VIETNAM NATION UNIVERSITY, HA NOI

**INTERNATIONAL SCHOOL**

**INS2080**

**PROJECT REPORT**

VINHOME RESIDENT MANAGEMENT

**Group Information**

**Name Student’s ID**

**Le Minh Hai 20070923**

**Nguyen Thi Lan Anh 20070054**

**Lecturer**

**Tran Thi Oanh**

Contents

[**1.Overview** 3](#_Toc92873665)

[**2.Business Narratives** 3](#_Toc92873666)

[**3.ERD** 4](#_Toc92873667)

[**4.Create physical database in MySql** 7](#_Toc92873668)

[**5.Inserting data** 10](#_Toc92873669)

[**6.Retrieving the database** 12](#_Toc92873670)

[**7.Conclusion** 24](#_Toc92873671)

# **1.Overview**

Population growth combined with economic growth has boosted infrastructure requirements to the sky-high state. Consequently, the need for a data administration solution in that sector also flourished.

Every apartment requires an effective database to keep track of all pertinent information, such as citizens’ personal information, their interests, what infrastructures they would like to have, etcetera. Citizens are now looking for apartments that satisfy their needs of entertainment, convenience services, and security. A specialized citizens management system will make the managers’ job easier and more effective to satisfy customers’ requirements.

Creating a management database is a challenging and complicated task due to the high required standards by both parties.

This report illustrates the process of creating and using the database to manage residents in Vinhome Smartcity building and provide them with the most suitable service and advertisement. This database also helps managers get the bill of electricity and water, data of using services to have the optimal plan for the next real estate project.

The Vinhome operator will get the data from the customer who bought the house from the house contracts, the entertainment services which they registered on Vin’s application, the parking registration and the data of using electricity and water from providers.

# **2.Business Narratives**

- Each apartment will have a separate apartment ID, with area, number of bedrooms, status (owned, leasing, no buyer, ..)

- Each apartment can belong to only one owner. Owners include name, ID/passport number, gender, age, phone number.

- Each apartment can have only one payment representative. Each agent has its own name and phone number

- Each apartment can have many members, each member has their own name, ID/passport/birth certificate code, age, gender

- This apartment building has three service residents, which are electricity and water services, parking services, and free use of service utilities.

- Utility service including ID and title of service (badminton, tennis, swimming, ...) and no charge

- A member can register for more than one service. Each time people register to use the utilities on the application, the registration date is recorded.

- Parking service including service ID, title (bicycle, motorbike, car), service (/month)

- One apartment can use multiple parking services. Every month when residents register to park their car, the number plate of the car will be recorded by the system.

- electricity and water service including service ID, title (electricity, water) service price. (2000 VND / 1 electricity number; 17000 VND / 1 water block)

- An apartment can use many electricity and water services. All electricity and water usage amounts for each month are recorded.

# **3.ERD**

There are 9 step to create an ERD

* Step1 and 2 – Identify Strong Entities & Attributes, attach attributes to entities

* Based on the business narratives we can have a list of strong entities and its attributes, the attributes underlined is identifier.

APARTMENT: aprtID, area ,numberOfBedRoom

OWNER: id, fullname, gender, age, phone

PAYPERSON: fullname, phone

LIVING\_RESIDENT: id, fullname, gender, age, phone

ENT\_SERVICE (entertainment service): entID, title

ELEC\_WATER\_SERVICE (electricity and water service): ewID, title, price

PARKING\_SERVICE: pID, title, price

PARKING\_LINE: LicensePLate, SUBCR\_TIME

*Another attribute :*

SUBCR\_TIME in the using of ELEC\_WATER\_SERVICE and ENT\_SERVICE

Quantity in the using of ELEC\_WATER\_SERVICE

TotalCost in the using off ELEC\_WATER\_SERVICE

* Step 3 and 4 - Identifying relationship and determine cardinality and participation constraints

One OWNER may own many APARTMENT

One APARTMENT may be owned by One OWNER

One PAYPERSON may pay for the fee of many APARTMENT

One APARTMENT may be paid the fee by One PAYPERSON

One APARTMENT may have many LIVING\_REDIDENT

One LIVING\_RESIDENT must be in One APARTMENT

One APARTMENT may use many ELEC\_WATER\_SERVICE

One ELEC\_WATER\_SERVICE may be used by many APARTMENT

One APARTMENT may have many PARKING\_LINE

One PARKING\_LINE must be belong to One APARTMENT

One PARKING\_SERVICE may include in many PARKING\_LINE

One PARKING\_LINE must be included in One PARKING\_SERVICE

One LIVING\_RESIDENT may use many ENT\_SEVICE

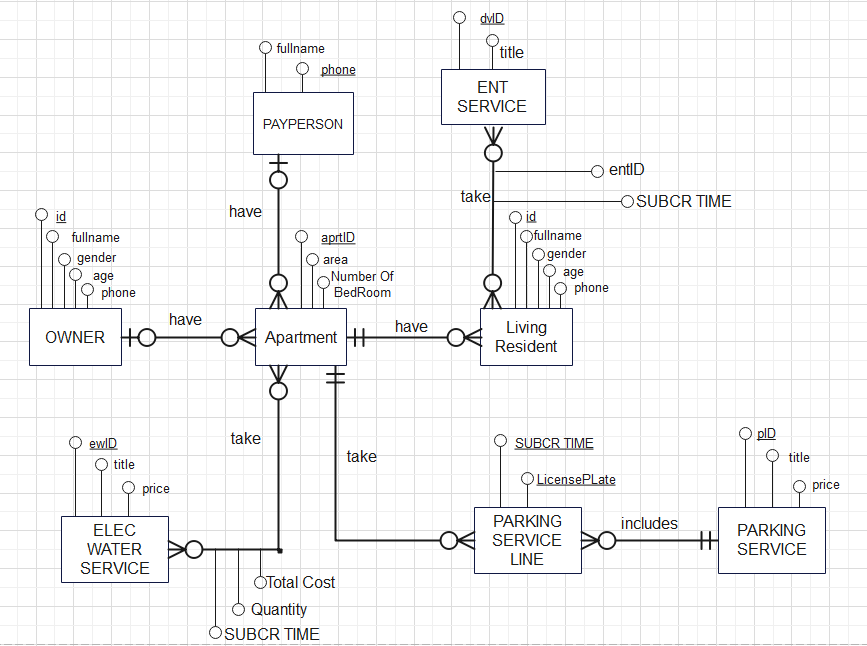
One ENT\_SERVICE may be used by many LIVING\_RESIDENT

Cardinality is yellow highlighted

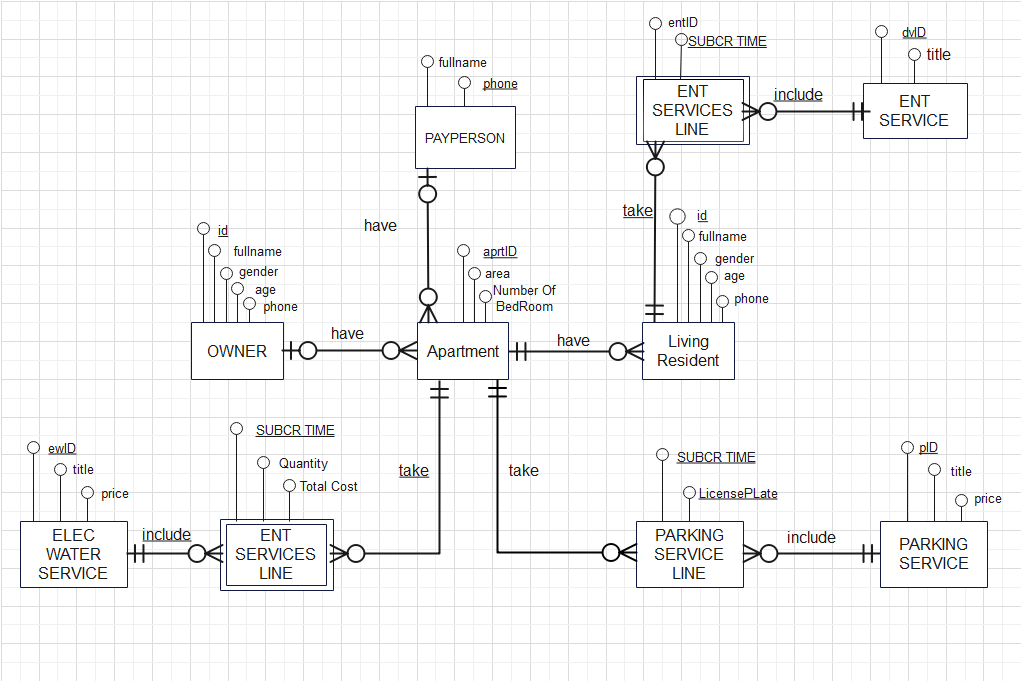
Participation is green highlighted

Relationship is red highlighted

* Step 5 - Attach ALL remaining attributes to entities or relationships



* Step 6 and 7 - expand M:M and reattach attributes



* Step 8 - Conversion to Relational Schema

PAYPERSON (fullname,phone)

OWNER (id,fullname,gender,age,phone)

APARTMENT(aprtID,area,numberOfBedRoom,id,phone)

foreign key (id) references OWNERR (id)

foreign key (phone) references PAYPERSON (phone)

LIVING\_RESIDENT(id,fullname,gender,age,phone,aptrID)

foreign key aprtID references APARTMENT (aprtID)

ENT\_SERVICE(entID,title)

ELEC\_WATER\_SERVICE (ewID,title,price)

PARKING\_SERVICE (pID,title,price)

ENT\_SERVICES\_LINE(entID,SUBCR\_TIME,id)

Foreign key entID references ENT\_SERVICE (entID)

Foreign key id references LIVING\_RESIDENT (id)

PARKING\_SERVICE\_LINE(LicensePLate,SUBCR\_TIME,TypeOfVehicle,Apartment)

Foreign key TypeOfVehicle REFERENCES PARKING\_SERVICE (title)

Foreign key Apartment REFERENCES APARTMENT (aprtID)

ELEC\_WATER\_SERVICE\_LINE(SUBCR\_TIME,TypeOfServices,Apartment,Quantity,CostPerUnit,TotalCost)

Foreign key TypeOfServices REFERENCES ELEC\_WATER\_SERVICE (title)

