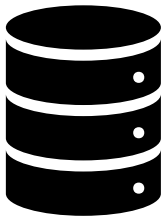


INFO 6210
DMDD PROJECT

Project Name:	Travel Agency Management System
Course Name:	INFO 6210 Data Management and Database Design
Year:	Spring 2021
Team Name:	Mind Ink Bots

This is the tracking document for our travel agency management project. We'll use this document to share details, resources, and schedules for our project. We will keep this document up-to-date as we work.

Team Members

Name	Role	Contact information
Parth Joshi	Member	joshi.pa@northeastern.edu
Dipti Dubey	Member	dubey.di@northeastern.edu
Shivani Vats	Member	vats.s@northeastern.edu
Prasanna Pimparwar	Member	pimparwar.p@northeastern.edu

Objectives

- ✓ To develop a system that automates the processes and activities of a travel agency.
- ✓ Capture customer's personal information and preferences for travel bookings and record the details of the final bookings made by the customers.
- ✓ Generate reports to facilitate ease in making travel decisions and preferences.
- ✓ To eliminate the broker fee and provide affordable deals to our customers.
- ✓ Provide an online booking system that allows the customers to check their reservations and know the status of the bookings from a convenient location at their own ease.

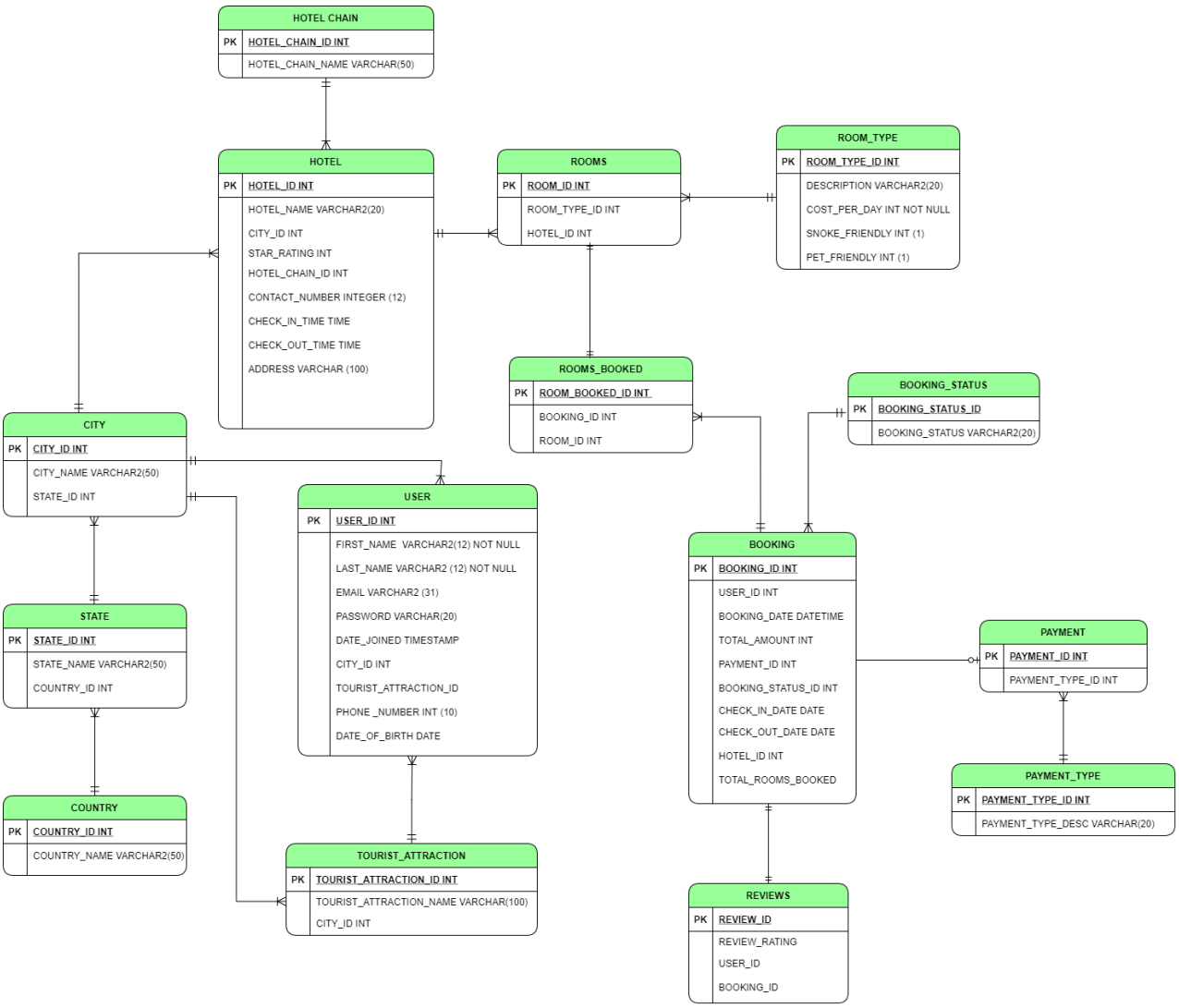
Problem Statement

Today, the world of data is constantly evolving every minute and has created a completely new dimension of challenges for both customers and agents across the globe to accurately record, store, update, and track data on a regular basis. A customer has to approach various agencies to find details of places and book hotels. This requires a lot of time and effort, may not fetch desired information from these agencies and often be misguided. Hence, it becomes tedious for a customer to plan a journey and have it executed properly since travel agencies cannot always possess optimal bookings to satisfy the customer needs. Thus, our aim here is to design a model that provides a hassle-free experience for booking a travel destination, maintain accurate user preferences and curate optimum travel packages.

