Hotel Reservation System

Name: Nicholas Wendt
Student ID: 923609710
GitHub username: Mystic2122

Milestone	Date Submitted
M1	09/20/2024
M2	10/2/2024
M3	10/16/2024

Table of Contents

Project Description

1. Motivations for Creating the Database

When looking through the list of possible projects, I chose to create a hotel management database because it seemed like a good, real-world application that wouldn't be too hard to implement. I was also able to come up with enough entities for this database by brainstorming ideas with a friend. I think it would be interesting to try and figure out how to optimize room availability and implement dynamic pricing. Some hotel management systems may lack or have inefficient forecasting capabilities, leading to lost revenue and operational inefficiencies. The goal of this database is to address these challenges by offering hotels insights into seasonal demand, improving both guest satisfaction and hotel profitability.

---

2. High-Level, Non-Technical Description

This hotel database system will streamline and centralize hotel operations such as room bookings, guest management, billing, staffing, and other hotel services. A key feature will be predictive analytics, allowing hotels to adjust pricing and staffing based on expected demand. This system will improve the operational efficiency of the hotel.

---

3. Unique Features

The **Predictive Pricing** feature will allow hotels to forecast peak demand periods based on historical booking data, enabling more informed decisions about pricing and staffing. This predictive pricing feature could also take into account nearby events that may draw more tourists than normal to a specific city such as concerts, conventions, and major sporting events.

Another unique feature will be a loyalty program that not only tracks frequent stays but also personalizes offers based on guest preferences and behavior. These features build upon existing systems but add a personalized engagement that current tools often lack.

---

4. Existing Software Tools/Products

- **Opera Property Management System (PMS)**: Opera PMS is widely used for hotel database management. Integrating my database's predictive analytics would allow hotels to better optimize pricing and staffing during peak and off-peak seasons, boosting profitability.
- **Cloudbeds**: Cloudbeds could benefit from the personalized loyalty program feature, which would increase customer retention by offering promotions and discounts to specific guests. In addition, the predictive insights would help optimize room availability and pricing, addressing a current gap in the software.

---

5. Use Cases

- **Use Case 1: Dynamic Pricing Suggestion**
    - **Actor**: Hotel Manager
    - **Scenario**: The manager logs into the system to check room availability for the upcoming month. The system analyzes historical booking data and predicts an increase in demand.
    - **Outcome**: The system suggests a price increase for specific rooms, helping the hotel increase revenue during peak demand.
- **Use Case 2: Personalized Guest Experience**
    - **Actor**: Guest
    - **Scenario**: A returning guest books a room online. The system recognizes their previous stays and preferences (e.g. a preference for a room with a view).
    - **Outcome**: The system automatically assigns a room matching the guest's preferences and applies a loyalty discount, improving the guest's experience.
- **Use Case 3: Staff Scheduling Optimization**
    - **Actor**: HR Manager

- **Scenario**: The HR manager uses the system to generate a staff schedule for the upcoming month. Based on predicted high-occupancy periods, the system suggests increasing staff availability during certain weeks.
- **Outcome**: The HR manager adjusts the schedule, ensuring that there will be enough staffing during peak periods and avoiding under or overstaffing.

Functional Requirements

1. Guest

1.1 Each guest shall have a unique guest_id.
1.2 Each guest shall make at least 1 reservation.
1.3 Each guest can create at most 1 profile.
1.4 Each guest shall have a first and last name.
1.5 Each guest shall have a date of birth.
1.6 Each guest shall have at most 1 email on file.
1.7 Each guest shall have one or more addresses on file.
1.8 Each guest can be a part of at most 1 rewards program.
1.9 Each guest shall have at most one phone number on file.

2. Room

2.1 Each room shall have a unique room_id.
2.2 Each room shall be assigned to at most 1 reservation per day.
2.3 Each room shall have exactly 1 room type (e.g. 2 queen beds, 1 king bed, penthouse suite).
2.4 Each room shall be associated with exactly 1 hotel branch.
2.5 Each room shall have at least 1 housekeeper assigned to it per day.
2.6 Each room shall have at most 1 mini bar.
2.6 Each room shall have a 0, 1, or many amenities.

