# 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.

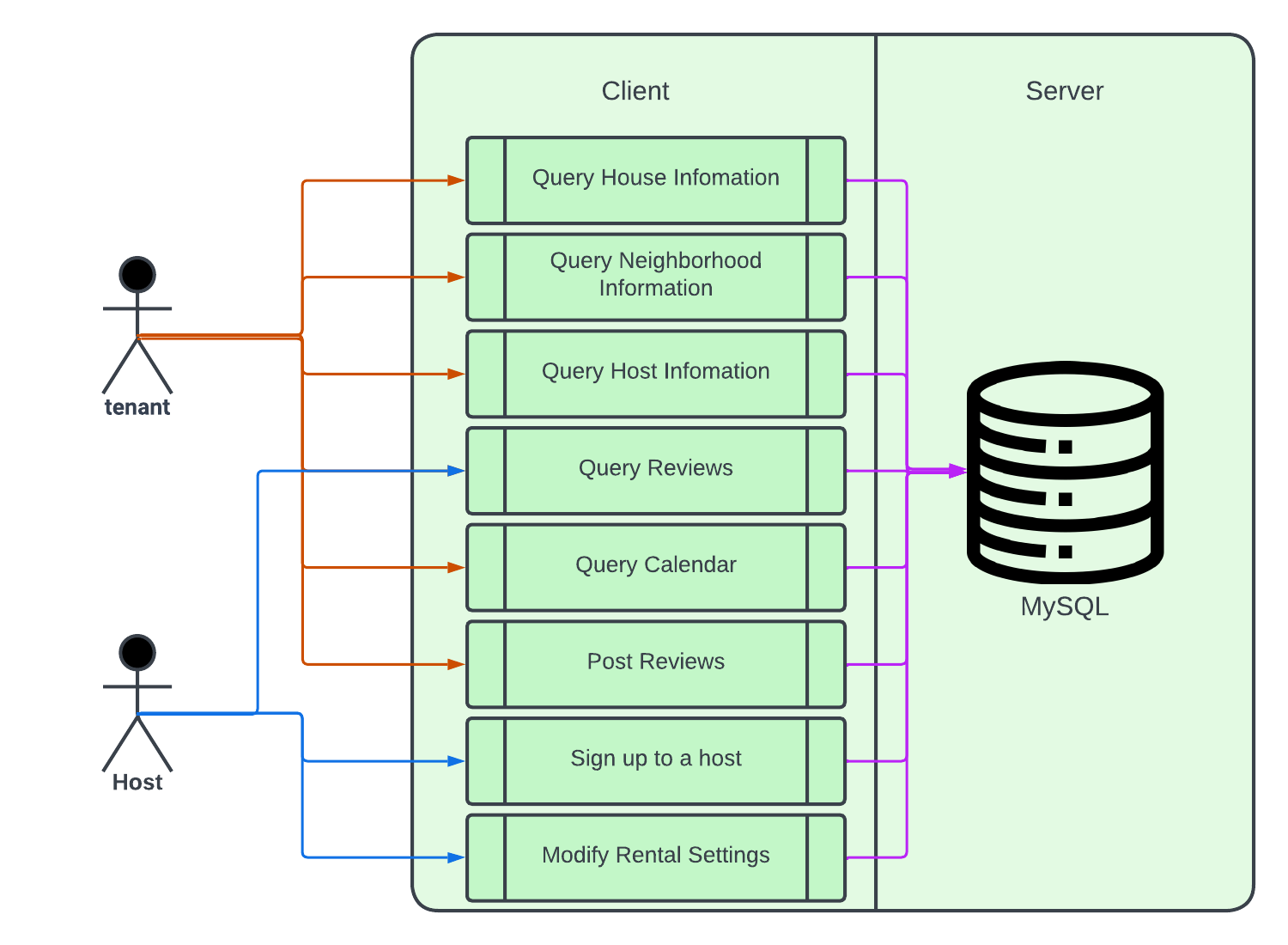
****

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.

