# The Analyze System to the Rental Housing Market

1. Introductuion

The rental housing market has always been one of the focuses of people's attention. In big cities, the rent is high and the difficulty of renting is increasing. In order to better understand the status of the rental housing market, analyzing rental data has become a common practice. With the development of big data technology and data analysis methods, more and more people are able to obtain and analyze rental data, and draw valuable insights and conclusions from it.

In this project, we aim to conduct an analysis of the rental housing market. I will use the Toronto rental housing market dataset provided by Airbub, to try to gain a deeper understanding of the rental market and provide useful insights. The project mainly including two parts. One is a CLI Application, and the other one is the data mining about rental housing market.

The results of the project will be presented in this document.

2. CLI Application

**2.1 Architecture Design**

The CLI Client is divided into two parts, the client part and the server part. In the client part it is a python CLI program, which will lead user to query and modify data. In the server part it is a MySQL database. It will accept commands sent by CLI program and return the query results to it. Fig. 1 shows the detailed architecture.

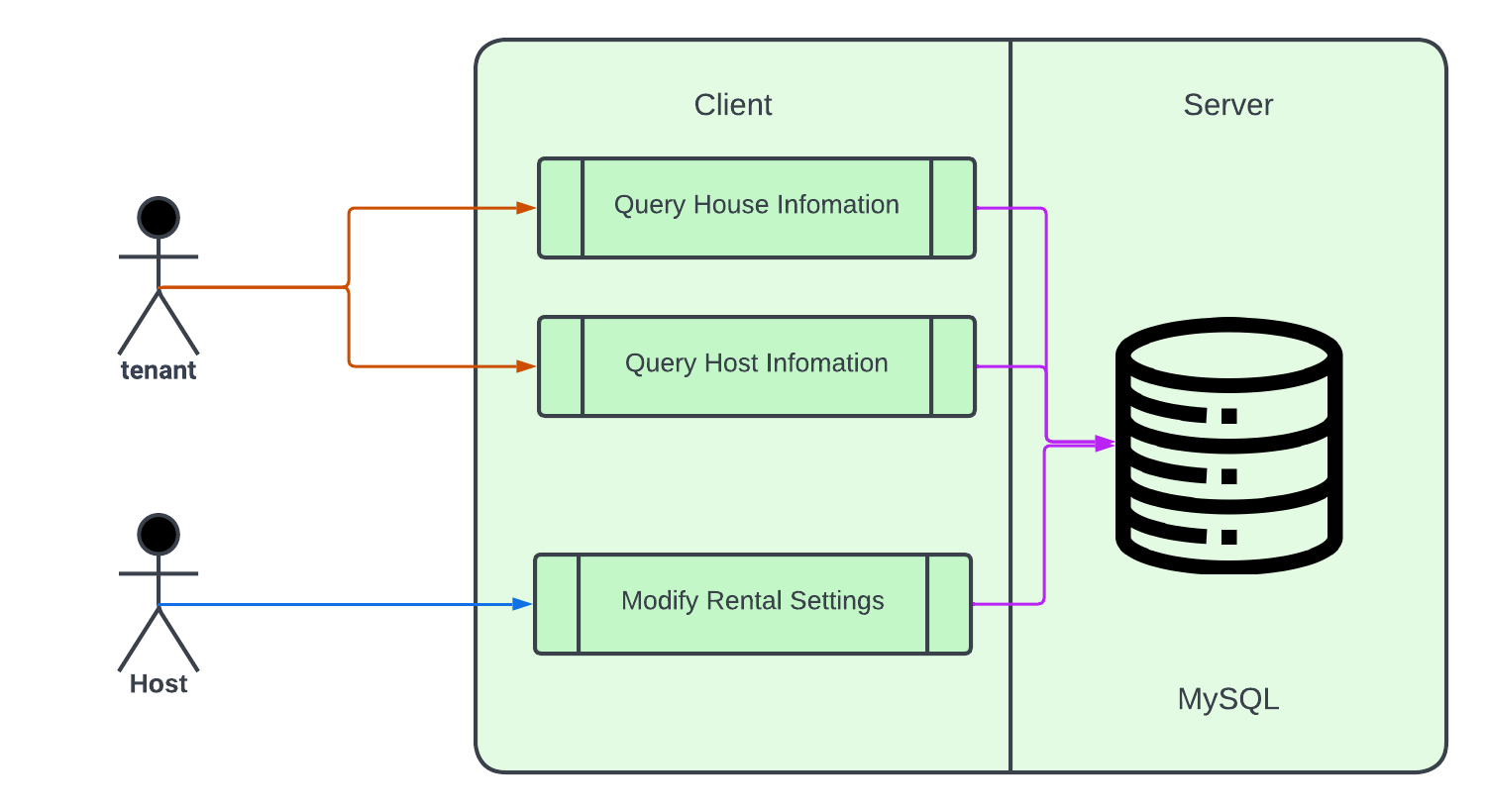
****

Fig. 1 The Client-Server architecture

The Client including sevenal functions. For tenant, you can query information about house, neighborhood, host. You can query reviews and calendar. For host, you can sign up as a host, and you can add rental information and modify rental information. Fig. 2 is the detailed flow of the client.

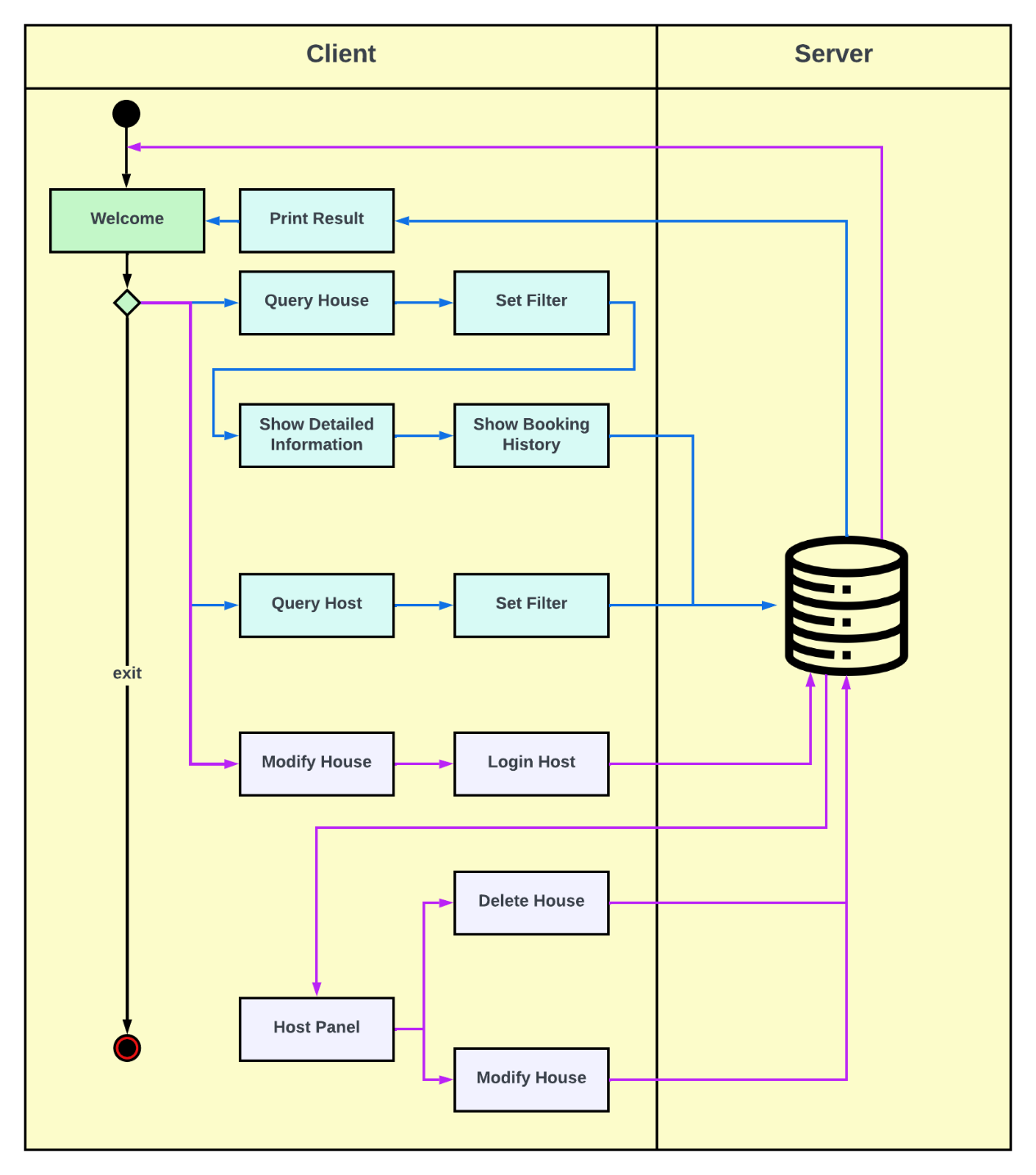


Fig. 2 Flow

When start the client, you can choose to query house, query host or login to host panel.

