**CS 6360.003: Database Design**

**Final Project**

Project Title: Hotels.com

Team-1(hotels.com-1)

Team Members:

1. Soham Milind Kulkarni : SMK190000
2. Sneha Ramachandran Nair : SRN180001
3. Shaurya Prashant Bavishi : SPB180003
4. **REQUIREMENT COLLECTION:**
5. HOTELS.com facilitates the bookings of different types of rooms, party halls and conference halls in hotels.
6. There are two kinds of users - Owner and Customer. A person can be both owner and customer but must have separate accounts for each of these roles. A person can have at most one customer account and one owner account.
7. To create an account, Customer/Owner needs to include details like username and password for creation of account.
8. Customers can save the credit/debit card details for multiple cards in their account. Each card will have a card number, owner name, expiry date and billing address.
9. Customer account will also keep track of total points which will indicate the booking points that the customer has earned so far. These points can be redeemed while making new booking.
10. Owners can list one or more hotels that they own from their owner account on Hotels.com. All the hotels listed by hotel owners are available for booking on Hotels.com.
11. HOTELS.com saves information about various hotels like name, rating, certification, languages spoken and a binary attribute that indicates whether pets are allowed or not. It also saves details like check-in and check-out time, cancellation period allowed for booking, size of the hotel and documents required at the time of check-in.
12. Every hotel stores the information about nearby bus stop, airport and train station. It also stores the distance between the hotel and these source locations.
13. Hotels may provide amenities like Gaming area, Parking Facility, Laundry, Restaurant, Swimming Pool, Bar and Fitness Centre. Database will contain Boolean values for these features which specifies whether a particular hotel has these amenities or not.
14. Every hotel is associated with a location. Location contains different landmarks. These landmarks are provided as places of attraction to hotels in that particular area.
15. Hotels consist of Conference halls whose name, capacity and cost of booking (per hour) is stored.
16. Hotels also consist of Party Halls. In addition to the above attributes (of conference halls), a Boolean attribute that indicates whether outside catering is allowed or not is provided.
17. Customers can book rooms in hotels. Every room has a Room\_Type\_ID which uniquely identifies the room type. It stores details like Area (specifies area of the room in Sq.Ft), a Boolean attribute specifying whether smoking inside the room is allowed or not and the view room offers (like Sea facing, Garden view, Lake view etc.)
18. Hotels categorizes their rooms into 3 types:

Suite: it can have King Corner or Queen Corner bed.

Economy: it can have Twin or Full bed.

Deluxe: it can have King or Queen Bed

For each type of room, information like number of beds, number of bathrooms, number of extra beds the room can hold, number of people - adult and children it can accommodate and cost per night. The basic cost/night for each type of room is different. This cost is subjected to vary at the time of booking.

1. Rooms may provide amenities like furniture (desk, chair, iron board, closet), entertainment utilities (TV, Music system, cable network), food appliances (microwave, refrigerator, tea/coffee maker), general appliances (AC, heater, hair dryer, telephone) and utilities to improve accessibility. All these attributes holds numeric values indicating their availability in quantity.

16. There exists a contact point from HOTELS.com that addresses the complaints that customers or owners may have. The database stores the complaint message and other details about the complaint such as date and type of complaint.

