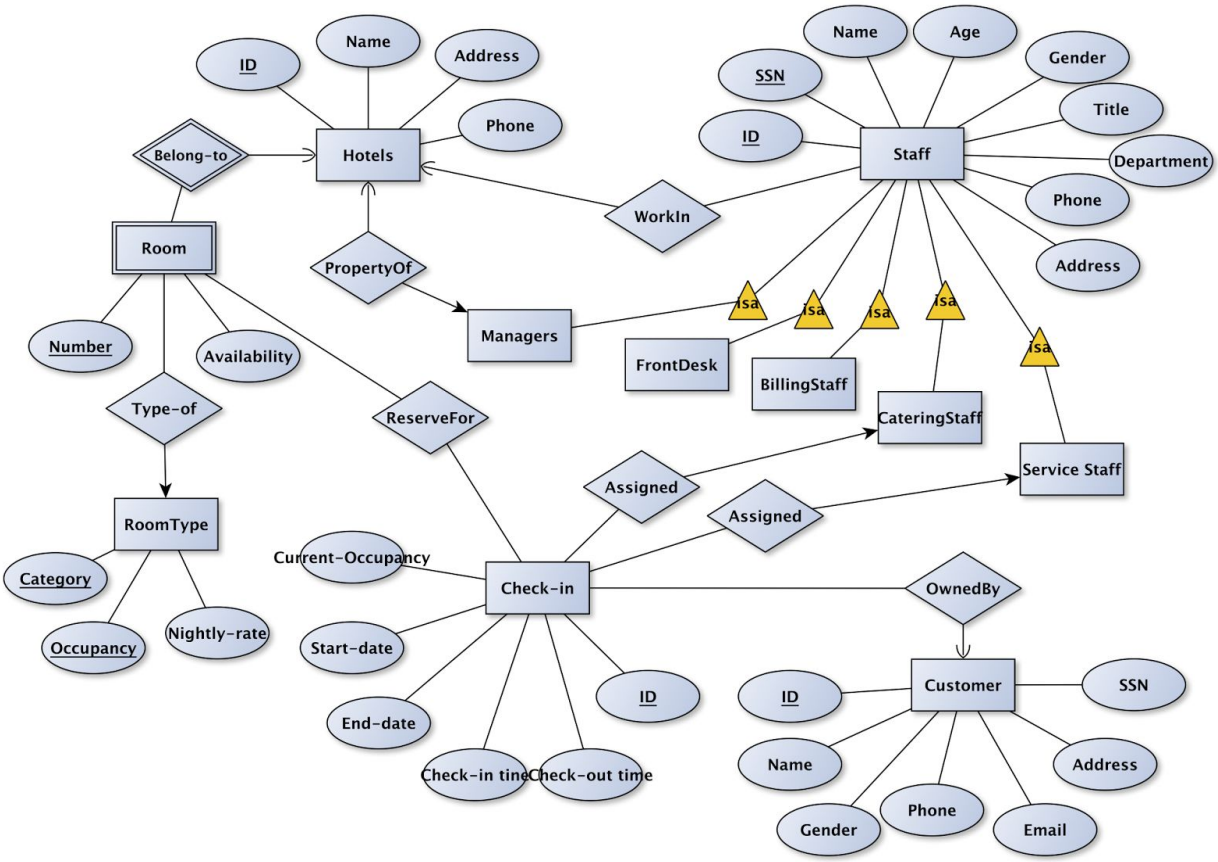
The service staff shall view the information about service staff, service records, check-in/check-out information, and hotel room. The data viewed by the service staff is listed as below.

- The table of staff with name, title, and department.
- The table of service records with type and amount.

