

**Symbiosis Institute of Technology**

**A DBMS Project Report on**

REAL ESTATE RESIDENTIAL PROPERTY MANAGEMENT SYSTEM

Submitted by

Tripti Singh 18070124074

Samarth Srivastava 18070124070

Under the Guidance of

**Prof. Shruti Patil**

# **Department of Information Technology**

**SYMBIOSIS INSTITUTE OF TECHNOLOGY, PUNE**

INDEX

|  |  |
| --- | --- |
| 1. Introduction-Problem Statement |  |
| 2. Objectives |  |
| 3. System Architecture |  |
| 4. Functionality and Module Description |  |
| 5. Functional Requirements |  |
| 6. Entities and their relationships. |  |
| 7. E-R /EER diagram |  |
| 8. Relational schema |  |
| 9. Keys |  |
| 10. Codd’s Rule |  |
| 11. Kind of anomalies in relational schema |  |
| 12. Functional dependencies of each relation |  |
| 13. Functional dependency charts of each relation |  |
| 14. Normalization of relational schema |  |
| 15. Database Implementation |  |
| 16. Query Execution |  |
| 17. Functions |  |
| 18. Procedures |  |
| 19. Triggers |  |
| 20. Views |  |

INTRODUCTION

The residential market in India has cumulative sales of residential apartments standing at 2,44,830 units and new apartment launches at 2,33,387 units**.** The housing sector, which currently contributes to nearly 5-6% of India’s [GDP](https://indiancompanies.in/india-gdp-gross-domestic-product/)**,** will nearly double its contribution to an estimated 11.2% by 2021.

Yet, it becomes cumbersome to search for the house of your dreams. It is not an unheard truth that word of mouth and tiresome day visits to finalise a property are the traditional methods of booking a house.

Our project envisions to revolutionize this by introducing a smart online real estate management system for renting, selling, or booking a property. This is the perfect software for anyone who is looking to relocate themselves; be it a student, a bachelor, or a beloved family.

It is a healthy and safe space that aggregates the best of real estate agents catering you to the ultra-luxury residential segment from the comfort of your homes. We ensure a happy housing for all. Real Estate is the business of buying, selling, and renting land, buildings, and offices. Every person wants his house should be best in a location with all facilities. General companies concentrate on its product and infrastructure are bought from Real Estate company. Due to real estate company wants to competition in this industry, every deliver the best service and makes its customer satisfied.

Online Property Management System is a software which takes care of everything that a real estate company want to do. Property Management System is developed to mediate real estate Companies introducing 0Brokerge cost for the customers

It is very strong and easy to use that makes quick booking and account handling process. The real estate Property management system is a web-based software and you can access it from anywhere.

This ensures the service to client 24X7. Clients can view and book their favourite property online just after few clicks. They can also pay the money online and get bills online.

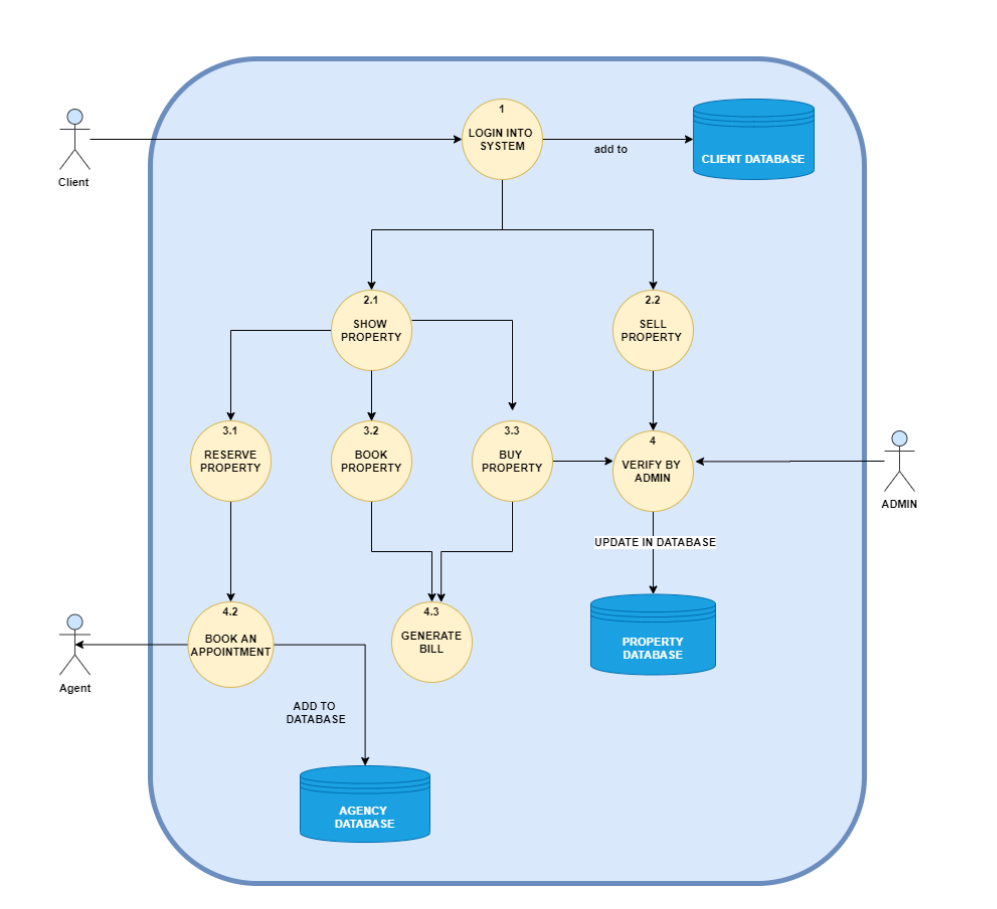
OBJECTIVES

This project is computerizing the manual process of booking a real estate. Front end is implemented using MySQL. The objective of this project is to help in booking or selling of a real estate residential property online. Online payment and verification make it more client friend. This system would definitely go to reduce labour and make business more profitable and promising to clients.

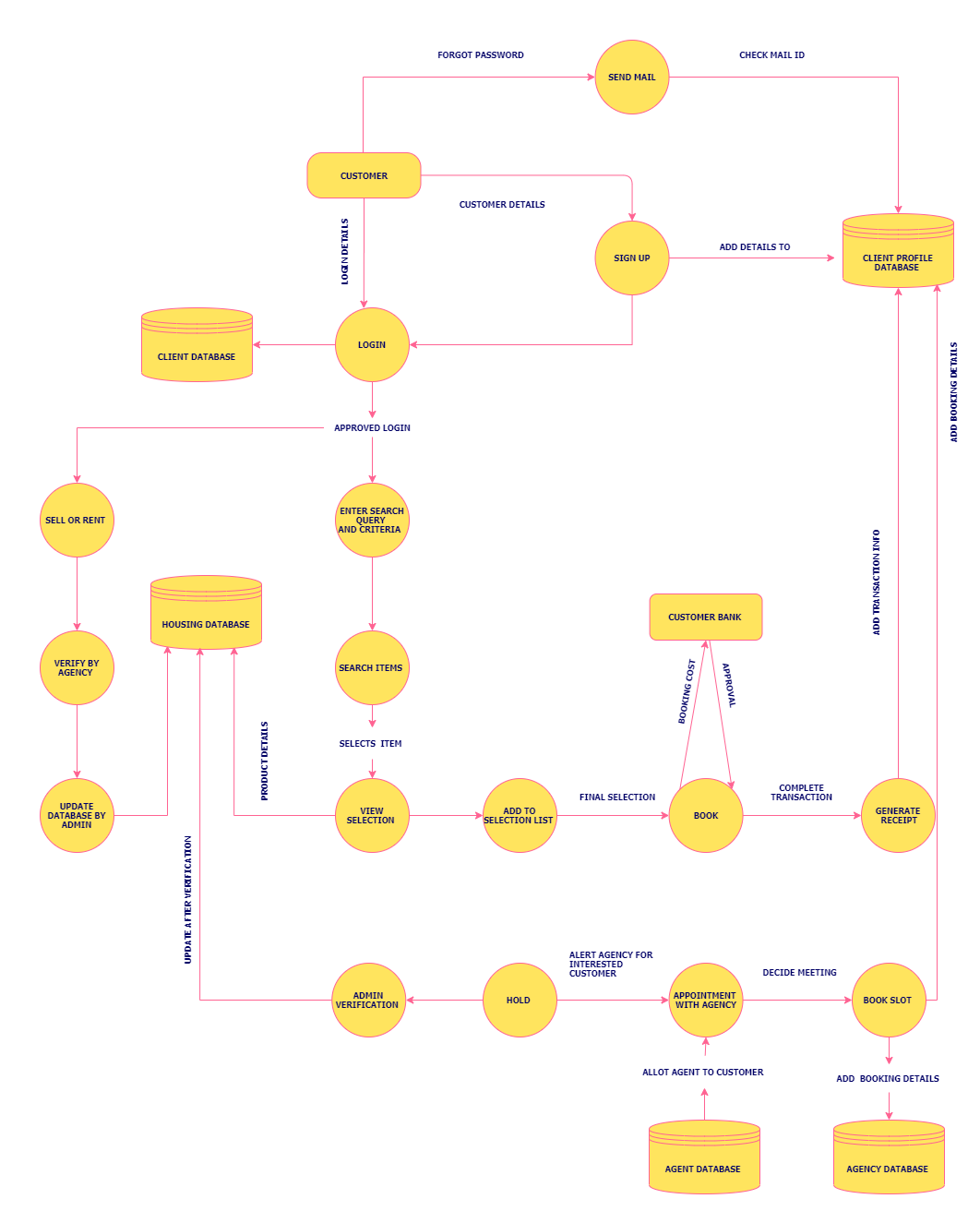
The services provided by our software solution are:

* User can book a property of his choice from the catalogue of choices in our database virtually.
* Authorised user (Owner) can put up his property for sale on our portal to get potential buyers or renters.
* Security of deposit and safe booking is ensured. No two people can book for the same place. Hence, ensures fraud control by dealers by avoiding any type of ambiguity in transactions.
* Real Estate Agents can collaborate with us to make a deal with a suitable or interested user. We collaborate with independent can property dealers to provide the best
* The user has a personalised profile with the updates on the selling and selected house status. The client can login using user id and password.
* The client can contact the agency at any time in case of a query.
* We ensure that the property displayed in our system is verified and recognised by the government.
* All transactions and entry in database are verified by admin.
* We offer a unique feature to the clients where they can reserve a house, book a meeting with agency have a good look at the property and then proceed to book the house.
* The client can customise his/her search to get the best residential property by choosing from a variety of filters that we offer.

SYSTEM DESIGN OF REAL ESTATE MANAGEMENT SYSTEM



DATA FLOW DETAILED DIAGRAM OF REAL ESTATE MANAGEMENT SYSTEM



FUNCTIONALITY DESCRIPTION

The system has three main users i.e. Admin, Agency, and Clients.

The Real Estate Management System has Admin who manages the workflow and integrity of the entire system.

The users can log into the system using user id and password. When security is verified these users can make then change which needs to be the one in the system.

Admin can change the details of property if needed or if there are any changes. Search the property to recommend the clients.

Search is based on property type, value, location, or status. The admin would manage the client’s details handle their query. The manager would also view the information of clients and if required can edit it too.

Admin has the right to view payment details in payment table. Any issue related to payment is handled by admin.

Our second end user is a client. Clients can log in the system and verify their identity. Clients can view the listed property. Clients can also sort and save their favourite property to buy or rent.

Clients need to verify their identity before they book any property. When their identity is verified, they can book and make the payment. System generated receipt would send to their email.

MODULE DESCRIPTION

The core modules of the system are discussed below:

### ***Admin Profile:***

Admin profile is assigned to a superuser having full access to the system. Admin module contains Admin Id, name, address, contact no. Admin can view the property details, property status and update, delete, modify as per requirement.

Admin can check the payment status, view, and manage the client details. Any issue in client’s details or in property details gets reported to the admin.

### ***Client Profile:***

The client profile is for those who want to buy, sell, or rent the property for their uses. Client profile contains Client id, name, address, contact details, etc. Clients can view property. They can personalise their searches with enlisted categories to get the best possible choice for their search.

The client can either buy or rent an available house. The client can book online if he/she is satisfied with the proceedings and the details listed or he can proceed to view the property physically by setting up the schedule with the agency and then proceed to book . They can make payment online or offline.

### ***Housing Details:***

House Property details hold the data about the property in property location, total value, owner etc. The property gives an idea about the property which helps the clients to choose the property.

### ***Search Property***

To customize the search as per needs we provide a search and filter option. Clients may be confused to search their desired property. Clients can search property by area, the basic amenities, the facilities available and more on. Property can be searched by property status by the admin. The users are only presented with available properties.

Property can also be searched using property value and much more. If there is some legal problem with the property. Then it would be unavailable to rent or buy.

### ***Availability:***

Searched property can be viewed as available or not. The client can book the property only if it is available. Sold property can be hidden from the page. This would decrease the conflict between buyers and seller. Hence, the admin maintains the view level of properties for the buyer clients.

### ***Booking:***

Once the property is searched and the client finds desire property. This module helps the client to book the property. This module collects the information and checks the information from the backend.

If data is verified, then client redirect to the payment page.

### ***Payment Details:***

After filling the booking details clients need to pay the money using cash/net banking/ATM card.

Payment portal would use a payment gateway to clear payment. Once payment will clear receipt of payment will be generated automatically.

### ***Add / update / delete Users:***

Only admin can add, update, and delete the client details and others user details. Admin has special permission to do so. User authentication according to his/her role i.e. as a client, agent, or admin.

This module takes care of every profile is our system. No user is allowed to use our system without a profile. Every profile must authenticate using system procedure. This would ensure the security of our system.

### ***Add Owner Details:***

Admin can add details of the landlord to every property. As it would help the clients to reach landlord or owner directly only if owner and client are ready to do so.