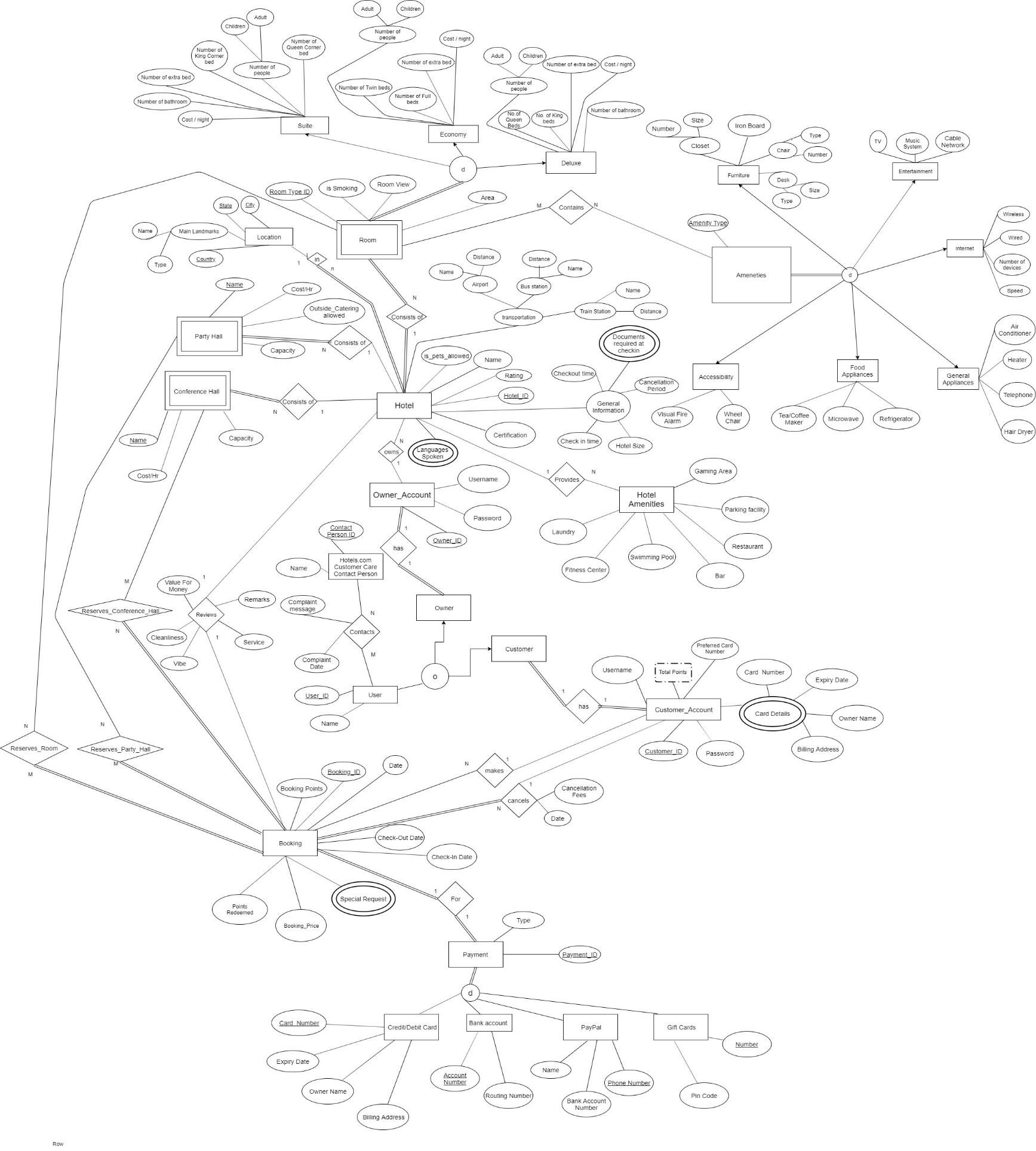
17. Customer can make a booking for any types of rooms, party hall or conference halls by selecting the required type, check-in date, check-out date. He can also make special requests if any. Customer will get booking points on every booking. He/she can redeem these points to avail discounts and offers.

18. Customer also has a facility to cancel the previously made booking. There is a fee associated with cancellation depending on the number of days between booking date and cancellation date.

19. Customer can make payment via different methods like credit/debit card, bank account

Paypal and gift cards.

1. Customer can submit only one review corresponding to the booking they made like remarks, and their rating (on a scale of 1-10) for cleanliness, vibe, service and value for money.

**EER DIAGRAM:** 

**C) MAPPING TO REALATIONAL MODEL AND NORMALIZATION:**

*Note : the method of mapping selected for generalization is noted in brackets according to the conventions of the slides*

1)HOTEL

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Hotel\_ID | Name | Rating | Certificati-on | Is\_Pets\_Allowed | Bus\_station\_distance | Airport\_distance | Train\_station\_distance |
| Check-in Time | Check-out Time | Hotel size | Cancellat-ion Period | Bus\_station\_name | Airport\_name | Train\_station\_ name | Country(FK refers LOCATION (Country)) |
| State(FK refers LOCATION (State)) | City(FK refers LOCATION (City)) | Hotel\_Owner\_ID(FK refers OWNER (Owner\_ID)) |  |  |  |  |  |

2) PARTY HALL

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Hotel\_ID (FK refers HOTEL (Hotel\_ID)) | Name | Cost/hour | Outside\_Catering\_allowed | Capacity |

3) BOOKING

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| BookingID | Booking\_Points | BookingDate | Check-in Date | Check-out Date |
| Hotel\_ID (FK refers HOTEL (Hotel\_ID)) | Review\_remarks | Review\_Cleanliness | Review\_Vibe | Review\_Service |
| Review\_Value\_for\_money | Payment\_ID(FK refers PAYMENT (Payment\_ID)) | Customer\_ID (FK refers CUSTOMER ACCOUNT (Customer\_ID)) | Cancellation\_fees | Cancellation\_date |
| Booking\_Price |  |  |  |  |

4) HOTEL AMENITIES

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Hotel ID (FK refers HOTEL (Hotel\_ID)) | Gaming Area | Restaurant | Bar | Swimming Pool | Fitness Center | Laundry | Parking Facility |

5) CONFERENCE HALL

|  |  |  |  |
| --- | --- | --- | --- |
| Hotel\_ID(FK refers HOTEL (Hotel\_ID)) | Name | Cost/hour | Capacity |

