



AIRBNB CLONE

DATABASE SYSTEMS

ABSTRACT

This project involves the creation of an Airbnb clone using a MySQL database. The database includes ten entities, such as User, Listing, Booking, Review, Message, Payment, Wish List, Neighborhood, Notification, and Language, with various attributes to capture necessary information. The system aims to provide a platform for users to search and book accommodations, communicate with hosts, and leave reviews. It includes API integration and leveraging advanced database features such as Joins, Views, Stored Procedures, Triggers, and Functions. It also includes features like wish lists, neighborhood recommendations, and notifications. This project report provides a detailed description of the entities and their attributes, their relationship between them, types of attributes, and the overall system architecture. It also highlights the importance and benefits of database management systems in web application development.

Table of Contents

Introduction	3
Online Connectivity & AWS Server Integration.....	3
Entities and Attributes.....	4
User	4
Listing	4
Booking	5
Review.....	5
Message	6
Payment	7
Wish List	7
Neighborhood.....	7
Notification.....	8
Language	8
City	8
Country.....	9
State	9
Location.....	9
Payment Method	9
House Rules	10
Type of Properties.....	10
Amenities	11
Credentials	11
Relationships between Entities.....	12
Types of Attributes	13
Advanced Database Features.....	14
Conclusion.....	15

Introduction

The project is a comprehensive platform that replicates the functionality of the popular accommodation booking service. It offers a user-friendly interface for browsing and booking listings, managing reservations, and facilitating communication between hosts and guests. The project incorporates advanced database features, including joins, stored procedures, triggers, views, and functions, to ensure efficient data management and seamless operations. These features enable us to optimize data retrieval, enforce data integrity, automate tasks, and enhance code reusability. The project's entity-relationship (ER) diagram visually represents the relationships and attributes of the entities involved, providing a clear overview of the data structure. With the integration of API, our Airbnb clone project allows seamless connectivity with external systems and services, enabling users to access and interact with the platform through various channels. Through the combination of advanced database features, API integration, and a well-defined data model, our AIRBNB project delivers a robust and user-centric accommodation booking experience.

Online Connectivity & AWS Server Integration

The project utilizes a robust database hosted on the AWS (Amazon Web Services) server. This online connection ensures that our database is readily accessible and provides seamless functionality for our application. By leveraging the power of AWS, we can securely store and retrieve data, allowing users to access and interact with listings, bookings, reviews, user profiles, and other essential information. The AWS server ensures high availability, scalability, and reliability, enabling our AIRBNB Clone to deliver a reliable and efficient platform for users to explore and engage in the world of accommodations and travel experiences.

Entities and Attributes

The following is a description of each entity and its attributes:

User

The User entity stores information about the users who use the platform. It has the following attributes:

- **User ID (Primary key):** A unique identifier for each user.
- **First name:** The user's first name.
- **Last name:** The user's last name.
- **Phone number:** The user's phone number.
- **Profile picture:** A picture of the user.
- **Verification status:** The user's verification status, which can be verified through email, phone number, or government ID.
- **Host status:** A boolean value that indicates whether the user is a host or not.
- **Reviews:** The total number of reviews and the average rating for the user.
- **Listing ID (foreign key to Listing entity):** The ID of the listing that the user has created.
- **Email address (foreign key to Credentials entity):** The user's email address.

Listing

The Listing entity stores information about the properties that are listed on the platform. It has the following attributes:

- **Listing ID (Primary key):** A unique identifier for each listing.
- **Title:** The title of the listing.
- **Description:** A description of the listing.
- **Number of guests:** The maximum number of guests allowed in the listing.
- **Number of bedrooms:** The number of bedrooms in the listing.

- **Number of beds:** The number of beds in the listing.
- **Number of bathrooms:** The number of bathrooms in the listing.
- **Price per night:** The price per night for the listing.
- **Availability calendar:** The availability of the listing, including the start and end date.
- **Address (foreign key to Location entity):** The address of the listing.
- **Host ID (foreign key to User entity):** The ID of the user who created the listing.
- **Neighborhood ID (foreign key to Neighborhood entity):** A unique neighborhood ID for each city.

Booking

The Booking entity stores information about the bookings made by the guests. It has the following attributes:

- **Booking ID (Primary key):** A unique identifier for each booking.
- **Check-in date:** The date on which the guest checks in.
- **Check-out date:** The date on which the guest checks out.
- **Number of guests:** The number of guests staying in the listing.
- **Total cost:** The total cost of the booking.
- **Listing ID (foreign key to Listing entity):** The ID of the listing that is being booked.
- **Guest ID (foreign key to User entity):** The ID of the user who made the booking.

Review

- **Review ID (Primary key):** This attribute uniquely identifies each review and allows for easy retrieval of specific reviews from the database.
- **Rating:** This attribute captures the numerical rating given by a guest to a listing. It could range from 1 to 5, with 5 being the highest rating.

- **Comment:** This attribute captures the written feedback given by a guest about their experience with a listing. It could be a short or long comment describing their stay, the host, the location, and other aspects of the experience.
- **Listing ID (foreign key to Listing entity):** This attribute establishes a relationship between the Review and Listing entities. It links each review to the listing it pertains to, enabling easy retrieval of all reviews for a particular listing.
- **Guest ID (foreign key to User entity):** This attribute establishes a relationship between the Review and User entities. It links each review to the guest who wrote it, enabling easy retrieval of all reviews written by a particular guest.
- **Host ID (foreign key to User entity):** This attribute establishes a relationship between the Review and User entities. It links each review to the host of the listing being reviewed, enabling easy retrieval of all reviews written about a particular host.

Message

- **Message ID (Primary key):** This attribute uniquely identifies each message and allows for easy retrieval of specific messages from the database.
- **Sender ID (foreign key to User entity) (User ID):** This attribute establishes a relationship between the Message and User entities. It links each message to the user who sent it, enabling easy retrieval of all messages sent by a particular user.
- **Receiver ID (foreign key to User entity) (User ID):** This attribute establishes a relationship between the Message and User entities. It links each message to the user who received it, enabling easy retrieval of all messages received by a particular user.
- **Message content:** This attribute captures the text of the message.
- **Timestamp:** This attribute captures the time when the message was sent.

Payment

- **Payment ID (Primary key):** This attribute uniquely identifies each payment and allows for easy retrieval of specific payments from the database.
- **Amount:** This attribute captures the amount of the payment.
- **Date:** This attribute captures the date when the payment was made.
- **Payment method ID (foreign key to Payment Method entity):** This attribute captures the method of payment used (e.g., credit card, PayPal, etc.).
- **Booking ID (foreign key to Booking entity):** This attribute establishes a relationship between the Payment and Booking entities. It links each payment to the booking it pertains to, enabling easy retrieval of all payments for a particular booking.

Wish List

- **Wish List ID (Primary key):** This attribute uniquely identifies each wish list and allows for easy retrieval of specific wish lists from the database.
- **User ID (foreign key to User entity):** This attribute establishes a relationship between the Wish List and User entities. It links each wish list to the user who created it, enabling easy retrieval of all wish lists created by a particular user.
- **Listing ID (foreign key to Listing entity):** This attribute establishes a relationship between the Wish List and Listing entities. It links each wish list to the listings it contains, enabling easy retrieval of all wish lists containing a particular listing.

Neighborhood

- **Neighborhood ID (Primary key):** This attribute uniquely identifies each neighborhood and allows for easy retrieval of specific neighborhoods from the database.
- **Neighborhood name:** This attribute captures the name of the neighborhood.

- **Listing ID (foreign key to Listing entity):** This attribute establishes a relationship between the Neighborhood and Listing entities. It links each neighborhood to the listings located within it, enabling easy retrieval of all listings in a particular neighborhood.

Notification

- **Notification ID (Primary key):** This attribute uniquely identifies each notification and allows for easy retrieval of specific notifications from the database.
- **User ID (foreign key to User entity):** This attribute establishes a relationship between the Notification and User entities. It links each notification to the user who received it, enabling easy retrieval of all notifications received by a particular user.
- **Message:** This attribute captures the text of the notification.
- **Timestamp:** This attribute captures the time when the notification was sent.

