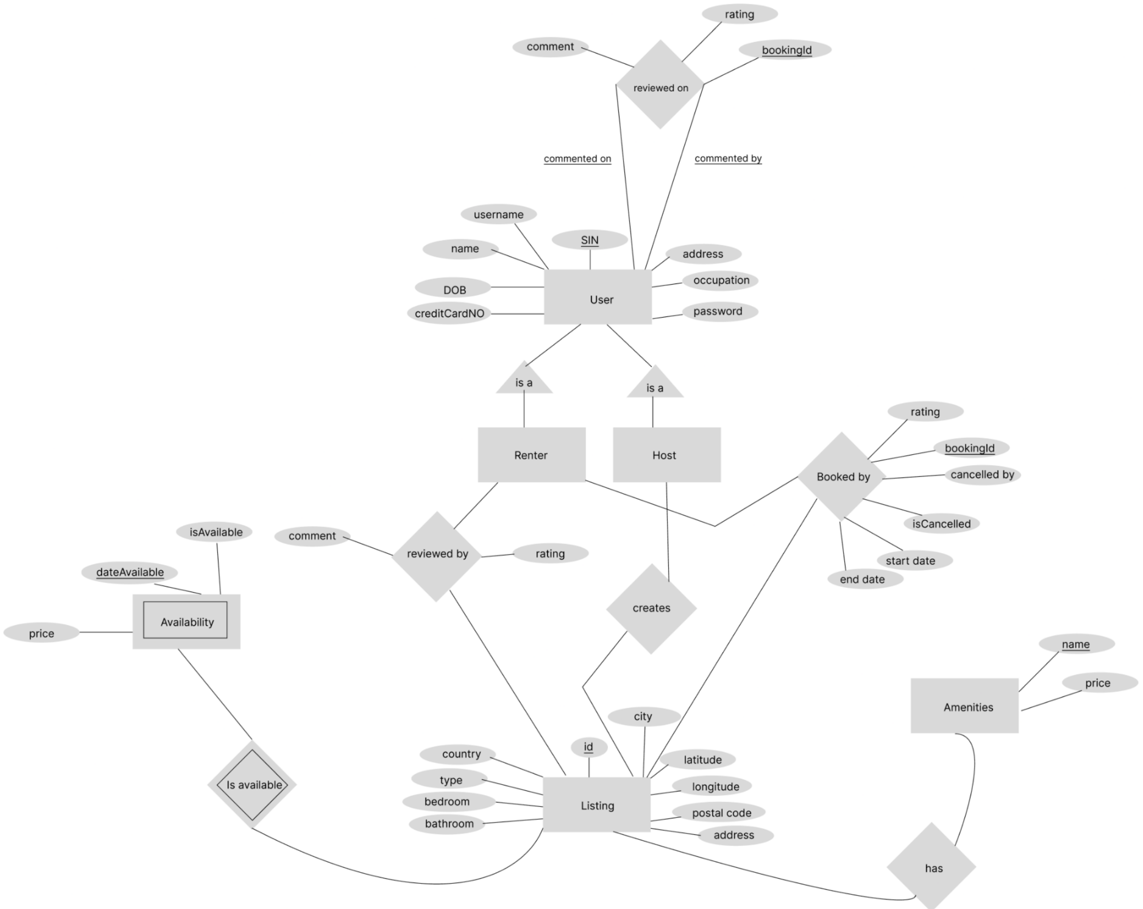
MyBnB Design Document

By: Dhruv Patel

Purpose of the Project:

The project aimed to develop a database system for an Airbnb-like program but will have additional features available to hosts and renters. This platform facilitates users in listing their properties, booking accommodations, and managing the associated data. The database system enables efficient data storage, retrieval, and manipulation for various aspects of the platform, including user information, property listings, bookings, and reviews.

E/R Diagram



Conceptual Problems and Justifications:

Creating the database structure through the translation of E/R diagrams into relational tables posed notable challenges. The process necessitated wise decision-making and adept problem-solving in the process of doing entity representation. One of them was the task of establishing primary keys for the tables, it demanded a lot of attention to prevent the emergence of duplications that might mess with the system constraints. This obstacle was overcome by improving our DDL statements to correspond with the requirements of specific operations, this is something we learned later on.

Addressing the issues of availability and bookings entailed strategic thinking. The Availability table functions as a repository for property availability information, while the BookedBy table does the role of preserving booking information. The usage of a dateAvailable column within the Availability table, coupled with the BookedBy table for bookings, resulted in a logical and structured organization of data. Although each listing is included with availability by day in the table, having a date range from users was deemed a more user-centric approach.