6) OWNER ACCOUNT

|  |  |  |  |
| --- | --- | --- | --- |
| Owner\_ID | Username | Password | User\_ID (FK refers USER (User\_ID)) |

7) CUSTOMER ACCOUNT

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Customer\_ID | Username | Password | Preferred Card Number | User\_ID (FK refers USER (User\_ID)) |

8)AMENITIES

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Amenity\_Type | Closet Size | Number of Closet | Iron Board | Chair Type | Number of Chairs | Desk  Type | No. of Desks |
| TV | Music System | Cable Network | Wireless Network  (Wi-Fi) | Wired Network | No. of allowed devices to connect | Internet Speed | AC |
| Heater | Telephone | Hair Dryer | Microwave | Coffee Maker | Refrigerator | Wheelchair support | Visual Fire alarm |

9) PAYMENT (Option 8A)

|  |  |
| --- | --- |
| Payment\_ID | Payment Type |

10) CREDIT / DEBIT CARD DETAILS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Payment\_ID(FK refers PAYMENT(Payment\_ID)) | CardNumber | ExpiryDate | Owner\_Name | Billing\_Address |

11) BANK\_ACCOUNT\_DETAILS

|  |  |  |
| --- | --- | --- |
| PaymentID(FK refers PAYMENT(Payment\_ID)) | AccountNumber | RoutingNumber |

12) PAYPAL\_DETAILS

|  |  |  |  |
| --- | --- | --- | --- |
| PaymentID(FK refers PAYMENT (Payment\_ID)) | Name | BankAccountNumber | PhoneNumber |

13)  GIFTCARDS

|  |  |  |
| --- | --- | --- |
| PaymentID(FK refers PAYMENT(Payment\_ID)) | Number | PinCode |

14) ROOM (8A))

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Hotel\_ID (FK refers HOTEL (Hotel\_ID)) | Room\_type\_ID | Is\_smoking\_allowed | Room View | Area |

15) SUITE

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Room\_type\_ID (refers ROOM (Room\_type\_ID)) | Cost per night | Number of King Corner Bed | Number of Queen Corner Bed | Number of Bathroom | Number of Extra Bed | Children | Adults |

16) ECONOMY

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Room\_type\_ID (refers ROOM (Room\_type\_ID)) | Cost per night | Number of Full Beds | Number of Twin Beds | Number of Bathroom | Number of Extra Bed | Children | Adults |

17) DELUXE

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Room\_type\_ID(FK refers ROOM(Room\_type\_ID)) | Cost per night | Number of King Bed | Number of Queen Bed | Number of Bathroom | Number of Extra Bed | Children | Adults |

18) LOCATION

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| State | City | Country | main\_landmark\_Name | main\_landmark\_Type |

19) USER

|  |  |  |  |
| --- | --- | --- | --- |
| User\_ID | Name | Customer\_Flag | Owner\_Flag |

20) HOTELS.com\_CONTACT CARE PERSON

|  |  |
| --- | --- |
| Contact Person ID | Name |

21) ROOM - AMENITIES  (Relationship)

|  |  |
| --- | --- |
| Room Type ID(FK refers ROOM(Room\_type\_ID)) | Amenity Type |

22)CARD DETAILS (multivalued attribute)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Username(FK refers CUSTOMER\_ACCOUNT(Username)) | Card Number | Expiry Date | Owner Name | Billing Address |

23) LANGUAGES\_SPOKEN (multivalued attribute)

|  |  |
| --- | --- |
| Hotel\_ID(FK refers HOTEL (Hotel\_ID)) | Languages\_Spoken |

24)  DOCUMENTS\_REQUIRED (multivalued attribute)

|  |  |
| --- | --- |
| Hotel\_ID(FK refers HOTEL(Hotel\_ID)) | DocumentsRequired |

25) SPECIAL\_REQUEST (multivalued attribute)

|  |  |
| --- | --- |
| Booking\_ID(FK refers BOOKING (Booking\_ID)) | SpecialRequest |

26) COMPLAINT (Relationship)

|  |  |  |  |
| --- | --- | --- | --- |
| Contact Person ID (refers  HOTELS.com CONTACT CARE PERSON (Contact\_Person\_ID)) | User ID(FK refers USER(USER\_ID)) | Complaint Date | Complaint Message |

27) RESERVES - ROOM (Relationship)

|  |  |
| --- | --- |
| Booking\_ID(FK refers BOOKING (Booking\_ID)) | Hotel\_ID (FK refers HOTEL(Hotel\_ID)) |

28) RESERVES - PARTY\_HALL (Relationship)

|  |  |
| --- | --- |
| Booking\_ID(FK refers BOOKING (Booking\_ID)) | Hotel\_ID(FK refers HOTEL(Hotel\_ID)) |

29) RESERVES - CONFERENCE\_HALL (Relationship)

|  |  |
| --- | --- |
| Booking\_ID(FK refers BOOKING (Booking\_ID)) | Hotel\_ID (FK refers HOTEL(Hotel\_ID)) |