3. Employee

3.1 Each employee shall have a unique employee_id.
3.2 Each employee shall have at least 1 employee role (e.g., host, manager, receptionist, maid).
3.3 Each employee shall have a first and last name.
3.4 Each employee shall belong to exactly 1 hotel branch.
3.5 Each employee shall track their hours worked per week.
3.6 Each employee shall track their overtime hours worked per week.

4. Hotel Branch

4.1 Each hotel branch shall have a unique hotel_id.

4.2 Each hotel branch shall have one location associated with it.
4.3 Each hotel branch shall have a unique parking_lot_id (foreign key).
4.4 Each hotel branch shall have 0, 1, or many amenities.
4.5 Each hotel branch shall have 0, 1, or many restaurants.
4.6 Each hotel branch shall have its own inventory table.
4.7 Each hotel branch shall have a phone number.

5. Reservation
5.1 Each reservation shall have a unique reservation_id.
5.2 Each reservation shall have exactly 1 room type associated with it.
5.3 Each reservation shall have a check-in time and date.
5.4 Each reservation shall have a check-out time and date.
5.5 Each reservation shall belong to exactly 1 hotel branch.
5.6 Each reservation shall reserve at most 1 parking spot.
5.7 Each reservation shall have exactly 1 bill associated with it.
5.8 Each reservation shall be able to create at most 1 hotel review.
5.9 Each reservation can be canceled up to a certain date.

6. Rewards Program
6.1 Each rewards program shall have a unique reward_program_id.
6.2 Each rewards program shall have perks associated with it.
6.3 Each rewards program shall have increased rewards with higher tiers.

7. Parking Lot
7.1 Each parking lot shall have a unique parking_lot_id.
7.2 Each parking lot shall have some number of parking spaces.
7.3 Each parking lot can have 0, 1, or many valets.

8. Employee Roles
8.1 Each employee role shall have a unique employee_role_id.
8.2 Each employee role shall have a name.
8.3 Each employee role shall have a salary.

9. Staff Schedule
9.1 Each schedule slot shall have a unique schedule_id.
9.2 Each schedule slot shall be filled with at least 1 employee_id.
9.3 Each employee can not be scheduled for more than 40 hours per week.
9.4 Each hotel has its own staff schedule.

10. Linen Service

    10.1 Each linen service shall have a unique linen_service_id.
    10.2 Each linen service shall be used by at least 1 hotel.
    10.3 Each linen service shall have contact information.

11. Amenities

    11.1 Each amenity shall have a unique amenity_id.
    11.2 Each amenity shall have a name.
    11.3 Each amenity shall be available to at least 1 hotel.
    11.4 Each amenity shall have a cost associated with it.

12. Restaurant

    12.1 Each restaurant shall have a unique restaurant_id.
    12.2 Each restaurant_id shall be associated with exactly 1 hotel branch.
    12.3 Each restaurant shall have hours of operation.

13. Room Service

    13.1 Each room service request has a unique room_service_id.
    13.2 Each room service request has a cost associated with it.
    13.3 Each room service request shall be billed to the bill_id associated with the reservation.

14. Accounting

    14.1 Each bill shall have a unique bill_id.
    14.2 Each bill_id shall be associated with exactly 1 reservation_id.
    14.3 Each charge to the reservation will be added to the bill and paid at the end of the reservation.

15. Events

    15.1 Each event shall have a unique event_id.
    15.2 Each event shall occur at exactly 1 hotel branch.
    15.3 Each event shall have a date.
    15.4 Each event shall have a maximum number of attendees.

16. Mini-bar
    16.1 Each mini-bar shall have a unique mini_bar_id.
    16.2 Each mini bar shall be associated with exactly 1 room.
    16.3 The cost of each item removed from the mini-bar shall be added to the bill_id.

