CSCI 5408

DATA MANAGEMENT AND WAREHOUSING

Database Builder Project

Group 28

Group Members:

Aditya Maheshbhai Purohit - B00952865

Aditya Pattani - B00966361

GitLab Assignment Links:

https://git.cs.dal.ca/apurohit/CSCI5408_F23_B00952865_AdityaMaheshbhai_Purohit/-/tree/main/DB%20Builder

https://git.cs.dal.ca/pattani/csci5408_f23_b00966361_aditya_pattani/-/tree/main/DB%20Builder

Table of Contents

1. Background Research

As a part of our initial research, we have identified several hotel websites [1] [2] [3] [4] [5] and have designed our mini world based on them. The information gathered and the entities identified from each of these websites and knowledge bases have been added to the below Table 1:

Table 1: Information gathered from various hotel related source [1] [2] [3] [4] [5].

Sr No.	Source	URL	Information	Comments
1	Radisson Blu Hotels	https://www.radissonhotels.com/en-us/brand/radisson-blu	**Entities**: - Property - Room - RoomType - Amenities - Guest - Reservation - Deal - MeetingSpace - Invoice - Payment Detail - Membership - Employee - ServiceRequest	- Owns multiple hotel **properties**. - Each property has multiple **rooms** and has a type (superior, deluxe). - Each room type has some max capacity, bed type and room amenities attached to it. - **Guests** make **reservations** of room. - There are **Deals** available. - Can also book **meeting space** instead of rooms. - Reservation will have **invoice** and **payment details**. - There are different **memberships** available and various benefits linked to it. - There are **employees** in the hotel. - There can be multiple complaints/service requests from

				specific rooms like fix light, TV, etc.
2	Zen Hotels	https://blog.zenhotels.com/where-to-store-valuables-in-a-hotel-or-hostel/	**Entities:** - Locker	- A hotel can have multiple lockers. These can be in the rooms or can be in general.
3	Schindler	https://www.schindler.com/en/industries/hotel.html	**Entities:** - Elevator	- A hotel can have multiple elevators with different configurations based upon the requirements and budget of the hotel
4	The Barrington Hotel	https://www.thebarringtonhotel.ca/	**Entities**: - Room - Amenities - Guest - Reservation - Deal - Membership - Reviews - ParkingLot - ParkingSpot - Event - ItemStock	- There are different **rooms** with different **facilities**. - **Guests** can book a room. - There are various **deals** & **membership** available. - Guests can leave **reviews** about their experience. - It has multiple parking lots. - Each parking lot can have multiple parking spots. - There can be multiple **events** organized by the hotel
6	Taj Hotels	https://www.tajhotels.com/en-in/restaurants/	**Entities:** - Restaurant	- There can be multiple **restaurants** in a single hotel.

We have created another table to describe our reasoning behind the existence of each of these entities:

Table 2: Description of entities along with their source websites [1] [2] [3] [4] [5].

Sr. No.	Entity	Source	Why is this an entity?
1	Property	https://www.radissonhotels.com/en-us/brand/radisson-blu	Property describes an individual hotel venue where multiple rooms and amenities are available. A hotel company can have multiple properties.
2	Room	https://www.radissonhotels.com/en-us/brand/radisson-blu	Room exists in a hotel. There can be many rooms, but each only rented by an individual or a group of people.
3	RoomType	https://www.radissonhotels.com/en-us/brand/radisson-blu	Room Type describes the class of the room. It can be a suite, penthouse, studio, standard, executive suite, etc.
4	Amenities	https://www.radissonhotels.com/en-us/brand/radisson-blu	This entity includes all the amenities provided by the hotel. Included but not limited to heating, hot water, TV, Air conditioning, swimming pool, spa etc.
5	Guest	https://www.radissonhotels.com/en-us/brand/radisson-blu	An individual who will be looking at available rooms and booking a room at a hotel if they find it suitable.
6	Reservation	https://www.radissonhotels.com/en-us/brand/radisson-blu	A reservation will be made by a guest and will include the check-in and check-out times.
7	Deal	https://www.radissonhotels.com/en-us/brand/radisson-blu	Hotel can have multiple deals. This deal can be different

			for its different properties.
8	MeetingSpace	https://www.radissonhotels.com/en-us/brand/radisson-blu	There can be various meeting spaces in a single hotel property.
9	Invoice	https://www.radissonhotels.com/en-us/brand/radisson-blu	A reservation for a hotel room has an invoice.
10	Payment Detail	https://www.radissonhotels.com/en-us/brand/radisson-blu	Includes how a guest makes payment for the room or any of the services availed by them.
11	Membership	https://www.radissonhotels.com/en-us/brand/radisson-blu	This entity contains records of guests who have subscribed to any of the membership programs offered by the hotel.
12	Employee	https://www.radissonhotels.com/en-us/brand/radisson-blu	An employee that works at a hotel. This entity will contain details such as department, salary, address, phone number etc.
13	Review	https://www.thebarringtonhotel.ca/	Reviews of the hotel. Only a guest can leave a review for the hotel that they booked.
14	ParkingLot	https://www.thebarringtonhotel.ca/	A hotel can have multiple parking lots for their guest.
15	ParkingSpot (Weak)	https://www.thebarringtonhotel.ca/	Each parking lot can have various parking spots. It is a weak entity because it cannot exist without a parking lot, and we can't identify a parking spot without knowing which lot it belongs to.