* **NOTES ON NORMALIZATION PROCESS**

1. To make the table in 1 NF, we created a different table for multivalued attributes - LANGUAGES\_SPOKEN, DOCUMENTS\_REQUIRED, SPECIAL\_REQUEST.
2. Then all the tables are in 3NF. (since we followed the rules of mapping from EER diagram to relational schema)

**D) SQL CREATE TABLE QUERIES:**

1. **Create Query for HOTEL**:

CREATE SEQUENCE hotel\_seq START WITH 1;

 DROP TABLE HOTEL if exists HOTEL;

  CREATE TABLE HOTEL (

  Hotel\_id integer DEFAULT hotel\_seq.nextval not null,

  Name  char(30)  not null,

  Rating  int(30) not null,

  Certification  varchar(50) not null,

  Is\_Pets\_Allowed  boolean not null,

  Bus\_station\_distance  int not null,

  Airport\_distance  int not null,

  Train\_station\_distance  int not null,

  Check\_in\_Time  date,

  Check\_out\_Time  date,

  Hotel\_size  int not null,

  Cancellation\_Period  int not null,

  Bus\_station\_name  varchar(10) not null,

  Airport\_name  varchar(10) not null,

  Train\_station\_ name  varchar(10) not null,

  Country varchar(50) not null,

  State varchar(50) not null,

  City varchar(50) not null,

  primary key (Hotel\_id)

);

ALTER TABLE HOTEL ADD CONSTRAINT FK\_State\_Hotel FOREIGN KEY (State) REFERENCES LOCATION(State) ON DELETE CASCADE;

ALTER TABLE HOTEL ADD CONSTRAINT FK\_Country\_Hotel FOREIGN KEY (Country) REFERENCES LOCATION(Country) ON DELETE CASCADE;

ALTER TABLE HOTEL ADD CONSTRAINT FK\_City\_Hotel FOREIGN KEY (City) REFERENCES LOCATION(City) ON DELETE CASCADE;

**2. Table for AMENITIES (ROOM)**

DROP TABLE if exists AMENITIES;

CREATE TABLE AMENITIES (

Amenity\_Type integer not null,

Closet\_size varchar(10) DEFAULT medium,

Number\_of\_closet integer DEFAULT 1,

Iron\_Board char(1) DEFAULT F,

Chair\_type varchar(10),

Number\_of\_chairs integer DEFAULT 1,

Desk\_Type varchar(10),

Number\_of\_Desks integer DEFAULT 0,

TV char(1) DEFAULT False,

Music\_system char(1) DEFAULT F,

Cable\_network varchar(20),

Wireless\_network char(1) DEFAULT F,

Wired\_network char(1) DEFAULT F,

Number\_of\_allowed\_devices\_to\_connect integer DEFAULT(2),

Internet\_speed varchar(10),

AC char(1) DEFAULT F,

Heater char(1) DEFAULT F,

Telephone char(1) DEFAULT F,

Hair\_Dryer char(1) DEFAULT F,

Microwave char(1) DEFAULT F,

Tea\_or\_Coffe\_maker char(1) DEFAULT F,

Refrigerator char(1) DEFAULT F,

Wheelchair\_support char(1),

Visual\_fire\_alarm char(1),

Primary key (Amenity\_Type)

);

**3. Table for Owner Account**

DROP TABLE if exists OWNER\_ACCOUNT;

CREATE TABLE OWNER\_ACCOUNT (

Owner\_ID integer not null,

Username varchar(25) not null,

Password varchar(25) not null,

User\_ID integer not null,

Primary key (Owner\_ID)

);

ALTER TABLE OWNER\_ACCOUNT ADD CONSTRAINT FK\_UserID\_Owner FOREIGN KEY (User\_ID) REFERENCES USER(User\_ID) ON DELETE CASCADE;

**4. Table for Party Hall**

DROP TABLE if exists PARTY\_HALL;

CREATE TABLE PARTY\_HALL (

Hotel\_ID integer not null,

Name varchar(25) not null,

Cost\_per\_hour integer not null,

Outside\_catering\_allowed char(1) DEFAULT F,

Capacity integer,

Primary key (Hotel\_ID, Name)

);

ALTER TABLE PARTY\_HALL ADD CONSTRAINT FK\_Hotel\_ID\_PartyHall FOREIGN KEY (Hotel\_ID) REFERENCES HOTEL(Hotel\_ID) ON DELETE CASCADE;

**5. Table for Booking**

DROP TABLE if exists BOOKING;

CREATE TABLE BOOKING (

Booking\_ID integer not null,

Booking\_Points integer DEFAULT 0,

Booking\_Date date not null,

Chek-in\_date date not null,

Check-out\_date date not null,

Hotel\_ID integer not null,

Review\_remarks varchar(100),

Review\_Cleanliness integer,

Review\_ Vibe integer,

Review\_Service integer,

Review\_Value\_For\_Money integer,

Payment\_ID integer not null,

Customer\_ID integer not null,

Cancellation\_fees integer,

Cancellation\_Date integer DEFAULT 2,

Booking\_Price integer not null,

Primary key(Booking\_ID)