### Add Employees / Business Associates/ Agents:

Real estate business needs a lot of manpower. This system would help to manage the manpower too. Admin can add employee details and their business associates. They are added to the agency database as agents.

FUNCTIONAL REQUIREMENTS

#### **Login/Sign up:**

Users can register their self in the system. Admin will verify their profile to work properly. Once a user is registered with the system, he/she can proceed to sell, buy or rent a property on the online portal.

Admin will view the property and verify that property only if that would be real with reasonable price.

#### **Validation:**

Validation is very important in the system. Invalid data can corrupt the valid data. So, we need to apply validation in each module. Validation would ensure the safety and security of data.

#### **Client Record:**

Client data should be secure. We should take care of clients’ data privacy in our mind. The client is the basic unit of our business. Client data and record would help us to provide them better information.

#### **Add Property:**

This function allows the admin and client to add the property details. But only admin can verify the property details. Without verification property, details would not reflect in the system.

#### **List of Property with details:**

All properties should be listed and reflected client. The client can view these listed properties and book the desired property. The client can also filter out the search to get the most suitable property matching the requirements.

#### **Payment options:**

Clients can pay the money or property via any mode they wish to do. They can pay via cash, credit cards, debit cards, online net banking, online wallets etc. The system should not restrict them to some payment option.

#### **Feedback and Suggestion:**

Feedback and suggestions are key to improve in business. Therefore, we need to take the feedback from our customers to improve our services.

### Non-functional Requirement of online property management system:

#### **Improve Search option:**

Search option should be rich enough to provide the property details to clients as per their requirement and wish. Search option should search in the whole database. Efficient search option attracts the client to buy and rent the property

#### **Support Multi-User:**

Nowadays, every system work in an environment of multi-users. As per requirement of the system, our system is developed in multi user’s environment.

#### **Fast:**

Search option should be fast enough to produce a result of the search in seconds. Every module should to compatible with other modules. There should not be any lag or delay in processing the data.

#### **Available(24X7):**

Property Management System should be available to clients 24X7. The user can access the system whenever they need to access, wherever they want to access provided having an internet connection and device to access.

# IDENTIFYING ENTITIES AND THEIR RELATIONSHIPS

ENTITIES:

* Client
* Admin
* Agent
* House
* Portal
* Agency
* Transaction

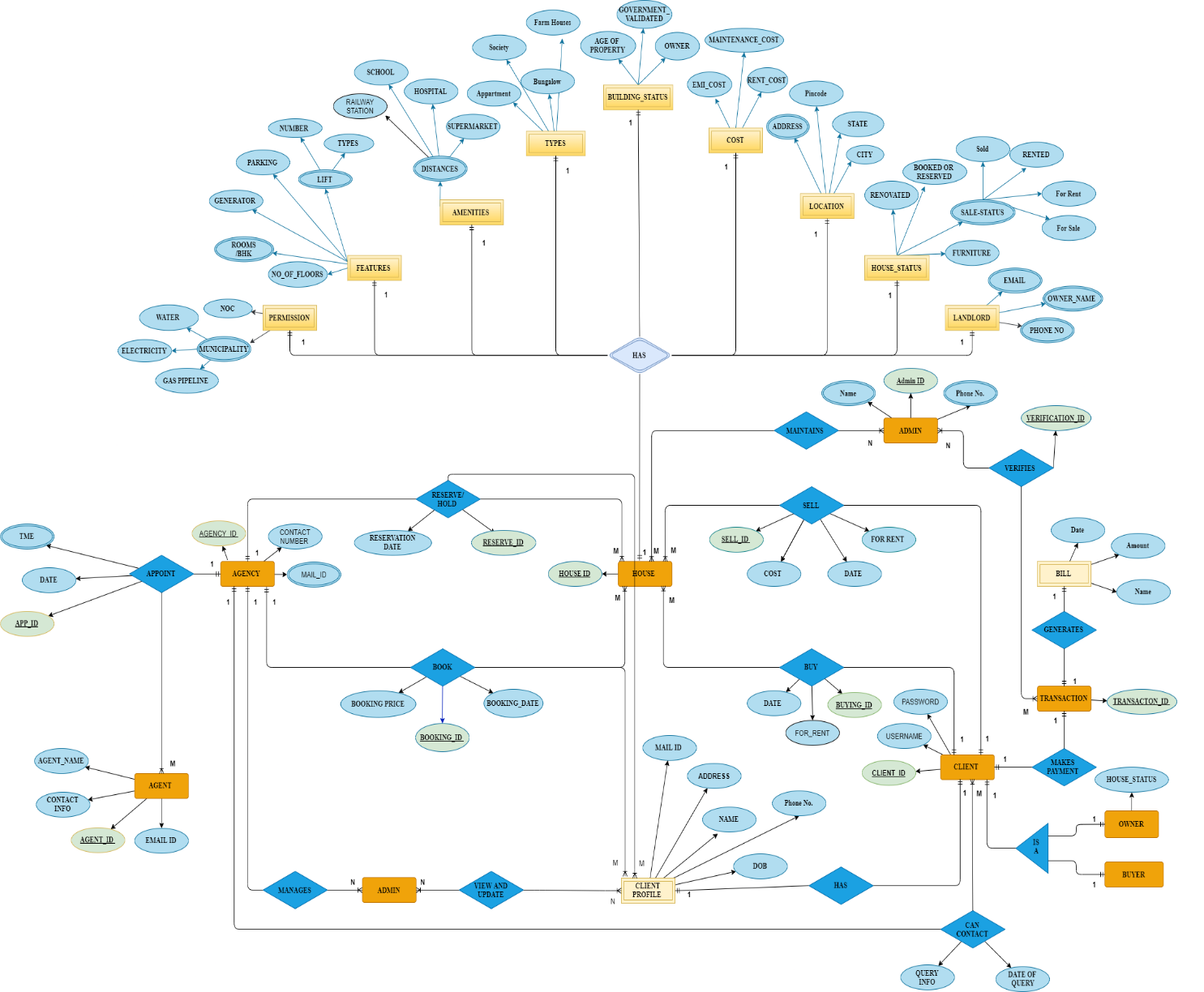
WEAK ENTITIES:

* Bill
* Cost
* Types
* Location
* Features
* Amenities
* Permission
* Client Profile
* House Status
* Building Status
* Buyer
* Owner

# RELATIONSHIPS

|  |  |  |  |
| --- | --- | --- | --- |
| **Entity 1** | **Entity 2** | **Relationship** | **Relationship Type** |
| Client | Client Profile | has (Strong Relationship) | 1: 1 |
| Client | Portal | signs\_up | Many: 1 |
| Client | Agency | can\_contact | Many: 1 |
| Client | Transaction | makes\_payment | 1: 1 |
| Client | House | sells | 1: Many |
| Client | House | buys | 1: Many |
| Client | Owner | is\_a | 1: 1 |
| Client | Buyer | is\_a | 1: 1 |
| Admin | Agency | manages | 1: 1 |
| Admin | Portal | manages | 1: 1 |
| Admin | House | maintains | 1: Many |
| Admin | Client Profile | views\_and\_updates | 1: Many |
| Admin | House | verifies | 1: Many |
| Transaction | Bill | generates | 1: 1 |
| House | Agency | booked\_through | Many: 1 |
| House | Agency | reserved\_through | Many: 1 |
| House | Cost | has (Weak Relationship) | 1: 1 |
| House | Types | has (Weak Relationship) | 1: 1 |
| House | Owner | has (Weak Relationship) | 1: 1 |
| House | Location | has (Weak Relationship) | 1: 1 |
| House | Features | has (Weak Relationship) | 1: 1 |
| House | Amenities | has (Weak Relationship) | 1: 1 |
| House | Permission | has (Weak Relationship) | 1: 1 |
| House | House Status | has (Weak Relationship) | 1: 1 |
| House | Building Status | has (Weak Relationship) | 1: 1 |
| Agency | Agent | appoints | 1: Many |

# ER DIAGRAM



# RELATIONAL SCHEMA

CLIENT

|  |  |  |  |
| --- | --- | --- | --- |
| CLIENT\_ID | ***PORTAL\_ID*** | USERNAME | PASSWORD |

CLIENT PROFILE

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***CLIENT\_ID*** | NAME | ADDRESS | PHONE NO. | EMAIL ID |  |

BUYER

|  |  |
| --- | --- |
| ***BUYING\_ID*** | ***CLIENT\_ID*** |

OWNER

|  |  |  |
| --- | --- | --- |
| ***SELL\_ID*** | ***CLIENT\_ID*** | HOUSE\_STATUS |

ADMIN

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ADMIN\_ID | ***PORTAL\_ID*** | ***CLIENT\_ID*** | ***AGENCY\_ID*** | ***HOUSE\_ID*** | PHONE |

HOUSE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| HOUSE\_ID | ***SELLING\_ID*** | ***BOOKING\_ID*** | ***BUYING\_ID*** | ***RESERVE\_ID*** |

AGENT

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***CLIENT\_ID*** | NAME | ADDRESS | PHONE NO. | EMAIL ID |

AGENCY

|  |  |  |
| --- | --- | --- |
| AGENCY\_ID | CONTACT NO. | EMAIL ID |

APPOINT

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| APP\_ID | Time | Date | ***AGENCY\_ID*** | ***AGENT\_ID*** |

CAN CONTACT

|  |  |  |  |
| --- | --- | --- | --- |
| ***CLIENT\_ID*** | ***AGENCY\_ID*** | QUERY INFO | DATE OF QUERY |

TRANSACTION

|  |  |
| --- | --- |
| TRANSACTION\_ID | ***CLIENT\_ID*** |

BILL

|  |  |  |  |
| --- | --- | --- | --- |
| ***TRANSACTION\_ID*** | NAME | DATE | AMOUNT |

VERIFIES

|  |  |  |
| --- | --- | --- |
| VERIFICATION\_ID | ***ADMIN\_ID*** | ***TRANSACTION\_ID*** |

RESERVE/HOLD

|  |  |  |  |
| --- | --- | --- | --- |
| RESERVE\_ID | RESERVATION\_DATE | ***AGENCY\_ID*** | ***HOUSE\_ID*** |

BOOK

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| BOOKING\_ID | BOOKING\_DATE | BOOKING\_PRICE | ***AGENCY\_ID*** | ***HOUSE\_ID*** |

SELLS

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SELLING\_ID | COST | DATE | FOR\_RENT | ***HOUSE\_ID*** | ***CLIENT\_ID*** |

BUYS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| BUYING\_ID | DATE | FOR RENT | ***HOUSE\_ID*** | ***CLIENT\_ID*** |

LANDLORD (WEAK ENTITY)

|  |  |  |  |
| --- | --- | --- | --- |
| HOUSE\_ID | NAME | EMAIL ID | PHONE NO. |

**PERMISSION (WEAK ENTITY)**

|  |  |  |
| --- | --- | --- |
| HOUSE\_ID | NOC | MUNICIPALITY |

**COST (WEAK ENTITY)**

|  |  |  |  |
| --- | --- | --- | --- |
| HOUSE\_ID | EMI\_COST | RENT\_COST | MAINTENANCE\_COST |

**AMENITIES (WEAK ENTITY)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| HOUSE\_ID | SCHOOL | HOSPITALS | SUPERMARKET | RAILWAY STATION |

**FEATURES (WEAK ENTITY)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| HOUSE\_ID | NO\_OF\_FLOORS | GENERATOR | PARKING | ROOMS /BHK | LIFT |

**TYPES (WEAK ENTITY)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| HOUSE\_ID | FARMHOUSES | APARTMENT | SOCIETY | BUNGALOW |

**BUILDING STATUS (WEAK ENTITY)**

|  |  |  |  |
| --- | --- | --- | --- |
| HOUSE\_ID | OWNER | GOVERNMENT\_VALIDATED | AGE\_OF\_PROPERTY |

**HOUSE STATUS (WEAK ENTITY)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| HOUSE\_ID | RENOVATED | BOOKED/ RESERVED | SALE\_STATUS | FURNITURE |

**LOCATION (WEAK ENTITY)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| HOUSE\_ID | ADDRESS | STATE | CITY | PINCODE |