17. Hotel Review
    17.1 Each hotel review shall have a hotel_review_id.
    17.2 Each hotel review shall be associated with exactly 1 hotel branch.
    17.3 Each hotel review shall include a rating.
    17.4 Each hotel review shall have a date when it was published.

18. Predictive Pricing
    18.1 Each pricing instance shall have a unique pricing_id.
    18.2 Predictive pricing shall be based on multiple factors (occupancy rates, season, booking patterns, events).
    18.3 Predictive pricing shall automatically update prices based on criteria (lower prices if occupancy rates are low, raise prices if booking rates are high).
    18.4 Predictive pricing shall allow manual overrides by hotel management.
    18.5 Predictive pricing shall not fall below a specified threshold.

19. Promo Offers
    19.1 Each promo offer shall have a promo_offer_id.
    19.2 Promo offers shall only be available to guests with a profile.
    19.3 Each promo offer shall only be used once per customer.
    19.4 Each promo offer shall have a description.
    19.5 Each promo offer shall have a start and end date.

20. Maintenance
    20.1 Each maintenance request shall have a unique maintenance_id.
    20.2 Each maintenance request shall only be associated with exactly 1 room.
    20.3 Each maintenance request shall be associated with exactly 1 employee assigned to the request.
    20.4 Each maintenance request shall have a priority level.

21. Inventory

21.1 Each item in inventory shall have an item_id.
21.2 Each type of item in the inventory shall have a count,
21.3 Inventory shall be automatically ordered when the inventory count falls below a specified threshold.

Non-Functional Requirements

Performance Requirements

1.1 The system shall process reservation requests in a reasonable time.

1.2 The database shall be able to handle a sufficient amount users without a decrease in performance.

1.3 Room availability queries shall return results in a reasonable time depending on the number of rooms the hotel has.

Scalability Requirements

2.1 The system shall be able to scale and support additional hotel branches without requiring major changes.

2.2 The database shall allow for the addition of up to 1,000 hotel branches and 100,000 rooms without impacting performance.

2.3 The system shall be able to scale to accommodate an increase in user traffic during peak seasons.

Availability Requirements

3.1 The system shall be available 99.9% of the time, except for during scheduled maintenance periods.

3.2 Scheduled maintenance shall occur during non-peak hours.

3.3 In the event of a system failure, the system shall recover and be fully operational within an acceptable amount of time.

Security Requirements

4.1 All guest and payment data shall be encrypted.

4.2 Access to employee and admin functions shall be restricted by employee roles.

4.3 All user sessions shall time out after 15 minutes of inactivity.

Usability Requirements

5.1 The system shall have a user-friendly interface for hotel staff and guests.

5.3 All error messages shall have a description.

Data Integrity Requirements

6.1 The system shall automatically validate input data (e.g., guest names, dates, payment details) to prevent invalid entries.

