Renting Home System

Dhirubhai Ambani Institute of Information and Communication Technology, Gandhinagar

M.Sc. Information Technology

Submitted to:
Prof. Minal Bhise

Submitted by:
Rithik Kumar Singh
202212087

ACKOWLEDGEMENT

The support and direction of several people play a significant role in every project's success. I would like to take this opportunity to thank everyone who contributed to the successful completion of this project.

I will always be grateful to my professor, Minal Bhise, for being so eager to offer me sound counsel and provide the framework in which I carried out this endeavour. I was able to fully comprehend this project and all its manifestations thanks to her persistent advice and readiness to share her enormous expertise, which also enabled me to finish the tasks that were allocated to me.

I also like to thank my teaching assistant, Mr. Maulik Sarvaiya, for all his time and work this semester. His insightful advice and recommendations were quite helpful to me.

Finally, I want to thank Mr. Ramesh Prajapati, our lab assistant, who made sure everything ran well and ensuring that our lab work was not hampered by any technical issues.

Rithik Kumar Singh
202212087

TABLE OF CONTENTS

SYSTEM REQUIREMENT SPECIFICATION (SRS)

DESCRIPTION

The house renting issue is one amongst the basic parts of society. It is extremely difficult to seek out an appropriate house in big city areas if people look for it physically. On the opposite hand, the house owner also has to rent the house.

It will be difficult to search out an individual just to hold a lease register a building, and thus they lose money.

Increased number of tenants and Landlords makes management difficult especially for the landlords who are losing huge sums of cash through tenants who evade rent.

Management has become difficult due to issues that include:

1. Data growth: Data increases day to day. Storing and maintaining all data manually is extremely difficult.
2. Currently most landlords/property managers use the manual system in recording and maintaining their property and customers data.
3. Data security is not assured in an exceedingly manual way, data is recorded on books/papers which can easily get damaged resulting in loss of information.

This project traverses lots of areas starting from business concept to computing field, and required to perform several researches to be ready to achieve the project objectives.

This system allows users to search for properties which are available for rent in a particular city.

It contains agent's details, so that if needed the tenant can easily contact them. As the tenants register themselves, their details get permanently stored in our database.

The database keeps the history and renting property details of clients. So as the client moves to new house the details regarding same gets stored in our database.

As the properties for rent are posted, every detail about it is asked such as location, area, price, number of rooms, availability of utilities like water, electricity, gas etc. so that its easier for clients to select the property according to their needs.

There is also a nearby property select feature which allow clients to search for place to live near the preferred location.

The details about the properties are updated as soon any changes are made, thus making it easier for clients to not face any problem.

The database has a filter too, which allows clients to search the property based on price, size, area etc. and sort the properties from lowest to highest price.

The System Administrator is responsible for adding, editing, updating, and deleting the data.

Product Functions:

User's registrations, posting properties for renting and details regarding them, editing deleting and updating details, finding properties according to the filters such as location, price, area etc. are the functions of the system.

Tenant, agent, and owner can register themselves providing their details such as first name, last name, phone number (there can be multiple numbers), gender, occupation, date of birth and email. Agent knows the property available for renting. Therefore, the tenant can contact the agent to find out about the properties. Tenant has the freedom to choose or not choose a property. The property also needs to be register with all the details about it such as number of rooms, rent and location. Location consists of city, pin code, state, country, and landmark. An owner can own several properties but each property belong to only one owner.

Information about the payment is also stored such as payment number, amount, payment date, payer id and payee id.

The registered users can login to the system by providing login id and password.

There are different user categories such as end user, database administrator and application programmer. Every type of user can access the database according to the permissions they are given. Users will query and edit the data in the database.

Privileges are assigned according to the user type. End user can only access and edit their data whereas administrator have access to all the things. He manages the database and make changes in it accordingly.

ADVANATAGES OF THE SYSTEM:

- The application can be accessed 24/7.
- Security is usually a problem on the web. Therefore, we try to produce a secure and trustworthy system.
- This online house rental solution is fully functional and versatile
- It is very easy to use.
- Eco-friendly: The monitoring of the Housing activity and therefore the overall business becomes easy and includes the smallest amount of paper work.

The database is expected to store very large amount of data including every information as possible related with clients, agents, and properties.

The Rental Management System is employed to simply identify the appropriate place to live. Hence this method is best applicable for the above reasons making House rental a simple process through an internet system.

The proposed renting house system is supposed to work very efficiently with many features. The application will provide faster and improved opportunities to get houses, as well as ensuring the availability of houses for rent in the greatest number of areas.

REQUIREMENT COLLECTION PHASE

Requirement Gathering is a process in which we understand and identify a business's project technical requirements and then proceed with a well-defined plan. It is one of the essential phases in a project plan.

1. Interview: It is one of the most effective techniques for requirement gathering. In this method, we talked to the user and clients who are unable to give out detailed information as they are not aware of the system development and related functionalities and tried extract relevant information.

   Relevant stakeholders were identified and they were asked lots of questions like what are the overall outcomes clients want from the product, what are company's business goals, what are the actionable and measurable objectives company needs to achieve to realize those goals and outcomes.

2. Brainstorming: It was used in requirement gathering phase to get as many ideas as possible from group of people and was used to identify possible solutions to the complex issues

3. Survey: A survey was conducted to collect information and requirements within a short period of time. First the goal of the survey was determined and thereafter the questionnaire was draft. Once questions list was ready, it was delivered to the user as well as the stakeholders for answers.

4. Observation: The team was observed in working environment and get ideas about the software. The process flow, steps, sore points and opportunities for improvement were identified and observation was documented.