文本

描述已自动生成

If you press 1, it will enter housing market, and you need to fill the filters.

文本

描述已自动生成

It will show the results.

文本

描述已自动生成

You can input the index of result to show detailed information.

文本

描述已自动生成

Then you can press 1 to show the renting history of this house.

文本

中度可信度描述已自动生成

Else if you press 2, you can query the host list. You need to fill the filters.

文本

描述已自动生成

Then you can find the result of it.

文本

描述已自动生成

Else if you press 3, you can login in to host panel, and your house will list below.

文本

描述已自动生成

You can modify your house.

文本

描述已自动生成

Or you can delete your house.



**2.2 ER Model**

In this section, we create an ER model for our rental system, which includes some entities, attributes, constrains to show the relationship between different entities in our system, as shown in Fig. 3. The data for our model has been collected from Airbnb website, which contain the data of Toronto rental housing market. We cleaned some noise data in it.

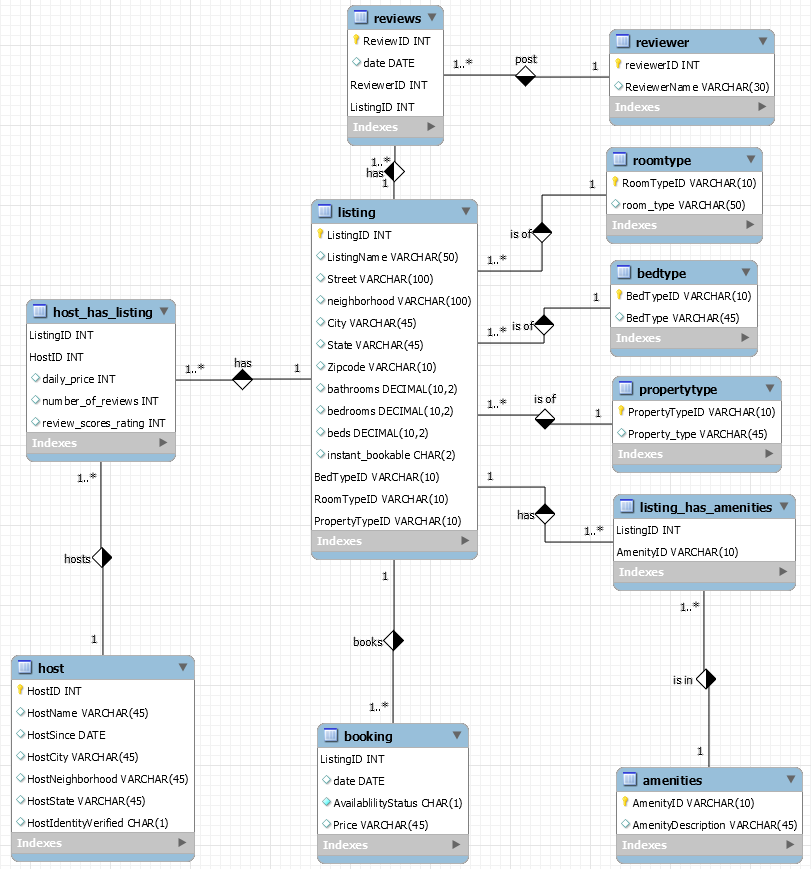


Fig. 3 ER Diagram

The entities present in the ER model are listing, booking, amenities, host, amenities, bedtype, propertytype, roomtype, reviews and reviewer.

Listing in this setting is the detailed information about house. It include many useful attributes. The primary key is ListingID. It has three foreign keys, which is BedTypeID, RoomTypeID, PropertyTypeID. And they refers to roomtype.RoomTypeID, propertytype.PropertyTypeID, bedtype.BedTypeID respectively.

Booking record the booking records of houses. The primary key of booking is ListingID, which is also a foreign key refers to listing.ListingID.

Amenities record the amenities a house have. It connect to listing by using a relationship “listing\_has\_amenities” to build a N-to-N connection. The primary key is AmenityID.