16	Event (Weak)	https://www.thebarringtonhotel.ca/	Contains a list of events and their schedules hosted at the hotel. It is a weak entity because it cannot exist without a Hotel and we can't identify any event without knowing the hotel property.
17	ItemStock	https://www.thebarringtonhotel.ca/	Contains the inventory details of the hotel.
18	Restaurant	https://www.tajhotels.com/en-in/restaurants/	Contains all the restaurant chains that exist in the hotel premises.
19	Elevator	https://www.schindler.com/en/industries/hotel.html	A single hotel property can have multiple elevators.
20	Locker	https://blog.zenhotels.com/where-to-store-valuables-in-a-hotel-or-hostel/	This entity contains details for all the locker services provided by the hotel, be it spa / gym lockers or storage lockers.
21	ServiceRequest	https://www.radissonhotels.com/en-us/brand/radisson-blu	Contains service requests raised by the guests. This can include air conditioning servicing, electronic device servicing. A service request is associated with a room / service and a guest.

Summary

As per our background research of various sources [1] [2] [3] [4] [5], we observed that there are some hotels that operate individually but some have a hotel chain, with multiple properties under that single hotel brand. We decided to keep the **hotel-chain** as our mini-world for the data modelling. A hotel can have multiple rooms of different types. These rooms can be reserved by

the guests using various payment methods and can also use deals if available. They also get an invoice for their payment. Some hotels also offer subscriptions or memberships to the guests so that they can get more benefits if they are frequent visitors.

Hotels are more than just rooms. It has elevators of various type and also have in-room and hotel-level amenities like swimming pool and gym. The hotel may also operate various restaurants under it. There are employees that work for a hotel property and the restaurants of that hotel.

A guest can leave reviews for their experience. They may also raise service-requests during their stay to get support from the hotel staff.

A hotel also usually has 1 or more parking lots, and each lot has multiple parking slots. The hotel also may have in-room or generic lockers for storing valuable items of guests. Moreover, these hotels also organize various events at their restaurants meeting spaces or halls.

Apart from the entities identified above we have also identified the relationships as shown below [1] [2] [3] [4] [5]:

1. A Property **has** multiple deals for their guests.
2. A Property **has** multiple room types available and rooms, and each room **has a** room type.
3. A guest **makes** a reservation which **is for** a room.
4. A guest also **writes** a review.
5. Guest also **subscribes** to membership.
6. A guest also **raises** service request which **is for** their room.
7. Each reservation **has an** invoice, and that invoice **has a** payment method.
8. A Property **has** elevators for convenience.
9. A Property **has** various meeting space for various events.
10. A Property **hosts** multiple events.
11. A Property **has** various amenities like swimming pool & gym.
12. A Property also **keeps** their item stocks.
13. A Property **has** restaurants for their guests.
14. Property also **contains** parking lot which further **has** parking slots.
15. Multiple employees **work at** a property.
16. A single Property **has** multiple lockers.

2. Initial Conceptual Model

Using the information we gathered in the previous step i.e. background research, we have designed an initial version of the conceptual model following Chen notation [6] and drawn using the draw.io [7] tool.