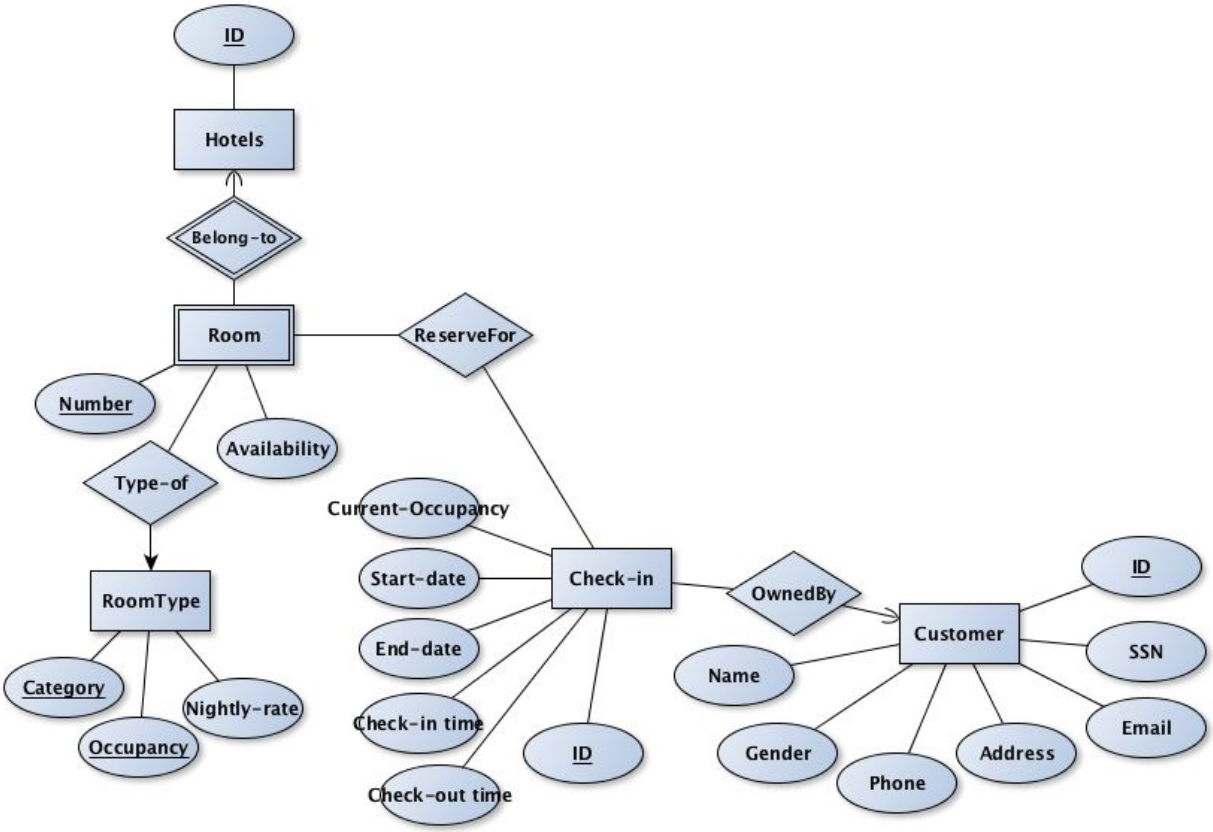
- The table of customers with name and gender.
- The table of rooms with hotel id, room number, and availability.
- The table of room types with category, occupancy, and nightly-rate.
- The table of check-in information with current occupancy, start date, end date, check-in time, and check-out time.