Bedtype, propertytype, roomtype record several kinds of types about bed, property and rooms for house. Their primary key is bedtypeid, propertytypeid, roomtypeid respectively.

Host record information about landlord. The primary key is HostID. It connects listing by using a relationship “host\_has\_listing” to build a 1-to-N connection.

Reviews record the information of reivews for houses and the primary key is reviewID. It has two foreign keys. one is listing id refers to listing.ListingID and the other is reviewerid refers to reviewer.reviewerid.

Reviewers record the information of reviewers. The primary key is Reviewerid.

**2.3 Relational Model**

图形用户界面, 文本, 应用程序

描述已自动生成

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图形用户界面

描述已自动生成

图形用户界面, 文本, 应用程序

描述已自动生成

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图形用户界面, 文本, 应用程序

描述已自动生成

图形用户界面, 应用程序

描述已自动生成

图形用户界面, 应用程序

描述已自动生成

图形用户界面, 文本, 应用程序

描述已自动生成

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

**2.4 Relational Schema**

According to our ER model, we created a relational database with normalized entities to integrate the information of rental housing market. We will also adopt an appropriate data type to store our data and then develop some SQL code to establish all the SQL objects. For instance, we provide a translation of the ER model to a relational model by writing the necessary SQL codes to create a database, several tables, primary keys, and foreign keys as follows.

*CREATE DATABASE IF NOT EXISTS `airbnb` /\*!40100 DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci \*/ /\*!80016 DEFAULT ENCRYPTION='N' \*/;*

*USE airbnb;*

*DROP TABLE IF EXISTS `amenities`;*

*CREATE TABLE `amenities` (*

*`AmenityID` varchar(10) NOT NULL,*

*`AmenityDescription` varchar(45) DEFAULT NULL,*

*PRIMARY KEY (`AmenityID`)*

*) ENGINE=InnoDB DEFAULT CHARSET=utf8;*

*DROP TABLE IF EXISTS `bedtype`;*

*CREATE TABLE `bedtype` (*

*`BedTypeID` varchar(10) NOT NULL,*

*`BedType` varchar(45) DEFAULT NULL,*

*PRIMARY KEY (`BedTypeID`)*

*) ENGINE=InnoDB DEFAULT CHARSET=utf8;*

*DROP TABLE IF EXISTS `booking`;*

*CREATE TABLE `booking` (*

*`ListingID` int NOT NULL,*

*`date` date DEFAULT NULL,*

*`AvailablilityStatus` char(1) NOT NULL,*

*`Price` varchar(45) DEFAULT NULL,*

*PRIMARY KEY (`ListingID`),*

*CONSTRAINT `fk\_Booking\_Listing1` FOREIGN KEY (`ListingID`) REFERENCES `listing` (`ListingID`)*

*) ENGINE=InnoDB DEFAULT CHARSET=utf8;*

*DROP TABLE IF EXISTS `host`;*

*CREATE TABLE `host` (*

*`HostID` int NOT NULL,*

*`HostName` varchar(45) DEFAULT NULL,*

*`HostSince` date DEFAULT NULL,*

*`HostCity` varchar(45) DEFAULT NULL,*

*`HostNeighborhood` varchar(45) DEFAULT NULL,*

*`HostState` varchar(45) DEFAULT NULL,*

*`HostIdentityVerified` char(1) DEFAULT NULL,*

*PRIMARY KEY (`HostID`)*

*) ENGINE=InnoDB DEFAULT CHARSET=utf8;*

*DROP TABLE IF EXISTS `host\_has\_listing`;*

*CREATE TABLE `host\_has\_listing` (*

*`ListingID` int NOT NULL,*

*`HostID` int NOT NULL,*

*`daily\_price` int DEFAULT NULL,*

*`number\_of\_reviews` int DEFAULT NULL,*

*`review\_scores\_rating` int DEFAULT NULL,*

*PRIMARY KEY (`ListingID`,`HostID`),*

*KEY `fk\_Host\_has\_Listing\_Host1\_idx` (`HostID`),*

*KEY `fk\_Host\_has\_Listing\_Listing1\_idx` (`ListingID`),*

*CONSTRAINT `fk\_Host\_has\_Listing\_Host1` FOREIGN KEY (`HostID`) REFERENCES `host` (`HostID`),*

*CONSTRAINT `fk\_Host\_has\_Listing\_Listing1` FOREIGN KEY (`ListingID`) REFERENCES `listing` (`ListingID`)*