 Proposed Solution

- ✓ To overcome the above problems, a database model can be used. The issues are resolved as follows:
- ✓ We, the Mind Ink Bots, have brainstormed to come up with a database design that brings out the user-friendly aspect of the entire hotel booking process.
- ✓ The database design allows us to sort the hotels on the basis of sentiment analysis conditional to the past customer reviews, which will be managed by the BOOKING and REVIEW ENTITY in our model.
- ✓ This database design enables one to infer or obtain the User and Hotel details by just including their auto-generated UNIQUE IDs, which in our case would be the PRIMARY KEY.
- ✓ Also, the database has been designed in a manner that caters to delivering optimum travel packages since the role of the agent has been entirely eliminated.
- ✓ Unlike the travel agencies prevalent in the market, our clients have the privilege and flexibility to book any hotel within the proximity of their favorite travel attraction.

DATA MODEL E-R DIAGRAM:



USER ENTITY

ATTRIBUTES	DOMAIN	COMMENTS
USER_ID	INTEGER	PRIMARY KEY, AUTOGENERATED, UNIQUE
FIRST_NAME	VARCHAR (12)	NOT NULL
LAST_NAME	VARCHAR (12)	NOT NULL
EMAIL_ID	VARCHAR (31)	UNIQUE KEY, NOT NULL
PASSWORD	VARCHAR (20)	NOT NULL
PHONE_NUMBER	INTEGER	UNIQUE, NOT NULL
TOURIST_ATTRACTION_ID	INTEGER	OPTIONAL
DATE_JOINED	TIMESTAMP	CURRENT_TIMESTAMP
CITY_ID	INTEGER	A Foreign Key which REFERENCES CITY_ID from the CITY Entity. This is used to fetch details of cities.
DATE_OF_BIRTH	DATE	OPTIONAL

COUNTRIES

ATTRIBUTES	DOMAIN	COMMENTS
COUNTRY_ID	INTEGER	PRIMARY KEY, AUTO GENERATED, UNIQUE
COUNTRY_NAME	VARCHAR (20)	NOT NULL

STATES

ATTRIBUTES	DOMAIN	COMMENTS
STATE_ID	INTEGER	PRIMARY KEY, AUTO GENERATED, UNIQUE
STATE_NAME	VARCHAR (20)	NOT NULL
COUNTRY_ID	INTEGER	A Foreign Key which REFERENCES COUNTRY_ID from the COUNTRY Entity. This is used to fetch details of countries.