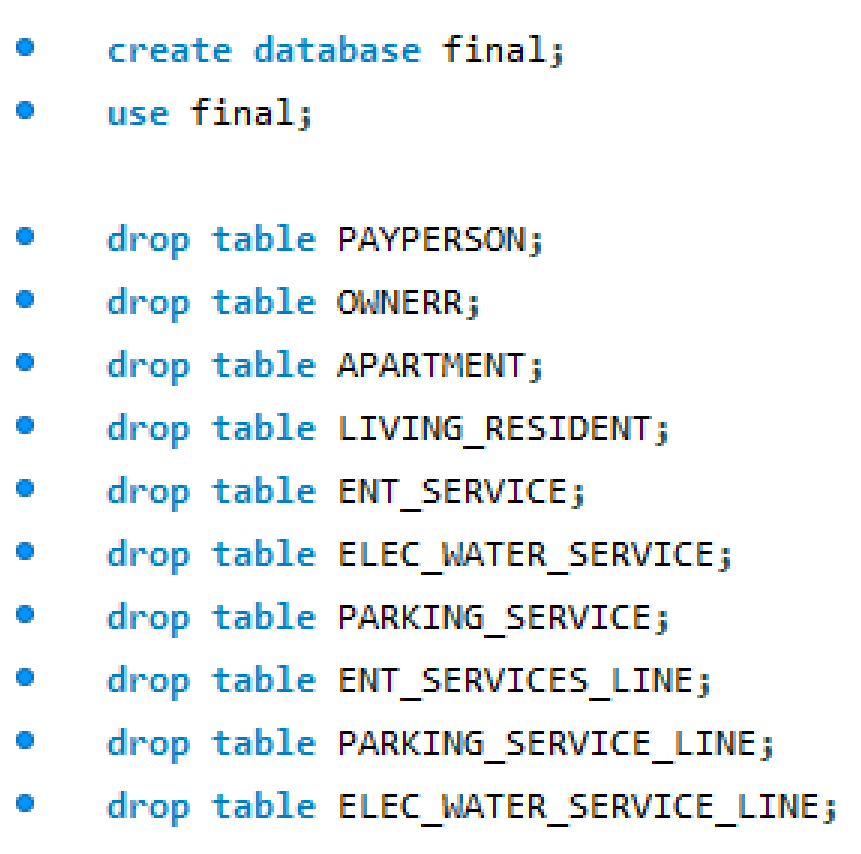
Foreign key Apartment REFERENCES APARTMENT (aprtID)

Foreign key CostPerUnit REFERENCES ELEC\_WATER\_SERVICE (price)

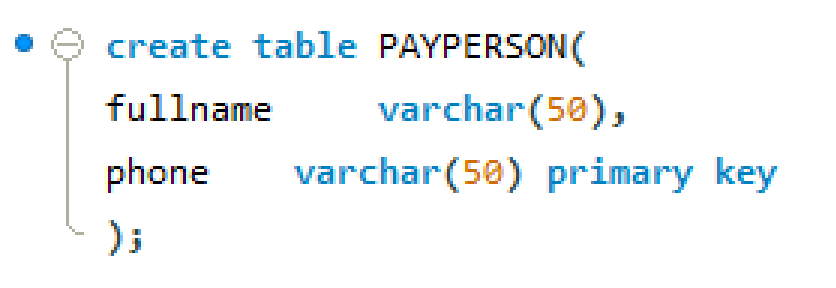
* Step 9 - Testing the erd ( by inserting some data)

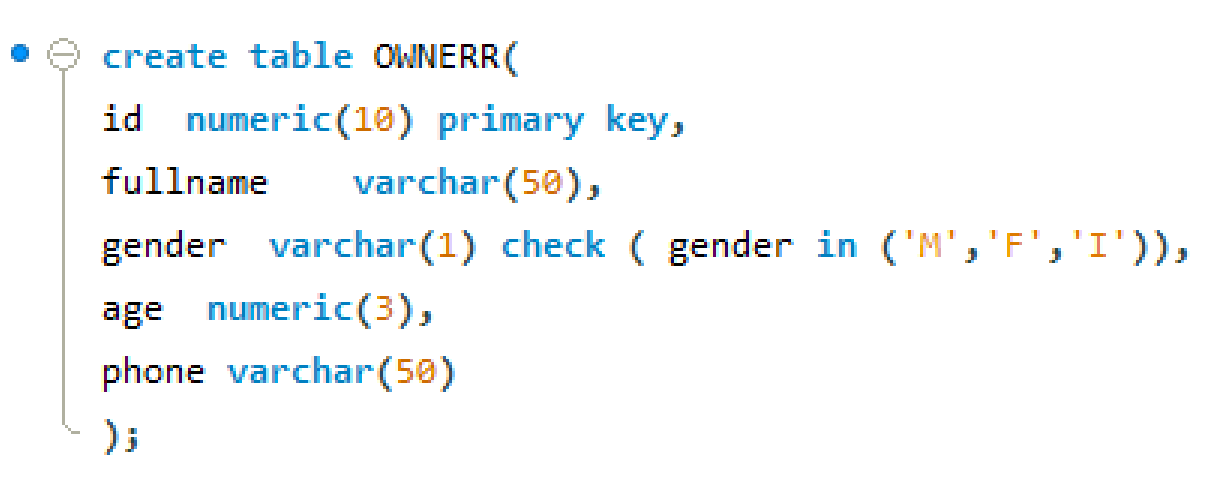
# **4.Create physical database in MySql**

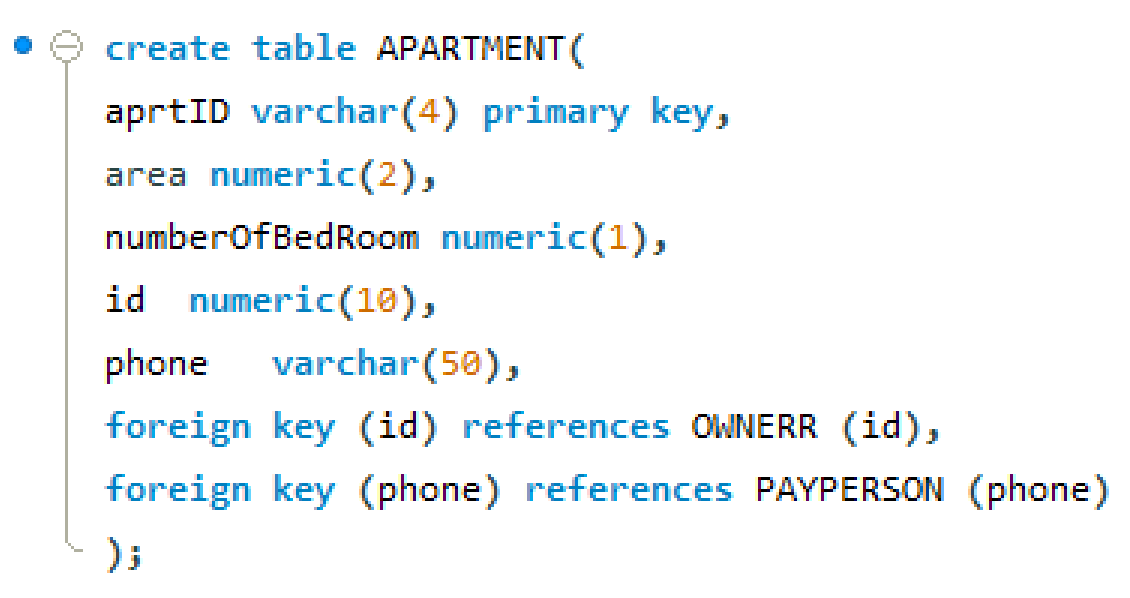
* Creating a database and writing codes to drop tables if they have any errors.

