DALHOUSIE UNIVERSITY

Data Management, Warehousing and Analytics

Jaskaran Singh

MACS

B00948857

js356337@dal.ca

Assignment 1

PROBLEM 1

Gitlab Repository link:

https://git.cs.dal.ca/singh16/csci5408_s23_b00948857_jaskaran_singh.git

Problem 1

Step 1: Resources

S.No.	Name	URL	Description
1	BBCanada	https://m.bbcanada.com/	Information about entities involved.
2	BNBFinder	https://www.bnbfinder.com/	Services offered in bed and breakfast hotels.
3	Vrbo	https://www.vrbo.com/travel/campaign/bedandbreakfast-vrbo	Information about entities and attributes involved.
4	Breakfast & Bed	https://breakfastandabed.com/	Payments
5	White Stone Marketing	https://www.whitestonemarketing.com/best-bed-and-breakfast-websites	Importance of feedback from guests in hotels
6	NS Bed & Breakfast	https://www.nsbedandbreakfast.com/	Local services and facilities provided in NS region.
7	Expedia	https://www.expedia.ca/aa/BedAndBreakfast	Related to Meals offered.
8	Booking.com	https://www.booking.com/bed-and-breakfast/index.en-gb.html	Information about entities involved.

Step 2: Entities

Following are the entities related to the problem statement:

1. **Hotel**
   - This entity contains details of the hotel like name, address, phone number, etc.
   - It is a strong entity as it has hotel id as primary key to uniquely identify itself.
2. **Room**
   - It contains detail of the room in the hotel like room number, room type, availability.
   - As it depends on the hotel for its identification, it is a weak entity.
3. **Staff**
   - It contains detail of the employees working for the hotel.
   - Since every employee can be uniquely identified by itself based on employee id, it is a strong entity.

4. **Meal**
    - This entity contains information of the meals provided by the hotel.
    - It is a strong entity as every meal can be uniquely identified by meal id.
5. **Amenities**
    - It contains details of the amenities provided by the hotel.
    - It is a strong entity as it can be uniquely identified based on amenity id.
6. **Booking**
    - This entity contains detail of the booking.
    - It is a strong entity as every booking has booking id as primary key for its identification.
7. **Coupons**
    - This entity contains detail of the coupons provided by the hotels for facilities and services around the hotel.
    - It is a strong entity as it has coupon id for its unique identification.
8. **Payment**
    - It contains detail of the payment for the booking done by the customer.
    - It is a strong entity as it has payment id as primary key for its identification.
9. **Customer**
    - It contains detail of the customer who is booking the hotel.
    - It is a strong entity as it has customer id as primary key for its unique identification.
10. **Feedback**
    - This entity contains detail of the review and rating provided by the customer for the hotel.
    - It is a weak entity as it depends on customer and hotel for its unique identification.

Step 3 & 4: <u>Conceptual ERD using Chen's model</u>

- Following ERD has been prepared using https://draw.io/ .
- All entities and relationships between them have been mentioned with cardinality.
- Entities are mentioned using light pink color.
- Attributes are mentioned in blue color.
- Relationships are mentioned in green color with cardinality.
- Primary key, weak entities, partial key, derived attribute, strong participation, etc. have been mentioned in the ERD using standard symbols and shapes for Chen model.
- Also, no fan trap or chasm trap has been found in the following conceptual data model.
- Hence, it is the final conceptual model with no design issue.