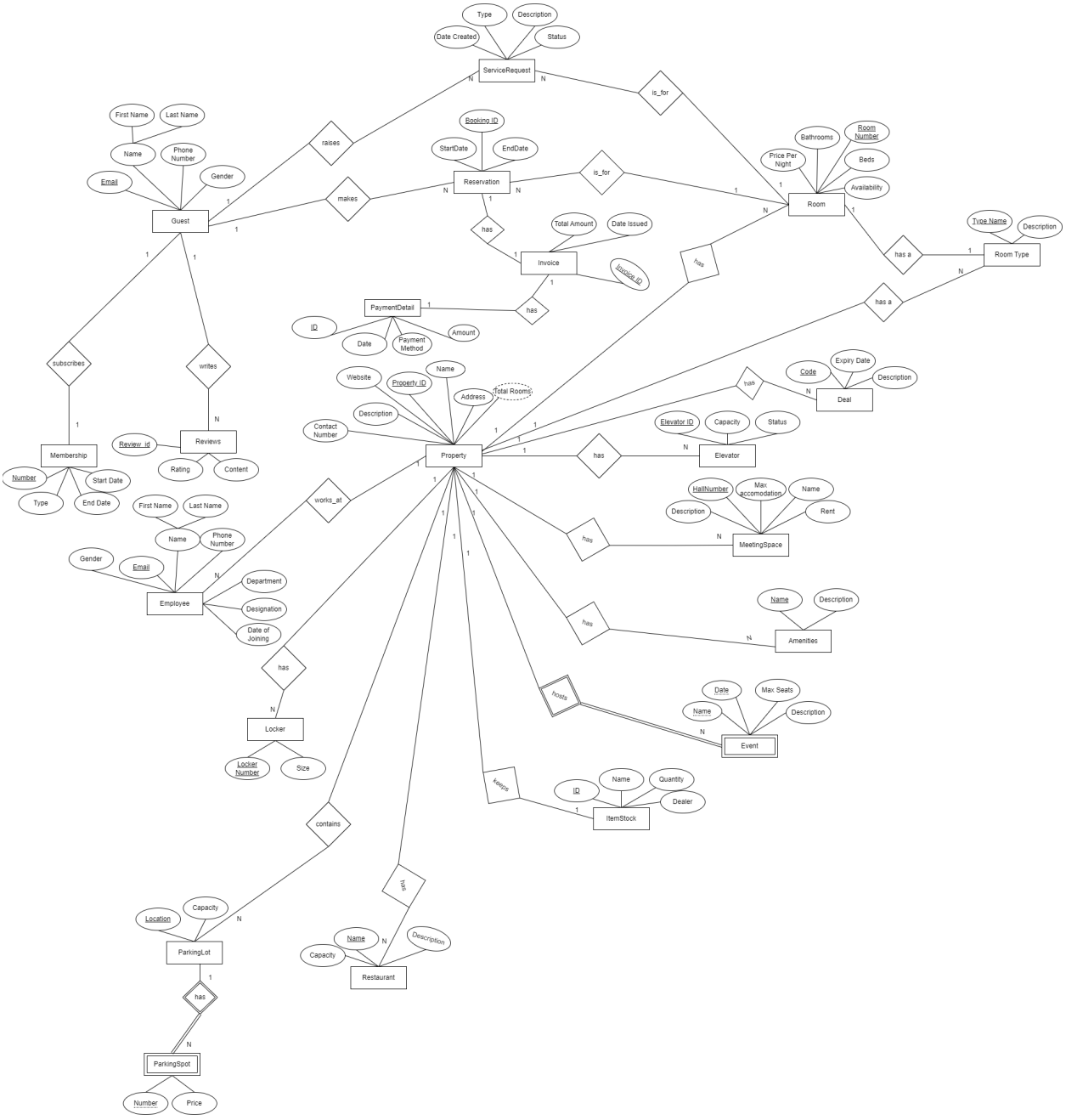

Figure 1: Initial Conceptual Model of Hotel [6] [7].

3. Identifying Design Issues

Historical Data: For our representation of the entities related to hotels, we will be accommodating a large number of employees. These employees can have some salary at one point of time but this can keep on changing as time goes.

If we don't store this data, we may not be able to get the idea of what the salary was for an employee few years back. To store historical data, we can create a separate entity - "employee_salary_hist" to store all the employee salary updates. The historical data will be stored in the employee_salary_hist entity which will be related to the Employee entity with an N:1 relationship using the Employee ID attribute.

Fan Traps: Initial version of the ER diagram suggests that a Guest would be able to make a Reservation at a property and would also be able to leave in separate Reviews. We would be able to identify a Reservation using a Guest and a Review using a Guest as well. Reservations have a "made-by" relationship with the Guest entity with a cardinality of N:1 and Review has a "written_by" relationship with the Guest entity with a cardinality of M:1. However, using only the Review entity, we will not be able to identify the Reservation it is referring to. This will cause a Fan trap in our design. Refer Figure 1 for more information.

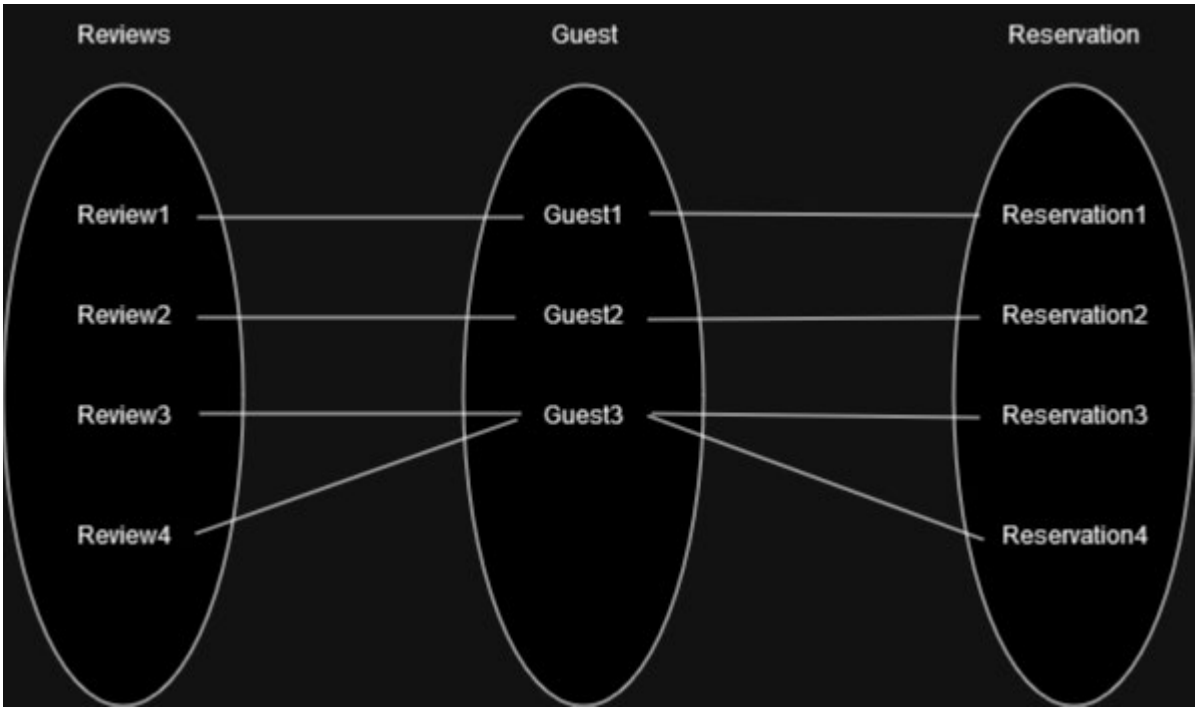

Figure 2: Identified a fan trap between the Guest, Reviews and Reservation entities [7].