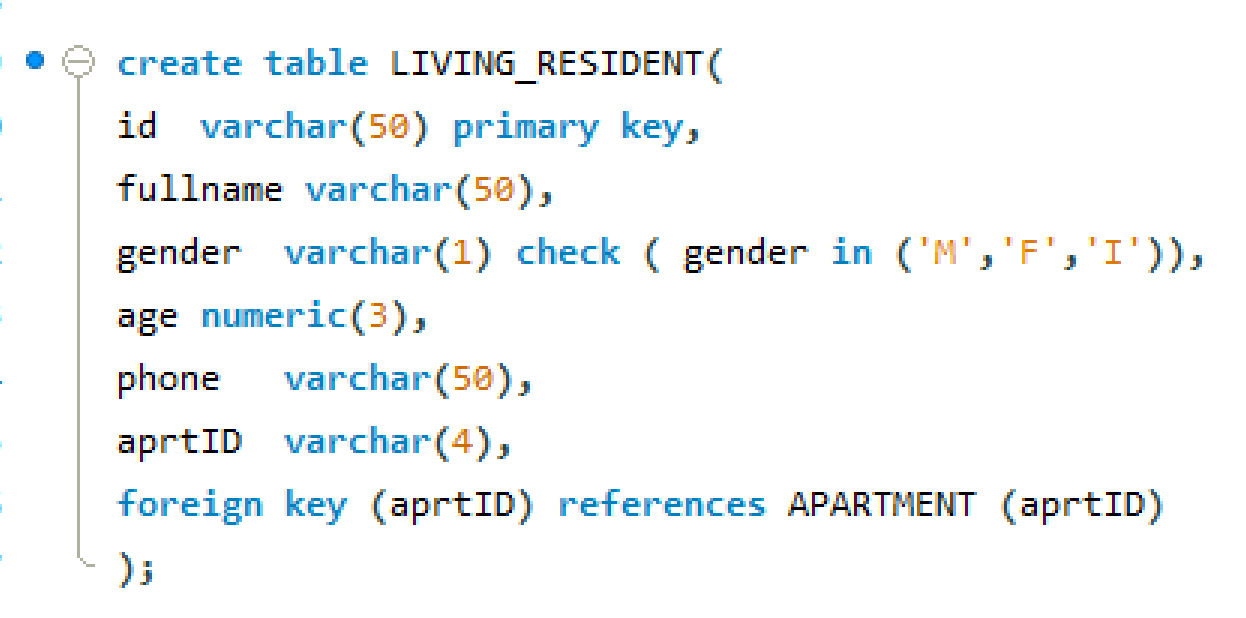


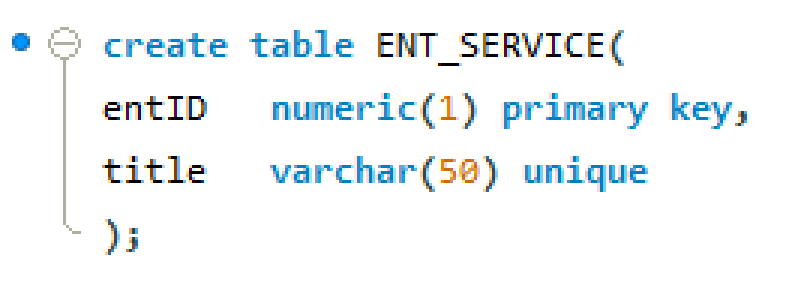
* Creating tables in database

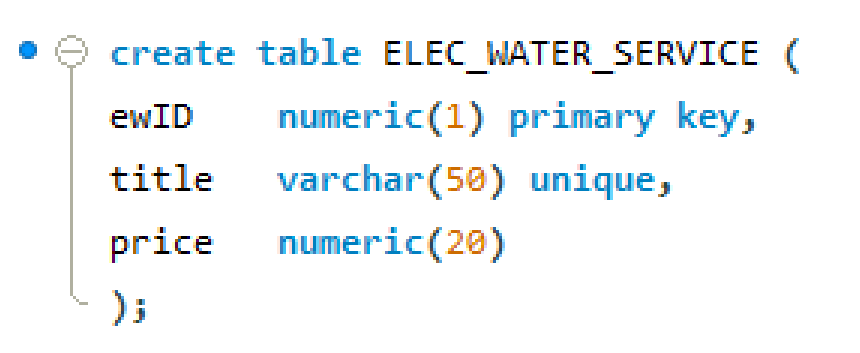
****

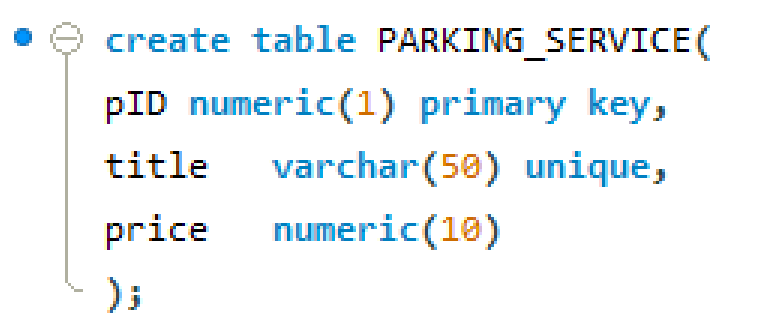


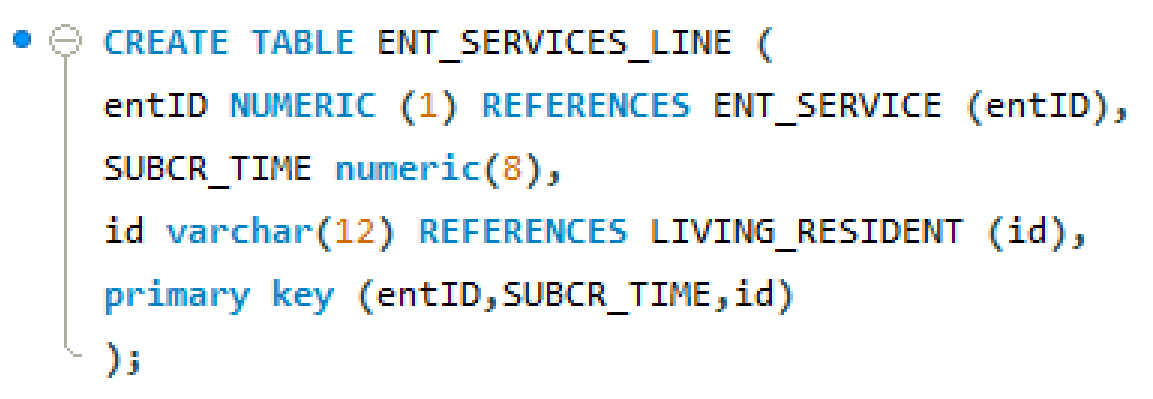


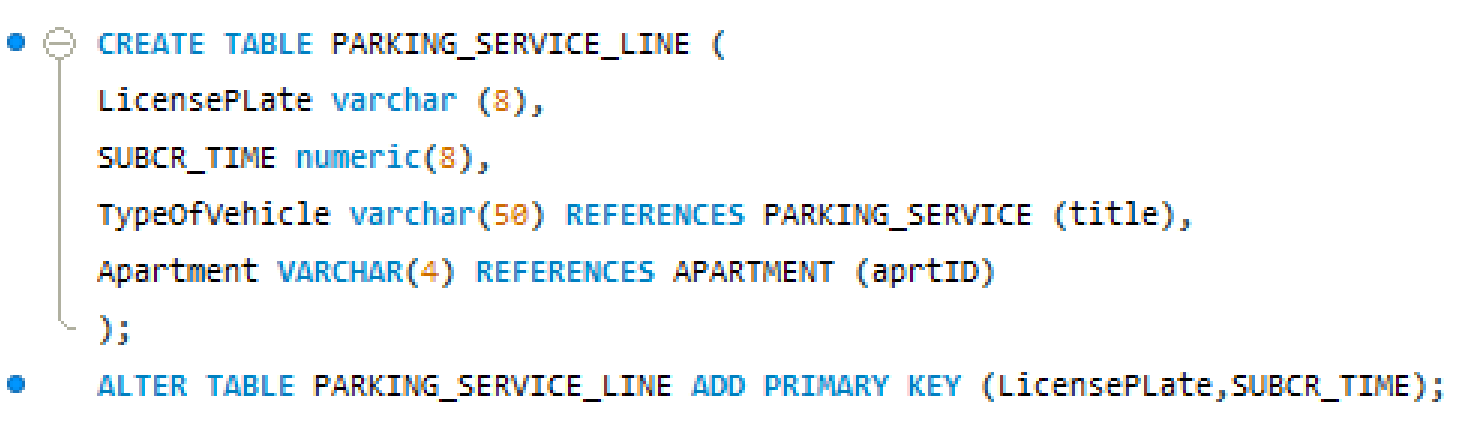


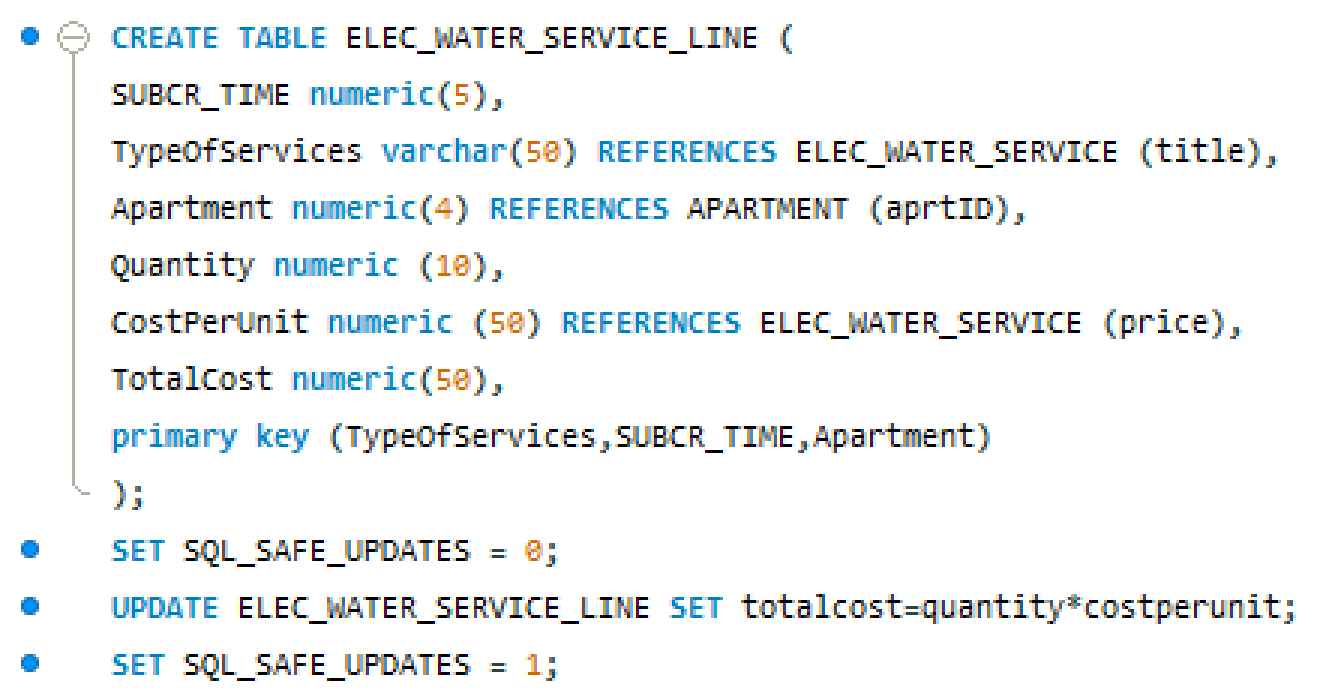












# **5.Inserting data**

This building has 10 floors and 100 apartments. Each table has from 2 to 280 rows. But this report shows you only 10 rows for each table.