CITY

ATTRIBUTES	DOMAIN	COMMENTS
CITY_ID	INTEGER	PRIMARY KEY, AUTO GENERATED, UNIQUE
CITY _NAME	VARCHAR (20)	NOT NULL
STATE_ID	INTEGER	A Foreign Key which REFERENCES STATE_ID from the STATE Entity. This is used to fetch details of States.

TOURIST ATTRACTIONS

ATTRIBUTES	DOMAIN	COMMENTS
TOURIST_ATTRACTION_ID	INTEGER	PRIMARY KEY, AUTO GENERATED, UNIQUE
TOURIST_ATTRACTION_NAME	VARCHAR (100)	NOT NULL
CITY_ID	INTEGER	A Foreign Key which REFERENCES CITY_ID from the CITY Entity. This is used to fetch details of cities.

HOTEL ENTITY

ATTRIBUTES	DOMAIN	COMMENTS
HOTEL_ID	INTEGER	PRIMARY KEY, AUTO GENERATED, UNIQUE
HOTEL_NAME	VARCHAR (20)	NOT NULL
ADDRESS	VARCHAR (100)	NOT NULL
CONTACT NUMBER	INTEGER (12)	NOT NULL
STAR_RATING	INTEGER	Describes the star rating of Hotel. NOT NULL
CITY_ID	INTEGER	A Foreign Key which REFERENCES CITY_ID from the CITY Entity. This is used to fetch details of cities.
HOTEL_CHAIN_ID	INTEGER	A Foreign Key which REFERENCES HOTEL_CHAIN_ID from the HOTELCHAIN Entity. This is used to fetch details of Hotel chains.
CHECK_IN_TIME	TIME	NOT NULL
CHECK_OUT_TIME	TIME	NOT NULL

HOTEL CHAIN ENTITY

ATTRIBUTES	DOMAIN	COMMENTS
HOTEL_CHAIN_ID	INTEGER	PRIMARY KEY, AUTO GENERATED, UNIQUE
HOTEL_CHAIN_NAME	VARCHAR (50)	NOT NULL

REVIEWS

ATTRIBUTES	DOMAIN	COMMENTS
REVIEW_ID	INTEGER	PRIMARY KEY, AUTO GENERATED
REVIEW_RATING	INTEGER	NOT NULL
BOOKING_ID	INTEGER	A Foreign Key which REFERENCES BOOKING_ID from the BOOKING Entity. This is used to fetch Booking Details.
USER_ID	INTEGER	A Foreign Key which REFERENCES USER_ID from the USER Entity. This is used to fetch User Details.

ROOMS

ATTRIBUTES	DOMAIN	COMMENTS
ROOM _ID	INTEGER	PRIMARY KEY, AUTO GENERATED
ROOM_TYPE_ID	INTEGER	A Foreign Key which REFERENCES ROOM_TYPE_ID from the ROOM TYPE Entity. This is used to fetch different room types.
HOTEL_ID	INTEGER	A Foreign Key which REFERENCES HOTEL_ID from the HOTEL Entity. This is used to fetch Hotel Details.

ROOM TYPE

ATTRIBUTES	DOMAIN	COMMENTS
ROOM_TYPE_ID	INTEGER	PRIMARY KEY, AUTO GENERATED
ROOM_TYPE_DESC	VARCHAR (20)	NOT NULL
ROOM_OCCUPANCY	INTEGER	NOT NULL
COST_PER_DAY	INTEGER	NOT NULL
SMOKE_FRIENDLY	INTEGER	OPTIONAL
PET_FRIENDLY	INTEGER	OPTIONAL

ROOM BOOKED

ATTRIBUTES	DOMAIN	COMMENTS
ROOM_BOOKED_ID	INTEGER	PRIMARY KEY, AUTO GENERATED, UNIQUE
BOOKING_ID	INTEGER	A Foreign Key which REFERENCES BOOKING_ID from the BOOKING Entity. This is used to fetch Booking Details.
ROOM_ID	INTEGER	NOT NULL