In order to mitigate the fan trap, we can update the relationships between Guest → Reservation and Guest → Reviews. In the updated model, a Guest will be able to write a Review and the Review in turn will exist for a particular Reservation (See figure 2).

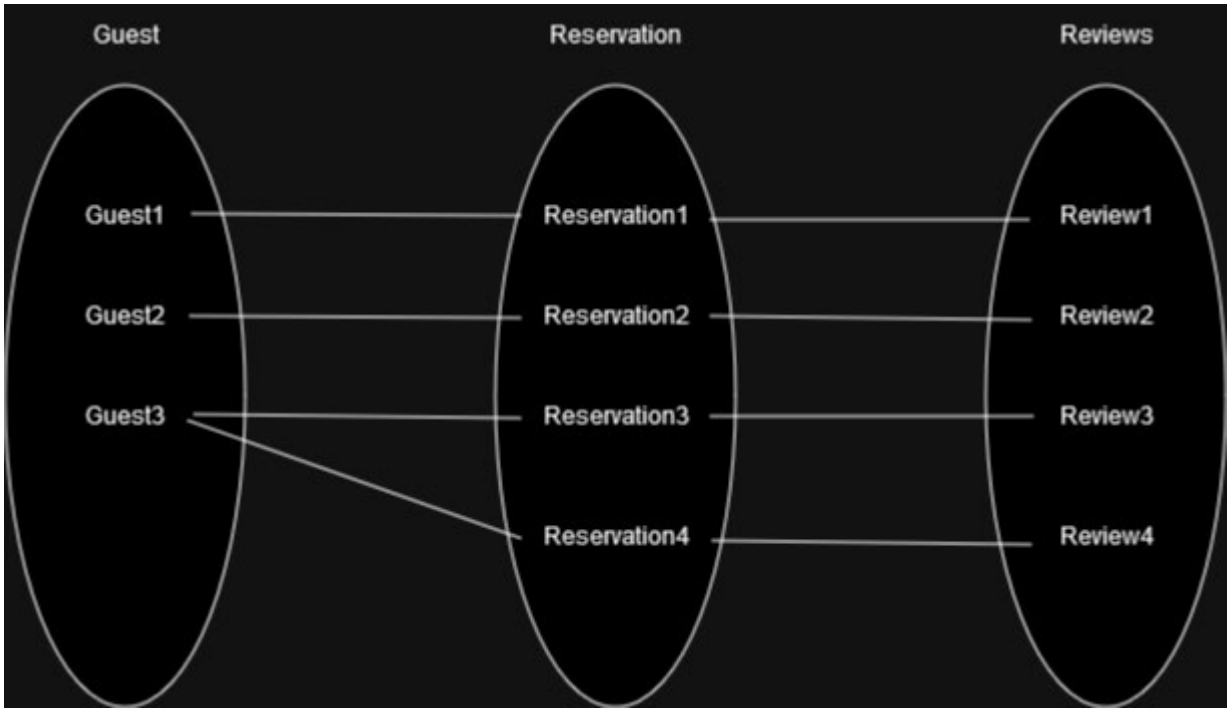

Figure 3: Updated relationship between the Guest, Reviews and Reservation entities to fix the fan trap [7].

The updated data model will have a 1:N relationship from Guest with the Reservation entity. Consequently, the relationship between Guest and Reviews will be replaced by a 1:1 relationship between the Reservation entity and the Review entity.

Chasm Traps: While creating the ER diagram, we did not notice any Chasm traps in our model. This is due to the fact that all the relationships exists and thus there are no missing paths.

Moreover, this trap may occur after the resolution of Fan Trap, but in our ER it's not the case. We don't have any possibility of guest reviewing without reservation, so such extra relationship between guest and review directly, is not needed in the Figure 3 anymore.