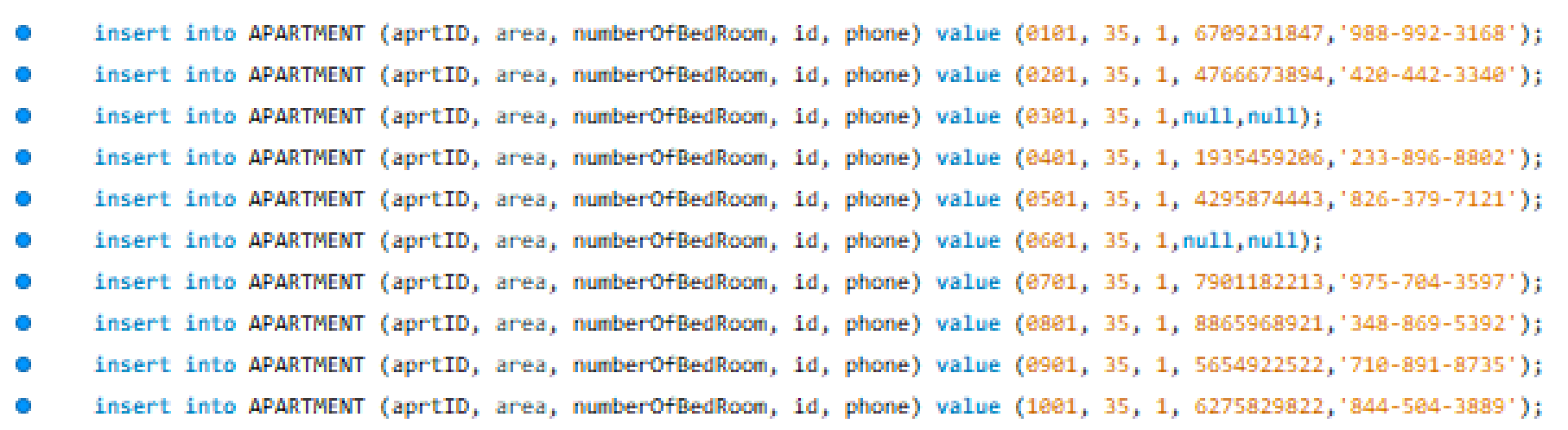
* PAYPERSON TABLE



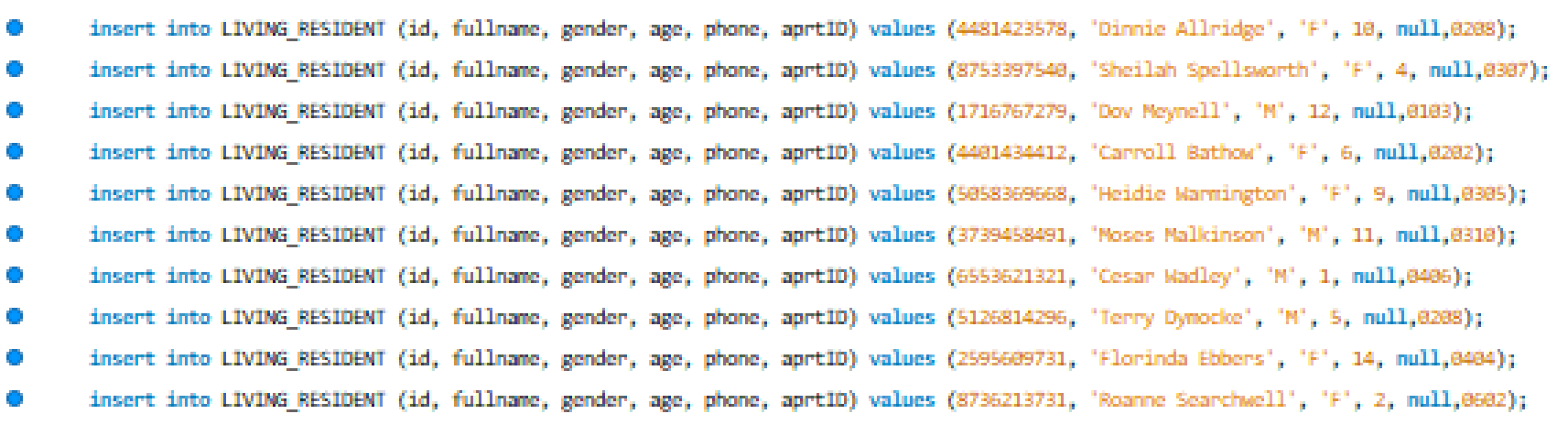
* OWNER TABLE



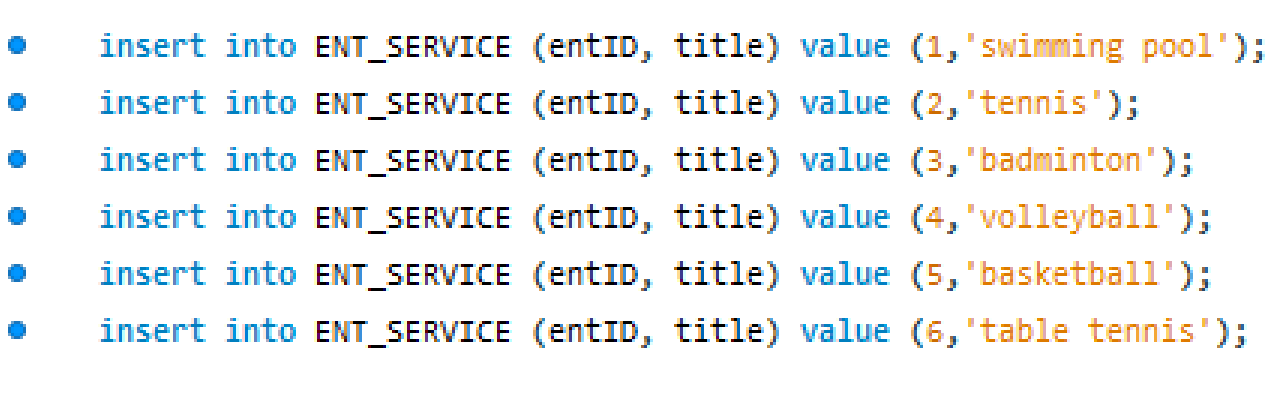
* APARTMENT TABLE



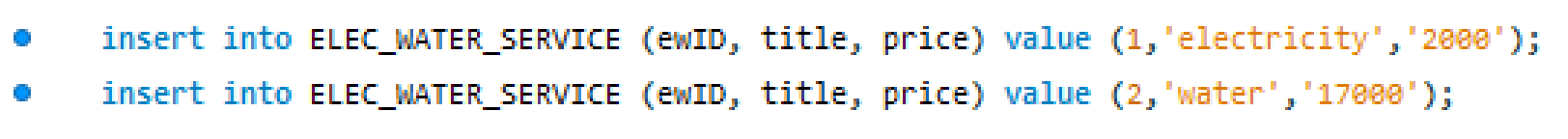
* LIVING RESIDENT TABLE



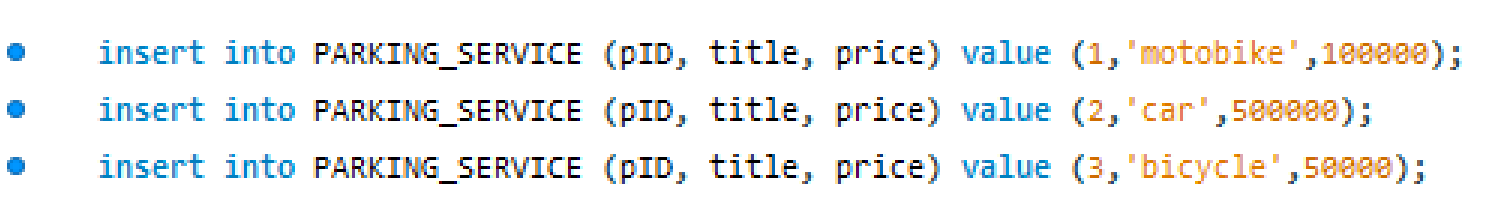
* ENT\_SERVICE TABLE



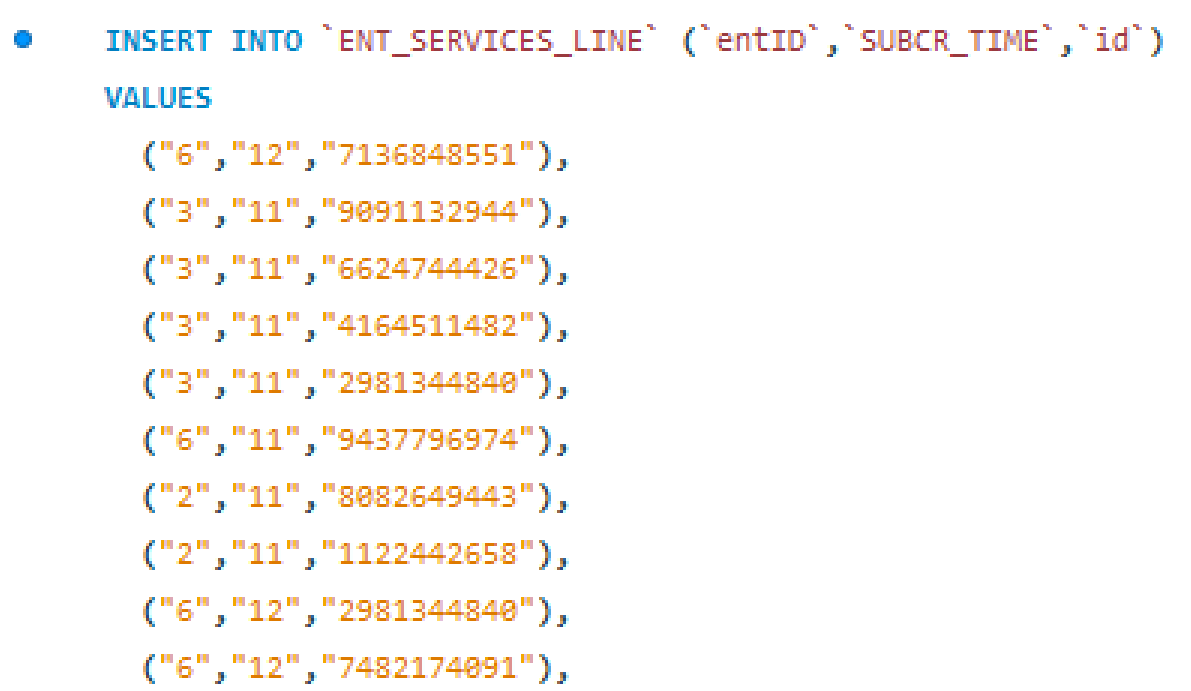
* ELEC\_WATER\_SEVICE TABLE



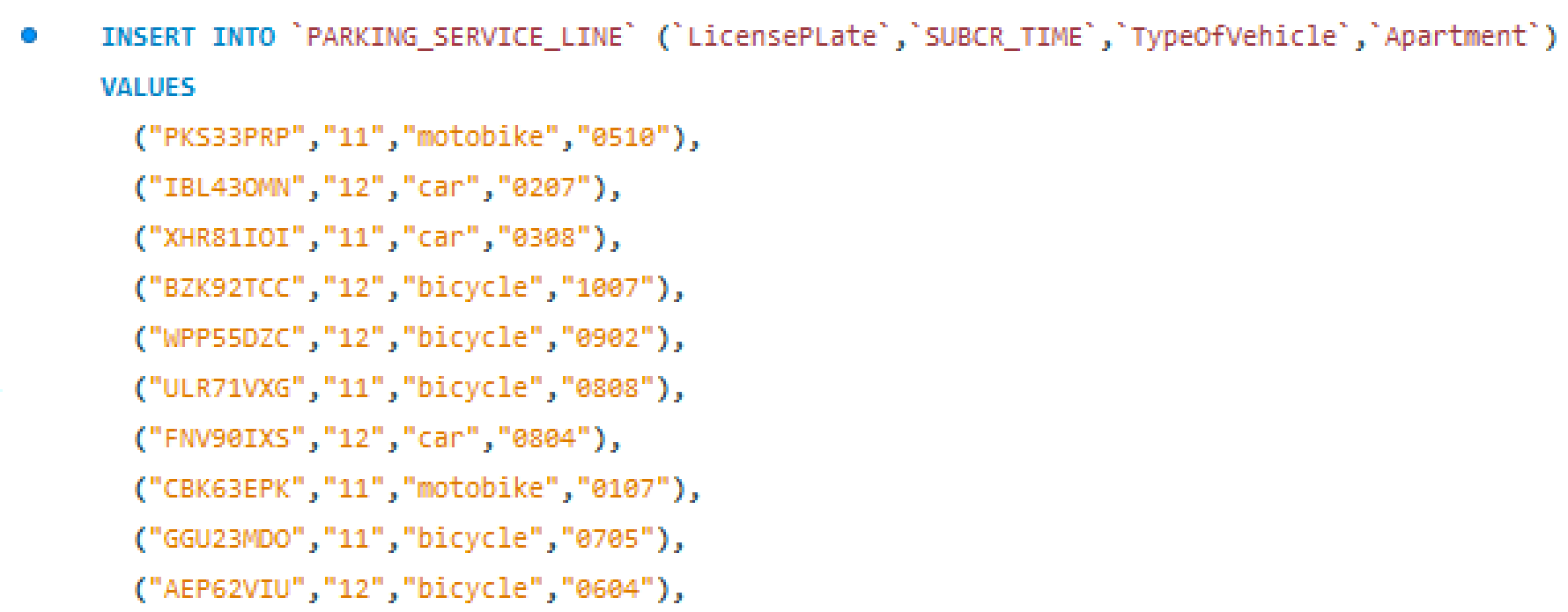
* PARKING\_SEVICE TABLE



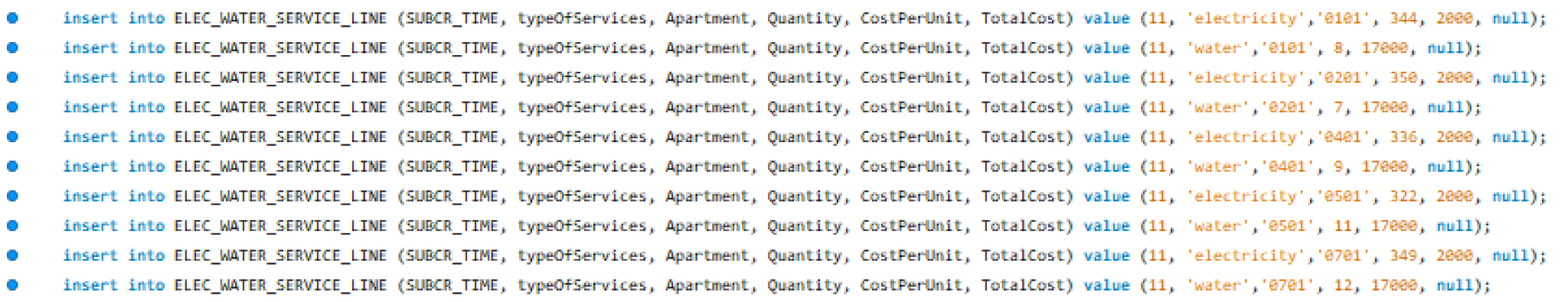
* ENT\_SEVICE\_LINE TABLE



* PARKING\_SEVICE\_LINE TABLE



* ELEC\_WATER\_SEVICE TABLE



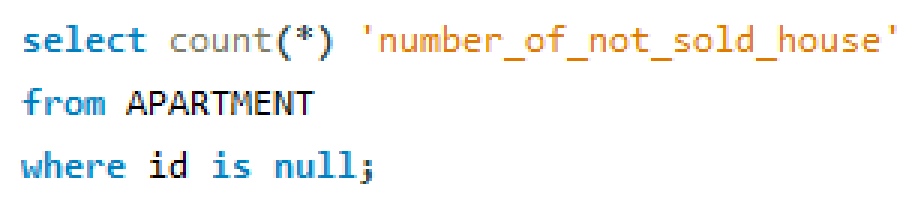
# **6.Retrieving the database**

This part has 10 questions showing us how to use the data effectively. The output not only help Vin managing the residents in this building but also getting the insight of the customers who bought house to boost the sale of apartments.