*) ENGINE=InnoDB DEFAULT CHARSET=utf8;*

*DROP TABLE IF EXISTS `listing`;*

*CREATE TABLE `listing` (*

*`ListingID` int NOT NULL,*

*`ListingName` varchar(50) DEFAULT NULL,*

*`Street` varchar(100) DEFAULT NULL,*

*`neighborhood` varchar(100) DEFAULT NULL,*

*`City` varchar(45) DEFAULT NULL,*

*`State` varchar(45) DEFAULT NULL,*

*`Zipcode` varchar(10) DEFAULT NULL,*

*`bathrooms` decimal(10,2) DEFAULT NULL,*

*`bedrooms` decimal(10,2) DEFAULT NULL,*

*`beds` decimal(10,2) DEFAULT NULL,*

*`instant\_bookable` char(2) DEFAULT NULL,*

*`BedTypeID` varchar(10) NOT NULL,*

*`RoomTypeID` varchar(10) NOT NULL,*

*`PropertyTypeID` varchar(10) NOT NULL,*

*PRIMARY KEY (`ListingID`,`BedTypeID`,`RoomTypeID`,`PropertyTypeID`),*

*KEY `fk\_Listing\_Bed Type1\_idx` (`BedTypeID`),*

*KEY `fk\_Listing\_Room Type1\_idx` (`RoomTypeID`),*

*KEY `fk\_Listing\_Property Type1\_idx` (`PropertyTypeID`),*

*CONSTRAINT `fk\_Listing\_Bed Type1` FOREIGN KEY (`BedTypeID`) REFERENCES `bedtype` (`BedTypeID`),*

*CONSTRAINT `fk\_Listing\_Property Type1` FOREIGN KEY (`PropertyTypeID`) REFERENCES `propertytype` (`PropertyTypeID`),*

*CONSTRAINT `fk\_Listing\_Room Type1` FOREIGN KEY (`RoomTypeID`) REFERENCES `roomtype` (`RoomTypeID`)*

*) ENGINE=InnoDB DEFAULT CHARSET=utf8;*

*DROP TABLE IF EXISTS `listing\_has\_amenities`;*

*CREATE TABLE `listing\_has\_amenities` (*

*`ListingID` int NOT NULL,*

*`AmenityID` varchar(10) NOT NULL,*

*PRIMARY KEY (`ListingID`,`AmenityID`),*

*KEY `fk\_Listing\_has\_Amenities\_Amenities1\_idx` (`AmenityID`),*

*KEY `fk\_Listing\_has\_Amenities\_Listing1\_idx` (`ListingID`),*

*CONSTRAINT `fk\_Listing\_has\_Amenities\_Amenities1` FOREIGN KEY (`AmenityID`) REFERENCES `amenities` (`AmenityID`),*

*CONSTRAINT `fk\_Listing\_has\_Amenities\_Listing1` FOREIGN KEY (`ListingID`) REFERENCES `listing` (`ListingID`)*

*) ENGINE=InnoDB DEFAULT CHARSET=utf8;*

*DROP TABLE IF EXISTS `propertytype`;*

*CREATE TABLE `propertytype` (*

*`PropertyTypeID` varchar(10) NOT NULL,*

*`Property\_type` varchar(45) DEFAULT NULL,*

*PRIMARY KEY (`PropertyTypeID`)*

*) ENGINE=InnoDB DEFAULT CHARSET=utf8;*

*DROP TABLE IF EXISTS `reviewer`;*

*CREATE TABLE `reviewer` (*

*`reviewerID` int NOT NULL,*

*`ReviewerName` varchar(30) DEFAULT NULL,*

*PRIMARY KEY (`reviewerID`)*

*) ENGINE=InnoDB DEFAULT CHARSET=utf8;*

*DROP TABLE IF EXISTS `reviews`;*

*CREATE TABLE `reviews` (*

*`ReviewID` int NOT NULL,*

*`date` date DEFAULT NULL,*

*`ReviewerID` int NOT NULL,*

*`ListingID` int NOT NULL,*

*PRIMARY KEY (`ReviewID`,`ReviewerID`,`ListingID`),*

*KEY `fk\_Reviews\_Reviewer1\_idx` (`ReviewerID`),*

*KEY `fk\_Reviews\_Listing1\_idx` (`ListingID`),*

*CONSTRAINT `fk\_Reviews\_Listing1` FOREIGN KEY (`ListingID`) REFERENCES `listing` (`ListingID`),*