4. Final Model (ERD)

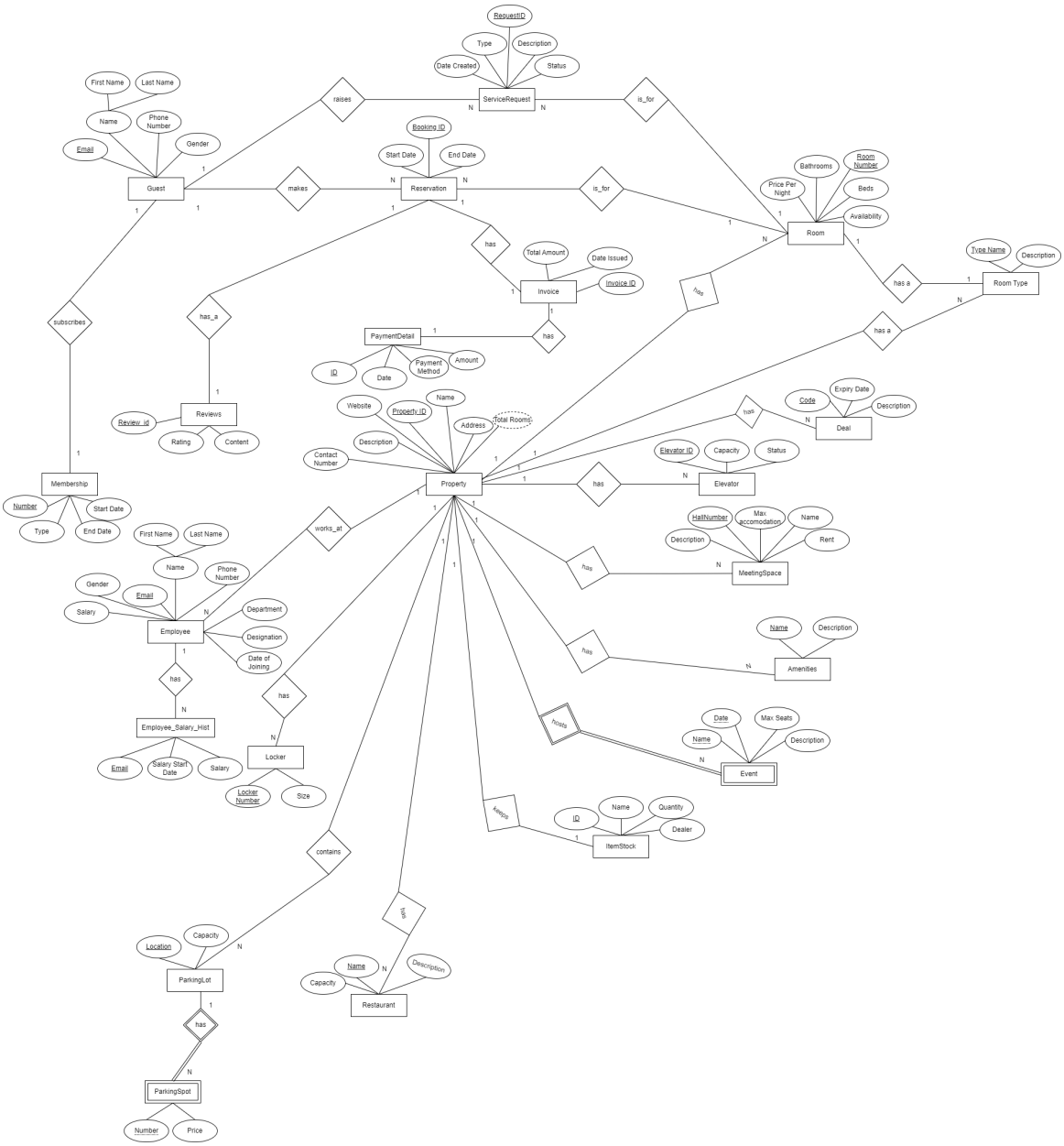

Figure 4: Final ERD after resolving design issues [7].

5. Logical Phase

Identifying Partial and Transitive dependencies

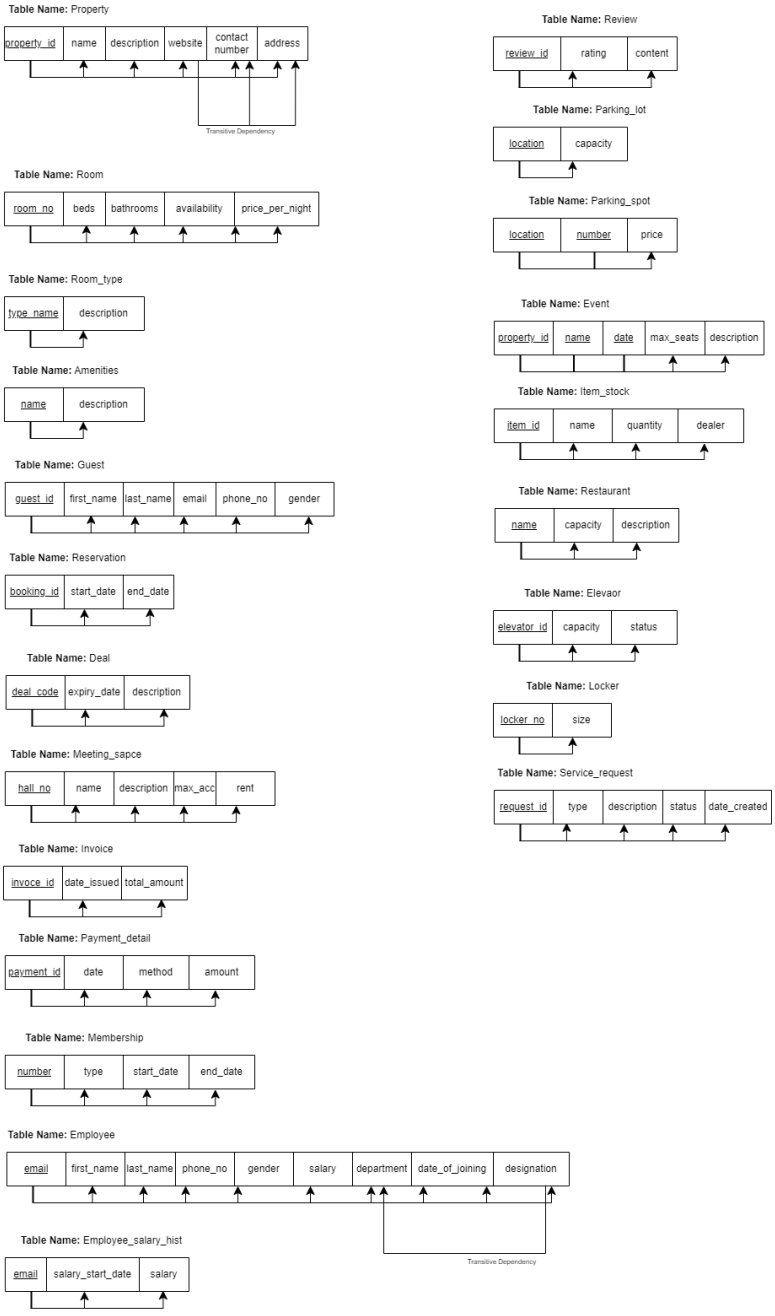

Figure 5: Partial & Transitive dependencies [7].

1NF

All the tables are in 1NF already has they don't have any columns which store multiple values. (Multi-value or composite attribute.)

2NF

All the tables are already in 2NF because its already in 1NF and there are no partial dependencies. There are no partial dependencies because there are no composite primary keys in the strong entities. In weak entities there are composite primary keys but there is again no partial dependency as all prime attributes are needed to uniquely identify any non-prime attribute.