Language

- **Language ID (Primary key):** This attribute uniquely identifies each language and allows for easy retrieval of specific languages from the database.
- **Language name:** This attribute captures the name of the language.
- **Listing ID (foreign key to Listing entity):** This attribute establishes a relationship between the Language and Listing entities. It links each language to the listings it pertains to, enabling easy retrieval of all listings.

City

- **Neighborhood ID (foreign key to Neighborhood entity):** A unique neighborhood ID for each city.
- **City:** The name of the city where the neighborhood is located.

Country

- **Neighborhood ID (foreign key to Neighborhood entity):** A unique neighborhood ID for each country.
- **Country:** The name of the country where the neighborhood is located.

State

- **Neighborhood ID (foreign key to Neighborhood entity):** A unique neighborhood ID for each state.
- **State:** The name of the state where the neighborhood is located.

Location

- **Address (Primary key):** The address of a listing or property.
- **Postal code:** The postal code or ZIP code associated with the address.

Payment Method

- **Listing ID (Foreign key to Listing entity):** This attribute establishes a relationship between the Payment Method and Listing entities.
- **Payment Method ID (Composite primary key):** This attribute uniquely identifies each payment method within a listing. It is part of the composite primary key along with the Listing ID.
- **User ID (Foreign key to User entity):** This attribute establishes a relationship between the Payment Method and User entities.
- **Credit Card:** This boolean attribute indicates whether the payment method is a credit card.
- **Debit Card:** This boolean attribute indicates whether the payment method is a debit card.

- **Cash on Hand:** This boolean attribute indicates whether the payment method is cash on hand.

House Rules

- **House Rules ID (Composite primary key):** A unique identifier for each set of house rules.
- **Listing ID (Foreign key to Listing entity):** This attribute establishes a relationship between the House Rules and Listing entities.
- **Smoking:** A boolean value indicating whether smoking is allowed or not.
- **Pets:** A boolean value indicating whether pets are allowed or not.
- **No shoes inside:** A boolean value indicating whether wearing shoes inside is prohibited.
- **Party/Events:** A boolean value indicating whether parties or events are allowed.
- **Visitors:** A boolean value indicating whether visitors are allowed.

Type of Properties

- **Property Type ID (Primary key):** A unique identifier for each type of property.
- **Listings ID (Foreign key to Listing entity):** This attribute establishes a relationship between the Type of Properties and Listing entities.
- **Apartment:** This attribute indicates whether the property type is an apartment.
- **Villa:** This attribute indicates whether the property type is a villa.
- **House:** This attribute indicates whether the property type is a house.
- **Cabin:** This attribute indicates whether the property type is a cabin.
- **Cottage:** This attribute indicates whether the property type is a cottage.
- **Loft:** This attribute indicates whether the property type is a loft.
- **Town House:** This attribute indicates whether the property type is a townhouse.

Amenities

- **Amenity ID (Primary key):** A unique identifier for each amenity.
- **Wi-Fi:** A boolean value indicating the availability of Wi-Fi.
- **Kitchen:** A boolean value indicating the availability of a kitchen.
- **Pool:** A boolean value indicating the availability of a pool.
- **AC:** A boolean value indicating the availability of air conditioning.
- **TV:** A boolean value indicating the availability of a television.
- **Washer:** A boolean value indicating the availability of a washer.
- **Parking:** A boolean value indicating the availability of parking.
- **Gym:** A boolean value indicating the availability of a gym.
- **Outdoor Spaces:** A boolean value indicating the availability of outdoor spaces.
- **Safety Features:** A boolean value indicating the presence of safety features.
- **Essentials:** A boolean value indicating the availability of essential amenities.

Credentials

- **Email (Primary key):** The email address associated with a user's account. It serves as the primary key for the User entity.
- **Password:** The password for a user's account, used for authentication and access control.