The usage of relationships between users and listings was of great significance. The UserCreatesListing table emerged as the connection between hosts and their respective listings. This choice ensured the establishment of an important relationship between users and the properties under their purview.

Using SQL queries to clean and drop the database and create tables with data was extremely helpful. This ensured not having to manually delete the database and create tables everytime we needed to restart.

The usage of foreign key constraints across tables was a pivotal way of safeguarding data integrity. Navigating the syntax and results of foreign keys and delete-on-cascade operations associated with them, presented its own share of challenges. Rigorous testing facilitated the refinement of DDL statements and foreign key-related actions, resulting in a consistent and efficient database structure.

Assumptions:

Payment Method: The system assumes that payment transactions will exclusively be facilitated through credit card transactions. Other payment methods, such as digital wallets or bank transfers, are not considered for payment.

Single User Account: It is assumed that each user, whether a renter or a host, operates under a single user account. This implies that a user can act as both a host and a renter interchangeably.

Identification with IDs: The system relies on unique identification through IDs for listings and bookings. This assumption ensures the efficient management of listings and reservations.

Date-Driven Search: The search functionality within the system operates on the premise that users are required to input specific dates they intend to book accommodations for. This approach streamlines the search process and aligns with users' booking preferences.

Access to Booking IDs: To facilitate the process of commenting or rating listings or hosts, users are expected to retrieve their booking IDs from the "View My Bookings" feature. This step serves as a prerequisite for users to effectively engage in providing feedback on their experiences.

Relation Schema:

Listing:

- listingId (Primary Key)
- city
- latitude
- longitude
- postalCode
- country
- type
- address
- bedrooms
- bathrooms

User:

- SIN (Primary Key)
- address
- occupation
- dob
- firstName
- lastName
- username
- password
- creditCardNO

Amenities:

- name (Primary Key)
- price

Availability:

- dateAvailable (Primary Key)
- price
- listingId (Primary Key, Foreign Key referencing Listing)
- isAvailable

BookedBy:

- bookingId (Primary Key)
- listingId (Foreign Key referencing Listing)
- renterSIN (Foreign Key referencing User)
- startDate
- endDate
- isCancelled
- cancelledBy (Foreign Key referencing User)

UserCreatesListing:

- hostSIN (Foreign Key referencing User)
- listingId (Primary Key, Foreign Key referencing Listing)

ListingReviewAndComments:

- bookingId (Primary Key, Foreign Key referencing BookedBy)
- renterSIN (Primary Key, Foreign Key referencing User)
- comment
- rating

ListingToAmenities:

- listingId (Primary Key, Foreign Key referencing Listing)
- amenity (Primary Key, Foreign Key referencing Amenities)

UserReviews:

- commentedOn (Primary Key, Foreign Key referencing User)
- commentedBy (Primary Key, Foreign Key referencing User)
- bookingId (Primary Key, Foreign Key referencing BookedBy)
- comment
- rating

Primary Keys:

- Listing: listingId
- User: SIN

- Amenities: name
- Availability: (dateAvailable, listingId)
- BookedBy: bookingId
- UserCreatesListing: listingId
- ListingReviewAndComments: (bookingId, renterSIN)
- ListingToAmenities: (listingId, amenity)
- UserReviews: (commentedOn, commentedBy, bookingId)

Foreign Keys:

- Availability: listingId (Referencing Listing)
- BookedBy: listingId (Referencing Listing), renterSIN (Referencing User), cancelledBy (Referencing User)
- UserCreatesListing: hostSIN (Referencing User), listingId (Referencing Listing)
- ListingReviewAndComments: bookingId (Referencing BookedBy), renterSIN (Referencing User)
- ListingToAmenities: listingId (Referencing Listing), amenity (Referencing Amenities)
- UserReviews: commentedOn (Referencing User), commentedBy (Referencing User), bookingId (Referencing BookedBy

Code

https://github.com/Priyankdave1/mybnb

DDL

Included separately.

System Limitations

1. The current system exposes identifiers (IDs) for listings and bookings, which might not be user-friendly and can result in a security risk.

2. The current search functionality only allows users to search based on certain properties and has limited filters.

3. Restricting payments to credit cards might exclude users who prefer alternative payment methods, such as digital wallets or bank transfers.

4. The user input is being validated but a result of invalid input may result in them having to run the command again and re enter their previous information, which limits the UX and user satisfaction.