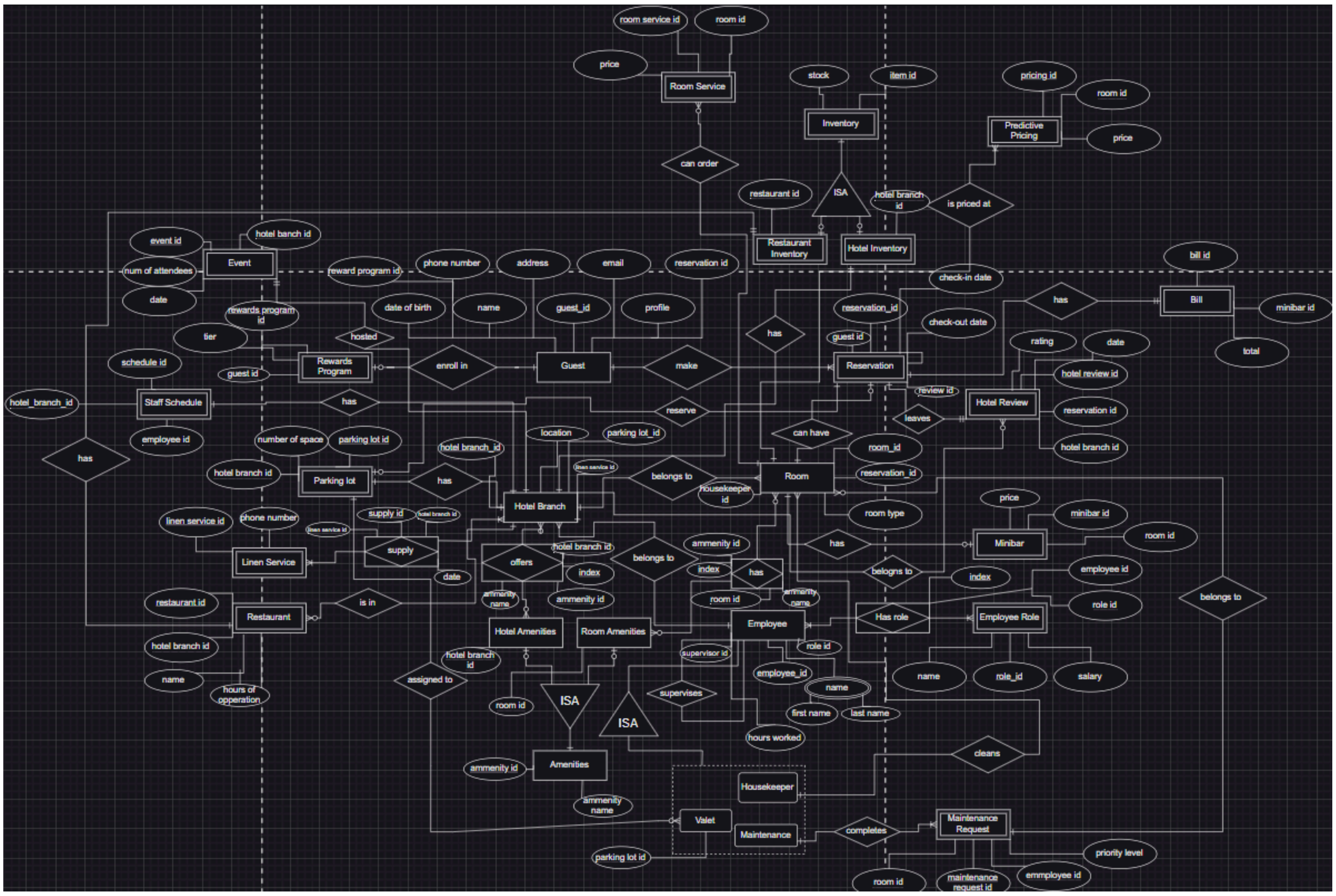
6.2 Any changes made to reservation records shall be tracked, with the ability to view the history of updates.