**OWNER (WEAK ENTITY)**

|  |  |  |  |
| --- | --- | --- | --- |
| HOUSE\_ID | NAME | EMAIL ID | PHONE NO. |

# KEYS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TABLE NAME** | **PRIMARY KEY** | **FOREIGN KEY** | **CANDIDATE KEY** | **ALTERNATE KEY** |
| 1. PORTAL | PORTAL\_ID | NULL | NULL | NULL |
| 1. CLIENT | CLIENT\_ID | PORTAL\_ID | 1) PASSWORD  2) CLIENT\_ID | PASSWORD |
| 1. CLIENT PROFILE | CLIENT\_ID | NULL | 1) CLIENT\_ID  2) PHONE NO.  3) EMAIL ID  4) ADDRESS | 1) PHONE NO.  2) EMAIL ID  3) ADDRESS |
| 1. BUYER | BUYING\_ID | CLIENT\_ID | NULL | NULL |
| 1. OWNER | OWNER\_ID | 1) SELL\_ID  2) CLIENT\_ID | NULL | NULL |
| 1. ADMIN | ADMIN\_ID | 1) PORTAL\_ID  2) CLIENT\_ID  3) AGENCY\_ID  4) HOUSE\_ID | 1) ADMIN\_ID  2) PHONE NO. | PHONE NO. |
| 1. HOUSE | HOUSE\_ID | 1) SELLING\_ID  2) BOOKING\_ID  3) BUYING\_ID  4) RESERVE\_ID | NULL | NULL |
| 1. AGENT | AGENT\_ID | 1) AGENCY\_ID | 1) AGENT\_ID  2) PHONE NO.  3) EMAIL ID | 1) PHONE NO.  2) EMAIL ID |
| 1. AGENCY | AGENCY\_ID | NULL | 1) AGENCY\_ID  2) CONTACT NO.  3) EMAIL ID | 1) CONTACT NO.  2) EMAIL ID |
| 1. APPOINT | APP\_ID | 1) AGENCY\_ID  2) AGENT\_ID | APP\_ID | NULL |
| 1. CAN CONTACT | NULL | 1) CLIENT\_ID  2) AGENCY\_ID | NULL | NULL |
| 1. TRANSACTION | TRANSACTION\_ID | CLIENT\_ID | TRANSACTION\_ID | NULL |
| 1. BILL | NULL | TRANSACTION\_ID | NULL | NULL |
| 1. VERIFIES | VERIFICATION\_ID | 1) ADMIN\_ID  2) TRANSACTION\_ID | VERIFICATION\_ID | NULL |
| 1. RESERVE/HOLD | RESERVE\_ID | 1) AGENCY\_ID  2) HOUSE\_ID | RESERVE\_ID | NULL |
| 1. BOOK | BOOKING\_ID | 1) AGENCY\_ID  2) HOUSE\_ID | BOOKING\_ID | NULL |
| 1. SELLS | SELLING\_ID | 1) HOUSE\_ID  2) CLIENT\_ID | SELLING\_ID | NULL |
| 1. BUYS | BUYING\_ID | 1) HOUSE\_ID  2) CLIENT\_ID | BUYING\_ID | NULL |
| 1. LANDLORD   (WEAK ENTITY) | HOUSE\_ID | NULL | 1) BILL\_ID  2) EMAIL ID  3) PHONE NO. | 1) EMAIL ID  2) PHONE NO. |
| 1. PERMISSION   (WEAK ENTITY) | HOUSE\_ID | NULL | HOUSE\_ID | NULL |
| 1. COST   (WEAK ENTITY) | HOUSE\_ID | NULL | HOUSE\_ID | NULL |
| 1. AMENITIES   (WEAK ENTITY) | HOUSE\_ID | NULL | HOUSE\_ID | NULL |
| 1. FEATURES   (WEAK ENTITY) | HOUSE\_ID | NULL | HOUSE\_ID | NULL |
| 1. TYPES   (WEAK ENTITY) | HOUSE\_ID | NULL | HOUSE\_ID | NULL |
| 1. BUILDING STATUS (WEAK ENTITY) | HOUSE\_ID | NULL | HOUSE\_ID | NULL |
| 1. HOUSE STATUS   (WEAK ENTITTY) | HOUSE\_ID | NULL | 1) HOUSE\_ID  2) ADDRESS | NULL |
| 1. LOCATION   (WEAK ENTITY) | HOUSE\_ID | NULL | 1) HOUSE\_ID  2) ADDRESS | ADDRESS |
| 1. OWNER   (WEAK ENTITY) | HOUSE\_ID | NULL | 1) HOUSE\_ID  2) EMAIL ID  3) PHONE NO. | 1) EMAIL ID  2) PHONE NO. |

# CODD’S RULE

The detailed implementation of Codd’s Rule as implemented in our project has been discussed below:

## **Rule 1: Information Rule**

All the data in our database is stored as a table distinctly. No data is left unaccounted for. Each data value belongs to a particular cell of a table. Hence, the real estate residential property management system satisfies the information rule. The data stored in a database, may it be user data or metadata, is a value of some table cell. Everything in a database is stored in a table format.

## **Rule 2: Guaranteed Access Rule**

## All the data can be accessed uniquely from the table name, primary key and a given attribute. The relational representation clearly shows the presence of primary key in every table. Hence, all the data is accessible and thus it satisfies the guarantied access rule.

**Rule 3: Systematic treatment of NULL values**

## The real estate residential property management system ensures to assign value Null to all the fields except the fields storing primary key. Hence it obeys the rule of Systematic Treatment of NULL Values.

**Rule 5: Comprehensive data Sub-Language rule**

A database can only be accessed using a language having linear syntax that supports data definition, data manipulation, and transaction management operations. This language can be used directly or by means of some application. If the database allows access to data without any help of this language, then it is considered as a violation. The database has been designed using MySQL a powerful well-defined language which can carry out all sorts of queries enlisted above.

**Rule 6: View updating Rule**

All the views of a database, which can theoretically be updated, must also be updatable by the system. The real estate residential property management system designed to update values at any point of time by access permissions.

**Rule 7: High Level Insert, Update and Delete Rule**

The real estate residential property management system is designed to support high-level insertion, updating, and deletion. It also supports union, intersection, and minus operations to yield sets of data records.

ANOMALIES

By carefully following the rules of designing a database. We made sure to avoid any anomaly that could occur during the development and execution of the database. Some points we made sure to abide by to prevent anomalies in our database are:

* Assigning meaningful and self-explanatory names to the relational schema.
* Reducing repetitive values and use in tuples.
* Making very less use of null values unless necessary.
* Not allowing possibility of surplus tuples.

NORMALISATION

Normalisation focuses on refining the database by removing ambiguity and redundancy. We carried out the process of normalisation in a series of steps.

Our goal was to achieve

* A data redundancy free database.
* To store only related data in the same table.
* Organise data with optimal efficiency.

Once the ER diagram was designed and the schematic diagram was ready the next step was to normalise the dataset.

*1st normalised form: Remove redundant data*

We made sure no similar data was placed in multiple groups. After careful evaluation, no data was found to be repeated. Each data entry has been placed in unique rows and only similar group of data is placed together avoiding conflict in data.

*2nd normalised form: Remove partial dependencies*

We checked for any partial dependencies in our data. Since every table was assigned a unique primary key and we made use of no composite keys. The data was cleared out of the partial dependencies.

*3rd normalised form: Remove transitive dependencies*

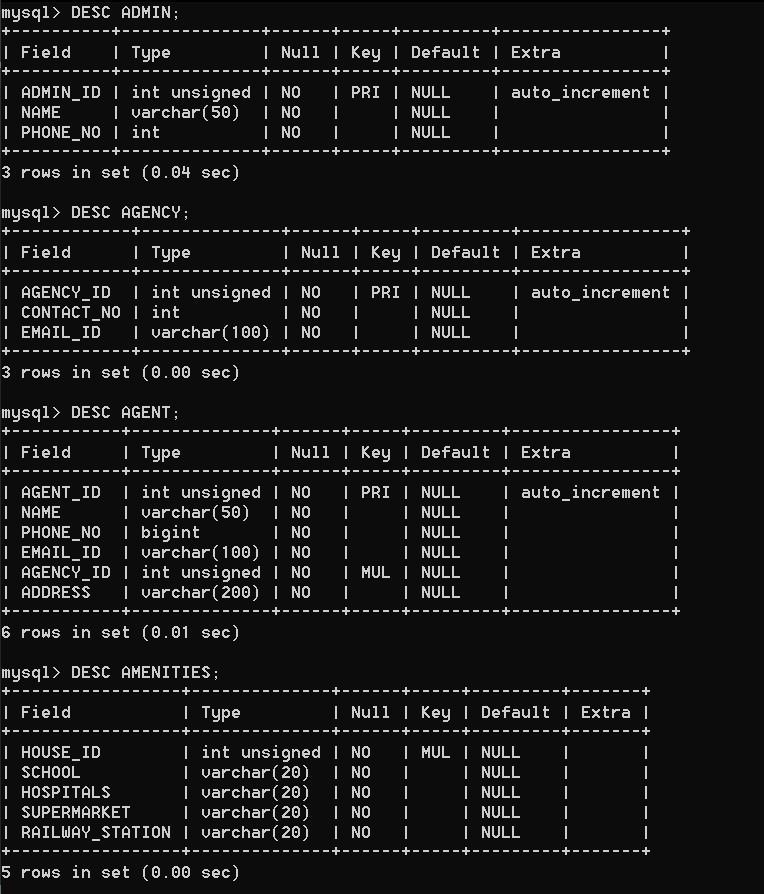
To make sure that the admin could easily access the reserve and booking details for the client and

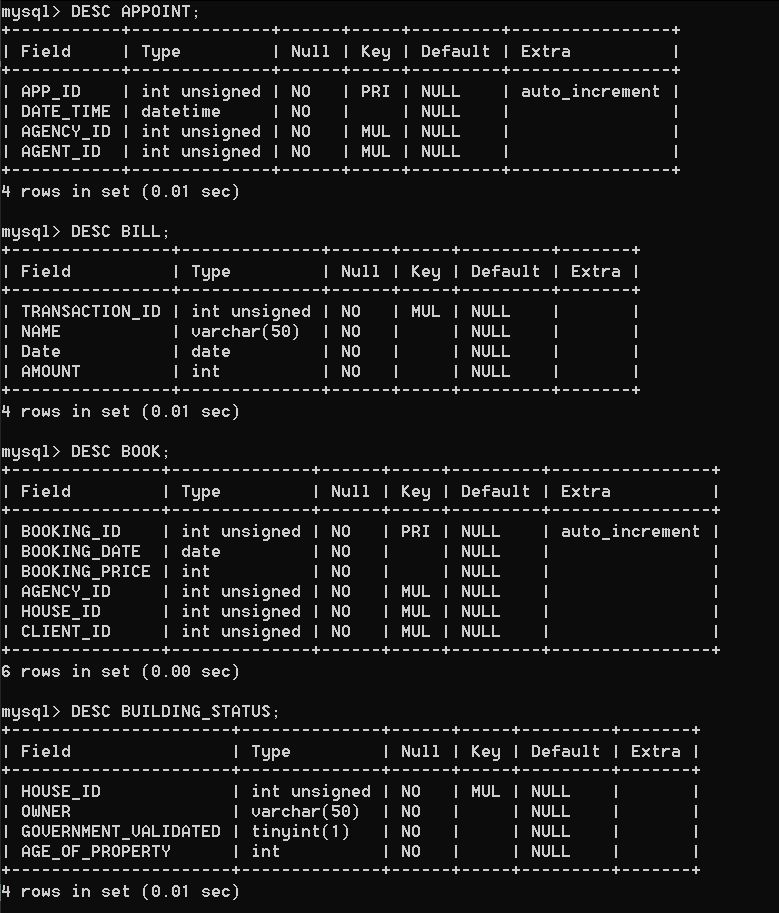
Housing point of view, we made sure to eliminate the transitive dependency by creating a new column as client and making the required changes in our ER diagram. Further a careful analysis revealed that all the transitive data had been eliminated. Client and client profile were separated, so was agent and agency this prevented any sort of ambiguity in the database.