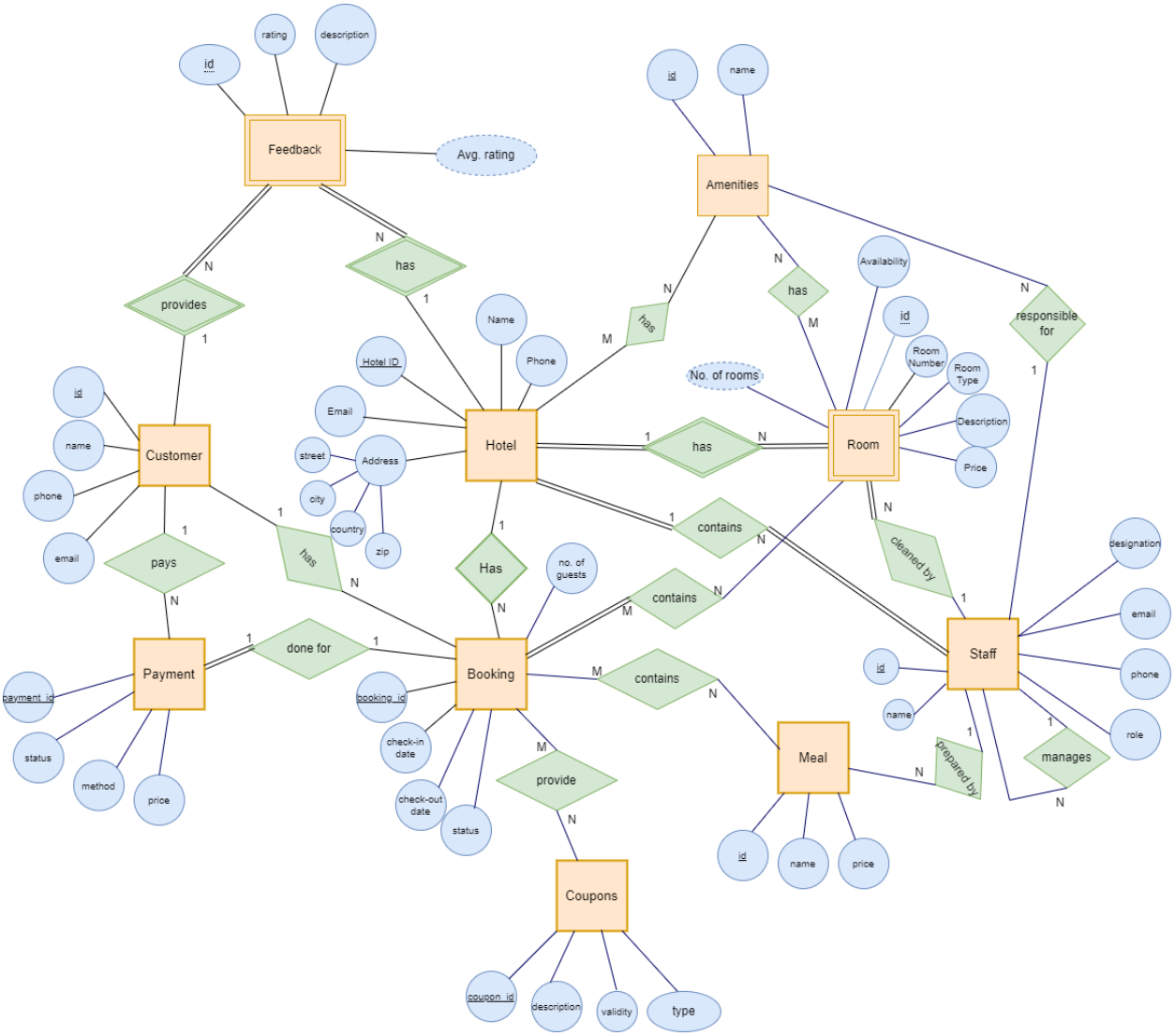

Figure 1 – ERD using Chen model

Step 5 & 6: Logical model and normalization

Now, considering each entity of the data model as a table, now amenities table can be normalized as below:

Amenties(ID, name, hotels(), rooms()) → 1NF → Amenity (ID, name)

Hotel(ID, name,…,amenities()) → Amenties_in_hotel (amenity ID, hotel_Id)

Room(ID, hotelID,..,amenities()) → Amenties_in_room (amenity ID, hotel_Id, room_id)

→ Hotel(ID, name,…)

→ Room(ID, hotelID,..)

Since Room and Hotel tables have n:m cardinality relationships with amenities, it has been normalized into the above 5 tables.

Similarly, normalization has been performed in below tables:

Meals(ID , name, booking()) → 1NF → Meal (ID, name, price)

Booking(ID,.., Meals()) → Meals_in_booking (booking ID, Meal ID)

→ Booking(ID,..)

Coupons(ID , description, validity, type, bookings()) → 1NF → Coupon (ID, description, validity, type)

Booking(ID,.., Coupons()) → Coupons_in_booking (booking ID,coupon ID)

→ Booking(ID,…..)

Rooms(ID , hotel ID,..., bookings()) → 1NF → Rooms (ID, hotel ID,…)

Booking(ID,.., Rooms()) → Rooms_in_booking (booking ID, ID, hotel ID)

→ Booking(ID,…..)

- After performing normalization, below are the final tables with attributes and their dependencies.

Table 1: Hotel

Hotel
Hotel ID (PK)
Name Phone Email Street name City Country Zip

1. Hotel ID is the primary key
2. Name, phone, email, street name, city, country and zip are dependent on hotel ID.

Table 2: Room

Room
Hotel ID, Room ID (PK)
Room number Availability Room type Description Price Staff_ID(FK)

1. Hotel ID and Room ID both make primary key for the room table.
2. Room number, availability, room type, description and price are dependent on the primary key.
3. Staff_ID is the foreign key which reference staff table.