Backup and Recovery Requirements

7.1 The system shall perform automated daily backups of the entire database at non-peak hours.

7.2 Backup data shall be stored in at least two separate locations.

7.3 The system shall support full database restoration from backups in the event of data corruption or loss.

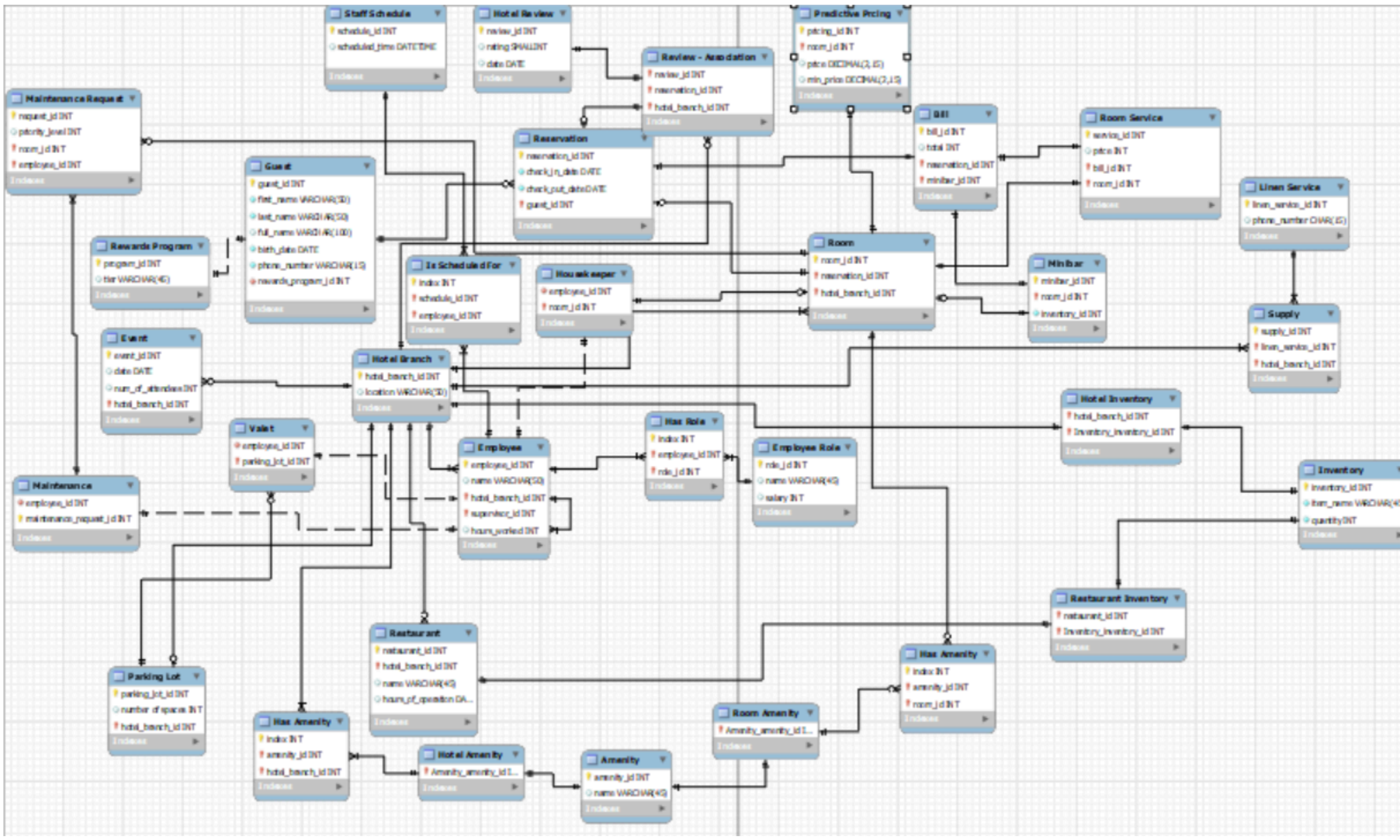

room service id
room id
price
Room Service
stock
item id
pricing id
room id
Inventory
Predictive Pricing
price
can order
restaurant id
ISA
hotel branch id
is priced at
event id
hotel banch id
Restaurant Inventory
Hotel Inventory
bill id
Event
num of attendees
reward program id
phone number
address
email
reservation id
check-in date
date
rewards program id
date of birth
name
guest_id
profile
reservation_id
has
Bill
minibar id
tier
hosted
has
guest id
check-out date
schedule id
Rewards Program
enroll in
Guest
make
Reservation
rating
date
total
guest id
hotel review id
hotel_branch_id
Staff Schedule
has
reserve
review id
Hotel Review
leaves
reservation id
employee id
number of space
parking lot id
location
parking lot_id
can have
room_id
hotel branch id
has
hotel branch id
hotel branch_id
Parking lot
has
linen service id
belongs to
Room
reservation_id
Hotel Branch
housekeeper id
room type
price
linen service id
phone number
supply id
hotel branch id
minibar id
room id
linen service id
supply
ammenity id
has
Minibar
Linen Service
date
hotel branch id
offers
belongs to
index
has
index
belongs to
index
employee id
ammenity name
ammenity id
room id
ammenity name
restaurant id
is in
Has role
Employee Role
role id
belongs to
Restaurant
Hotel Amenities
Room Amenities
Employee
role id
hotel branch id
supervisor id
hotel branch id
employee_id
name
role_id
salary
name
hours of opperation
assigned to
name
ISA
supervises
first name
last name
room id
ISA
hours worked
cleans
amenity id
Amenities
Housekeeper
ammenity name
Valet
Maintenance
completes
Maintenance Request
parking lot id
priority level
room id
maintenance request id
emmployee id