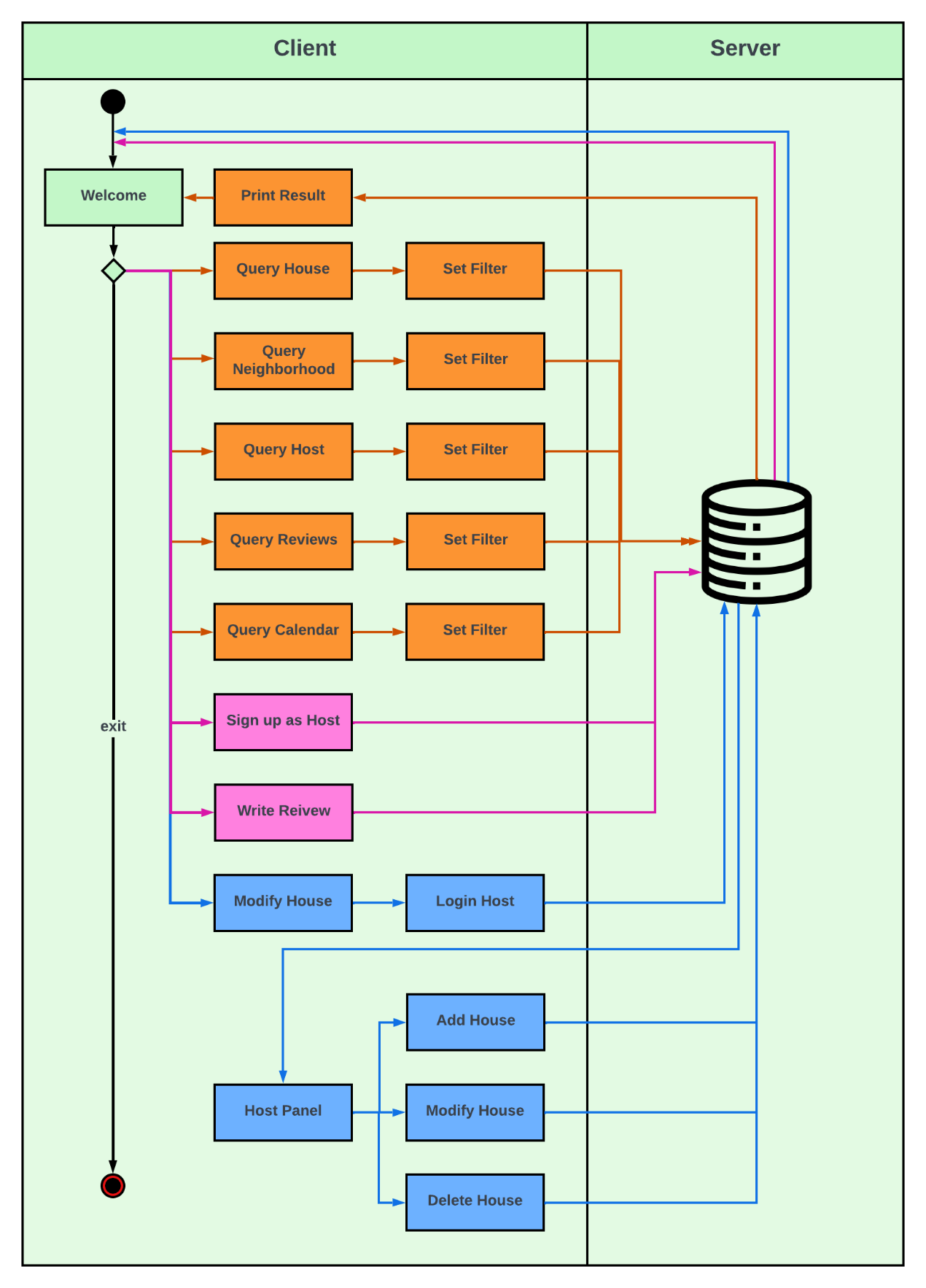


Fig. 2 Flow

**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.

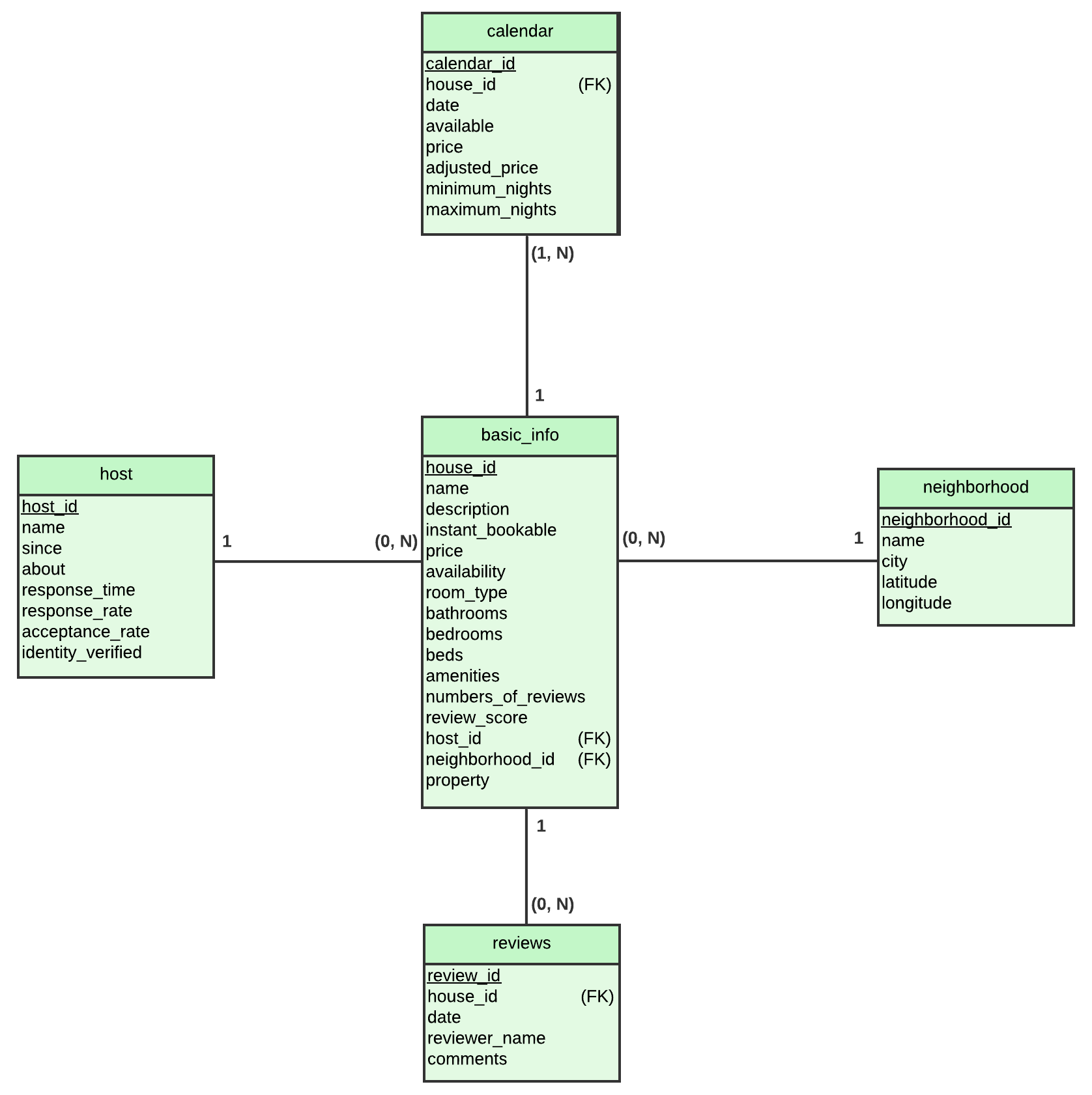


Fig. 3 ER Diagram

**2.3 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 `656project` /\*!40100 DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci \*/ /\*!80016 DEFAULT ENCRYPTION='N' \*/;*

*USE 656project;*

*DROP TABLE IF EXISTS `neighborhood`;*

*CREATE TABLE `neighborhood` (*

*`neighborhood\_id` bigint NOT NULL AUTO\_INCREMENT,*

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

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

*`latitude` double DEFAULT NULL,*

*`longitude` double DEFAULT NULL,*

*PRIMARY KEY (`neighborhood\_id`),*

*KEY `name` (`name`)*

*) ENGINE=InnoDB AUTO\_INCREMENT=418 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;*