6. DDL (Data Definition Language)

```
-----------------------------------------------------
-- CSCI 5408 - Data Management, Warehousing & Analytics
-- Authors: Aditya Pattani (B00966361)
--          Aditya Maheshbhai Purohit (B00952865)
--
-- This code contains all the entities and relationships
-- described in the DB Builder Report
-----------------------------------------------------

-- Dropping and Creating the database
DROP DATABASE IF EXISTS CSCI5408_GROUP28;

CREATE DATABASE IF NOT EXISTS CSCI5408_GROUP28;

USE CSCI5408_GROUP28;

-- Dropping and Creating the tables
DROP TABLE IF EXISTS PROPERTY;

CREATE TABLE PROPERTY (
         PROPERTY_ID    INT PRIMARY KEY NOT NULL
        ,NAME           VARCHAR(100)    NOT NULL
        ,DESCRIPTION    VARCHAR(255)    NOT NULL
        ,WEBSITE        VARCHAR(255)    NOT NULL
        ,CONTACT_NUMBER VARCHAR(14)     NOT NULL
        ,ADDRESS        VARCHAR(255)    NOT NULL
);

DROP TABLE IF EXISTS ROOM_TYPE;
```

```
CREATE TABLE ROOM_TYPE (
         TYPE_NAME      VARCHAR(50) PRIMARY KEY NOT NULL    DEFAULT
"Single"    UNIQUE
        ,DESCRIPTION    VARCHAR(255)            NOT NULL
);

DROP TABLE IF EXISTS MEMBERSHIP;

CREATE TABLE MEMBERSHIP (
         NUMBER         INT PRIMARY KEY NOT NULL
        ,TYPE           VARCHAR(50)     NOT NULL    DEFAULT "Solo"
        ,START_DATE     DATE            NOT NULL
        ,END_DATE       DATE
);

DROP TABLE IF EXISTS ROOM;

CREATE TABLE ROOM (
         ROOM_NO            INT PRIMARY KEY NOT NULL
        ,BEDS               INT             NOT NULL
        ,BATHROOMS          INT             NOT NULL
        ,AVAILABILITY       SMALLINT        NOT NULL
        ,PRICE_PER_NIGHT    INT             NOT NULL
        ,PROPERTY_ID        INT             NOT NULL
        ,ROOM_TYPE          VARCHAR(50)     NOT NULL
        ,FOREIGN KEY (PROPERTY_ID)
            REFERENCES PROPERTY(PROPERTY_ID)
        ,FOREIGN KEY (ROOM_TYPE)
            REFERENCES ROOM_TYPE(TYPE_NAME)
);

DROP TABLE IF EXISTS AMENITIES;

CREATE TABLE AMENITIES (
         NAME           VARCHAR(100)    PRIMARY KEY NOT NULL
        ,DESCRIPTION    VARCHAR(255)                NOT NULL
        ,PROPERTY_ID    INT                         NOT NULL
        ,FOREIGN KEY (PROPERTY_ID)
            REFERENCES PROPERTY(PROPERTY_ID)
);

DROP TABLE IF EXISTS GUEST;

CREATE TABLE GUEST (
         GUEST_ID           INT PRIMARY KEY NOT NULL
```

```
        ,FIRST_NAME         VARCHAR(255)    NOT NULL
        ,LAST_NAME          VARCHAR(255)    NOT NULL
        ,EMAIL              VARCHAR(100)    NOT NULL    UNIQUE
        ,PHONE_NUMBER       VARCHAR(14)     NOT NULL    UNIQUE
        ,GENDER             SMALLINT        NOT NULL
        ,MEMBERSHIP_NUMBER  INT
        ,FOREIGN KEY (MEMBERSHIP_NUMBER)
            REFERENCES MEMBERSHIP(NUMBER)
);

DROP TABLE IF EXISTS RESERVATION;

CREATE TABLE RESERVATION (
         BOOKING_ID     INT PRIMARY KEY NOT NULL
        ,START_DATE     DATE            NOT NULL
        ,END_DATE       DATE
        ,GUEST_ID       INT             NOT NULL
        ,ROOM_NO        INT             NOT NULL
        ,FOREIGN KEY (GUEST_ID)
            REFERENCES GUEST(GUEST_ID)
        ,FOREIGN KEY (ROOM_NO)
            REFERENCES ROOM(ROOM_NO)
);

DROP TABLE IF EXISTS DEAL;

CREATE TABLE DEAL (
         DEAL_CODE      VARCHAR(50) PRIMARY KEY NOT NULL
        ,EXPIRY_DATE    DATE
        ,DESCRIPTION    VARCHAR(255)            NOT NULL
        ,PROPERTY_ID    INT                     NOT NULL
        ,FOREIGN KEY (PROPERTY_ID)
            REFERENCES PROPERTY(PROPERTY_ID)
);

DROP TABLE IF EXISTS MEETING_SPACE;

CREATE TABLE MEETING_SPACE (
         HALL_NO            INT PRIMARY KEY NOT NULL
        ,NAME               VARCHAR(20)
        ,DESCRIPTION        VARCHAR(255)    NOT NULL
        ,MAX_ACCOMODATION   INT             NOT NULL
        ,RENT               INT             NOT NULL
        ,PROPERTY_ID        INT             NOT NULL
        ,FOREIGN KEY (PROPERTY_ID)
```

```
            REFERENCES PROPERTY(PROPERTY_ID)
);

DROP TABLE IF EXISTS PAYMENT_DETAIL;

CREATE TABLE PAYMENT_DETAIL (
         PAYMENT_ID     INT PRIMARY KEY NOT NULL
        ,DATE           DATE            NOT NULL
        ,METHOD         VARCHAR(10)     NOT NULL    -- Can make this an INT field
if we have a Payment_Method table
        ,AMOUNT         INT             NOT NULL
);

DROP TABLE IF EXISTS INVOICE;

CREATE TABLE INVOICE (
         INVOICE_ID     INT PRIMARY KEY NOT NULL
        ,DATE_ISSUED    DATE            NOT NULL
        ,TOTAL_AMOUNT   INT             NOT NULL
        ,RESERVATION_ID INT             NOT NULL
        ,PAYMENT_DETAIL INT             NOT NULL
        ,FOREIGN KEY (RESERVATION_ID)
            REFERENCES RESERVATION(BOOKING_ID)
        ,FOREIGN KEY (PAYMENT_DETAIL)
            REFERENCES PAYMENT_DETAIL(PAYMENT_ID)
);

DROP TABLE IF EXISTS EMPLOYEE;

CREATE TABLE EMPLOYEE (
         EMAIL              VARCHAR(255)    PRIMARY KEY NOT NULL
        ,FIRST_NAME         VARCHAR(255)                NOT NULL
        ,LAST_NAME          VARCHAR(255)                NOT NULL
        ,PHONE_NO           VARCHAR(14)                 NOT NULL    UNIQUE
        ,GENDER             SMALLINT                    NOT NULL
        ,SALARY             INT                         NOT NULL
        ,DEPARTMENT         VARCHAR(50)                 NOT NULL
        ,DATE_OF_JOINING    DATE                        NOT NULL
        ,DESIGNATION        VARCHAR(100)                NOT NULL
        ,WORKS_AT           INT                         NOT NULL
        ,FOREIGN KEY (WORKS_AT)
            REFERENCES PROPERTY(PROPERTY_ID)
);

DROP TABLE IF EXISTS EMPLOYEE_SALARY_HIST;
```

```
CREATE TABLE EMPLOYEE_SALARY_HIST (
         EMAIL              VARCHAR(255)    PRIMARY KEY NOT NULL
        ,SALARY_START_DATE  DATE                        NOT NULL
        ,SALARY             INT                         NOT NULL
        ,FOREIGN KEY (EMAIL)
            REFERENCES EMPLOYEE(EMAIL)
);

DROP TABLE IF EXISTS REVIEW;

CREATE TABLE REVIEW (
         REVIEW_ID              INT PRIMARY KEY NOT NULL
        ,RATING                 SMALLINT        NOT NULL
        ,CONTENTS               VARCHAR(255)    NOT NULL
        ,WRITTEN_BY             INT             NOT NULL
        ,WRITTEN_FOR_BOOKING    INT             NOT NULL
        ,FOREIGN KEY (WRITTEN_BY)
            REFERENCES GUEST(GUEST_ID)
        ,FOREIGN KEY (WRITTEN_FOR_BOOKING)
            REFERENCES RESERVATION(BOOKING_ID)
);

DROP TABLE IF EXISTS PARKING_LOT;

CREATE TABLE PARKING_LOT (
         LOCATION       VARCHAR(255)    PRIMARY KEY NOT NULL
        ,CAPACITY       INT                         NOT NULL
        ,FOR_PROPERTY   INT                         NOT NULL
        ,FOREIGN KEY (FOR_PROPERTY)
            REFERENCES PROPERTY(PROPERTY_ID)
);

DROP TABLE IF EXISTS PARKING_SPOT;

CREATE TABLE PARKING_SPOT (
         LOCATION       VARCHAR(255)    NOT NULL
        ,NUMBER         INT             NOT NULL
        ,PRICE          INT             NOT NULL
        ,PARKING_LOT    VARCHAR(255)    NOT NULL
        ,PRIMARY KEY (LOCATION, PARKING_LOT)
        ,FOREIGN KEY (PARKING_LOT)
            REFERENCES PARKING_LOT(LOCATION)
);
```

```
DROP TABLE IF EXISTS EVENT;

CREATE TABLE EVENT (
         NAME           VARCHAR(100)    NOT NULL    UNIQUE
        ,DATE           DATE            NOT NULL
        ,MAX_SEATS      INT             NOT NULL
        ,DESCRIPTION    VARCHAR(255)    NOT NULL
        ,PROPERTY_ID    INT             NOT NULL
        ,PRIMARY KEY (NAME, DATE, PROPERTY_ID)
        ,FOREIGN KEY (PROPERTY_ID)
            REFERENCES PROPERTY(PROPERTY_ID)
); -- To add primary Key

DROP TABLE IF EXISTS ITEM_STOCK;

CREATE TABLE ITEM_STOCK (
         ITEM_ID        INT PRIMARY KEY NOT NULL
        ,NAME           VARCHAR(255)    NOT NULL
        ,QUANTITY       INT                         DEFAULT 0
        ,DEALER         VARCHAR(100)    NOT NULL
        ,PROPERTY_ID    INT             NOT NULL
        ,FOREIGN KEY (PROPERTY_ID)
            REFERENCES PROPERTY(PROPERTY_ID)
);

DROP TABLE IF EXISTS RESTAURANT;

CREATE TABLE RESTAURANT (
         NAME           VARCHAR(255)    PRIMARY KEY NOT NULL
        ,CAPACITY       INT                         NOT NULL
        ,DESCRIPTION    VARCHAR(255)                NOT NULL
        ,PROPERTY_ID    INT                         NOT NULL
        ,FOREIGN KEY (PROPERTY_ID)
            REFERENCES PROPERTY(PROPERTY_ID)
);

DROP TABLE IF EXISTS ELEVATOR;

CREATE TABLE ELEVATOR (
         ELEVATOR_ID    INT PRIMARY KEY NOT NULL
        ,CAPACITY       INT             NOT NULL    DEFAULT 0
        ,STATUS         SMALLINT        NOT NULL    DEFAULT 1   -- 1 means
working and 0 means inactive
        ,PROPERTY_ID    INT             NOT NULL
        ,FOREIGN KEY (PROPERTY_ID)
```

```
            REFERENCES PROPERTY(PROPERTY_ID)
);

DROP TABLE IF EXISTS LOCKER;

CREATE TABLE LOCKER (
         LOCKER_NO      INT PRIMARY KEY NOT NULL
        ,SIZE           VARCHAR(10)     NOT NULL
        ,PROPERTY_ID    INT             NOT NULL
        ,FOREIGN KEY (PROPERTY_ID)
            REFERENCES PROPERTY(PROPERTY_ID)
);

DROP TABLE IF EXISTS SERVICE_REQUEST;

CREATE TABLE SERVICE_REQUEST (
         REQUEST_ID     INT PRIMARY KEY NOT NULL
        ,TYPE           VARCHAR(100)    NOT NULL
        ,DESCRIPTION    VARCHAR(255)    NOT NULL
        ,STATUS         SMALLINT        NOT NULL
        ,DATE_CREATED   DATE            NOT NULL
        ,GUEST_ID       INT             NOT NULL
        ,ROOM_NO        INT             NOT NULL
        ,FOREIGN KEY (GUEST_ID)
            REFERENCES GUEST(GUEST_ID)
        ,FOREIGN KEY (ROOM_NO)
            REFERENCES ROOM(ROOM_NO)
);
```