Relationships between Entities

The entities in the project have several relationships between them. The relationships can be summarized as follows:

- A User can have multiple Listings, Bookings, Reviews, Messages, Payments, Wish Lists, Notifications and can have one Credentials and Payment Method.
- A Listing can have multiple Bookings, Reviews, Languages, and can be located in one Neighborhood, can have one set of Amenities, can have one set of House Rules, can have one Type of Property, can have one Location.
- A Booking can have one Guest and one Listing.
- A Review can have one Guest and one Listing, and can be left by one Host.
- A Message can be sent by one User and received by one User.
- A Payment can be made for one Booking.
- A Wish List can have one User and multiple Listings.
- A Neighborhood can have multiple Listings, one City, one State, one Country.
- A Notification can be received by one User.
- A City can be associated with one Neighborhood.
- A State can be associated with one Neighborhood.
- A Country can be associated with one Neighborhood.
- A Location can be associated with one Listing.
- House Rules can be associated with one Listing.
- Type of Property can be associated with one Listing.
- Amenities can be associated with one Listing.
- A Credential can have single User.
- A Payment Method can be associated with one User.

Types of Attributes

The attributes in the project can be classified into several types, which are:

- **Primary key:** An attribute that uniquely identifies an entity in a table.
- **Composite key:** A composite primary key is a primary key that consists of multiple attributes or columns, uniquely identifying a record in a table when combined. In the given entities, an example of a composite primary key is the combination of attributes Listing ID and Payment Method ID in the Payment Method entity.
- **Foreign key:** An attribute that refers to the primary key of another table. For example, Listing ID in the Booking, Review, Wish List, Language, and Neighborhood tables is a foreign key that refers to the primary key of the Listing table.
- **Char/Varchar:** An attribute that stores textual data. First name, Last name, Email address, Password, Phone number, Profile picture, Verification status, Title, Description, Type of property, Location, Address, Amenities, House rules, Payment method, Notification type, Notification content, Language name, Neighborhood name, State, Country are of char types.
- **Integer:** An attribute that stores numerical data. For example, Number of guests, Number of bedrooms, Number of beds, Number of bathrooms, Price per night, Reviews (total number, average rating), Rating (out of 5), Total cost, Amount, and Timestamp are numeric attributes.
- **Boolean:** An attribute that stores true or false values. Host status and Amenities and House Rules have boolean attributes.
- **Date:** An attribute that stores date values. For example, Check-in date, Check-out date, Date, and Availability calendar (start date, end date) are date attributes.
- **Timestamp:** An attribute that store the current time. For example, In Message and Notification; Created At are timestamp attributes

Advanced Database Features

- **Views:** We utilize views to create virtual tables that present selected data from our database in a structured and organized manner. Views allow for simplified data retrieval and enhanced query flexibility.
- **Stored Procedures:** We employ stored procedures to store a set of pre-defined SQL statements that can be executed whenever needed. These procedures help streamline complex database operations, improve performance, and promote code reusability.
- **Triggers:** Triggers are utilized in our project to automatically execute specific actions or operations in response to certain events or conditions in the database. Triggers help enforce data integrity, implement business rules, and automate tasks.
- **Functions:** We make use of functions to encapsulate specific logic or calculations that can be reused throughout our database. Functions enhance code modularity, improve readability, and simplify complex operations.
- **Joins:** In our project, we extensively leverage different types of joins, such as inner join, outer join, and cross join, to combine data from multiple tables based on common columns. Joins enable us to retrieve related data and establish relationships between entities.
- **Referential Actions:** Our project also involves the implementation of referential actions, such as On Delete Cascade, which ensures that when a record is deleted from a parent entity, related records in child entities are automatically deleted as well. This feature helps maintain data integrity by keeping the database consistent and eliminating orphaned records, providing a seamless and efficient data management process.

Conclusion

In conclusion, the project involves creating an Airbnb clone using a MySQL database. The entities involved in the project are User, Listing, Booking, Review, Message, Payment, Wish List, Neighborhood, Notification, and Language, each with their respective attributes. The relationships between the entities were also described, as well as the types of attributes used in the project. By creating a robust database management system, the project aims to provide a seamless experience for users and hosts, allowing them to easily manage their bookings, listings, messages, and payments.