*DROP TABLE IF EXISTS `host`;*

*CREATE TABLE `host` (*

*`host\_id` bigint NOT NULL,*

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

*`since` date DEFAULT NULL,*

*`about` text,*

*`response\_time` varchar(45) DEFAULT NULL,*

*`response\_rate` varchar(45) DEFAULT NULL,*

*`acceptance\_rate` varchar(45) DEFAULT NULL,*

*`identity\_verified` varchar(5) DEFAULT NULL,*

*PRIMARY KEY (`host\_id`),*

*KEY `name` (`name`)*

*) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;*

*DROP TABLE IF EXISTS `basic\_info`;*

*CREATE TABLE `basic\_info` (*

*`house\_id` bigint NOT NULL,*

*`name` text CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci,*

*`description` text CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci,*

*`instant\_bookable` varchar(5) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci DEFAULT NULL,*

*`price` int DEFAULT NULL,*

*`availability` int DEFAULT NULL,*

*`room\_type` varchar(20) DEFAULT NULL,*

*`bathrooms` varchar(20) DEFAULT NULL,*

*`bedrooms` int DEFAULT NULL,*

*`beds` int DEFAULT NULL,*

*`amenities` text,*

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

*`review\_score` float DEFAULT NULL,*

*`host\_id` bigint DEFAULT NULL,*

*`neighborhood\_id` bigint DEFAULT NULL,*

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

*PRIMARY KEY (`house\_id`),*

*KEY `host\_id\_idx` (`host\_id`),*

*KEY `neighborhood\_id\_idx` (`neighborhood\_id`),*

*CONSTRAINT `host\_id` FOREIGN KEY (`host\_id`) REFERENCES `host` (`host\_id`),*

*CONSTRAINT `neighborhood\_id` FOREIGN KEY (`neighborhood\_id`) REFERENCES `neighborhood` (`neighborhood\_id`)*

*) ENGINE=InnoDB DEFAULT CHARSET=utf8;*

*DROP TABLE IF EXISTS `reviews`;*

*CREATE TABLE `reviews` (*

*`review\_id` bigint NOT NULL,*

*`house\_id` bigint DEFAULT NULL,*

*`date` date DEFAULT NULL,*

*`reviewer\_name` varchar(20) DEFAULT NULL,*

*`comments` text,*

*PRIMARY KEY (`review\_id`),*

*KEY `house\_id\_idx` (`house\_id`),*

*KEY `date` (`date`),*

*CONSTRAINT `house\_id` FOREIGN KEY (`house\_id`) REFERENCES `basic\_info` (`house\_id`)*

*) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;*

*DROP TABLE IF EXISTS `calendar`;*

*CREATE TABLE `calendar` (*

*`calendar\_id` int NOT NULL AUTO\_INCREMENT,*

*`house\_id` bigint DEFAULT NULL,*

*`date` date DEFAULT NULL,*

*`available` varchar(5) DEFAULT NULL,*

*`price` int DEFAULT NULL,*

*`adjusted\_price` int DEFAULT NULL,*

*`minimum\_nights` int DEFAULT NULL,*

*`maximum\_nights` int DEFAULT NULL,*

*PRIMARY KEY (`calendar\_id`),*

*KEY `house\_id\_idx` (`house\_id`),*

*KEY `date` (`date`),*

*CONSTRAINT `\_house\_id` FOREIGN KEY (`house\_id`) REFERENCES `basic\_info` (`house\_id`)*

*) ENGINE=InnoDB AUTO\_INCREMENT=3016467 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;*

**2.4 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/

**2.5 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 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 room\_type and null for other filter, expected correct selected result.

Input not existing room\_type and null for other filter, expected null result.

Test Objective: Test neighborhood 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 city and null for other filter, expected correct selected result.

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

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: Test reviews query

Test Cases:

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

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

Input existing house id and existing date, expected correct selected result.

Input existing house id and not existing date, expected null result.

Test Objective: Test calendar query

Test Cases:

Input existing house id and existing date, expected correct selected result.

Input existing house id and not existing date, expected null result.

Input not existing house id and existing date, 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: Sign up as host

Test Cases:

Input name and description, expected output “sign up successfully”.

Test Objective: Post review

Test Cases:

Input existing house\_id, name and comment, expected output “Post review successfully”.

Input not existing house\_id, name and comment, expected exception.

Test Objective: Add house

Test Cases:

Input information it asked, expected output “Add new house successfully”.

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