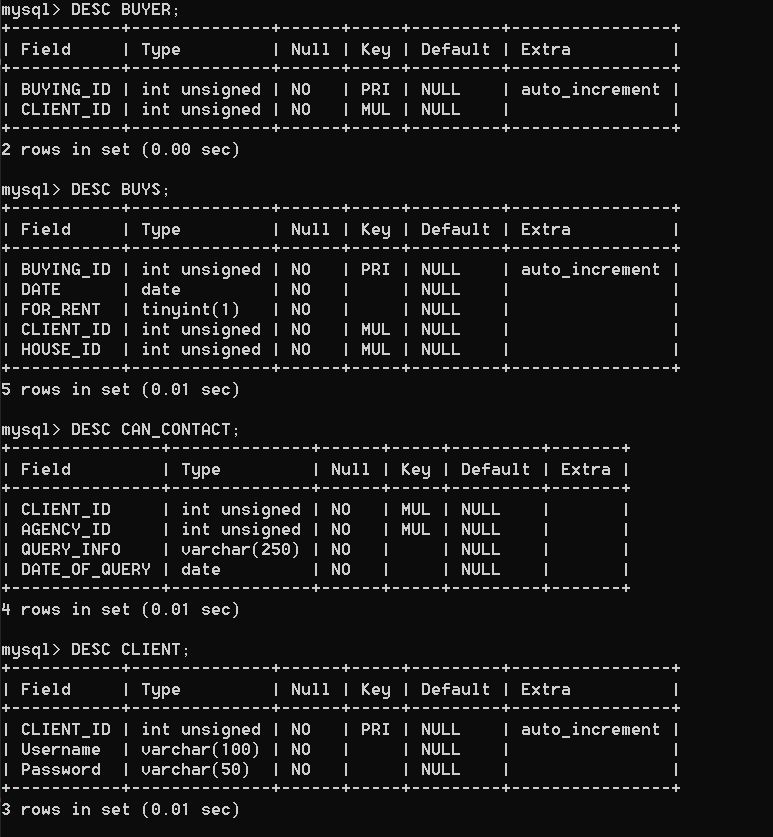
DATABASE IMPLEMENTATION

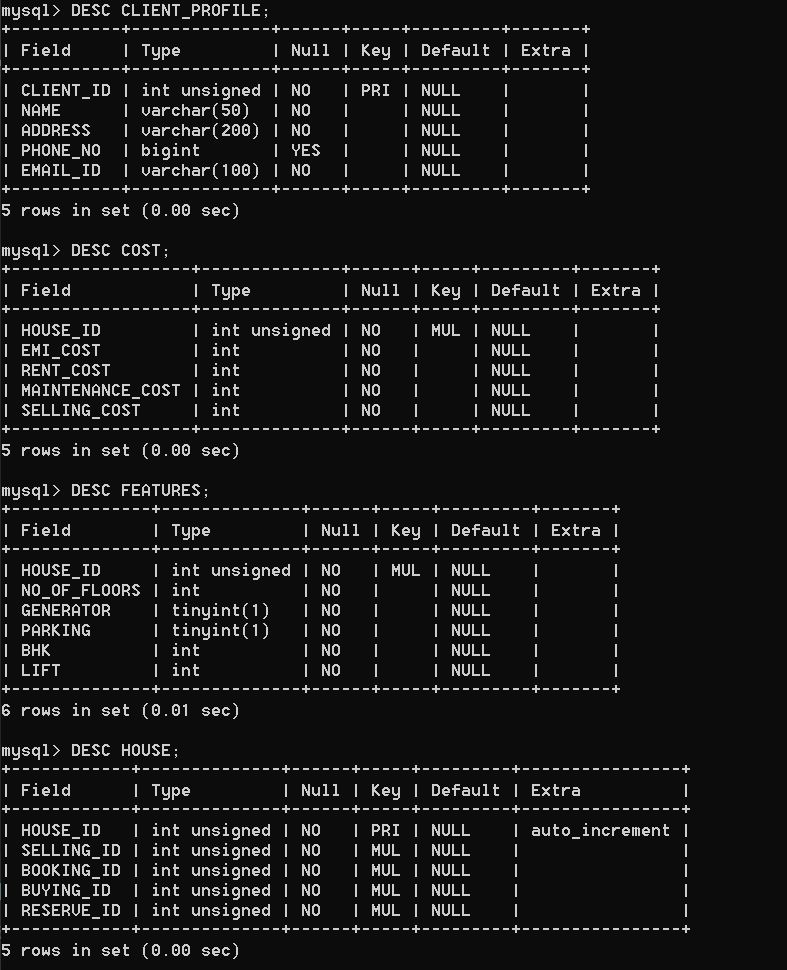
1. Table Descriptions**:**

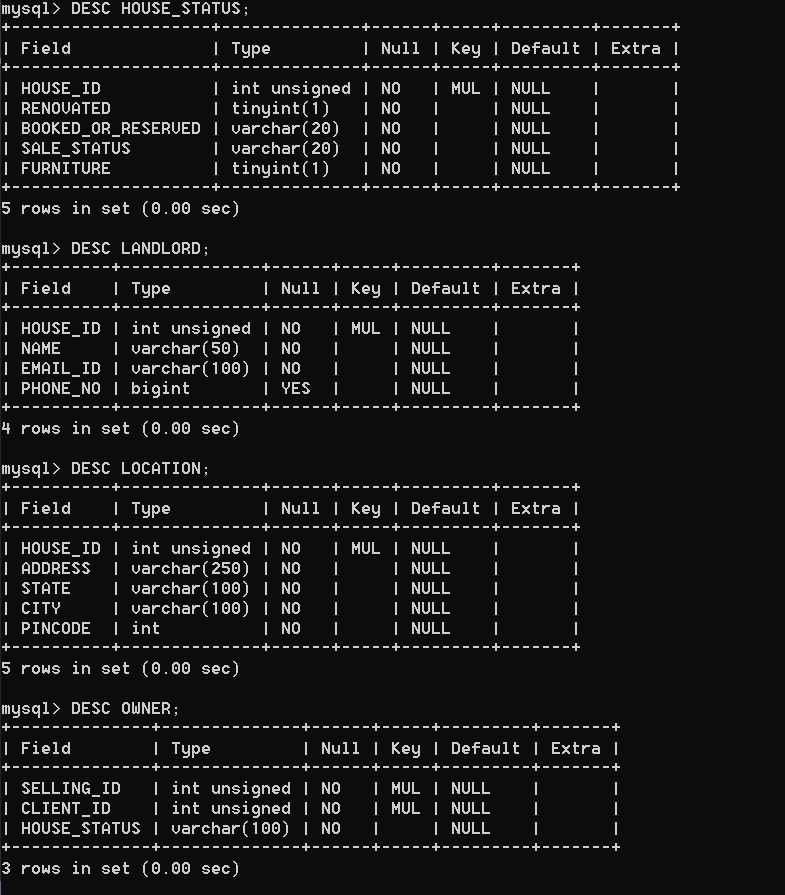
****

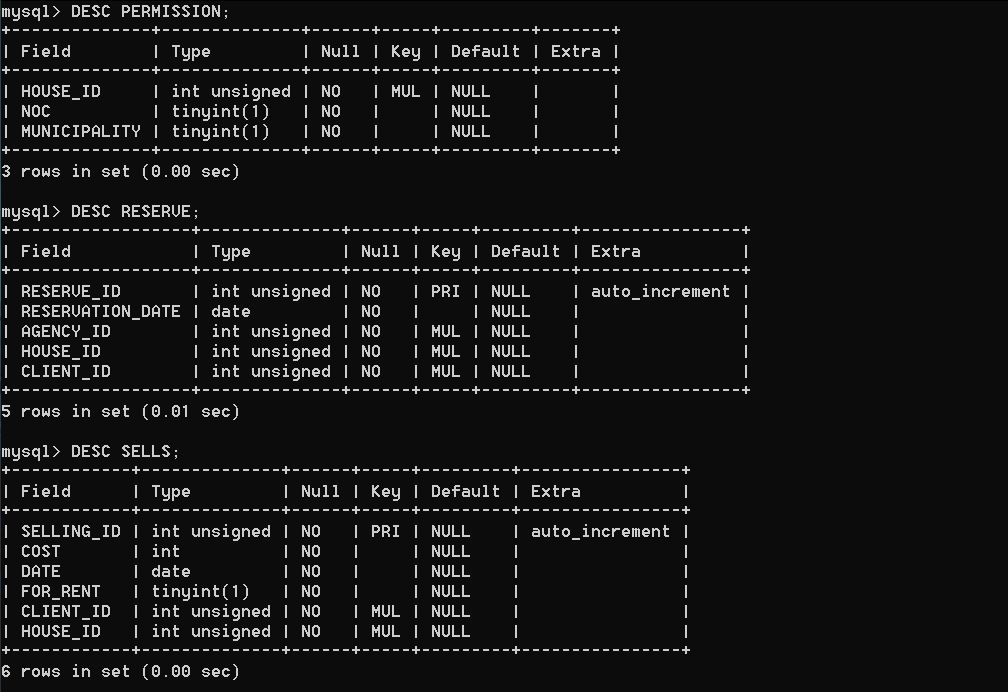
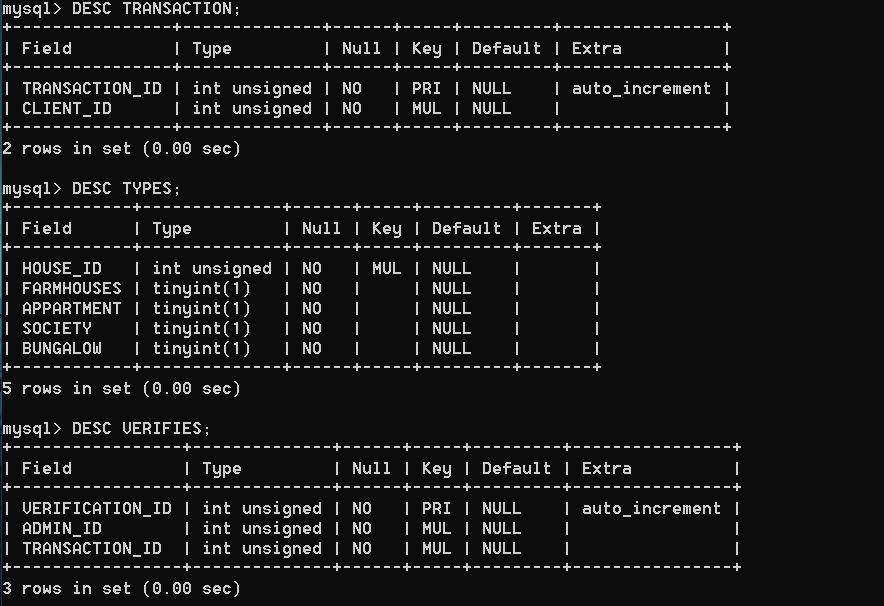
****