7) Local E/R Diagrams

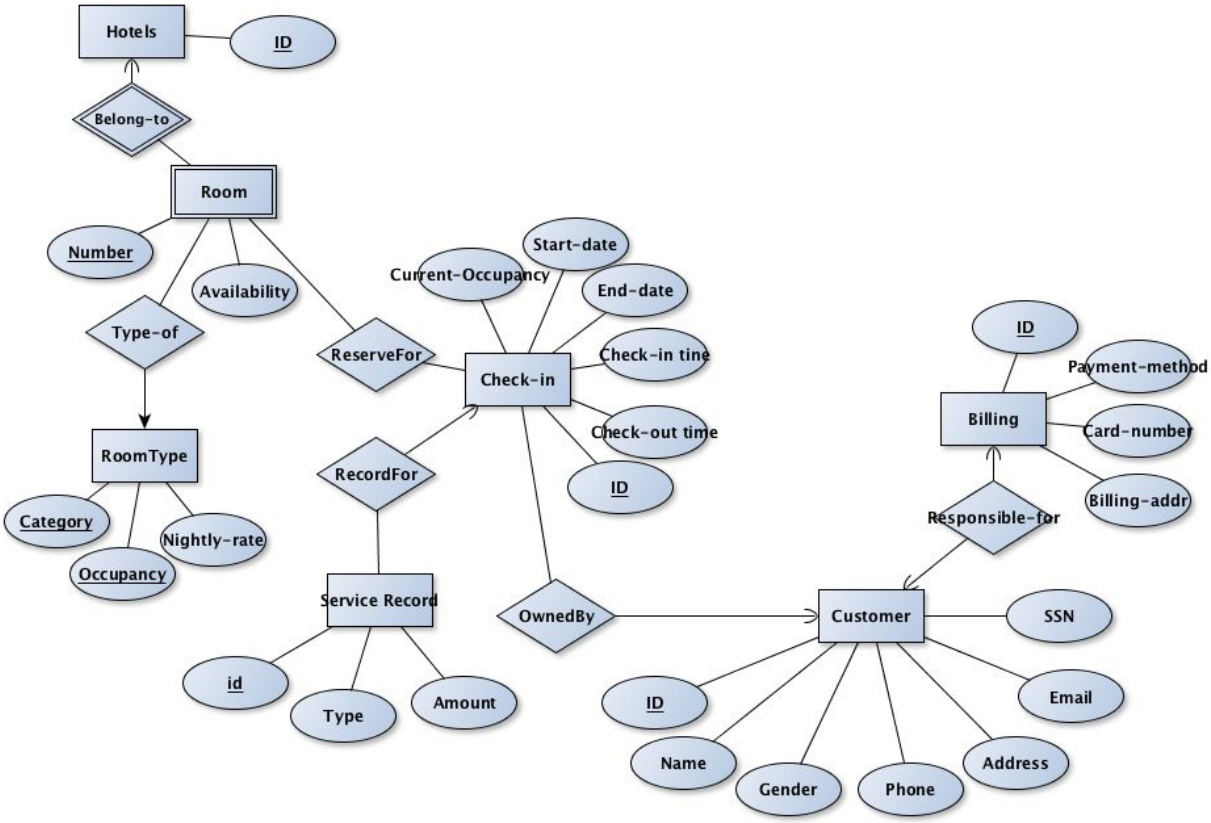
Manager



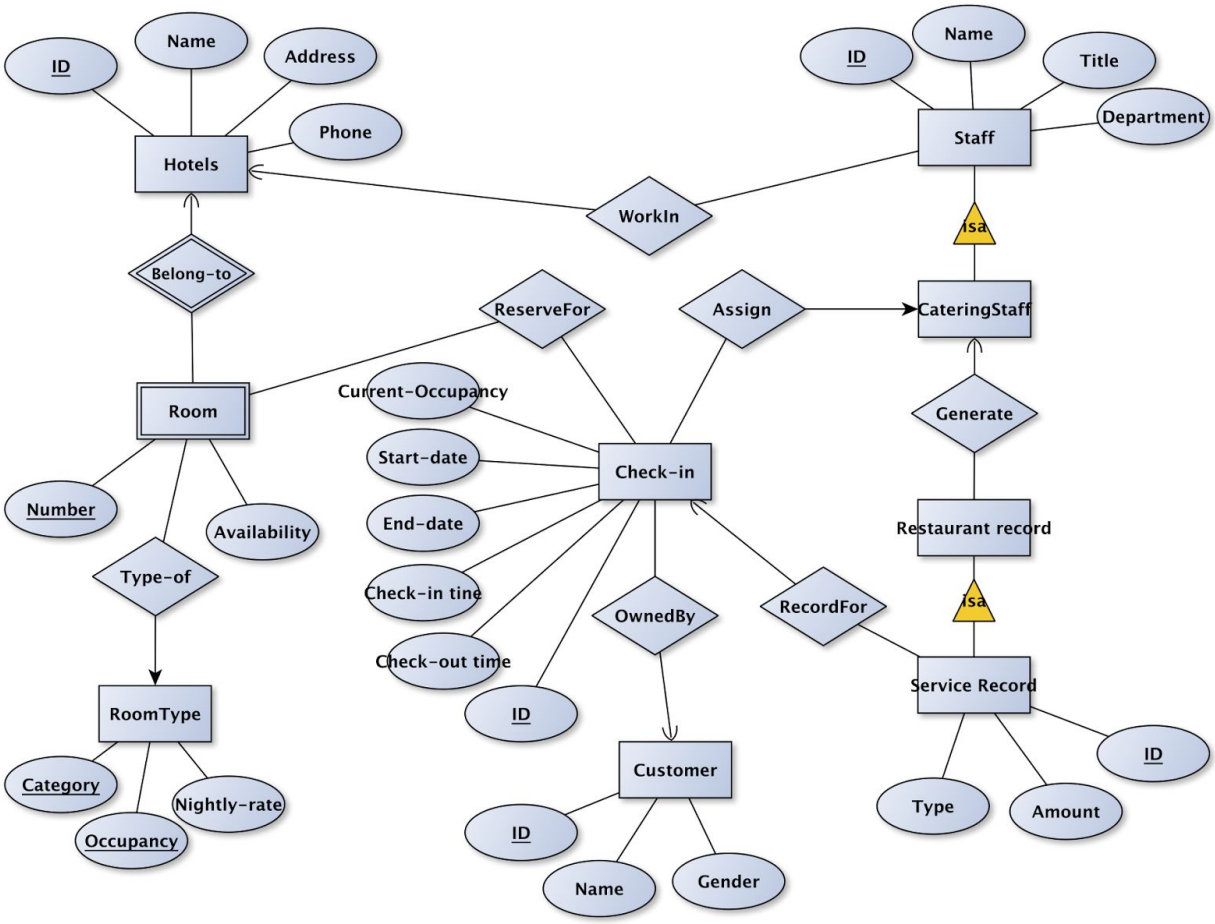
Front Desk Staff



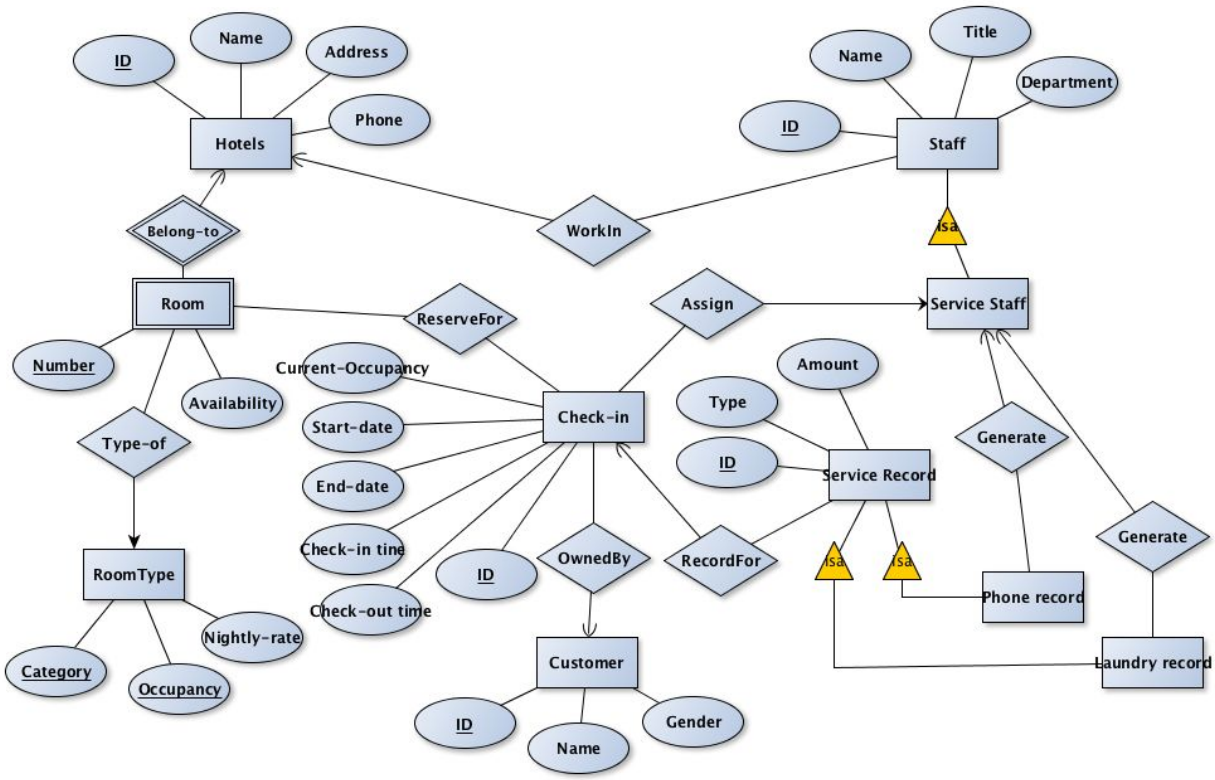
Billing Staff



Catering Staff



Service Staff



8) Local E/R Documentation

1. The local E/R diagram corresponds to each user class associated with the application: managers, front desk staff, billing staff, catering staff and service staff.
2. We distinguish front desk staff and billing staff because the front desk staff handles only check-in and check-out while the actual payment for the billing during the stay is delegated to billing staff. For example, the front desk staff can check out the customer only if they have come to billing staff and pay for all the bills.
3. We chose to represent room as a weak entity because room is only uniquely identified provided by the ID of the hotel it is in and its room number in real life.