);

ALTER TABLE BOOKING ADD CONSTRAINT FK\_Hotel\_ID\_Booking FOREIGN KEY (Hotel\_ID) REFERENCES HOTEL(Hotel\_ID) ON DELETE CASCADE;

ALTER TABLE BOOKING ADD CONSTRAINT FK\_Payment\_ID\_Booking FOREIGN KEY (Payment\_ID) REFERENCES PAYMENT(Payment\_ID) ON DELETE CASCADE;

ALTER TABLE BOOKING ADD CONSTRAINT FK\_Customer\_ID\_Booking FOREIGN KEY (Customer\_ID) REFERENCES CUSTOMER(Customer\_ID) ON DELETE CASCADE;

**6. Table for Hotel Amenities**

DROP TABLE if exists HOTEL\_AMENITIES;

CREATE TABLE HOTEL\_AMENITIES (

Hotel\_ID integer not null,

Gaming\_area char(1) DEFAULT F,

Restaurant char(1) DEFAULT F,

Bar char(1) DEFAULT F,

Swimming\_pool char(1) DEFAULT F,

Fitness\_Centre char(1) DEFAULT F,

Laundry char(1) DEFAULT F,

Parking\_facility char(1) DEFAULT F,

Primary key(Hotel\_ID)

);

ALTER TABLE HOTEL\_AMENITIES ADD CONSTRAINT FK\_Hotel\_ID\_HotelAmenities FOREIGN KEY (Hotel\_ID) REFERENCES HOTEL(Hotel\_ID) ON DELETE CASCADE;

**7. Table for Conference Hall**

DROP TABLE if exists CONFERENCE\_HALL;

CREATE TABLE CONFERENCE\_HALL (

Hotel\_ID integer not null,

Name varchar(25) not null,

Cost\_per\_hour integer not null,

Capacity integer,

Primary key(Hotel\_ID, Name)

);

ALTER TABLE CONFERENCE\_HALL ADD CONSTRAINT FK\_Hotel\_ID\_ConferenceHall FOREIGN KEY (Hotel\_ID) REFERENCES HOTEL(Hotel\_ID) ON DELETE CASCADE;

**8. Table for Customer Account**

DROP TABLE if exists CUSTOMER\_ACCOUNT;

CREATE TABLE CUSTOMER\_ACCOUNT (

Customer\_ID integer not null,

Username varchar(25) not null,

Password varchar(25) not null,

Preferred\_Card\_Number integer,

User\_ID integer not null,

Primary key (Customer\_ID)

);

ALTER TABLE CUSTOMER\_ACCOUNT ADD CONSTRAINT FK\_UserID\_Customer FOREIGN KEY (User\_ID) REFERENCES USER(User\_ID) ON DELETE CASCADE;

**9. Table for PAYMENT**

DROP TABLE if exists PAYMENT;

CREATE TABLE PAYMENT(

Payment\_ID integer not null,

Payment varchar(25) not null,

Primary key (Payment\_ID)

);

**10. Table for CREDIT / DEBIT CARD DETAILS**

DROP TABLE if exists CREDIT / DEBIT CARD DETAILS;

CREATE TABLE CREDIT / DEBIT CARD DETAILS(

Payment\_ID integer not null,

CardNumber varchar(25) not null,

ExpiryDate date not null,

Owner\_Name varchar(25) not null,

Billing\_Address varchar(25) not null,

Primary key(Payment\_ID,CardNumber)

);

ALTER TABLE CREDIT / DEBIT CARD DETAILS ADD CONSTRAINT FK\_Payment\_ID FOREIGN KEY (Payment\_ID) REFERENCES PAYMENT(Payment\_ID) ON DELETE CASCADE;

**11. Table for BANK\_ACCOUNT\_DETAILS**

DROP TABLE if exists  BANK\_ACCOUNT\_DETAILS;

CREATE TABLE  BANK\_ACCOUNT\_DETAILS(

Payment\_ID integer not null,

AccountNumber int not null,

RoutingNumber int not null,

Primary key(Payment\_ID, Account Number)

);

ALTER TABLE BANK\_ACCOUNT\_DETAILS ADD CONSTRAINT FK\_Payment\_ID FOREIGN KEY (Payment\_ID) REFERENCES PAYMENT(Payment\_ID) ON DELETE CASCADE;

**12. Table for PAYPAL\_DETAILS**

DROP TABLE if exists PAYPAL\_DETAILS;

CREATE TABLE PAYPAL\_DETAILS(

Payment\_ID  integer not null,

CardNumber varchar(25) not null,

ExpiryDate date not null,

Owner\_Name varchar(25) not null,

Billing\_Address varchar(25) not null,

Primary key(Payment\_ID)

