

# House Rental Database System

Group 2	
Ayush Mishra	002772710
Manogna Pallapothu	002776168
Padma Prasad Anaokar	002727445
Raushan Kumar Jha	002707535
Shrawani Karpe	002194600

## Business problems being addressed:

This project aims at managing and developing the database system of the House rental property business. As the database will be used by both business and their customers it will be best designed to suit both of their requirements. Since there are multiple attributes that contribute to this business our project will cover all different entities like users, property types, and relationships between them. It will also focus on best practices to create and design a database system for such type of business.

The purpose of this project is to create a detailed database system that will cover the different attributes to make an informed decision. The project will help to visualize the whole process with the help of EER. It can also be used to promote and share data efficiently. The project will focus on the functional aspects of creating a system that will be easy and powerful to store and retrieve the data and serves the business and customer needs.

## Database Design:

Following are the functionalities and key database design:

Different types of users can manage and find rental properties. Upon creating an account, users will be assigned a unique user ID and presented with two user type options: Landlord, Management Company.

Landlords can list their properties on the app by providing their details, such as name, property address, price, and address, and are assigned a unique ID number. They are then presented with two options: individually managed or driven by a management company. The first option is for landlords who manage their properties themselves, while the second option is for landlords who want a management company to take over all the management work for their property.

Management companies, on the other hand, are required to provide their name, address, and company ID. They have tie-ups with a number of broker firms and take over all property responsibilities, including complaints, repairs, cleanings, and all other maintenance duties.

Broker firms have their own unique agency ID, name, address, and brokerage rates. Each primary broker has their own broker ID, name, and work location, and may have an assistant. A brokerage firm can have one or more primary brokers associated with them, and a primary broker can have at most one assistant.

Customers looking for rentals can approach broker firms or management companies listed on the app, who have access to the properties listed by landlords. These entities help customers find suitable accommodation for their needs.

This is a database design that consists of multiple tables, each with its own primary key(s) and relationships with other tables through foreign keys. Here's a brief overview of the tables and their relationships:

1. **AppUser** - stores information about users of the app, including their unique User\_ID and Name.
2. **Landlords** - stores information about individual landlords, including their unique L\_ID, Name, Price, property\_address, and Landlord\_AppUser\_ID.  
Landlord\_AppUser\_ID is a foreign key that references User\_ID in the AppUser table, indicating the user account associated with the landlord.
3. **Management\_Company** - stores information about management companies, including their unique Company\_ID, Name, Address, and Management\_AppUser\_ID. Management\_AppUser\_ID is a foreign key that references User\_ID in the AppUser table, indicating the user account associated with the company.
4. **IndividuallyManaged\_Landlords** - stores information about landlords managed individually, including their unique IML\_ID, Name, and Address.
5. **ManagementDriven\_Landlords** - stores information about landlords managed by management companies, including their unique MDL\_ID, Name, and Address.
6. **BrokerFirm** - stores information about brokerage firms, including their unique Agency\_ID, Name, Address, and Brokerage\_Rates.
7. **Brokers** - stores information about brokers, including their unique Broker\_ID, Name, Work\_Location, and Agency\_ID. Agency\_ID is a foreign key that references Agency\_ID in the BrokerFirm table, indicating which brokerage firm the broker is associated with.
8. **AssistantBroker** - stores information about assistant brokers, including their unique ABroker\_ID, Name, and Broker\_ID. Broker\_ID is a foreign key that references Broker\_ID in the Brokers table, indicating which broker the assistant broker is assisting.
9. **Customer** - stores information about customers, including their unique Customer\_ID and Name.
10. **Property\_Details** – store information about Property and management company and act as associative entity.
11. **Assigns** - establishes a many-to-many relationship between Management\_Company and ManagementDriven\_Landlords tables, indicating which management company is assigned to manage which landlord. This table has two foreign keys: MDL\_ID references MDL\_ID in the ManagementDriven\_Landlords table, and Company\_ID references Company\_ID in the Management\_Company table.
12. **Assists** - establishes a many-to-many relationship between BrokerFirm and IndividuallyManaged\_Landlords tables, indicating which brokerage firm is assisting which individually managed landlord. This table has two foreign keys: Agency\_ID references Agency\_ID in the BrokerFirm table, and IML\_ID references IML\_ID in the IndividuallyManaged\_Landlords table.
13. **Works\_With** - establishes a many-to-many relationship between BrokerFirm and Management\_Company tables, indicating which brokerage firm is working with which management company. This table has two foreign keys: Agency\_ID references Agency\_ID in the BrokerFirm

table, and Company\_ID references Company\_ID in the Management\_Company table.

14. **Approaches** - establishes a many-to-many relationship between BrokerFirm and Customer tables, indicating which brokerage firm has approached which customer. This table has two foreign keys: Agency\_ID references Agency\_ID in the BrokerFirm table, and Customer\_ID references Customer\_ID in the Customer table.
15. **ReachesOut** - stores information about when a customer reaches out to a management company, including the unique Customer\_ID and Company\_ID. Customer\_ID is a foreign key that references Customer\_ID in the Customer table, indicating which customer reached out. Company\_ID is a foreign key that references Company\_ID in the Management\_Company table, indicating which management company was reached out to.

## Initial ERD:

