Abstract

This document provides an overview of the database management system designed to support a platform similar to Airbnb, enabling efficient accommodation booking, user interactions, and transaction management. The database follows a structured relational model, using MySQL and the Inno DB storage engine for robustness and data integrity. It consists of multiple interconnected tables that manage accommodations, bookings, payments, disputes, complaints, user management, and support services.

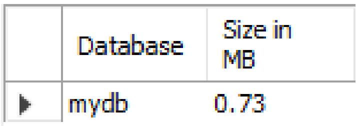
The system is structured to ensure scalability and efficiency. The Accommodation and Availability tables track listings and their availability, while User, Host, and Guest tables store relevant information about platform users. The Booking table links users with accommodations, integrating with Payment, Commission, and Discount tables to manage financial transactions. Dispute resolution is performed through the Complaint, Dispute, and Blacklist tables, ensuring user security and regulatory compliance. Additionally, the Support_Ticket table enables customer service operations.

Normalization techniques, including third normal form (3NF), were applied to minimize redundancy and optimize performance. The use of foreign keys enforces referential integrity between tables, preventing orphaned records and maintaining consistency. Indexing strategies improve query efficiency, particularly in high-traffic scenarios where rapid data retrieval is crucial for user interactions.

Metadata and Database Size

The system currently comprises 20 tables, with a total of 420 entries across all relations. The total database size is approximately 0.73 MB, calculated using:

```sql
SELECT table_schema "Database",
       ROUND(SUM(data_length + index_length) / 1024 / 1024, 2) "Size in MB"
FROM information_schema.tables
WHERE table_schema = 'mydb';
```



	Database	Size in MB
▶	mydb	0.73

This ensures transparency in storage requirements and helps optimize resource allocation. Regular maintenance, such as index optimization and archiving, is essential for sustained performance as the dataset grows.

The implemented database design provides a reliable and structured environment for managing accommodations, financial transactions, and user interactions. It ensures data integrity, security, and scalability, supporting the operational needs of an Airbnb-like system while maintaining efficiency in storage and retrieval processes.

Functionality Highlights

Key functionalities include:

- Secure user management, including host authentication and role-based access control.
- Listing and booking accommodations with real-time availability updates.
- Storing details about accommodations, such as descriptions, amenities, and pricing.
- Processing payments and applying discounts.
- Handling user disputes, complaints, and support tickets.
- Managing host commissions and financial transactions.
- Managing reservations, including check-in and check-out dates, statuses and payment details.
- Allows users to leave feedback and ratings for accommodations and hosts.
- Provides a structured way for users to report issues and receive assistance.

Conclusion

This implementation provides a well-structured and optimized database that ensures data consistency, query efficiency, and enables easy management of bookings, payments, and user interactions. Key functionalities include real-time availability tracking to prevent double bookings, a secure payment processing system with support for discounts and commissions, a review and rating system to enhance user trust, and a complaint and dispute resolution mechanism to ensure fair handling of conflicts. Additionally, the support ticket system allows users to report issues efficiently and receive adequate assistance, while blacklist management helps maintain platform security. These features collectively create a seamless experience for users, hosts, and administrators, ensuring smooth operations within the platform.