Entity Descriptions

- Guest (Strong)
  - Guest_id: pk, numeric
  - Name: alphanumeric
  - DoB: date
  - Address: alphanumeric
  - Phone number: alphanumeric
  - Email: alphanumeric
  - Rewards_program_id: fk, numeric
  - Reservation_id: fk, numeric
- Room (Strong)
  - Room_id: pk, numeric
  - Reservation_id: fk, numeric
  - Room type: alphanumeric
  - Housekeeper_id: fk, numeric
- Employee (Strong)
  - Employee_id: pk, numeric
  - Role_id: fk, numeric
  - Supervisor_id: fk (recursive), numeric
  - Name: alphanumeric
  - Hours worked: numeric
- Hotel Branch (Strong)
  - Hotel_branch_id: pk, numeric
  - Parking_lot_id: fk, numeric
  - Linen_service_id: fk, numeric
  - Location: alphanumeric
- Reservation (Weak)
  - Reservation_id: pk, numeric
  - Check-in date: date
  - Check-out date: date
  - Guest_id: fk, numeric
  - Review_id: fk, numeric
- Rewards Program (Weak)
  - Rewards_program_id: pk, numeric
  - Guest_id: fk, numeric
  - Tier: alphanumeric
- Parking Lot (Weak)
  - Parking_lot_id: pk, numeric
  - Number of spaces: numeric
  - Hotel_branch_id: fk, numeric
  - Employee_id (valet): fk, numeric
- Employee Roles (Weak)
  - Role_id: pk, numeric

  - Salary: numeric
  - Name of role: alphanumeric
- Staff Schedule (Weak)
  - Schedule_id: pk, numeric
  - Employee_id: fk, numeric
  - Hotel_branch_id: fk, numeric
  - Time scheduled: date/time
- Linen Service (Weak)
  - Linen service id: pk, numeric
  - Phone number: numeric
- Amenities (Strong)
  - Amenity_id: pk, numeric
  - Name: alphanumeric
- Restaurant (Weak)
  - Restaurant_id: pk, numeric
  - Holet_branch_id: fk, numeric
  - Name: alphanumeric
  - Hours of operation: time
- Room Service (Weak)
  - Room_service_id: pk, numeric
  - Room_id: fk, numeric
  - Price: numeric
- Billing (Weak)
  - Bill_id: pk, numeric
  - Minibar_id: fk, numeric
  - Room_service_id: fk, numeric
  - Total: numeric
- Minibar (Weak)
  - Minibar_id: pk, numeric
  - Room_id: fk, numeric
- Event (Weak)
  - Event_id: pk, numeric
  - Number of guests: numeric
  - Hotel_branch_id: fk, numeric
  - Date: date
- Hotel Review (Weak)
  - Hotel_review_id: pk, numeric
  - Reservation_id: fk, numeric
  - Hotel_branch_id: fk, numeric
  - Date: date
  - Rating: numeric
- Predictive pricing (Weak)
  - Pricing_id: pk, numeric
  - Room_id: fk, numeric

    - Price: numeric
- Maintenance Request (Weak)
    - Maintenance_request_id: pk, numeric
    - Room_id: fk, numeric
    - Employee_id (maintenance): fk, numeric
    - Priority level: numeric
- Inventory (Weak):
    - Item_id: pk, numeric
    - Stock: numeric