Improvements

1. Users might find it more intuitive to work with listing and booking names or labels. This can enhance the user experience by making interactions more comprehensible.

2. To provide a more customizable experience, additional filters could be introduced. For instance, filters for the number of rooms can significantly improve the search results and user satisfaction.

3. Incorporating a variety of payment options could cater to a wider user base.

4. Provide an interface that has better error prevention and identifies the user whose data is invalid before they submit so they don't lose their previous progress.

Conclusion:

In conclusion, our project aimed to create a property rental management system, with a focus on the database and logic of the system. This endeavor showcased the complexities of data modeling, relationship establishment, and user interaction. While challenges like key definitions and foreign relationships were encountered, they led us to refine our approach for data consistency.

The system now effectively handles property listing, booking, and user reviews. However, limitations in data privacy, UX and UI intricacy were recognized. To enhance the system, implementing more payment options, allowing users to customize their experience to enhance user satisfaction, and refining user interface.

The project highlights the significance of collaborative problem-solving, data-driven decisions, and ongoing advancements in software development. It provides a foundation for future iterations, capable of delivering improved functionalities, enriched user experiences for a more refined rental system.

User Manual:

Register:

Description: Registering allows users to create a new account in the system, granting them access to various features.

To register a new account execute the **register** command and follow prompts.

Prompts:
First name:
Last name:
Email:
Password (hidden input):

Result:
Upon successful registration, a new user account is created and can be used to log in to the system.

Login:

Description: Logging in allows users to access their account and utilize the application's features.

To log in to your account use the **login** command.

Prompts:

Email:
Password (hidden input):
SIN:

Result:
Successful login grants the user access to their account and other functionalities of the application.

Logout:

Description: Logging out ends the current user session. Some following actions will require logging in again.

To logout call the **logout** command

Delete Account:

Description: Deleting an account permanently removes the user's profile, data, and access to the application.

To delete your account call **delete-account** command.

Listing Management:

Description: Listing management involves creating, deleting, and viewing property listings that hosts want to offer for rent.

Create:

To create a new property listing: call the **create-listing** command with the following prompts
Prompts:
title: Title of the property.
description: Description of the property.
price: Price per night for the property.
availability: Availability dates for the property.

Delete a listing:

To delete a listing: call the **delete-listing** command and answer the following prompt
Prompts:
Listing ID: ID of the listing to be deleted.

Booking and Cancellation

Description: Booking management involves creating, canceling, and viewing bookings for properties that renters want to book.

Create Booking

To create a booking: Execute the **create-booking** command with the following prompts:

Prompts:

start-date: Start date of the booking in the format YYYY-MM-DD.
end-date: End date of the booking in the format YYYY-MM-DD.
The system will display a list of available properties during the specified date range. Choose a property by entering its Listing ID.

Cancel Booking

To cancel a booking: Execute the **cancel-booking** command and answer the following prompts:

Prompts:

Cancel as: Specify whether you're canceling as a host or renter.
Booking ID: ID of the booking to be canceled.

View Bookings:

To view your bookings: Execute the **view-booking-as-host or view-booking-as-renter** command.

Rating and Commenting

Description: This feature allows users to rate and comment on properties they have booked.

Rate and Comment on Listing

To rate and comment on a listing: Execute the **rate-and-comment-on-listing** command with the following option:

Options:

--bookingId (or -I): Booking ID of the listing you want to rate and comment on.

The system will prompt you to provide a rating (on a scale of 1 to 5) and a comment for the property.

Rate and Comment on User

Description: This feature allows users to rate and comment on other users (hosts or renters) based on their experiences.

Rate and Comment on Host/Renter

To rate and comment on a host or renter: View the bookings using the view-bookings-as-* command. Them execute the **rate-and-comment-user** command with the following options:

Options:

--accType (or -a): Specify whether you're rating and commenting as a host or renter.
--bookingId (or -b): Booking ID associated with the interaction.
The system will prompt you to provide a rating (on a scale of 1 to 5) and a comment for the user.

Availability and Pricing

Description: Availability and pricing management allows hosts to update the availability dates and pricing for their property listings.

Update Availability:

To update the availability dates for a property listing: call the **update-availability** command with the following options

Prompts:

Enter the Listing ID: Listing ID of the property.
Choose the type of action (remove/insert): Type of action to perform (remove or insert).
Enter the start date (YYYY-MM-DD): Start date for the pricing update.
Enter the end date (YYYY-MM-DD): End date for the pricing update.