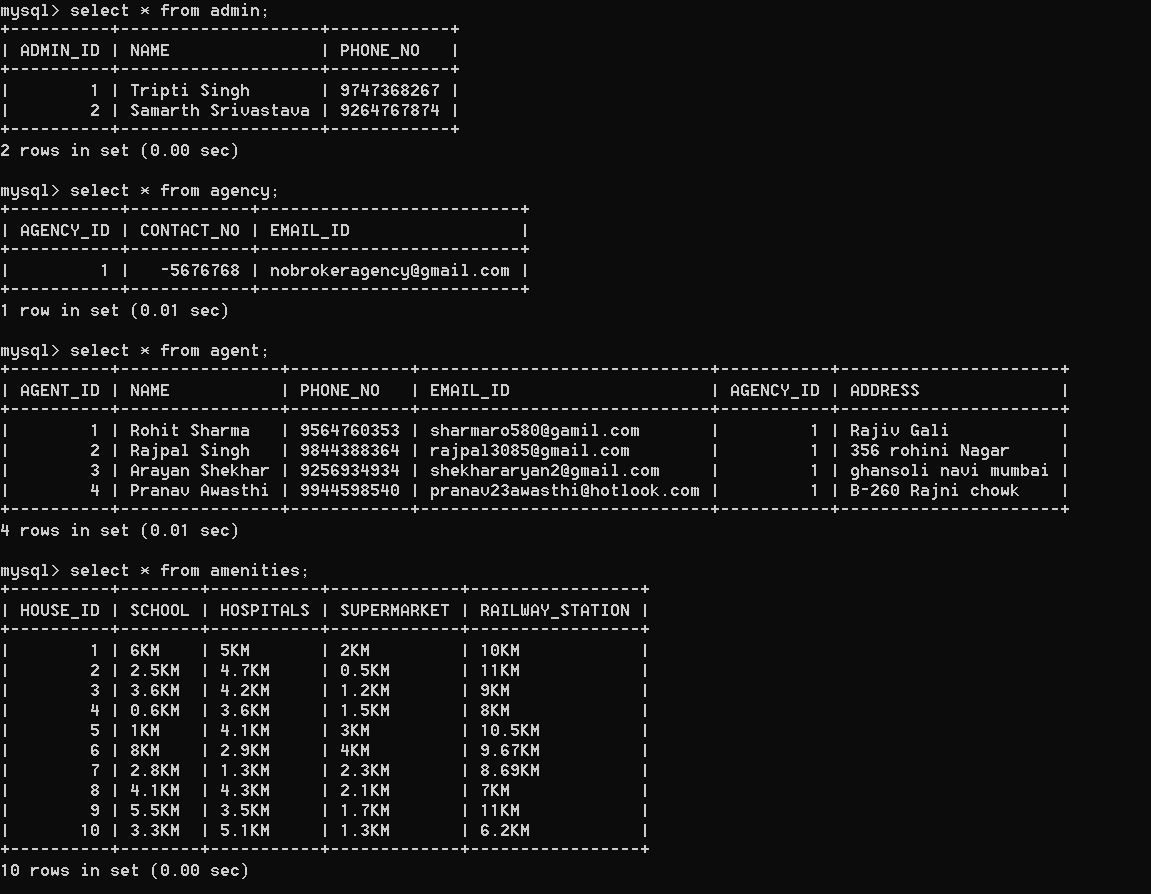
****

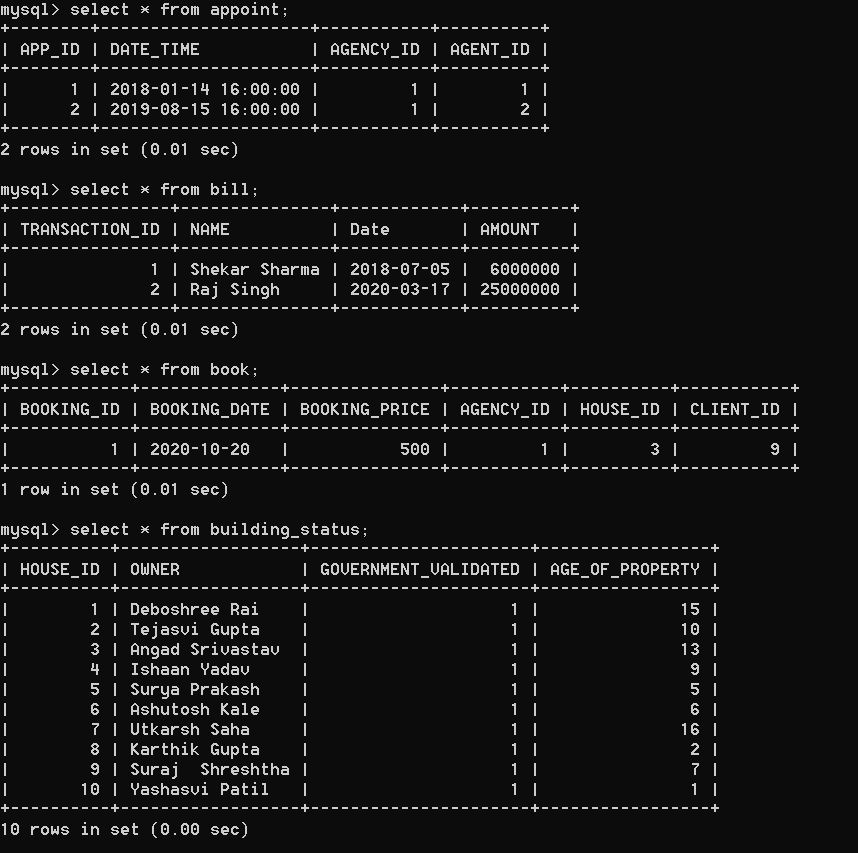
****

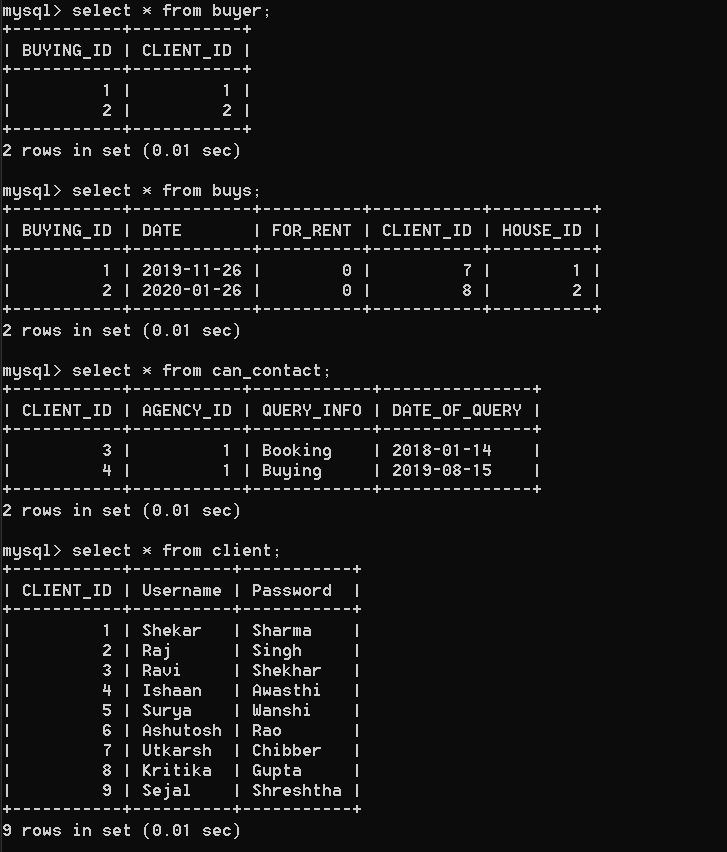
****

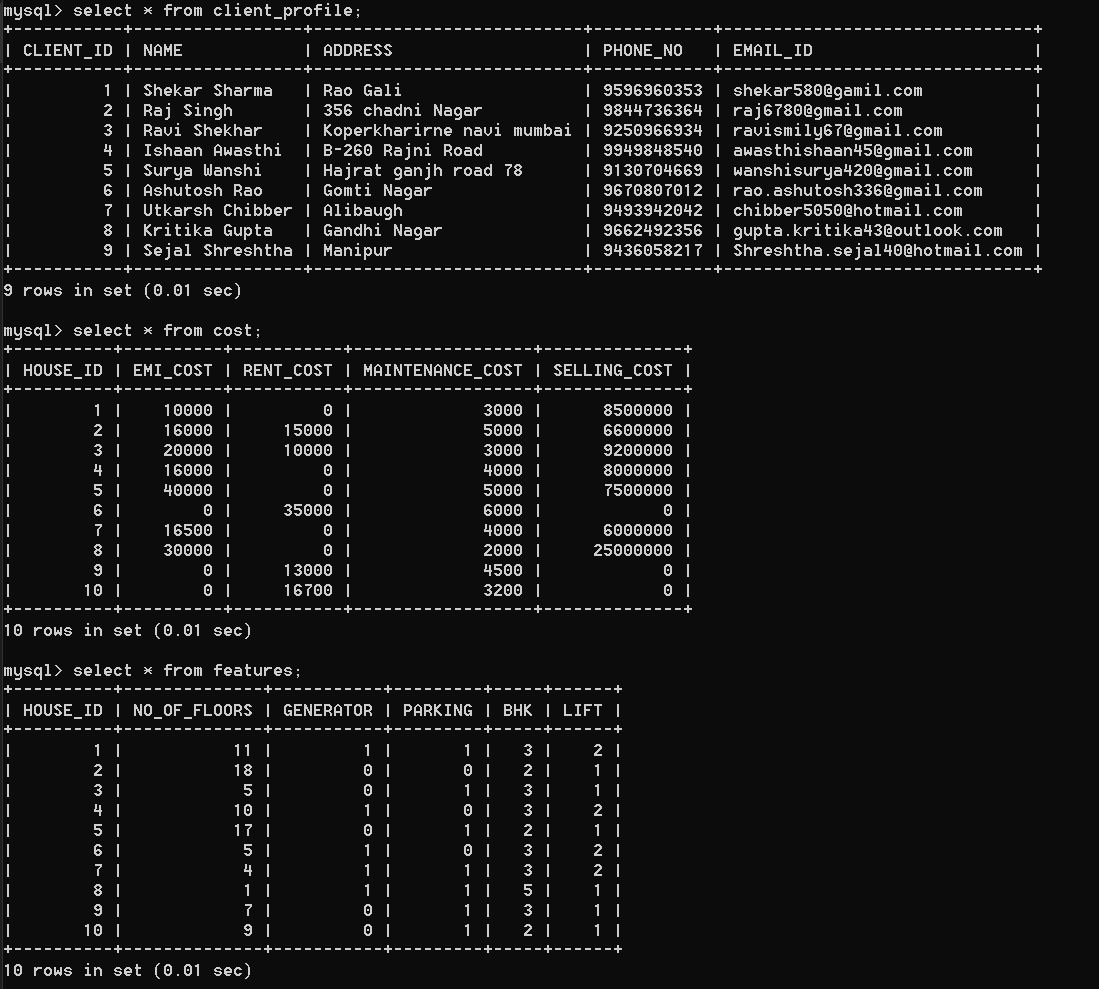
****

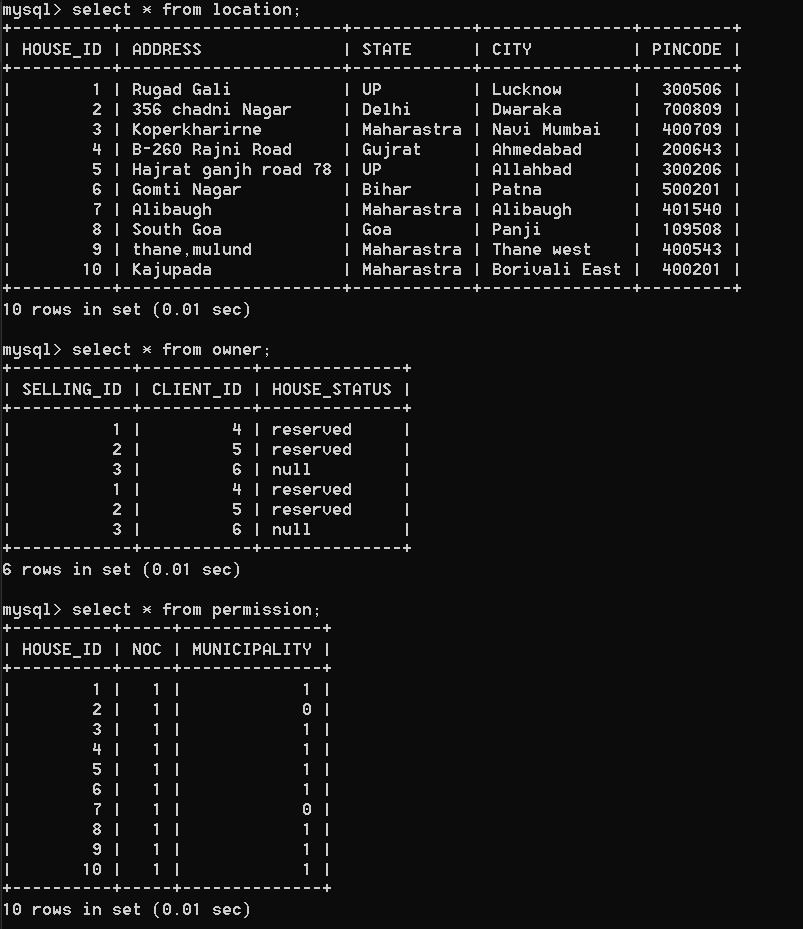
****

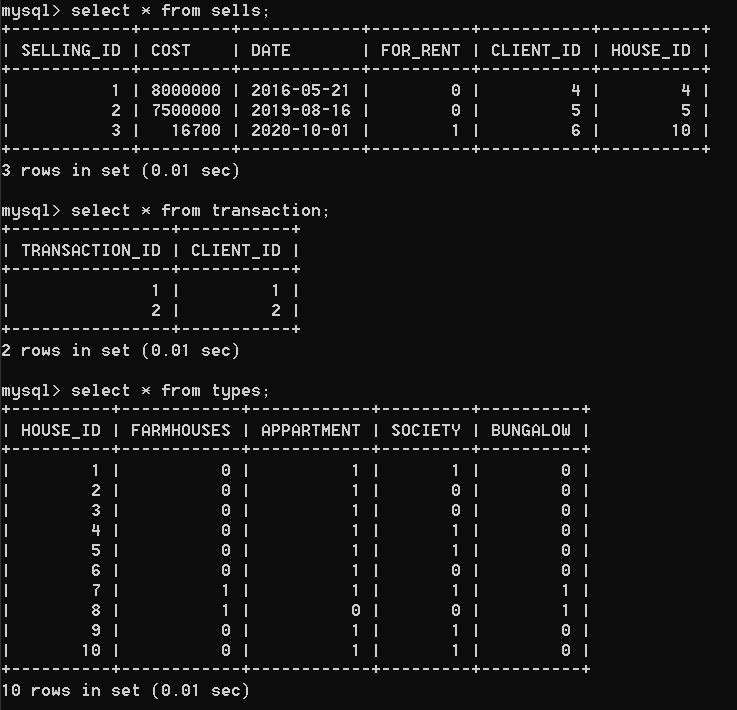
1. **** Table Data:

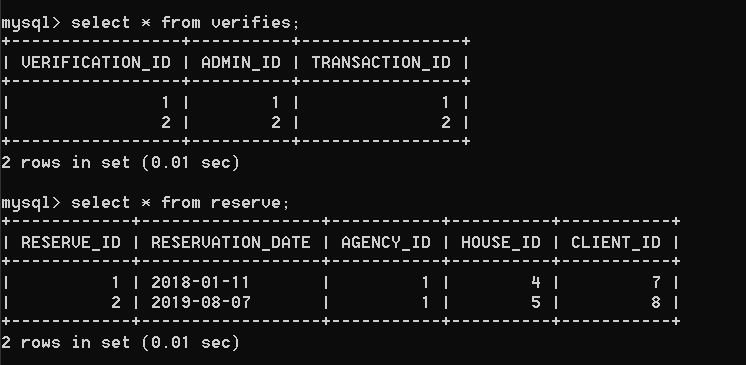
****

****

****

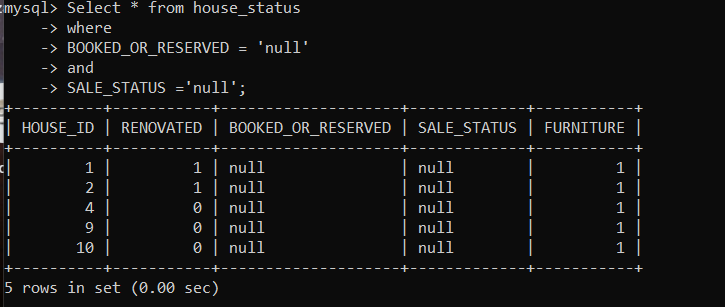
****

****

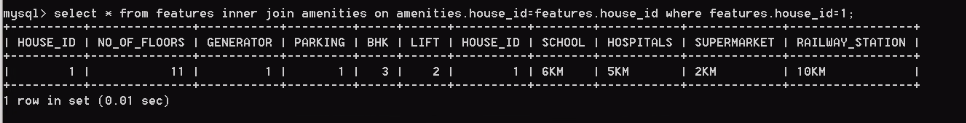
****

QUERY EXECUTION

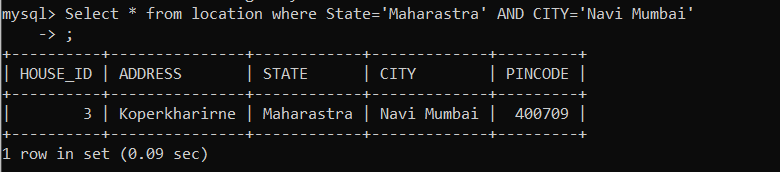
1. Print all the houses that are up for sale.



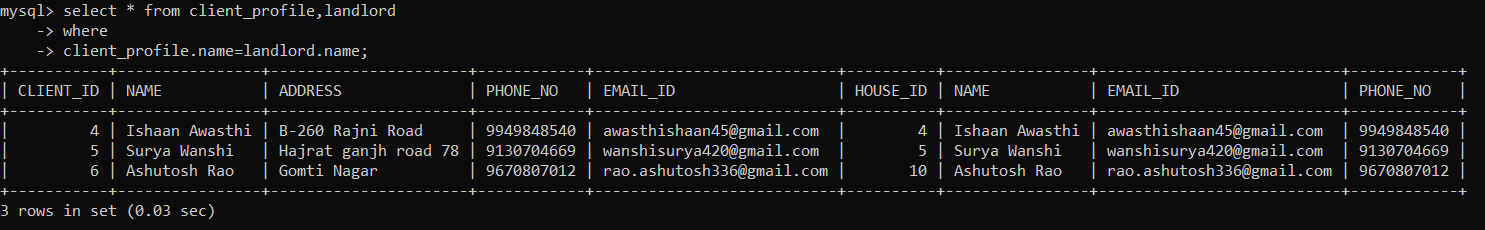
1. Display all the details of a house for given house id =1



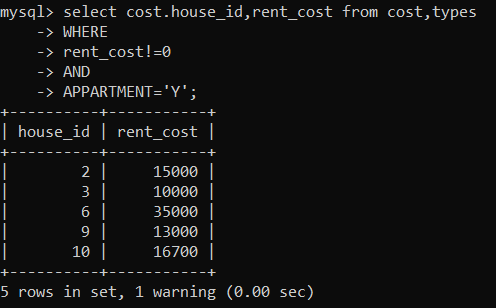
1. Show available houses in Maharashtra and Navi Mumbai city.



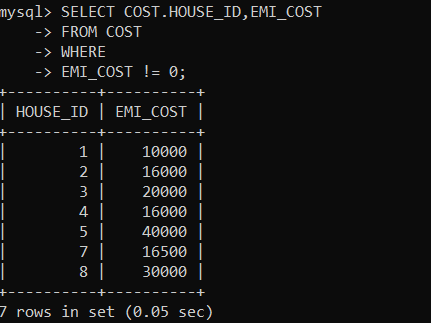
1. Show all the clients that the owners of houses.