*CONSTRAINT `fk\_Reviews\_Reviewer1` FOREIGN KEY (`ReviewerID`) REFERENCES `reviewer` (`reviewerID`)*

*) ENGINE=InnoDB DEFAULT CHARSET=utf8;*

*DROP TABLE IF EXISTS `roomtype`;*

*CREATE TABLE `roomtype` (*

*`RoomTypeID` varchar(10) NOT NULL,*

*`room\_type` varchar(50) DEFAULT NULL,*

*PRIMARY KEY (`RoomTypeID`)*

*) ENGINE=InnoDB DEFAULT CHARSET=utf8;*

**2.5 Data Importing**

A python program is provided to import data from csv files to the database. In order to use it, you need to change the host, user and password in the program. We also provide dataset. If you want to use it you need to put them in ./data/

**2.6 Testcases**

Test Objective: Test database connection

Test Cases:

Start the program, expecting welcome information printed.

Test Objective: Test house\_info query

Test Cases:

Input null for all filter, expected output is correct selected result.

Input existing id and null for other filter, expected correct selected result.

Input not existing id and null for other filter, expected null result.

Input existing name and null for other filter, expected correct selected result.

Input not existing name and null for other filter, expected null result.

Input existing neighborhood and null for other filter, expected correct selected result.

Input not existing neighborhood and null for other filter, expected null result.

Input existing instance\_bookable and null for other filter, expected correct selected result.

Input not existing instance\_bookable and null for other filter, expected null result.

Input existing bedrooms and null for other filter, expected correct selected result.

Input not existing bedrooms and null for other filter, expected null result.

Input existing bathrooms and null for other filter, expected correct selected result.

Input not existing bathrooms and null for other filter, expected null result.

Test Objective: Show details of house.

Test Cases:

Input existing index of result, expect detailed information of it.

Test Objective: Show house book history.

Test Cases:

Input 1 to show book history, expect book history shown.

Test Objective: Test host query

Test Cases:

Input null for all filter, expected output is correct selected result.

Input existing id and null for other filter, expected correct selected result.

Input not existing id and null for other filter, expected null result.

Input existing name and null for other filter, expected correct selected result.

Input not existing name and null for other filter, expected null result.

Input existing since and null for other filter, expected correct selected result.

Input not existing since and null for other filter, expected null result.

Input existing verified and null for other filter, expected correct selected result.

Input not existing verified and null for other filter, expected null result.

Test Objective: Login to host panel

Test Cases:

Input existing host id, expected owned house information.

Input existing host id, expected exception.

Test Objective: Change house information

Test Cases:

Input not existing index, expected output “Index don't exist”.

Input existing index and information it asked, expected output “Change house successfully”.

Test Objective: Delete house

Test Cases:

Input not existing index, expected output “Index don't exist”.

Input existing index and information it asked, expected output “Deleted Successfully!”.

3. Data Mining

**3.1 Data Cleaning**

表格

描述已自动生成

表格

描述已自动生成

表格

描述已自动生成

表格

描述已自动生成

表格

中度可信度描述已自动生成

表格

描述已自动生成

文本

中度可信度描述已自动生成

图表, 直方图

描述已自动生成

图形用户界面, 应用程序

描述已自动生成

图表, 直方图

描述已自动生成

图表, 直方图

描述已自动生成

表格

中度可信度描述已自动生成

表格

描述已自动生成



图表, 直方图

描述已自动生成

图表, 直方图

描述已自动生成

图表

描述已自动生成

图表, 折线图

描述已自动生成

手机屏幕截图

描述已自动生成

表格

描述已自动生成

图片包含 表格

描述已自动生成

手机屏幕截图

描述已自动生成

**3.2 Listing Exploration**

表格

描述已自动生成

图表, 折线图, 直方图

描述已自动生成

图表

描述已自动生成

图表, 条形图

描述已自动生成

图表, 条形图

描述已自动生成

文本

描述已自动生成

表格

描述已自动生成





图形用户界面, 图表, 散点图

描述已自动生成

图表, 散点图

描述已自动生成





图表, 散点图

描述已自动生成

图表

描述已自动生成





图形用户界面, 图表, 散点图

描述已自动生成

图表, 散点图

描述已自动生成



图形用户界面, 图表, 散点图

描述已自动生成

图表

描述已自动生成



图表, 散点图

描述已自动生成

图表, 折线图

描述已自动生成



文本

描述已自动生成





文本

描述已自动生成