);

ALTER TABLE PAYPAL\_DETAILS ADD CONSTRAINT FK\_Payment\_ID FOREIGN KEY (Payment\_ID) REFERENCES PAYMENT(Payment\_ID) ON DELETE CASCADE;

**13. Table for GIFTCARDS**

DROP TABLE if exists GIFTCARDS;

CREATE TABLE GIFTCARDS(

Payment\_ID int not null,

Number int not null,

Pincode int not null,

Primary key(Payment\_ID,Number)

);

ALTER TABLE GIFTCARDS ADD CONSTRAINT FK\_Payment\_ID FOREIGN KEY (Payment\_ID) REFERENCES PAYMENT(Payment\_ID) ON DELETE CASCADE;

**14**. **Table for Room**

DROP TABLE ROOM if exists ROOM;

  CREATE TABLE ROOM(

  Room\_type\_ID  int,

  Is\_smoking\_allowed Type boolean  FALSE,

  Room\_View  varchar(10) not null,

  Area  varchar(10) not null,

  primary key (Room\_type\_ID)

);

ALTER TABLE ROOM ADD CONSTRAINT FK\_Hotel\_ID FOREIGN KEY (Hotel\_ID) REFERENCES HOTEL(Hotel\_ID) ON DELETE CASCADE;

**15. Table for SUITE**

DROP TABLE SUITE if exists SUITE;

CREATE TABLE SUITE(

Cost per night  int,

Number\_of\_King\_Corner\_Bed int not null,

Number\_of\_Queen\_Corner\_Bed int not null,

Number\_of\_Bathroom int not null,

            Number\_of\_Extra\_Bed int not null,

Children int not null,

            Adults int not null,

Room\_type\_ID int primary key

);

ALTER TABLE ROOM ADD CONSTRAINT FK\_Room\_type\_ID FOREIGN KEY (Room\_type\_ID) REFERENCES ROOM(Room\_type\_ID) ON DELETE CASCADE;

**16. Table for ECONOMY**

DROP TABLE ECONOMYif exists ECONOMY;

CREATE TABLE ECONOMY(

Cost per night  int,

Number\_of\_Full\_beds   int not null,

Number\_of\_Twin\_Beds  int not null,

Number\_of\_Bathroom  int not null,

            Number\_of\_Extra Bed  int not null,

Children  int not null,

            Adults  int not null,

Room\_type\_ID int primary key

);

ALTER TABLE ROOM ADD CONSTRAINT FK\_Room\_type\_ID FOREIGN KEY (Room\_type\_ID) REFERENCES ROOM(Room\_type\_ID) ON DELETE CASCADE;

**17. Table for DELUXE**

DROP TABLE DELUXE exists DELUXE ;

CREATE TABLE DELUXE (

Cost\_per\_night   int,

Number\_of\_King\_Bed   varchar(10) not null,

Number\_of\_Bathroom  varchar(10) not null,

            Number\_of\_Extra\_Bed  int not null,

Children  int not null,

            Adults  int not null,

Room\_type\_ID int primary key

);

ALTER TABLE DELUXE ADD CONSTRAINT FK\_Room\_type\_ID FOREIGN KEY (Room\_type\_ID) REFERENCES ROOM(Room\_type\_ID) ON DELETE CASCADE;

**18. Table for LOCATION**

DROP TABLE LOCATION if exists LOCATION;

CREATE TABLE LOCATION(

State  varchar(10) not null,

City  varchar(10) not null,

Country  varchar(10) not null,

main\_landmark\_Name  varchar(10),

main\_landmark\_Type  varchar(10),

primary key (State,City,Country)

);

**19. Table for USER**

DROP TABLE USER if exists USER;

CREATE TABLE USER(

Name Type varchar(10) not null,

Customer\_Flag char(1) F,

Owner\_Flag char(1) F,

primary key (User\_ID),

);

**20. Table for Hotels.com\_**CONTACT\_CARE\_PERSON

DROP TABLE Hotels.com CONTACT\_CARE\_PERSON

if exists Hotels.com\_CONTACT\_CARE\_PERSON;

CREATE TABLE Hotels.com CONTACT\_CARE\_PERSON(

Contact Person ID int not null,

Name varchar(10) not null,

Primary key(Contact\_Care\_Person)

);

**21. Table for ROOM-AMENITIES**

DROP TABLE ROOM\_AMENITIES

if exists ROOM\_AMENITIES;

CREATE TABLE ROOM\_AMENITIES(

Room\_Type\_ID int not null,

Amenity varchar(10) not null

Primary key(Room\_Type\_ID, Amenity)

);

ALTER TABLE ROOM\_AMENITIES ADD CONSTRAINT FK\_Room\_type\_ID FOREIGN KEY (Room\_type\_ID) REFERENCES ROOM(Room\_type\_ID) ON DELETE CASCADE;