Update Pricing:

To update the pricing for a property listing: call the **update-price** command with the following options

Prompts:

Enter the Listing ID: Listing ID of the property.
Enter the start date (YYYY-MM-DD): Start date for the pricing update.
Enter the end date (YYYY-MM-DD): End date for the pricing update.
Enter the new price per night: New price per night for the property.

Note: Ensure that the provided start date and end date are in the format "YYYY-MM-DD" and that the listing ID corresponds to an existing property listing.

Search

Description: The search and filtering command allows users to search for property listings based on various criteria, including amenities, price range, availability, and location. The tool provides multiple options to refine search results according to the user's preferences.

To search, call the command with the options below and then choose from the query choice listed and follow the prompts.

Options:
--sortByPrice (-s): Specify the sorting order for search results based on the total price. Choose either asc (ascending) or desc (descending). Default: asc.
--amenity (-a): Specify amenities to filter by. Multiple amenities can be provided by using the option multiple times (e.g. -a Oven -a Microwave)
--price_min (-pmin): Specify the minimum price to filter by.
--price_max (-pmax): Specify the maximum price to filter by.
--start_date: Specify the start date for availability filtering (required).
--end_date: Specify the end date for availability filtering (required).

Choose from these choices:

- Search by Postal Code: Find listings within a certain postal code area.
- Search by Address: Find listings with a specific address.
- Search Listings Within Range: Find listings within a specified range(KM) of a given latitude and longitude.
- Change Filters: Modify the filtering options including amenities, price range, and availability dates.
- Clear Filters: Reset all filters to their default values except for the availability dates.
- Exit

Note: Follow the prompts and input guidelines provided by the command.

Reports

Report 1: Number of Bookings by City/Postal Code

Description: This report generates the number of bookings based on either city or postal code within a specified date range.

To generate the report: run the **report1-num-bookings-by-city-postalcode** command with the following options.

Options:

--start_date: Start date for the report period (YYYY-MM-DD).
--end_date: End date for the report period (YYYY-MM-DD).
--searchBy: Search by city or postal code.

Prompt:
City or Postal Code depending on what was selected in the option.

Report 2: Number of Listings in an Area

Description: This report calculates the number of listings in a specific country, city, or postal code.

To generate the report: run the **report2-num-listings-in-area** command with the following options.

Options:

--country: Country.
--city: (Optional) City.
--postalcode: (Optional) Postal code for the query.

Report 3: Host Ranking by Listings Owned

Description: This report ranks hosts based on the number of listings they own in a specific country or city.

To generate the report: run the **report3-host-ranking-by-listings-owned** command with the following options.

Options:

--country: Country.
--city: (Optional) City.

Report 4: Possible Commercial Hosts

Description: This report identifies possible commercial hosts who own a significant portion of listings in a country or city.

To generate the report: run the **report4-possible-commercial-hosts** command with the following options.

Options:

--country: Country.
--city: (Optional) City.

Report 5: Renter Ranking by Number of Bookings

Description: This report ranks renters based on the number of bookings they have made within a specified date range and city.

To generate the report: run the **report5-rank-renters-by-num-bookings** command with the following options.

Options:

--start_date: Start date for the report period (YYYY-MM-DD).
--end_date: End date for the report period (YYYY-MM-DD).
--city: (Optional) City.

Report 6: Most Cancellations

Description: This report identifies users with the most booking cancellations within a specified date range.

To generate the report: run the **report6-most-cancellations** command with the following options.

Options:

--run_for: Generate the report for hosts or renters.

Report 7: Show Common Noun Phrases in Comments

Description: This report identifies common noun phrases used in comments for a specific listing.

To generate the report: run the **report7-noun** command with the following options.

Options:

--listingId: Listing ID for the comments.

Host Toolkit:

Description: The toolkit provides features to generate pricing information, display essential amenities, and suggest prices with additional amenities for a specific property listing.

Actions:

1. Generate Price:
Description: Generate the suggested price per day for a property listing.
Usage: Select this option, enter the listing ID, and receive the suggested price.

2. Show Essential Amenities:
Description: Display the essential amenities based on the property type.
Usage: Select this option, and the essential amenities will be shown.

3. Generate Price Using Additional Amenities:
Description: Generate the suggested price per day for a property listing by adding extra amenities and compare it with the original suggested price.
Usage: Select this option, choose additional amenities, and receive the suggested price with the new amenities included.