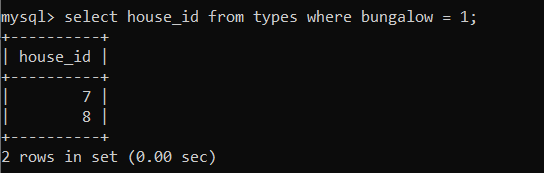
1. Show all the houses available on rent, apartment type and their rental cost.



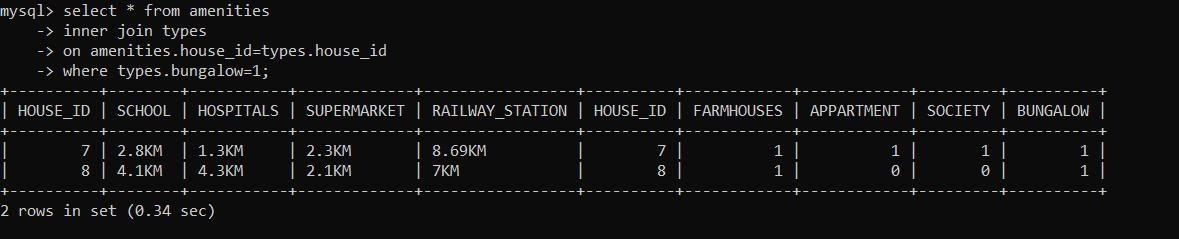
1. Show the EMI cost of houses



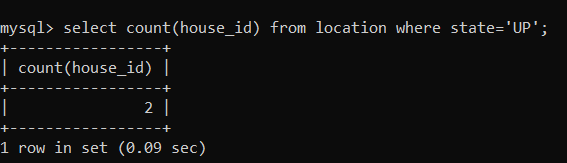
1. Show all the bungalow type houses available.



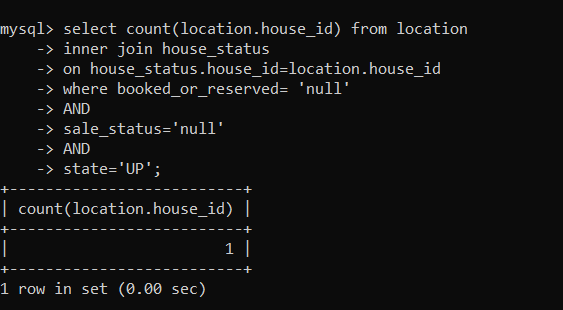
1. Show bungalow type houses with their amenities.



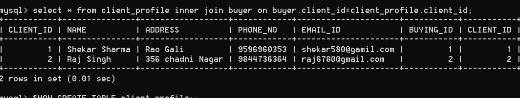
1. Count total no of houses available in UP.



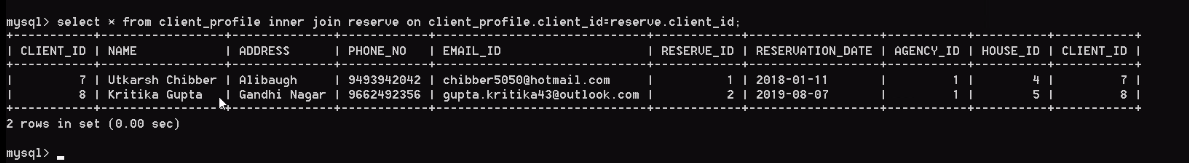
1. Count the total number of houses available in UP for buy.



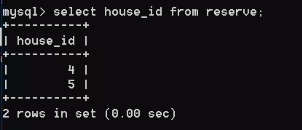
1. Client details who bought a house.



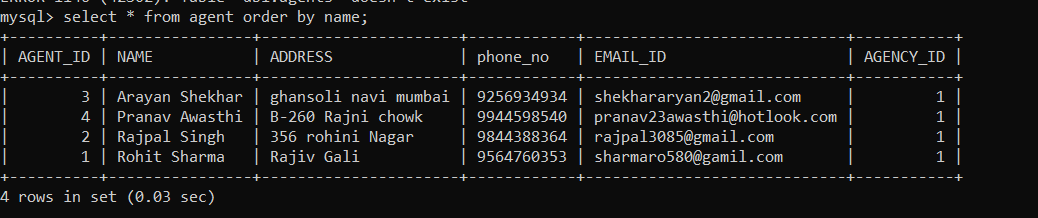
1. Client details who reserved house



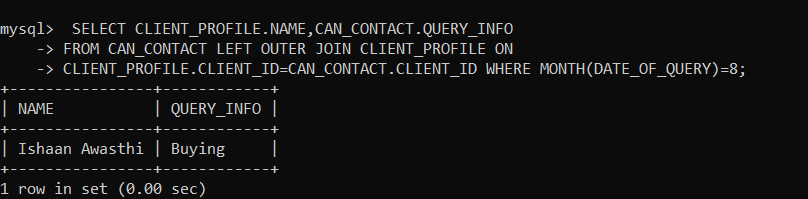
13.List the houses that have been reserved



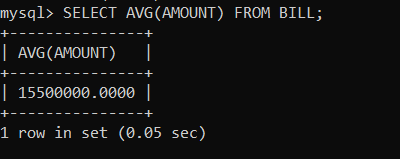
14. Show details of agents in the order of their names.



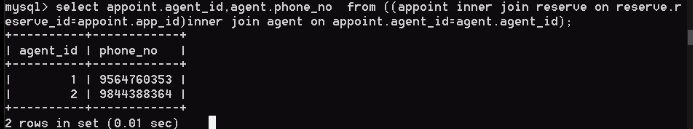
15.Show queries in month of August and Client name associated with it



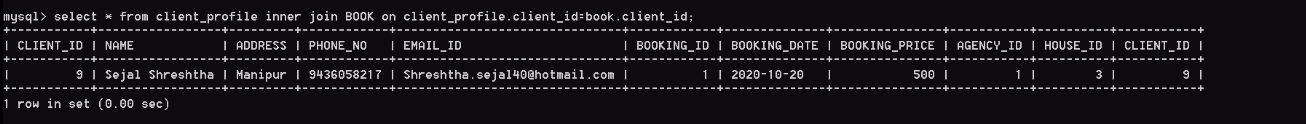
16. Get the average billing price of all the houses that have been sold.

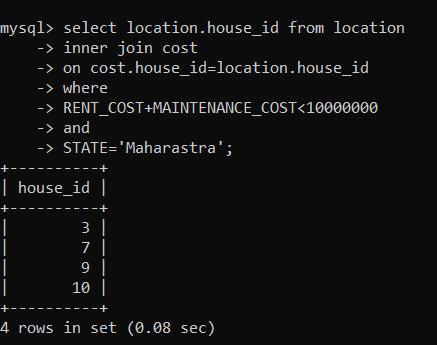


17. Show the phone number of the agent who has been assigned to the client for an appointment.



18. Show the client booking details.

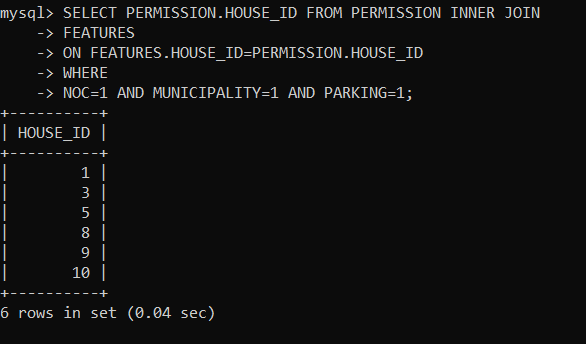


19. Show house in Maharashtra whose selling and maintenance cost is less than 1 crore.

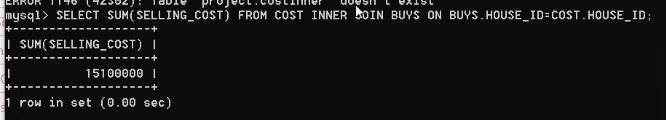
20. Get emails of landlords whose houses have been sold.

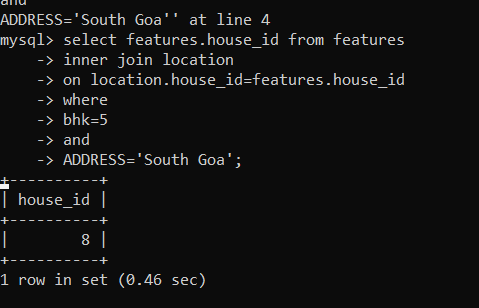


20.Check if the houses have been registered by NOC ,Muncipality and has parking.

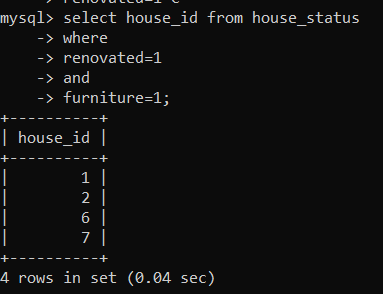


21. Find the total cost of the houses to be sold.

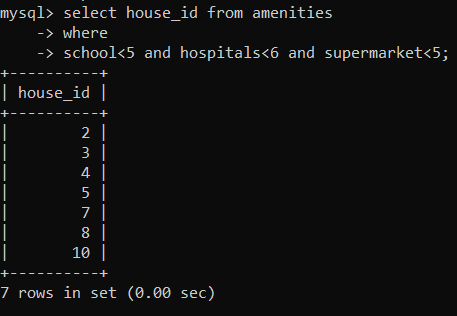


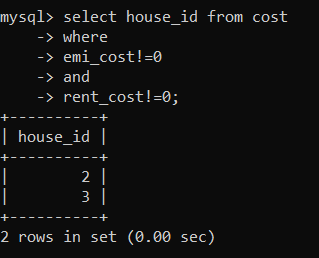
22. Show house details of a house with 5BHK in Goa. 

23.Select houses which have been renovated and furnished.



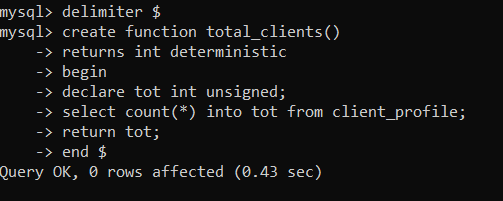
24. Getting houses based on their distances to school, hospitals and supermarket.

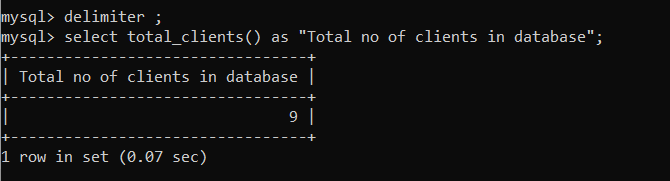


25. Get houses which are on rent and sale. 

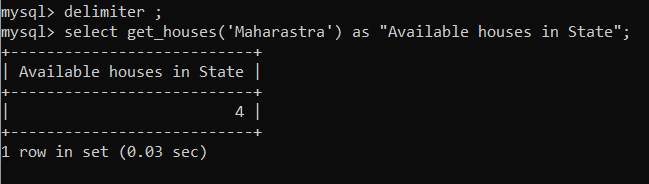
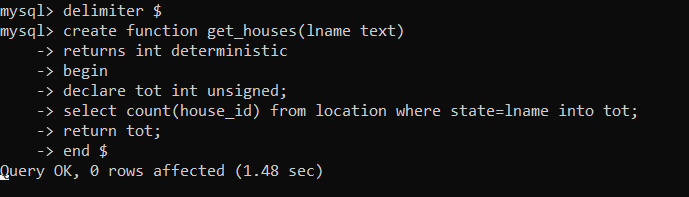
FUNCTIONS

1. Count the number of Clients we have in our System

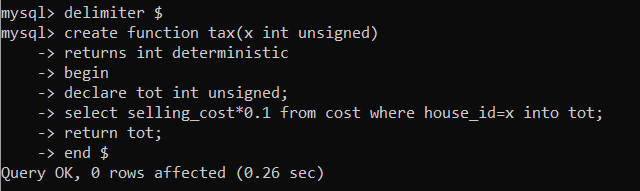


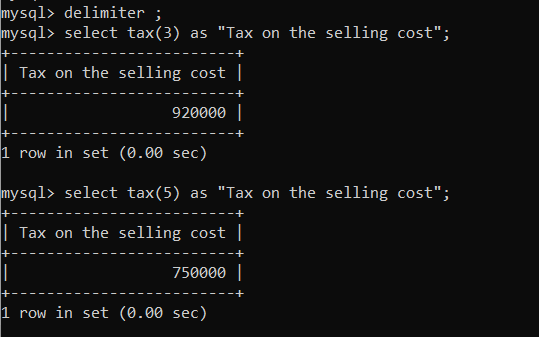


1. Get the available houses for buy in a state

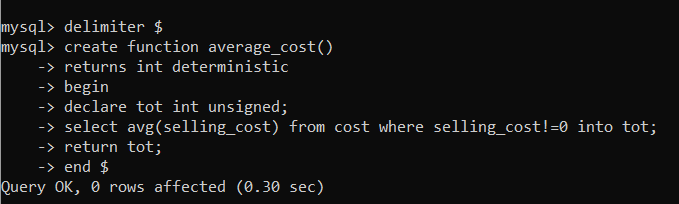


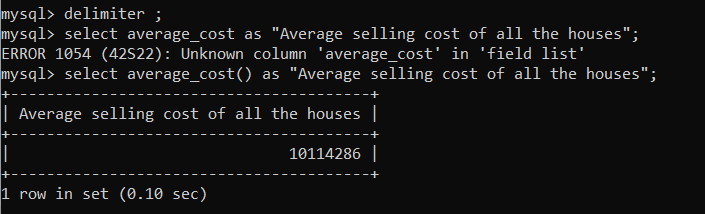
1. Apply 10% tax on the given house id as input and output the price.