4. A subclass design is used so that the database system can distinguish different types of staff (such as manager, front desk staff, etc.) and different type of service records (such as laundry bill, phone bill, etc.) which can have potentially different relationship in the database system.
5. The following shows the constraints we made on the relationship in the diagram:
    - *Belong-To*: weak relationship for weak entity room to borrow key from hotel. A hotel has multiple rooms, but a room belongs to exactly one hotel.
    - *TypeOf*: each room has exactly one type and a room type can apply to multiple rooms.
    - *PropertyOf*: A hotel can be owned by at most one manager, a manager can own exactly one hotel.
    - *WorkIn*: A hotel can have multiple staff, but a staff can only work for exactly one hotel.
    - *ReserveFor*: A room can have multiple check-in information (past and present) related to it. A check-in information can have more than one room booked for the check in.
    - *Assigned*: Catering staff can be assigned to serve customers in multiple check-in. A check-in can have at most one catering staff associated.
        - If the room booked for check-in is presidential room, then exactly one catering staff is assigned.
        - Otherwise, no catering staff is assigned and appeared as NULL for cater staff id in check-in relation.
    - *Assigned*: Service staff can be assigned to serve customers in multiple check-in. A check-in can have at most one Service staff associated.
        - If the room booked for check-in is presidential room, then exactly one service staff is assigned.
        - Otherwise, no catering staff is assigned and appeared as NULL for service staff id in check-in relation.
    - *OwnedBy*: A check-in information is owned by exactly one customer (if it is a group then one representative owns the check-in information). A customer can have multiple check-in information (past and present).
    - *Generate*: A check-in information is generated by exactly one front desk staff. A front desk staff can generate multiple check-in information.
    - *RecordFor*: A checked-in information can have multiple service records. A service record is only tie to one specific check-in information.