BOOKING ENTITY

ATTRIBUTES	DOMAIN	COMMENTS
BOOKING_ID	INTEGER	PRIMARY KEY, AUTO GENERATED, UNIQUE
BOOKING_DATE	DATETIME	NOT NULL
HOTEL_ID	INTEGER	A Foreign Key which REFERENCES HOTEL_ID from the HOTEL Entity. This is used to fetch Hotel Details.
USER_ID	INTEGER	A Foreign Key which REFERENCES USER_ID from the USER Entity. This is used to fetch User Details.
CHECK_IN_DATE	DATE	NOT NULL
CHECK_OUT_DATE	DATE	NOT NULL
BOOKING_STATUS_ID	INTEGER	A Foreign Key which REFERENCES BOOKING_STATUS_ID from the BOOKING Entity. This is used to fetch Booking status.
PAYMENT _ID	INTEGER	NOT NULL
TOTAL_AMOUNT	FLOAT	CALCULATED VALUE
TOTAL_ROOMS_BOOKED	INTEGER	CALCULATED VALUE

BOOKING STATUS

ATTRIBUTES	DOMAIN	COMMENTS
BOOKING_STATUS_ID	INTEGER	PRIMARY KEY, AUTO GENERATED
BOOKING_STATUS	VARCHAR (20)	NOT NULL

PAYMENT

ATTRIBUTES	DOMAIN	COMMENTS
PAYMENT_ID	INTEGER	PRIMARY KEY, AUTO GENERATED
PAYMENT_TYPE_ID	INTEGER	A Foreign Key which REFERENCES PAYMENT_TYPE_ID from the PAYMENT TYPE Entity. This is used to fetch different modes of payment.

PAYMENT TYPE

ATTRIBUTES	DOMAIN	COMMENTS
PAYMENT_TYPE_ID	INTEGER	PRIMARY KEY, AUTO GENERATED
PAYMENT_TYPE_DESC	VARCHAR (20)	NOT NULL, Describes the mode of payment

Business Rules

1. **USER** can book one or more **ROOMS**
2. Multiple **ROOMS** can be booked by the same user in the same **BOOKING**
3. Each **USER** can create one or multiple **BOOKINGS**
4. **HOTEL_CHAIN_ID** is linked to multiple **HOTELS**
5. Each **USER** can make one or more **RESERVATIONS**.
6. Each **RESERVATION** can only be made by one and only one **USER**.
7. Each **USER** can make one or more **PAYMENTS**.
8. Each **PAYMENT** can only be made by one and only one **USER**.
9. **USER** can book one or more rooms in one **BOOKING**
10. One **BOOKING** can only be processed by **ONE PAYMENT_ID**
11. Each **ROOM RESERVATION** can belong to one and only one **PAYMENT.**

12. Each **PAYMENT** can belong to one or more room reservations but only one **BOOKING_ID.**
13. One or more **ROOMS** can be accommodated in under one **BOOKING_ID**
14. Once the **BOOKING** is processed, the **BOOKING STATUS** will be set to **RESERVED.**
15. Once the **BOOKING** is done, the **BOOKING STATUS** will be set to **COMPLETED.**
16. A new **BOOKING_ID** will be created to book a different **HOTEL.**
17. Each **BOOKING** will have only one **REVIEW**
18. **NO REVIEW** can be given without a **COMPLETED BOOKING_STATUS.**
19. Each **ROOM ID** is **UNIQUE** and has one and only one room associated with it.
20. **ROOMS** cannot be assigned to more than one **ROOM RESERVATION** in the same time frame.
21. One **HOTEL CHAIN** can comprise of **MANY HOTELS**
22. **HOTEL** will be sorted and filtered on the basis of **TOURIST ATTRACTION**
23. Many **TOURIST ATTRACTIONS** can be in a **SINGLE CITY**
24. A **STATE** can have more than one **CITY**
25. A **CITY** is our main entity, linking **HOTELS** and **TOURIST ATTRACTION**
26. Each **ROOM_TYPE** in a hotel has the same **COST.**

SECURITY CONSTRAINTS: (User level Access/Permissions)

USER:

1. Has only READ access to the following: COUNTRY, STATE, CITY, BOOKING_STATUS, TOURIST ATTRACTION, ROOM_BOOKED, ROOM_TYPE and PAYMENT_TYPE tables.
2. Has UPDATE access to USER, BOOKING, PAYMENT.
3. Has WRITE access to BOOKING, REVIEW