Table 3: Amenity

Amenity
ID (PK)
Name Staff_ID(FK)

1. ID is the primary key for the meal table.
2. Staff_ID is the foreign key which reference staff table.

Table 4: Amenties_in_hotel

Amenties_in_hotel
Amenity_ID, hotel_Id (PK)

1. Amenity ID and hotel Id makes the composite key for the table.

Table 5: Amenties_in_room

Amenties_in_room
Amenity_ID, hotel_ID, room_ID (PK)

1. Amenity ID, room_ID and hotel_ID makes the composite key for the table.

Table 6: Staff

Staff
Staff ID (PK)
Designation Email Phone Role Manager ID Hotel_ID(FK)

1. Staff ID is the primary key for the table.
2. Designation, email, phone, role and price are dependent on the primary key.
2. Manager ID attribute has been defined for self-relationship of staff.
3. Hotel_ID is the foreign key which references Hotel table.

Table 7: Meal

Meal
ID (PK)
Name Price Staff_ID(FK)

1. ID is the primary key for the meal table.
2. Name and price are dependent on the primary key.
3. Staff_ID is the foreign key which reference staff table.

Table 8: Meals_in_booking

Meals_in_booking
booking ID, meal ID (PK)

1. Booking ID and meal_ID makes the composite key for the table.

Table 9: Booking

Booking
ID (PK)
Check_in_date Check_out_date No_of_guests Status Hotel_ID (FK) Room_ID (FK) Customer_ID (FK)

1. ID is the primary key for the booking table.
2. Check_in_date, check_out_date, status and no_of_guests are dependent on the primary key.
3. Hotel_ID is the foreign key which reference hotel table, similarly Room_ID references room table and customer_id references customer table.

Table 10: Coupon

Coupon
ID (PK)
description validity type

1. ID is the primary key for the coupon table.
2. Description, validity and type are dependent on the primary key.

Table 11: Coupons_in_booking

Coupons_in_booking
Booking ID, Coupon ID (PK)

1. Booking ID and coupon ID makes the composite key for the table.

Table 12: Payment

Payment
ID (PK)
Method Price Status Booking_ID(FK) Customer_ID (FK)

1. ID is the primary key for the payment table.
2. Method, price and status are dependent on the primary key.
3. Booking_ID and customer_ID are foreign keys referencing booking and customer table respectively.

Table 13: Customer

Customer
ID (PK)
Name Phone Email

1. ID is the primary key for the customer table.

2. Name, phone and email are dependent on the primary key.

Table 14: Feedback

Feedback
ID, Customer_id, Hotel_id (PK)
Rating Description

1. ID, customer_id and hotel_id makes the primary key for the feedback table.
2. Rating and description are dependent on the primary key.

Table 15: Rooms_in_booking

Rooms_in_booking
ID, Room ID, Hotel_ID (PK)

1. ID, Room_ID and Hotel_ID makes the primary key for the table.

Step 7 & 8: Physical model

'BedBreakfast' database has been created and DDL queries have been attached in the 'bed_breakfast_queries' file.

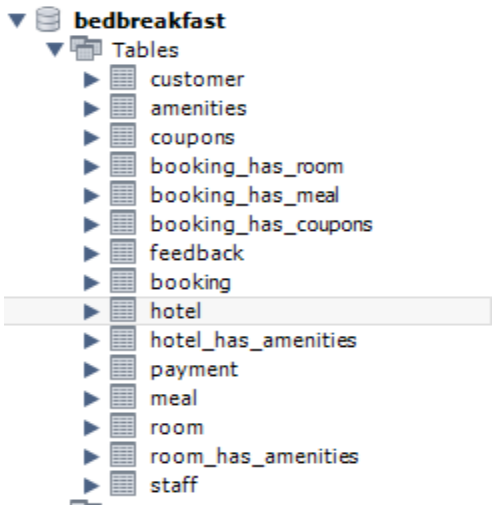

Figure 2: Tables created after executing queries in bed_breakfast_queries file.

References:

- **Draw.io, ERD Tool**
  Link: https://draw.io/

- **ERD with draw.io**
  Link: https://drawio-app.com/blog/entity-relationship-diagrams-with-draw-io/

- **Chen and Crow's Foot notation, Gleek**
  Link: https://www.gleek.io/er-diagrams