**22. Table for CARD\_DETAILS**

DROP TABLE CARD\_DETAILS

if exists CARD\_DETAILS;

CREATE TABLE CARD\_DETAILS(

Username varchar(50) not null,

Card\_Number int not null,

Expiry\_Date date not null,

Owner\_name varchar(50) not null,

Billing\_address varchar(100) not null

Primary key(Username)

);

ALTER TABLE CARD DETAILS ADD CONSTRAINT FK\_Username FOREIGN KEY (Username) REFERENCES CUSTOMER\_ACCOUNT(Username) ON DELETE CASCADE;

**23. Table for LANGUAGES\_SPOKEN**

DROP TABLE LANGUAGES\_SPOKEN

if exists LANGUAGES\_SPOKEN;

CREATE TABLE LANGUAGES\_SPOKEN(

Hotel\_ID int not null,

Languages\_Spoken varchar(10) not null

Primary key(Hotel\_ID)

);

ALTER TABLE LANGUAGES\_SPOKEN ADD CONSTRAINT FK\_Hotel\_ID  FOREIGN KEY (Hotel\_ID ) REFERENCES Hotel (Hotel\_ID) ON DELETE CASCADE;

**24. Table for DOCUMENTS\_REQUIRED**

DROP TABLE DOCUMENTS\_REQUIRED

if exists DOCUMENTS\_REQUIRED ;

CREATE TABLE DOCUMENTS\_REQUIRED (

Hotel\_ID int not null,

DocumentsRequired varchar(10) not null

Primary key(Hotel\_ID)

);

ALTER TABLE LANGUAGES\_SPOKEN ADD CONSTRAINT FK\_Hotel\_ID  FOREIGN KEY (Hotel\_ID ) REFERENCES Hotel (Hotel\_ID) ON DELETE CASCADE;

**25. Table for SPECIAL\_REQUEST**

DROP TABLE SPECIAL\_REQUEST

if exists SPECIAL\_REQUEST ;

CREATE TABLE SPECIAL\_REQUEST (

Booking\_ID int not null,

SpecialRequest varchar(10) not null,

Primary key(Booking\_ID)

);

ALTER TABLE LANGUAGES\_SPOKEN ADD CONSTRAINT FK\_Booking\_ID  FOREIGN KEY (Booking\_ID ) REFERENCES Booking(Booking\_ID) ON DELETE CASCADE;

**26. Table for COMPLAINT**

DROP TABLE COMPLAINT

if exists COMPLAINT;

CREATE TABLE COMPLAINT(

Contact\_Person\_ID int not null,

UserID int not null,

Complaint\_date date not null,

Complaint\_Message varchar(20) not null,

SpecialRequest varchar(10) not null

Primary key(Contact\_Person\_ID,UserID)

);

ALTER TABLE COMPLAINT ADD CONSTRAINT FK\_Contact\_Person\_ID   FOREIGN KEY (Contact\_Person\_ID ) REFERENCES HOTELS.com CONTACT CARE PERSON(Contact\_Person\_ID ) ON DELETE CASCADE;

ALTER TABLE COMPLAINT ADD CONSTRAINT FK\_UserID   FOREIGN KEY (UserID) REFERENCES USER(UserID) ON DELETE CASCADE;

**27. Table for**  **RESERVES - ROOM**

DROP TABLE RESERVES - ROOM

if exists RESERVES - ROOM;

CREATE TABLE RESERVES - ROOM(

Booking\_ID int not null,

Hotel\_ID int not null,

Primary key(Bookinf\_ID,Hotel\_ID)

);

ALTER TABLE RESERVES - ROOM ADD CONSTRAINT FK\_Booking\_ID    FOREIGN KEY (Booking\_ID) REFERENCES BOOKING(Booking\_ID ) ON DELETE CASCADE;

ALTER TABLE RESERVES - ROOM ADD CONSTRAINT FK\_UserID   FOREIGN KEY (Hotel\_ID) REFERENCES HOTEL(Hotel\_ID) ON DELETE CASCADE;

**28. Table for**  **RESERVES - PARTY\_HALL**

DROP TABLE RESERVES - PARTY\_HALL

if exists RESERVES - PARTY\_HALL;

CREATE TABLE RESERVES - PARTY\_HALL(

Booking\_ID int not null,

Hotel\_ID int not null,

Primary key(Bookinf\_ID,Hotel\_ID)

);

ALTER TABLE RESERVES - PARTY\_HALL ADD CONSTRAINT FK\_Booking\_ID    FOREIGN KEY (Booking\_ID) REFERENCES BOOKING(Booking\_ID ) ON DELETE CASCADE;

ALTER TABLE RESERVES - ROOM ADD CONSTRAINT FK\_Hotel\_ID   FOREIGN KEY (Hotel\_ID) REFERENCES HOTEL(Hotel\_ID) ON DELETE CASCADE;