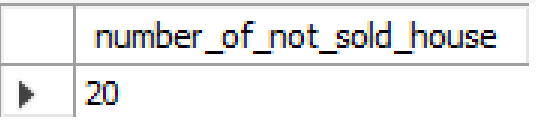
* Question 1

How many apartments are not sold? Classify them by area.

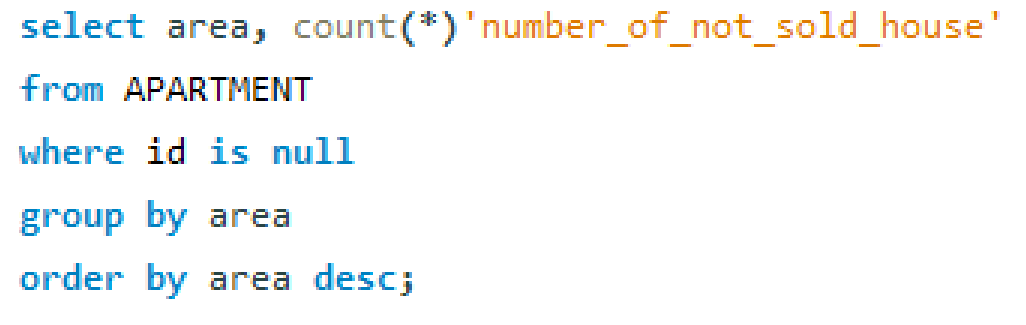
* Firstly, count how many house are not sold in total



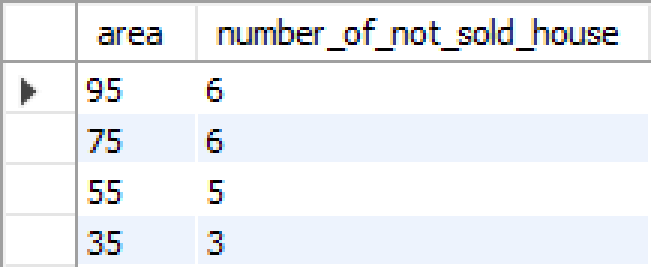
Output



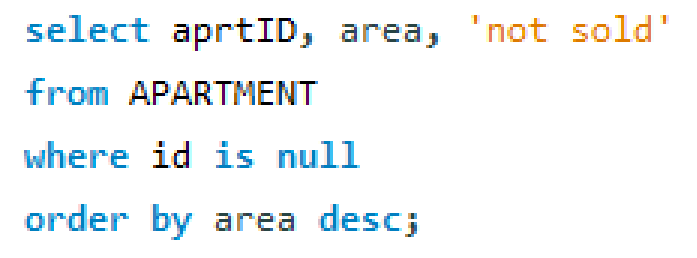
* Then count how many houses are not sold group by area



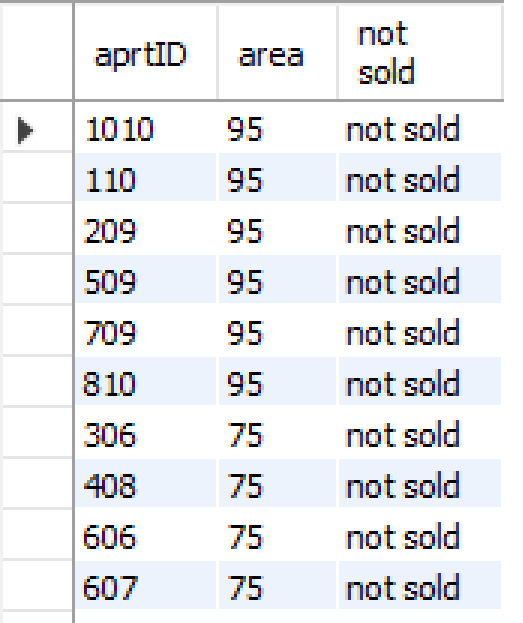
Output



* Define the apartment id of not sold apartments order by the descending area



Output ( first 10 row)

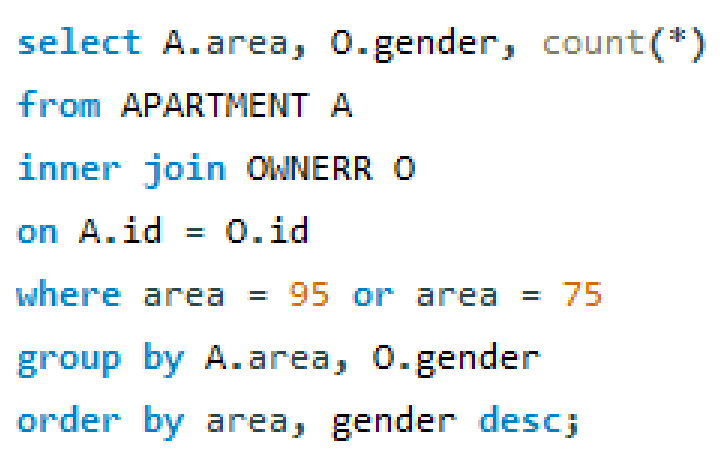


As we can see, apartments which are 75 or 95 m2 have the most number of apartments not sold. So, we need to find the insight of people who buy this type of apartment.

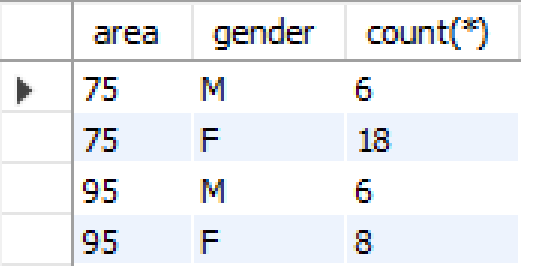
* Question 2

Who will buy the 75 or 95 m2 apartments ?