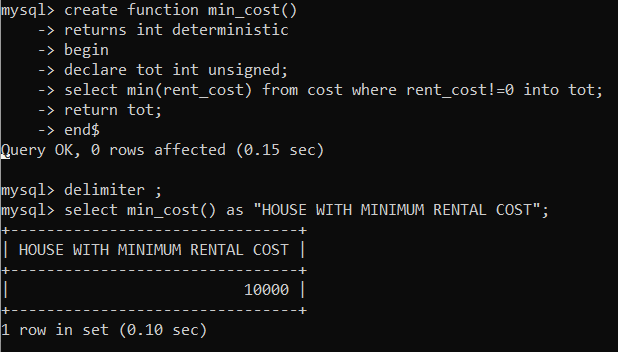


1. Get the average cost of the all the houses that are up for sale.



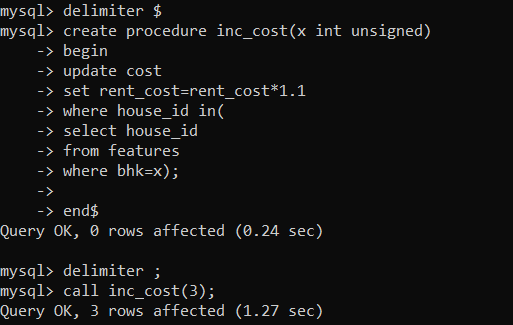


1. Get the house which has the minimum rental cost.

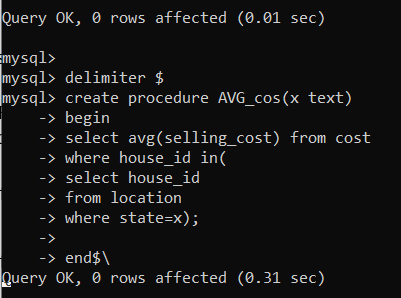


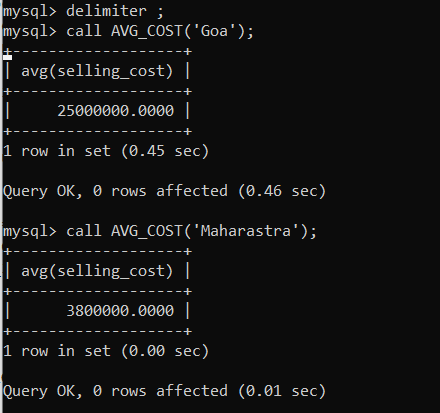
PROCEDURES

1. Update the rental cost to Increase by 10% for Houses with a given BHK number.

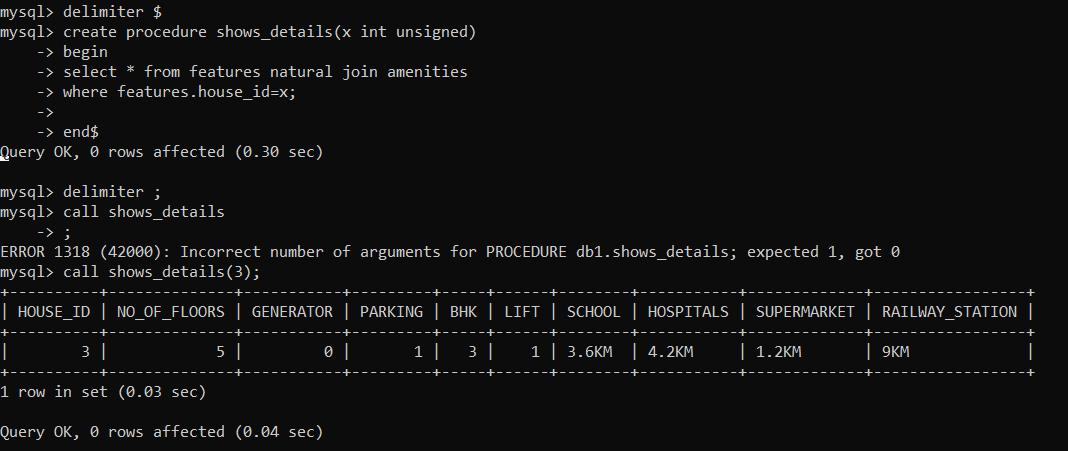


1. Find the Average age of the Properties available in a particular state.

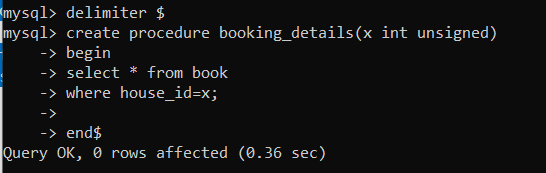


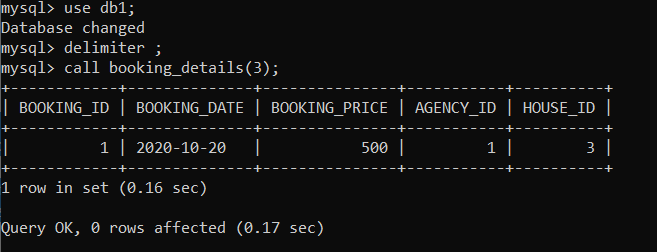


1. Display all the details of a particular house given the house id.

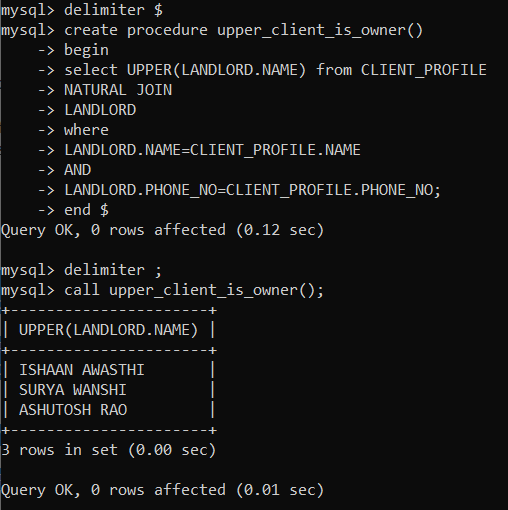


1. Show booking details of a particular house



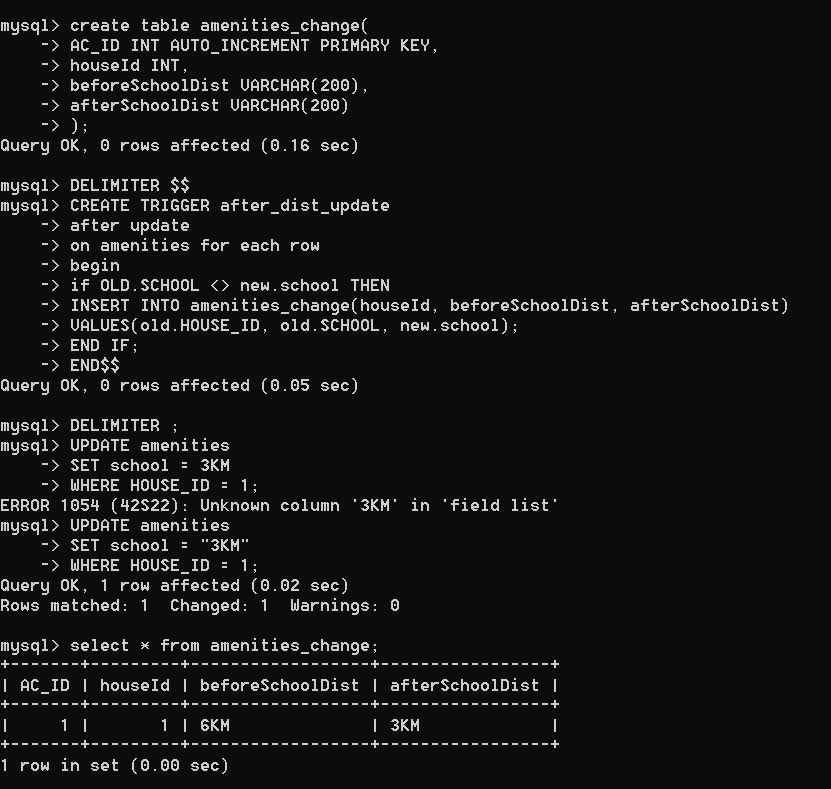


1. Procedure to print client who is an owner in uppercase.

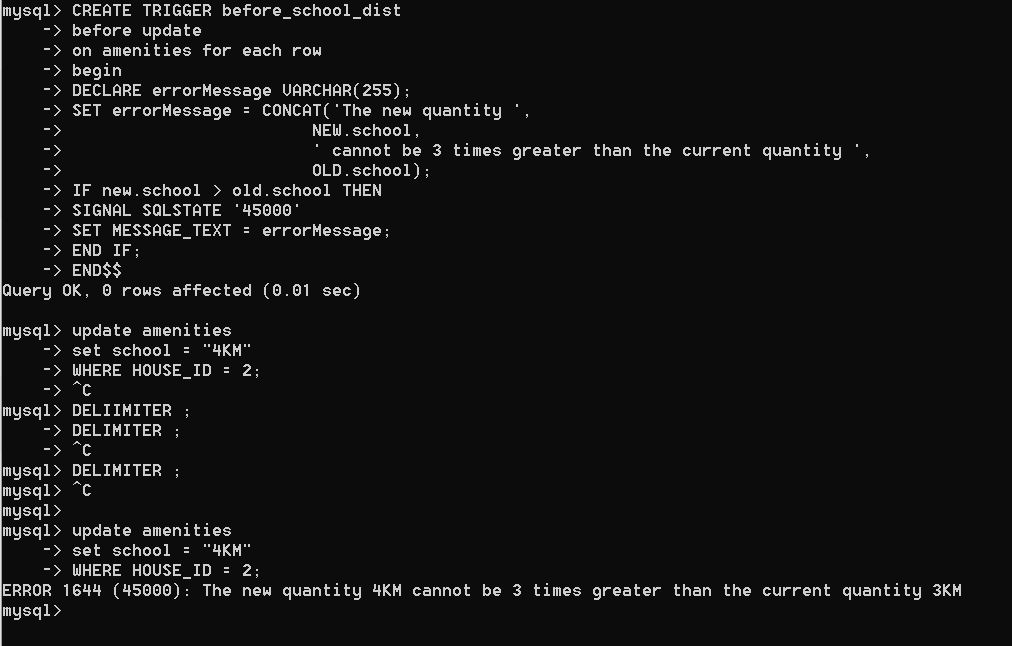


TRIGGERS

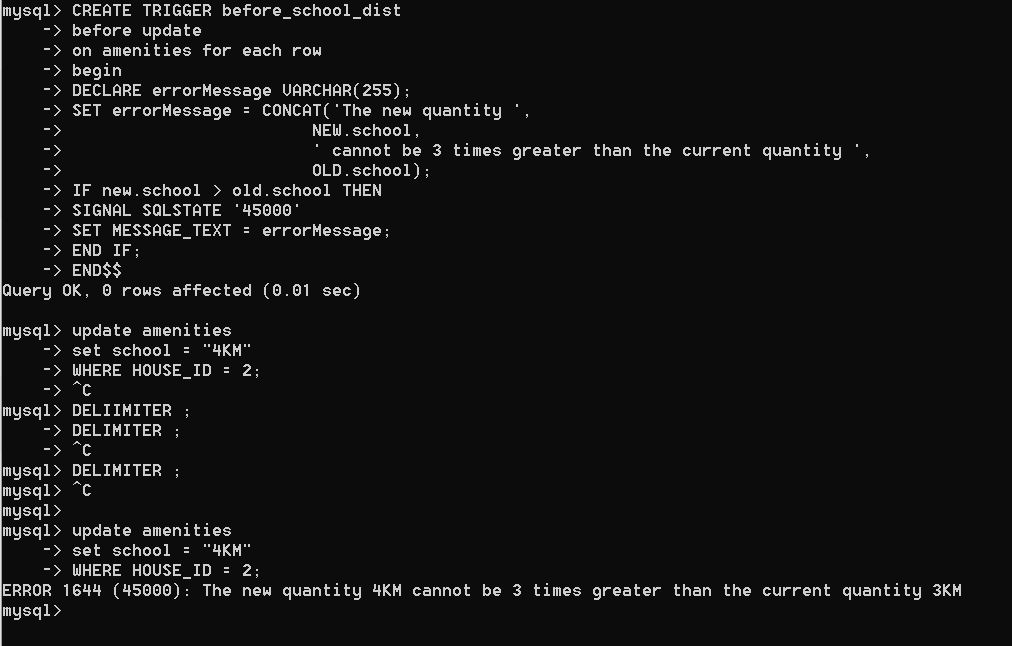
1. Implement an After Update Trigger on Amenities table to show the change in distance of the Schools



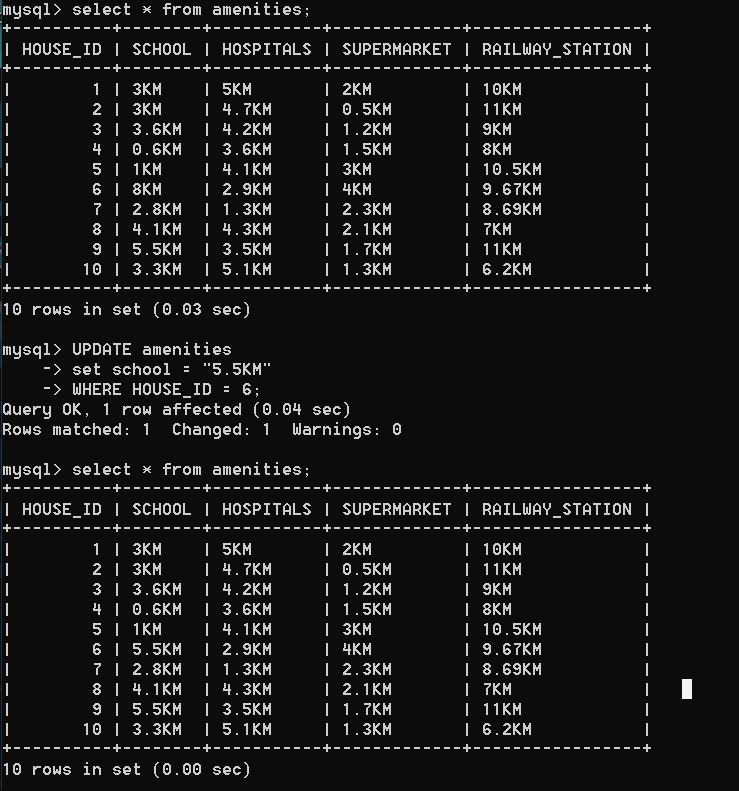
1. Implement a Before Update Trigger on Amenities table to check and update if the distance of the school