- *Generate*: A service staff can generate multiple laundry records. A laundry record only generated by exactly one service staff.
- *Generate*: A service staff can generate multiple phone records. A phone record only generated by exactly one service staff.
- *Generate*: A catering staff can generate multiple restaurant records. A laundry record only generated by exactly one restaurant staff.
- *Generate*: A billing staff can generate multiple bill accounts. A billing account can only have one billing staff associated with it.
- *ResponsibleFor*: A customer has exactly one billing account. A billing account has exactly one customer who owns it.

9) Local Relational Schemas

Manager

staff(id, ssn, name, age, gender, title, department, phone, address, hotel_id)

manager(id, hotel_id)

frontdesk_staff(id)

billing_staff(id)

catering_staff(id)

service_staff(id)

hotel(id, name, address, phone)

room(number, category, occupancy, availability, hotel_id)

room_type(category, occupancy, nightlyRate)

customer(id, name, ssn, gender, phone, email, address)

checkin(id, current_occupancy, start_date, end_date, start_time, end_time, cstaff_id, sstaff_id, customer_id)

reserve_for(hotel_id, room_number, checkin_id)

Front Desk Staff

staff(id, name, title, department, hotel_id)

catering_staff(id)

service_staff(id)

frontdesk_staff(id)

room(number, category, occupancy, availability, hotel_id)

room_type(category, occupancy, nightlyRate)

reserve_for(hotel_id, room_number, checkin_id)

customer(id, name, ssn, gender, phone, email, address)

checkin(id, current_occupancy, start_date, end_date, start_time, end_time, fdstaff_id, cstaff_id, sstaff_id, customer_id)

Billing Staff

staff(id, name, title, department, hotel_id)

billing_staff(id)

room(number, category, occupancy, availability, hotel_id)

room_type(category, occupancy, nightly_rate)

service_record(id, type, amount, checkin_id)

customer(id, name, ssn, gender, phone, email, address, billing_id)

checkin(id, current_occupancy, start_date, end_date, start_time, end_time, customer_id)

reserve_for(hotel_id, room_number, checkin_id)

billing(id, billing_addr, payment_method, card_number, customer_id)

Catering Staff

staff(id, name, title, department, hotel_id)

catering_staff(id)

room(number, category, occupancy, availability, hotel_id)

customer(id, name, gender)

checkin(id, current_occupancy, start_date, end_date, start_time, end_time, cstaff_id, customer_id)

service_record(id, type, amount, checkin_id)

restaurant_record(id, cstaff_id)

reserve_for(hotel_id, room_number, checkin_id)

Service Staff

staff(id, name, title, department, hotel_id)

service_staff(id)

room(number, category, occupancy, availability, hotel_id)

customer(id, name, gender)

checkin(id, current_occupancy, start_date, end_date, start_time, end_time, sstaff_id, customer_id)

service_record(id, type, amount, checkin_id)

phone_record(id, sstaff_id)

laundry_record(id, sstaff_id)

reserve_for(hotel_id, room_number, checkin_id)

10) Local Schema Documentation

The technical approach to convert E/R diagram to relation is described as following

1. For each entity displayed on the diagram, key(s) and attributes associated with the entity should be recorded in the schema.
2. For weak entity (specifically Room entity), the schema includes information to describe in part 1 plus the additional key(s) from the supporting relationship.
3. For many-many relationships, the key(s) for the entity associated with the relationship is included in the schema.
4. For many-one relationships (many-one relationship R from E to F), we can combine source entity and relationship into one schema that contains all the attributes and key(s) from source entity(E), attributes associated with relationship(R) and key(s) from target entity(F).

The method that converts subclass into schema follows the E/R viewpoint. The schema for subclass should contain its unique attributes as well as key(s) from its parent.