* We will count the customer who bought 75 and 95 m2 apartment grouping by gender

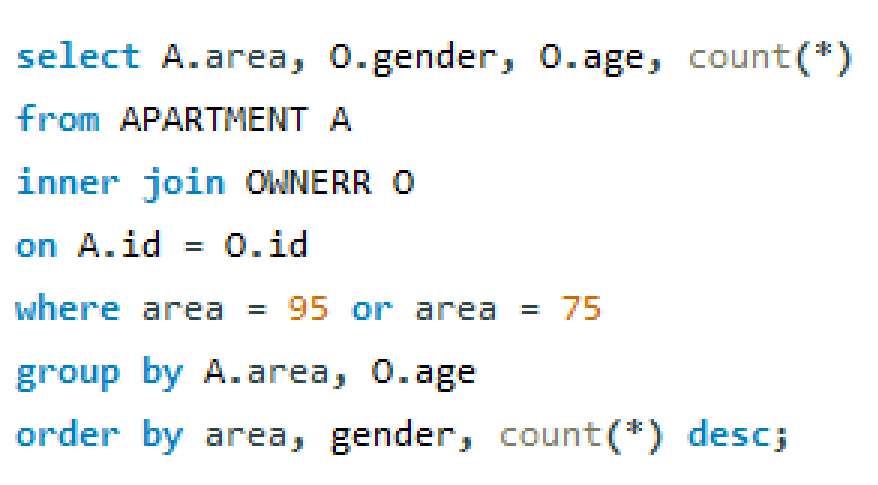


Output

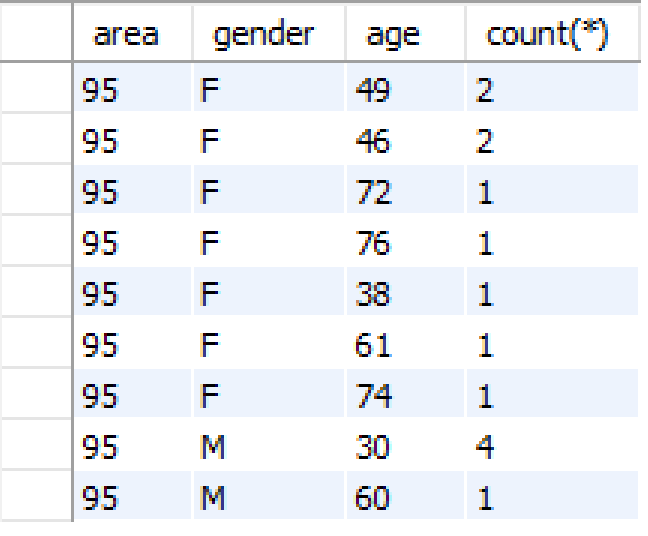
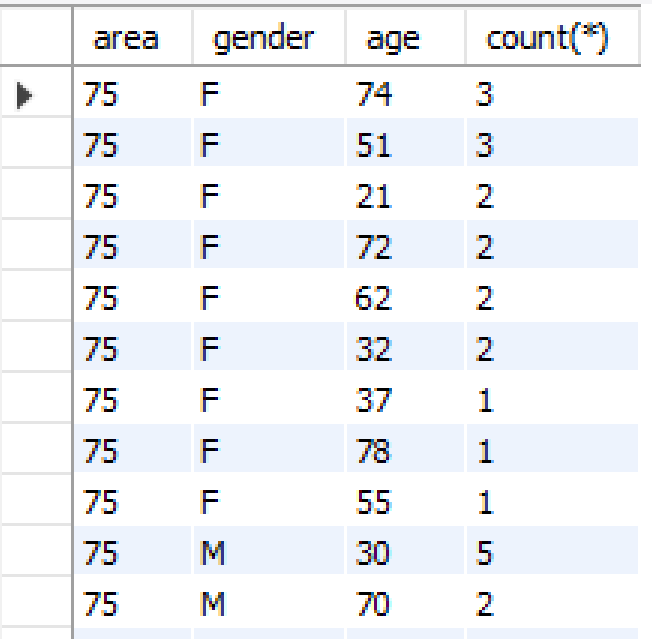


The result show that there are more females bought this kind of apartment than males.

* Count the customer who bought 75 ang 95 m2 apartment group by age



Output



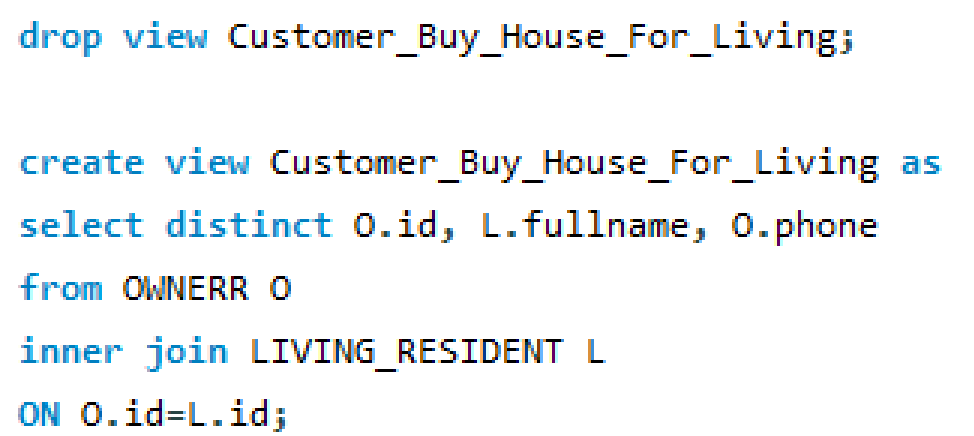
There is no significant dependency between the age of people and the number of people who bought this type of apartment. But we can see that almost customers who older than 30 bought those apartments.

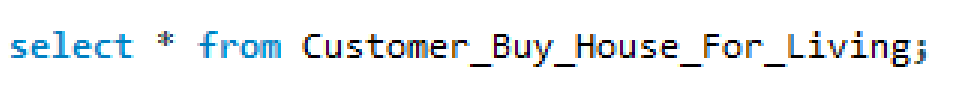
* After answering question 1 and 2, we get the insight of apartments which are not sold so we can send this information to salesmen to sell them.
* Question 3

Who are the owners and also live in their apartments?

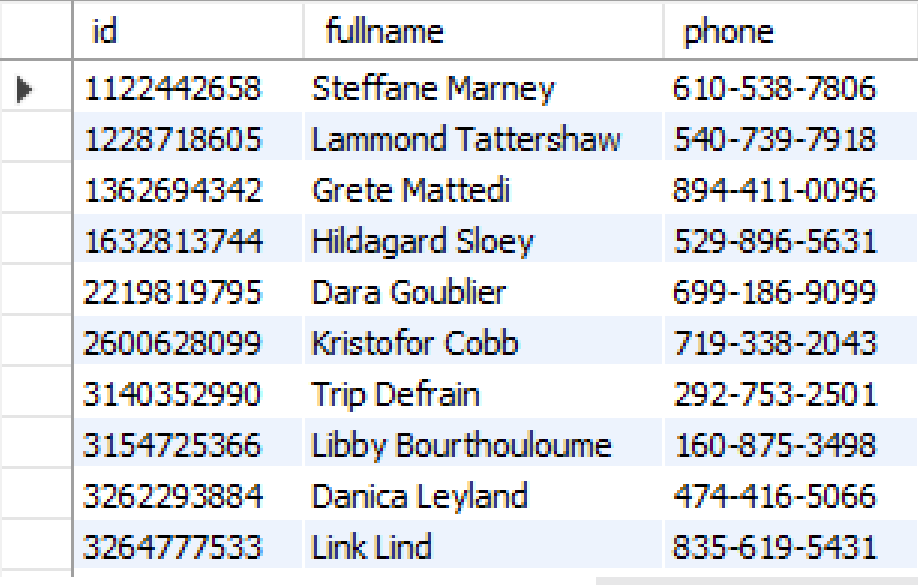
They may be the kind of customer who buys a house for living.

* We have to find the intersection of “id” between OWNERR table and LIVING\_RESIDENT table
* Create a view name customer\_buy\_house\_for\_living.





Output ( first 10 rows)

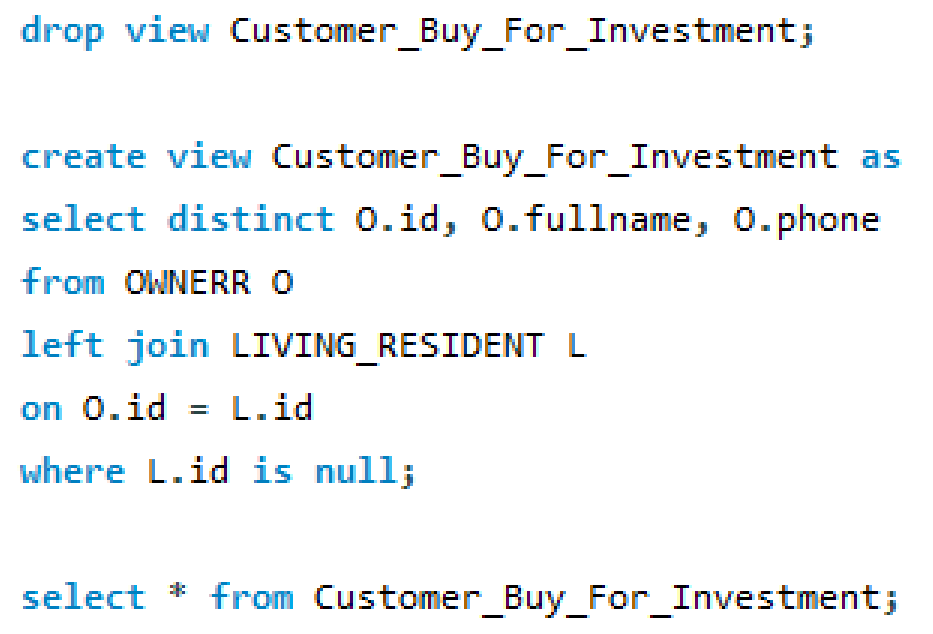


* Question 4

Who are the owners but do not live in the house they bought?

They may be investors who bought houses for rent or selling them for other people in short term to take profit.

* Finding the id have in ONWERR table but not in LIVING\_RESIDENT table.
* Create a view name Customer\_buy\_for\_invesment



Output ( first 10 rows)