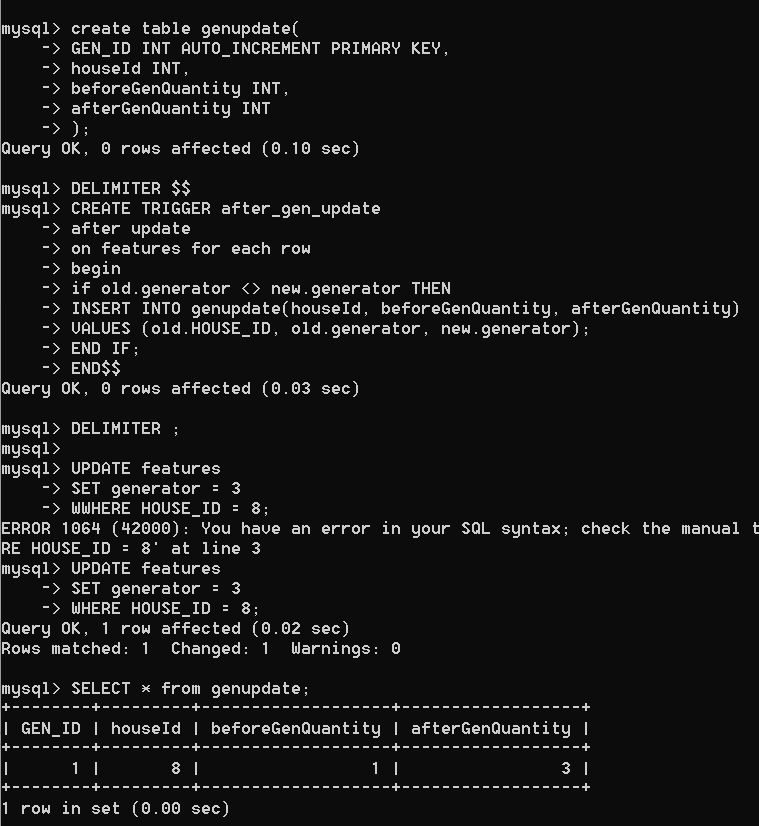
1. If the distance of the new school is greater than the current school

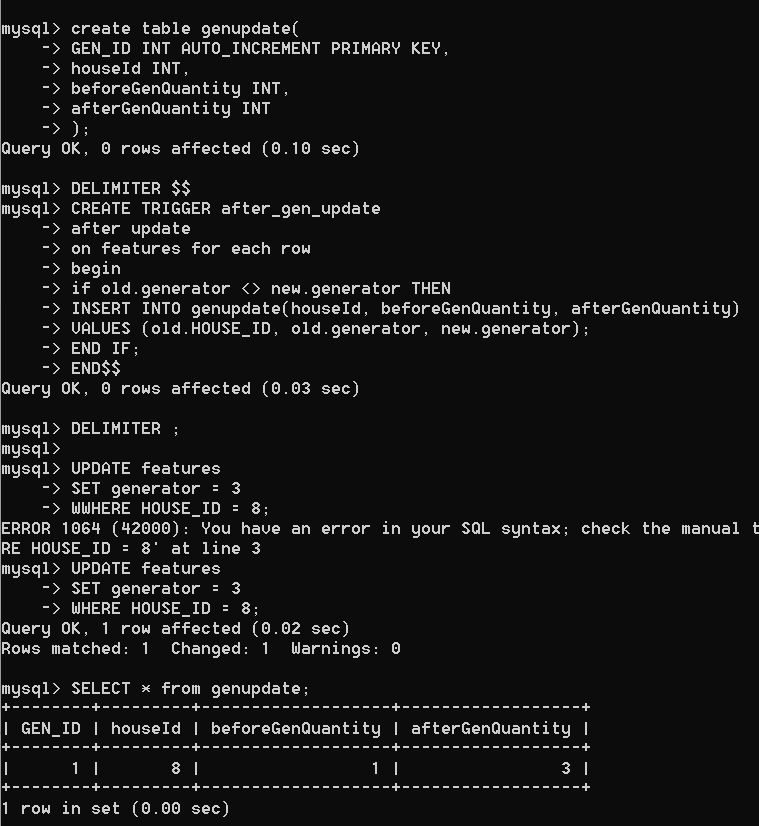


1. If the distance of new school is less than the current school, update in the database takes place

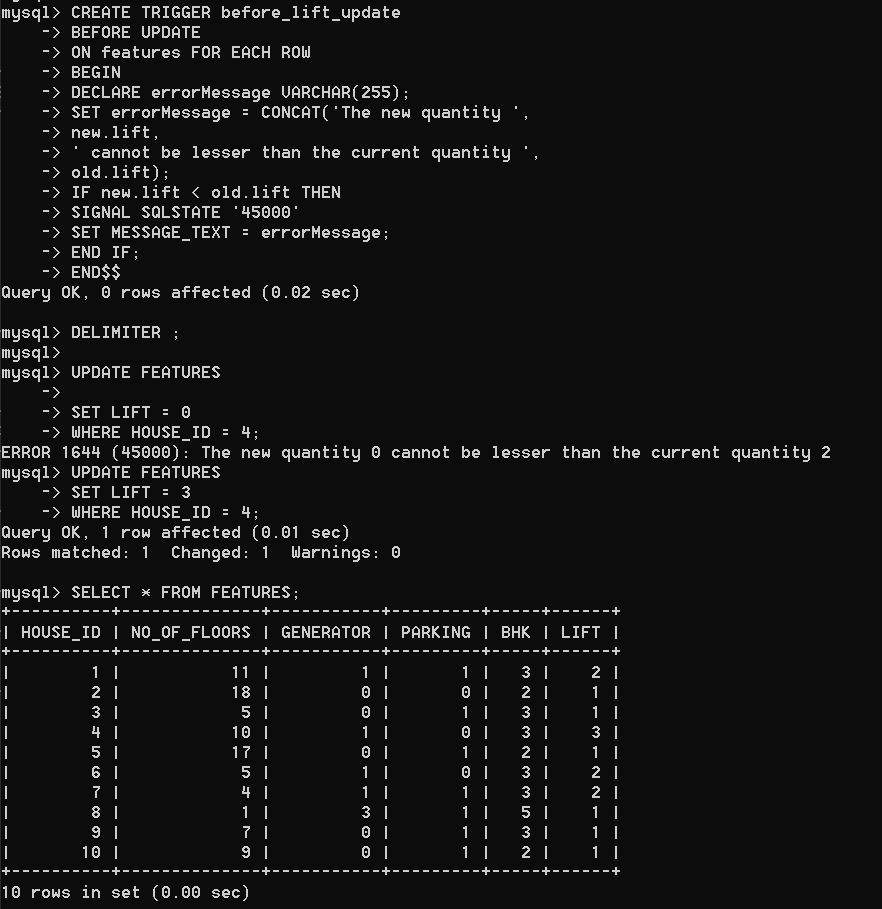


1. Implement an After Update Trigger on Features table to show the change in number of Generators.

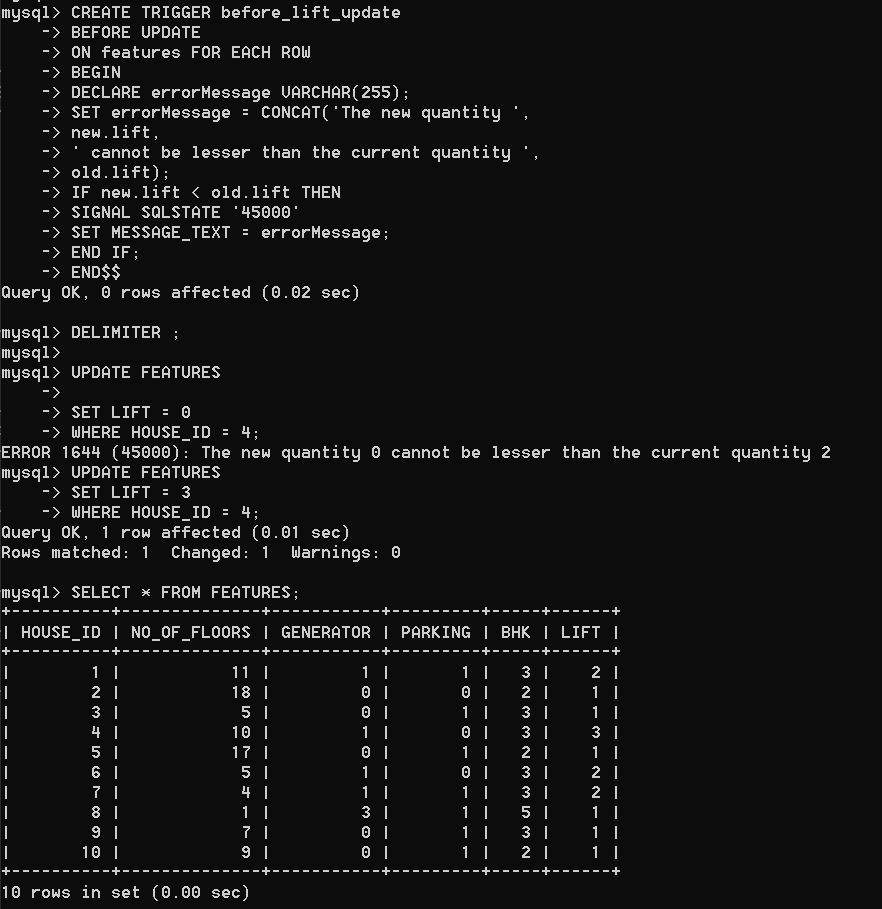




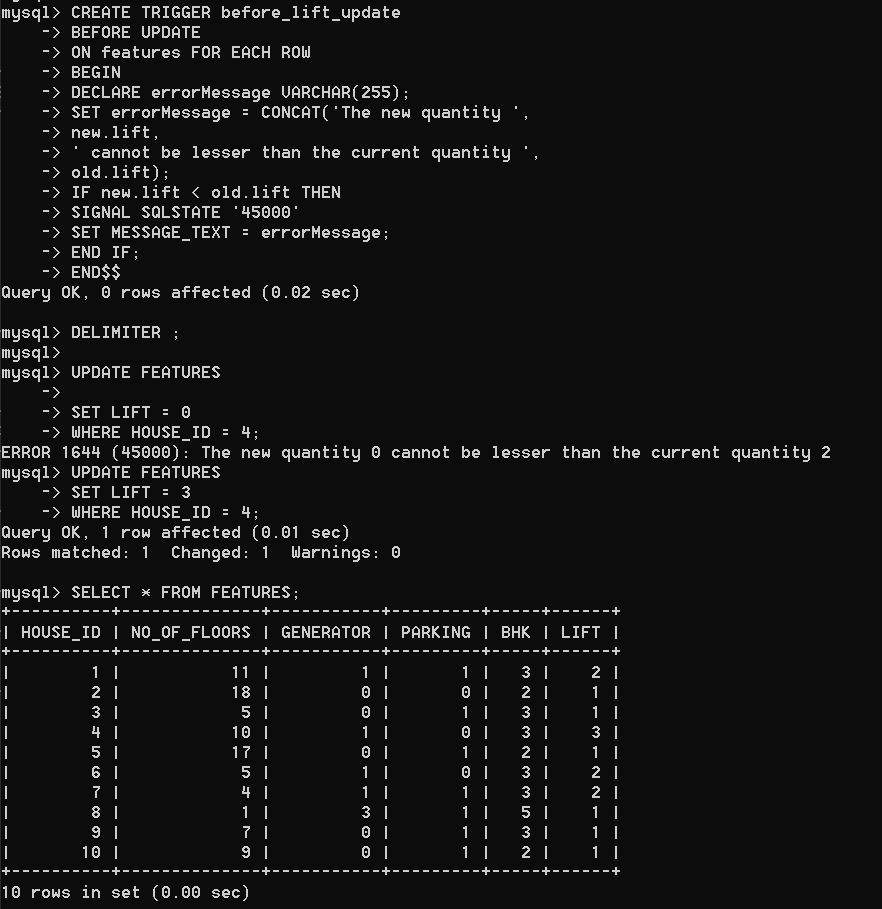
1. Implement a Before Update Trigger on Features table to check and update the number of lifts in the Apartment.



1. If new number of lifts < current number of lifts



1. If new number of lifts > current number of lifts



VIEWS

**VIEW AND ITS SIGNIFICANCE**

A view is like a window through which one can view or change information in the table.

Though it looks like a table, it has no physical significance i.e. it is virtually present.

The data of view is derived from the base tables. It is used to store the data definition and not the copy of the data.

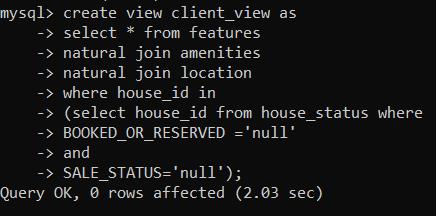
View is a significant part of database systems as it promotes simplicity by showing only the required details and security by preventing unauthorised users from seeing irrelevant information.

The client in our database management system can choose to view houses that are available for buying and can add to those houses by adding into the database about his/her property that they would like to sell.

Admin is responsible to manage the data in the entirety of the system. The agency can modify the reserve and book details of the system along with managing all the agent details that are provided for booking.

Here the client views all the available houses that he can buy and the necessary details that can help support the client’s selection:

**VIEW CREATION AND EXECUTION:**



THANKYOU