**29. Table for**  **RESERVES\_CONFERENCE\_HALL**

DROP TABLE RESERVES\_CONFERENCE\_HALL

if exists RESERVES\_CONFERENCE\_HALL;

CREATE TABLE RESERVES\_CONFERENCE\_HALL(

Booking\_ID int not null,

Hotel\_ID int not null,

Primary key(Booking\_ID,Hotel\_ID)

);

ALTER TABLE RESERVES - CONFERENCE\_HALL ADD CONSTRAINT FK\_Booking\_ID    FOREIGN KEY (Booking\_ID) REFERENCES BOOKING(Booking\_ID ) ON DELETE CASCADE;

ALTER TABLE RESERVES - CONFERENCE\_HALL ADD CONSTRAINT FK\_Hotel\_ID   FOREIGN KEY (Hotel\_ID) REFERENCES HOTEL(Hotel\_ID) ON DELETE CASCADE;

**E) PL/SQL:**

**Stored Procedures:**

* 1. *Procedure to increase the price of a Deluxe room type of a particular hotel by 20%:*

CREATE or REPLACE PROCEDURE Price\_Increase(Hotel\_id IN INT ) AS

thisDeluxeRoom Deluxe%ROWTYPE;

CURSOR Hotel\_Deluxe\_Rooms IS

SELECT D.\* FROM Deluxe D, Room R

WHERE D.Room\_Type\_ID=R.Room\_Type\_ID

AND R.Hotel\_ID=Hotel\_id;

FOR UPDATE;

BEGIN

OPEN Hotel\_Deluxe\_Rooms;

LOOP

  FETCH Hotel\_Deluxe\_Rooms INTO thisDeluxeRoom;

  EXIT WHEN (Hotel\_Deluxe\_Rooms%NOTFOUND);

  UPDATE Deluxe SET Cost\_per\_night = Cost\_per\_night \* 1.2

  WHERE CURRENT OF Hotel\_Deluxe\_Rooms;

END LOOP;

CLOSE Hotel\_Deluxe\_Rooms;

END;

* 1. *Procedure to double the price of party halls for all hotels located in a particular city.:*

create or replace PROCEDURE Party\_hall\_priceup\_by\_location(City IN VARCHAR(10)) AS

thisPartyHall PartyHall%ROWTYPE;

CURSOR Hotel\_by\_loc IS

SELECT P.\* FROM Hotel H, PartyHall P

WHERE H.Hotel\_ID=P.Hotel\_ID

AND H.City= City;

FOR UPDATE;

BEGIN

OPEN Hotel\_by\_loc;

LOOP

  FETCH Hotel\_by\_loc INTO thisPartyHall;

  EXIT WHEN (Hotel\_by\_loc%NOTFOUND);

  UPDATE PartyHall SET Cost\_Per\_Hour = Cost\_per\_hour \* 2

  WHERE CURRENT OF Hotel\_by\_loc;

END LOOP;

CLOSE Hotel\_by\_loc;

END;

**Triggers:**

1. *If user tries to cancel the booking after the specified cancellation period of the hotel,*

*then an error is raised saying cancellation is not allowed.*

*Cancellation Trigger:*

Create or replace TRIGGER Check\_Cancellation\_period

FOR UPDATE OF Cancellation\_Date ON BOOKING

COMPOUND TRIGGER

this\_Cancellation\_Period HOTEL.Cancellation\_Peiod%TYPE;

BEFORE STATEMENT IS

 Select DISTINCT HOTEL.Cancellation\_Peiod

From HOTEL, BOOKING

Where Booking.Hotel\_ID == Hotel.Hotel\_ID;

END BEFORE STATEMENT;

  AFTER STATEMENT IS

  BEGIN

    IF :NEW.Cancellation\_Date - :Check\_in\_date <

      this\_Cancellation\_Period

    THEN

      Raise\_Application\_Error(-20000, 'Cancellation is not allowed as it exceeds the      cancellation Period from check\_in\_date');

    END IF;

END AFTER STATEMENT

END Check\_Cancellation\_period;

1. *Trigger is fired when cost/night of economy room of a hotel is updated and is made more than deluxe room which raises an error:*

Create or replace TRIGGER Check\_Price\_difference

AFTER UPDATE OF cost\_per\_night ON ECONOMY

thisDeluxePrice DELUXE.Price%TYPE;

  BEGIN

    SELECT D.Cost\_per\_night INTO thisDeluxePrice

    FROM DELUXE D, ROOM R

    WHERE D.Room\_Type\_Id = R.Room\_Type\_Id

    AND R.Hotel\_ID = :NEW.Hotel\_ID

  IF :NEW.Cost\_per\_night > thisDeluxePrice

    THEN

      Raise\_Application\_Error(-20000, 'The price of Economy room cannot be more than Deluxe Room of the same hotel');

    END IF;

  END Check\_Price\_difference;