References

[1] "Radisson Blu." [Online]. Available: https://www.radissonhotels.com/en-us/brand/radisson-blu. [Accessed: 02-Nov-2023].

[2] "The Barrington Hotel - Halifax hotels," *The Barrington Hotel*. [Online]. Available: https://www.thebarringtonhotel.ca/. [Accessed: 02-Nov-2023].

[3] "Unique dining," *Taj*. [Online]. Available: https://www.tajhotels.com/en-in/restaurants/. [Accessed: 02-Nov-2023].

[4] "Hotels & resorts," *Manufacturer of elevators, escalators, & moving walkways*. [Online]. Available: https://www.schindler.com/en/industries/hotel.html. [Accessed: 02-Nov-2023].

[5] Z. Blog, "Where to store valuables in a hotel or hostel," *Zenhotels.com*. [Online]. Available: https://blog.zenhotels.com/where-to-store-valuables-in-a-hotel-or-hostel/. [Accessed: 02-Nov-2023].

[6] "Chen notation," *Vertabelo Data Modeler*, 02-Aug-2014. [Online]. Available: https://vertabelo.com/blog/chen-erd-notation/. [Accessed: 02-Nov-2023].

[7] "Flowchart maker & online diagram software," *Diagrams.net*. [Online]. Available: https://app.diagrams.net/. [Accessed: 02-Nov-2023].