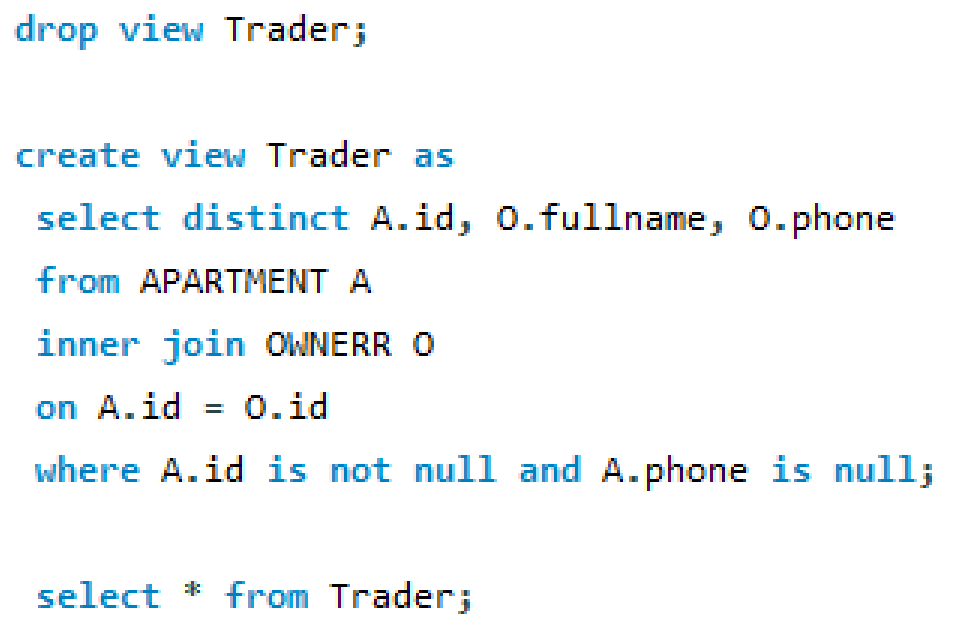


* Question 5

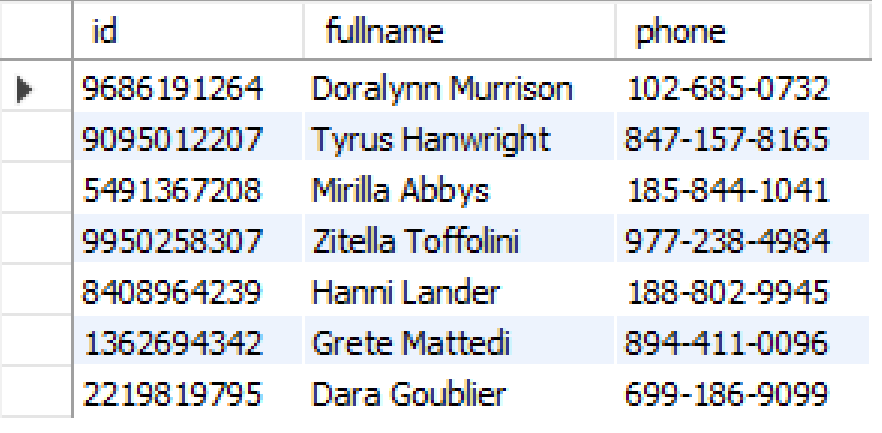
Who bought houses but nobody lives in their house?

Maybe they bought house for trading.

* Because that if a house has nobody lives in, they will not have a payperson. So in the APARTMENT table a house bought but no one lives in is a house whose ‘id’ is not null but ‘phone’ of the pay person is null.
* Create a view name Trader



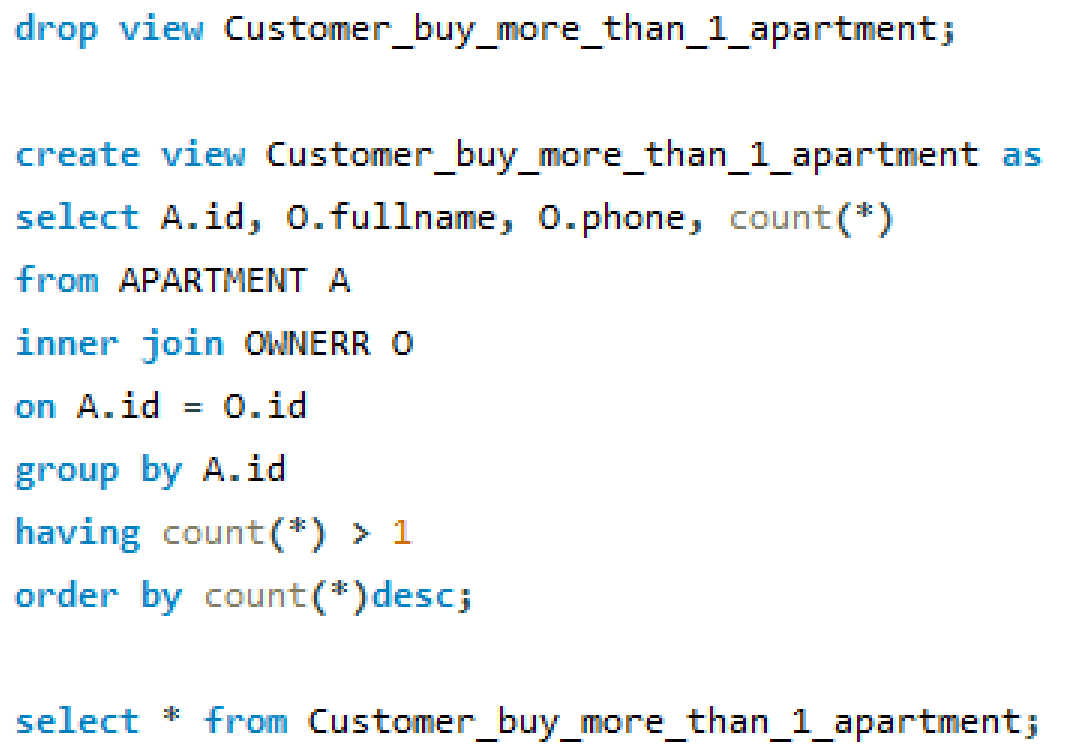
Output



* Question 6

How many customers buy more than 1 house?

* Count the “id” in the APARTMENT table which appears more than ones.
* Create a view name Customer\_buy\_more\_than\_1\_apartment.



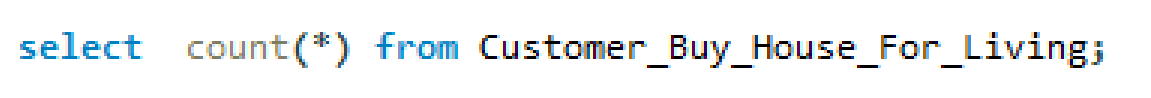
Output (first 10 rows)



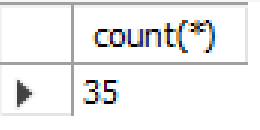
* Question 7

Count the number of all output in question 3, 4, 5, 6

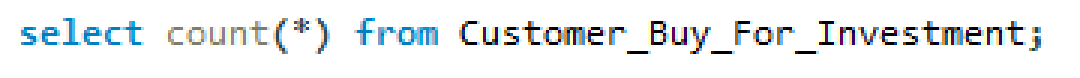
* How many customers buy house for living



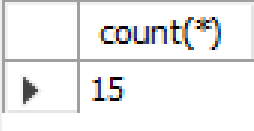
Output



* How many customers buy houses for investment



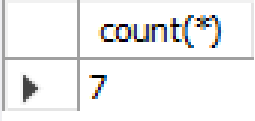
Output



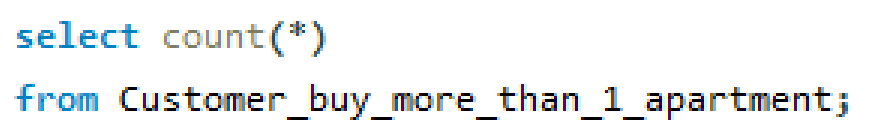
* How many customers buy houses for trading?



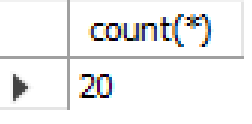
Output



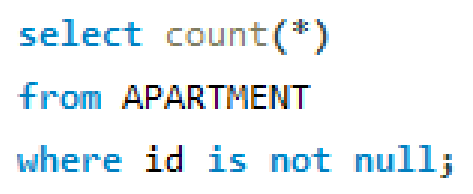
* How many people bought more than 1 house?



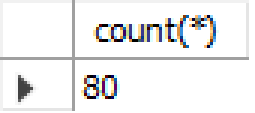
Output



* How many apartments sold?

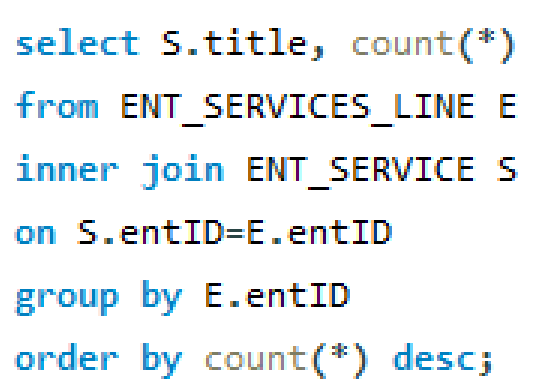


Output

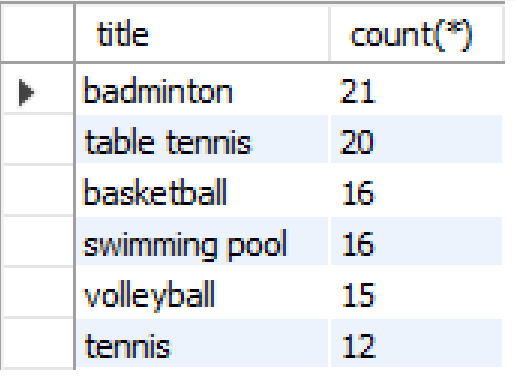


* Question 8

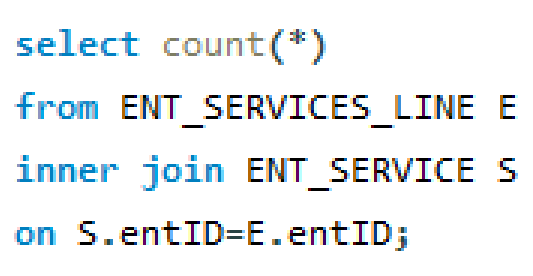
Rank the entertainment service by the quantity of residents using them.