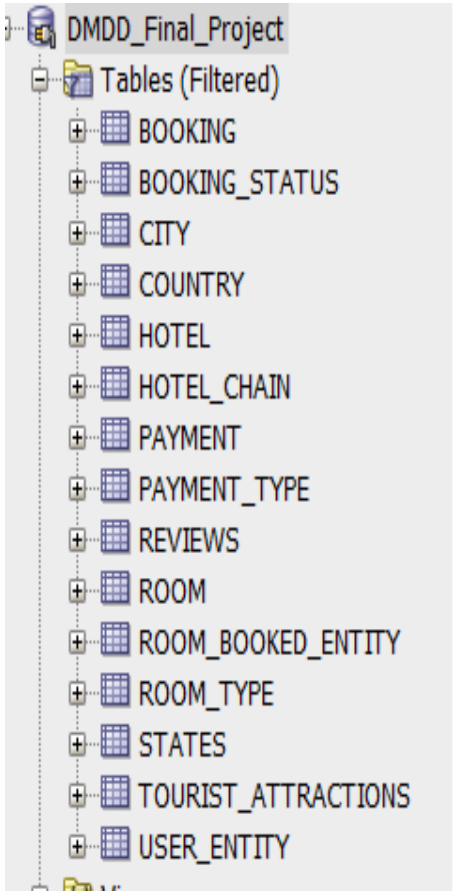
ADMIN:

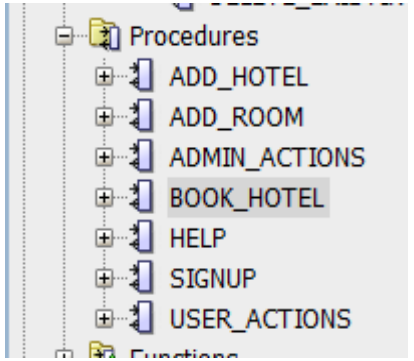
1. Has full access i.e. READ/WRITE/UPDATE all the entities in the database

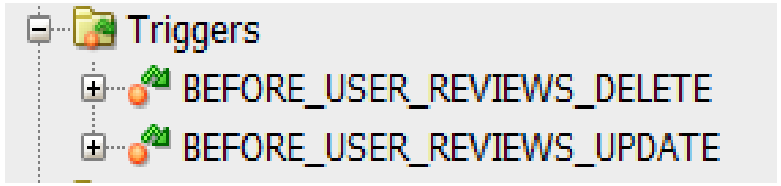
Features

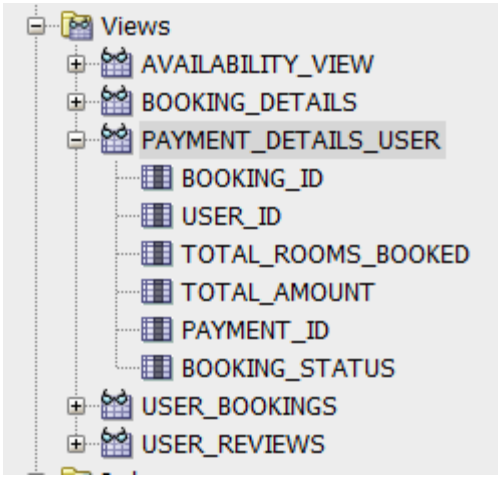
- **DDL**
- **DML**
- **PL/SQL PROCEDURE**
- **PACKAGES**
- **CUSTOM VIEWS**
- **TRIGGERS**
- **INDEXES**
- **USER ROLES**
- **REPORTS**

Database Objects






Triggers
BEFORE_USER_REVIEWS_DELETE
BEFORE_USER_REVIEWS_UPDATE


Views
AVAILABILITY_VIEW
BOOKING_DETAILS
PAYMENT_DETAILS_USER
BOOKING_ID
USER_ID
TOTAL_ROOMS_BOOKED
TOTAL_AMOUNT
PAYMENT_ID
BOOKING_STATUS
USER_BOOKINGS
USER_REVIEWS

DEMO

THANK YOU