Staff Schedule
schedule_id INT
scheduled_time DATETIME
Hotel Review
review_id INT
rating SMALLINT
date DATE
Predictive Pricing
pricing_id INT
room_id INT
price DECIMAL(2,15)
min_price DECIMAL(2,15)
Review - Association
review_id INT
reservation_id INT
hotel_branch_id INT
Maintenance Request
request_id INT
priority_level INT
room_id INT
employee_id INT
Reservation
reservation_id INT
check_in_date DATE
check_out_date DATE
guest_id INT
Bill
bill_id INT
total INT
reservation_id INT
minibar_id INT
Room Service
service_id INT
price INT
bill_id INT
room_id INT
Linen Service
linen_service_id INT
phone_number CHAR(15)
Guest
guest_id INT
first_name VARCHAR(50)
last_name VARCHAR(50)
full_name VARCHAR(100)
birth_date DATE
phone_number VARCHAR(15)
rewards_program_id INT
Rewards Program
program_id INT
tier VARCHAR(45)
Is Scheduled For
index INT
schedule_id INT
employee_id INT
Housekeeper
employee_id INT
room_id INT
Room
room_id INT
reservation_id INT
hotel_branch_id INT
Minibar
minibar_id INT
room_id INT
inventory_id INT
Supply
supply_id INT
linen_service_id INT
hotel_branch_id INT
Event
event_id INT
date DATE
num_of_attendees INT
hotel_branch_id INT
Hotel Branch
hotel_branch_id INT
location VARCHAR(50)
Valet
employee_id INT
parking_lot_id INT
Has Role
index INT
employee_id INT
role_id INT
Employee Role
role_id INT
name VARCHAR(45)
salary INT
Hotel Inventory
hotel_branch_id INT
Inventory_inventory_id INT
Inventory
inventory_id INT
item_name VARCHAR(45)
quantity INT
Maintenance
employee_id INT
maintenance_request_id INT
Employee
employee_id INT
name VARCHAR(50)
hotel_branch_id INT
supervisor_id INT
hours_worked INT
Restaurant Inventory
restaurant_id INT
Inventory_inventory_id INT
Restaurant
restaurant_id INT
hotel_branch_id INT
name VARCHAR(45)
hours_of_operation DA...
Has Amenity
index INT
amenity_id INT
room_id INT
Parking Lot
parking_lot_id INT
number of spaces INT
hotel_branch_id INT
Has Amenity
index INT
amenity_id INT
hotel_branch_id INT
Hotel Amenity
Amenity_amenity_id I...
Amenity
amenity_id INT
name VARCHAR(45)
Room Amenity
Amenity_amenity_id I...
Indexes

Normalization Techniques

1. Originally in the ERD, I had the Staff Schedule table having a 1 - 1 relationship with the Hotel Branch table. I laid it out this way originally because I wanted each hotel to have its own schedule. However, with my way of thinking, there were going to be columns for each time slot ready to be filled by employees. This violates the first normal form since it introduces repeating columns, rather than sorting them in rows. In order to fix this, I linked the Staff Schedule table to the Employee table. This allows the Staff Schedule table to show time slots for employees in rows rather than columns, using employee_id as a foreign key to show which employee will fill that timeslot. This allows the Staff Schedule table to adhere to the first normal form.
2. Originally, I had the Room Service table connected to the Room table. I thought this would make sense since there has to be a room to order room service. However, It was better to have the relationship go to the Bill table since each room service is something that is billed. This was done to enforce the third normal form since the original way this was implemented would've led to a transitive dependency (Room Service -> Room -> Bill).
3. To ensure the Hotel Review table adheres to the second normal form, I had to split it into 2 tables. This had to be done since the primary keys for the original table were hotel_review_id, reservation_id (to show which reservation the review was linked to), and hotel_branch_id (to show which hotel branch the review belonged to). Since these are all keys, the actual review rating needs to depend on all of these keys in order for the table to be in the second normal form. This isn't true since the rating doesn't depend on hotel_branch_id. In order to fix this, we need to separate this into 2 tables. One table with the keys (hotel_review_id, reservation_id, hotel_branch_id) and a second table with actual data for the rating (hotel_review_id, review, date).