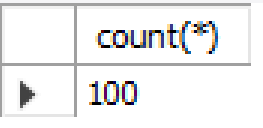
Output



* Total entertainment services used



Output



* After answering question 3, 4, 5, 6, 7, 8 we got the insight, what is the favorite entertainment service they want to use, the purpose they bought the apartment. Then we can send this information to the marketing department to enhance the sale, create the most suitable marketing strategy and send it to the designing department to create the most suitable building plan for the next real estate project.
* Question 9

How to execute a bill for an apartment include:

\_\_Name of payperson

\_\_Phone number of payperson

\_\_apartment ID

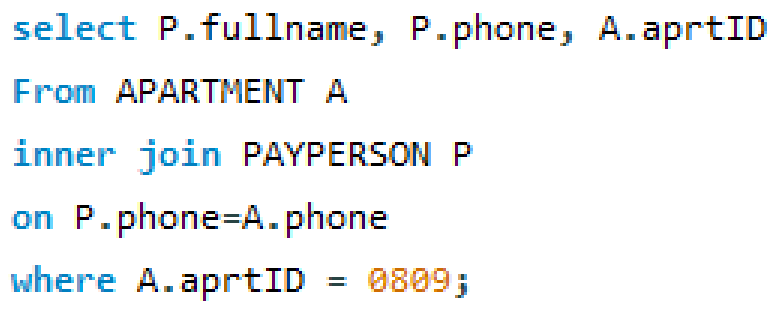
\_\_Month

\_\_\_\_\_Electrictity, quantity, cost per unit, total

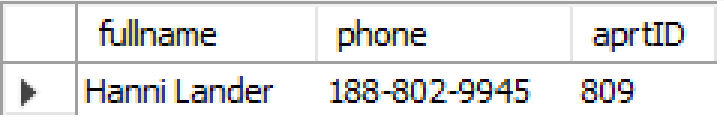
\_\_\_\_\_Water, quantity, cost per unit, total

\_\_\_\_\_(kind of vehicle), License Plate, cost

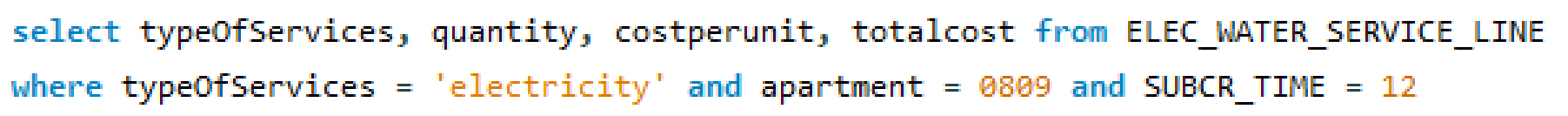
* For example, we will execute the bill for 0809 in December
* Getting the pay person's information of this apartment



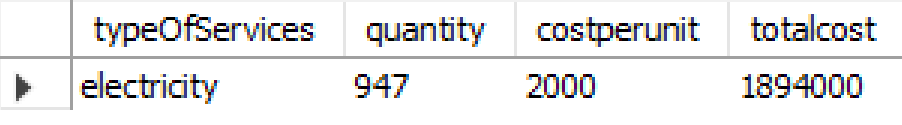
Output



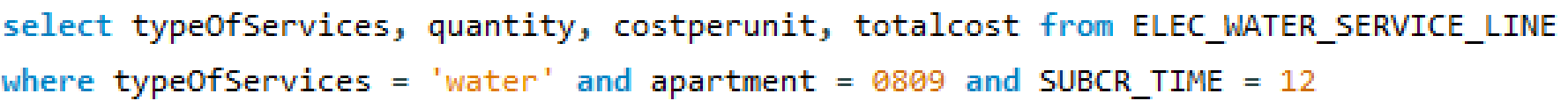
* Get electricity bill



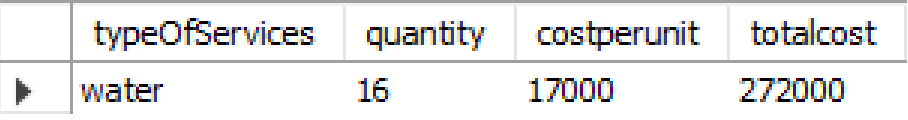
Output



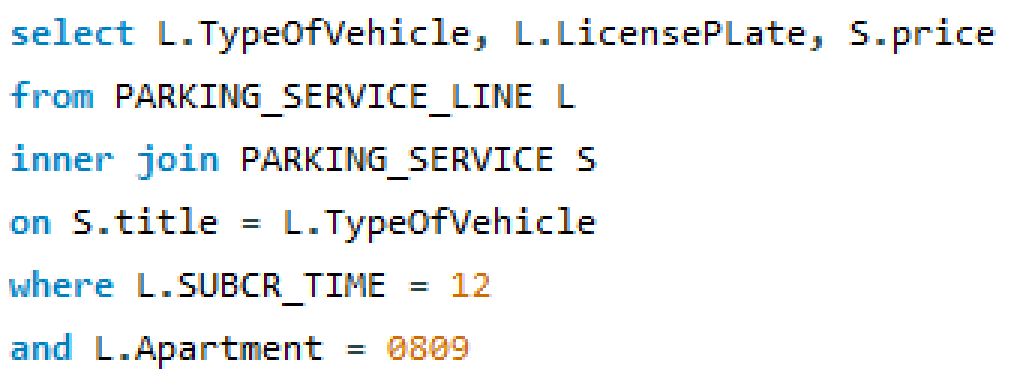
* Get water bill



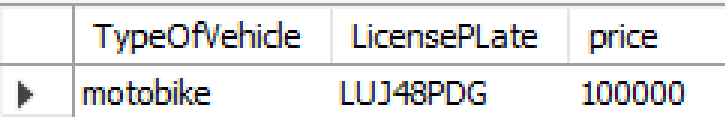
Output



* Get parking bill



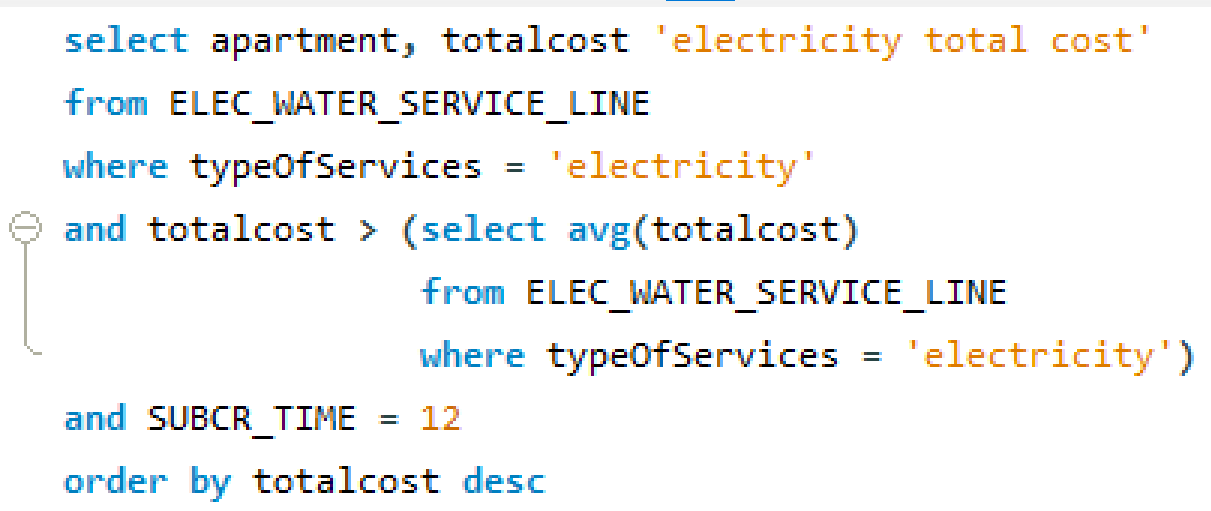
Output



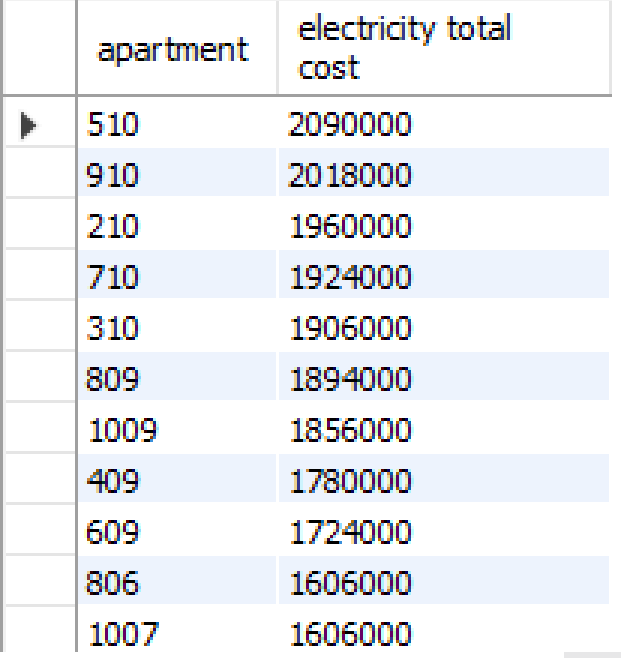
Put all of this output together we will have a completed bill .

* Question 10

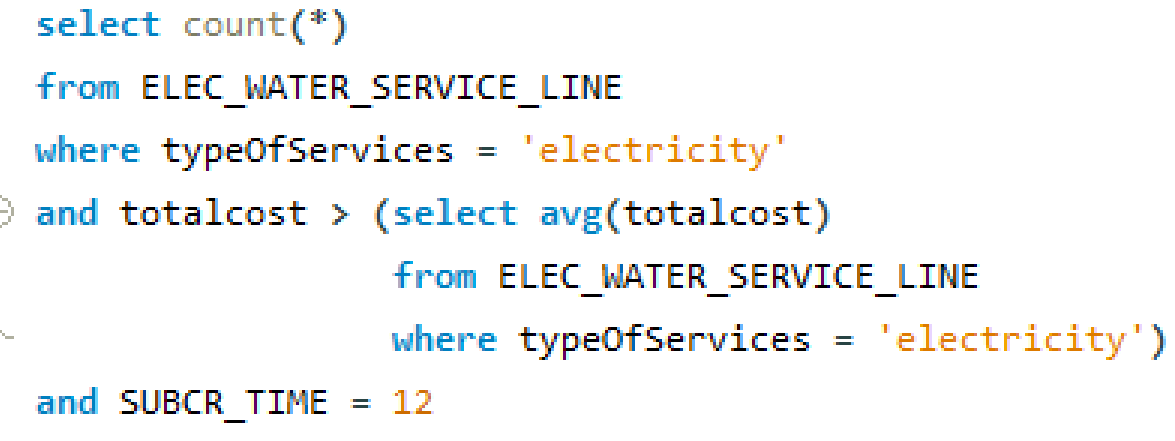
If an apartment’ cost of electricity much than average, they need to pay an extra 50,000 vnd fee for environment.



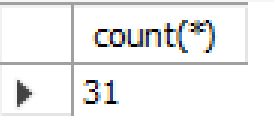
Output (first 10 rows)



* Count the number of them



Output



# **7.Conclusion**

* SQL techniques used in this project

SELECT FROM

WHERE

COUNT, AVERAGE

GROUP BY

HAVING

ORDER BY

INTERSECT

MINUS

INNER JOIN

VIEW

SELECTING LITERAL

SUBQUERIES

Our apartment management system is a practical solution for any building managers who seek accuracy, efficiency at a reasonable price. In our opinion, any technology or toolkits that lighten technical tasks should be highly valued, as our system is not just a convenience system, it's our future.

CONTRIBUTION

|  |  |
| --- | --- |
| Student’s name\_Student’s code | Contribution (part) |
| Nguyen Thi Lan Anh\_20070054 | 1, 2, 3, 4, 5, 6, 7 |
| Le Minh Hai\_ 20070923 | 1, 2, 3, 4, 5, 7 |

--------------------------------------THE END-------------------------------------------