BACKGROUND READING

Magic bricks**:** It is a high-end property portal that caters to a global market with its unique services.

The Magic bricks design is based on meticulous research, unique product developments, and innovative initiatives which has been readily accepted by the users.

To support their growing online traffic, Magic bricks migrated from the proprietary database to MariaDB.

With this migration, they refactored their application architecture to separate read and write database calls. This has allowed them to load balance of their heavy read calls across multiple instances of slaves without any worry of lag during data syncs.

Using MariaDB, they are now able to serve approximately 7 million page views and approximately 6 million API calls per day. MariaDB has not only helped to support this high volume of traffic but has also smoothened their database related operations.

On magicbricks all those owners who want to rent/sell their properties, buyers, tenants and agents can register and proceed accordingly. From its large database consisting of huge amounts of data of different properties and agents, it makes the experience of buyers, agents and tenants very smooth. Their database management system is such that it ensures smooth functioning and storage of data with no comptonization of security of data.

References

https://w3techs.com/sites/info/magicbricks.com

https://cio.economictimes.indiatimes.com/news/business-analytics/heres-how-magicbricks-achieved-60-higher-accuracy-in-lead-generation/74073651

https://mariadb.com/resources/blog/magicbricks-migrates-to-mariadb-to-support-its-high-volume-traffic/

Interview Plan

Interview Plan 1

System: Renting Home

Participant:
Yash khatri
Sanket Acharya

Date: 26/8/2022 **Time:** 11.00 am

Duration: 30 minutes

Place: Gandhinagar

Purpose of Interview:
Preliminary meeting to identify problems and requirements regarding the current database to understand how existing services can be improved and how security of data stored is ensured.

Agenda:
To discuss and deliberate upon the frequency of these services which are accessed to improve the user experience.
Initial ideas
Follow-up actions

Documents to be brought to the interview:
Rough plan of building database.
Documents related to current storage of data of users
Documents related to database system used

Interview plan 2

System: Renting Home

Participant:
Naresh Suthar
Antriksh Jain

Date: 27/8/2022 **Time:** 12.00 pm

Duration: 30 minutes

Place: Gandhinagar

Purpose of Interview:
To understand how existing services can be improved and how security of data stored is ensured.

Agenda:
To know about how the stored data is secured and how they are constantly improved.

Interview Summary 1

System: Renting Home

Participant:
Yash khatri
Sanket Acharya

Date: 26/8/2022 **Time:** 11.00 am
Duration: 30 minutes

Place: Gandhinagar

Purpose of Interview:
Preliminary meeting to identify problems and requirements regarding the current database to understand how existing services can be improved and how security of data stored is ensured.

1. Data increases day to day. Storing and maintaining all data is very difficult. So, performance gets affected and data security too.
2. Data is currently stored in relational or non-relation, graph-dB, Document DB.
3. Various types of data stores are used because of the need to meet scalability and performance.
4. Centralized data where data management (governance, quality, security) are easy.
5. The information about the user (buyer, sellers, agents and tenants) and properties as collected by the company is:
   (a) information supplied by end users and
   (b) information which is automatically tracked while navigation
6. Collect additional information at other times when users provide feedback, comments, change his/her content or responding to a survey, or any communications with the company.

Interview Summary 2

System: Renting Home

Participant:
Naresh Suthar
Antriksh Jain

Date: 27/8/2022 **Time:** 12.00 pm
Duration: 45 minutes

Place: Gandhinagar

Purpose of Interview:
To understand how existing services can be improved and how security of data stored is ensured.

1. Appropriate security measures are taken to gaurd against unauthorized access to, disclosure and destruction of data are taken.
2. The security measures include internal review of the data collection, storage and processing practices and security measures, including appropriate encryption and physical security measures to guard against unauthorized access to systems where store personal data is stored.
3. All the information gathered on TIL (Together I Learned) is securely stored within the Company controlled database. The database is stored on server which is secured behind the firewall, access to servers is password protected and is strictly limited.

Questionnaire

These questions were asked from general public:

1. Type of residential property you are looking for

- House/Villa
- PG/Hostel
- Shared Flat
- Flat
- Other:

2. What's your ideal budget

- up to Rs 5000
- Rs 6000 - Rs 10,000
- Rs 10,000 - Rs 15,000
- Above Rs 15,000

3. With which lease term are you comfortable with

- 3 months
- 6 months
- 11 months
- Other:

4. House/Apartment type you prefer

- 1 RK
- 1 BHK
- 2 BHK
- 3 BHK
- Other:

5. Number of bathrooms/washrooms preferred

- 1
- 2
- 3
- Other:

6. Can you offer one month's rent and security deposit in advance

- Yes
- No

7. Do you require parking space or not

- Yes
- No

8. Are you willing to pay for utilities (water, gas, and electricity)

- Yes
- No

9. Have you ever been evicted?

- Yes
- No

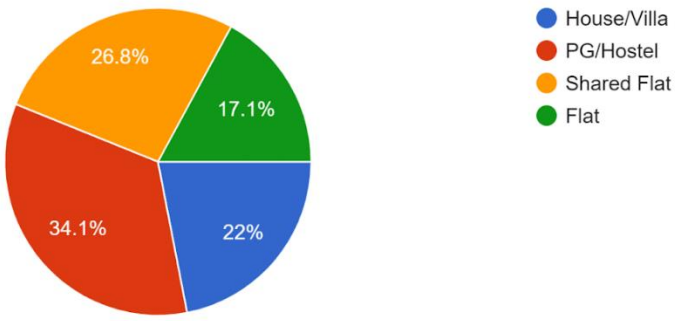
10. What age category do you fall in?

- below 18
- 18-22
- 23-50
- above 50

Responses

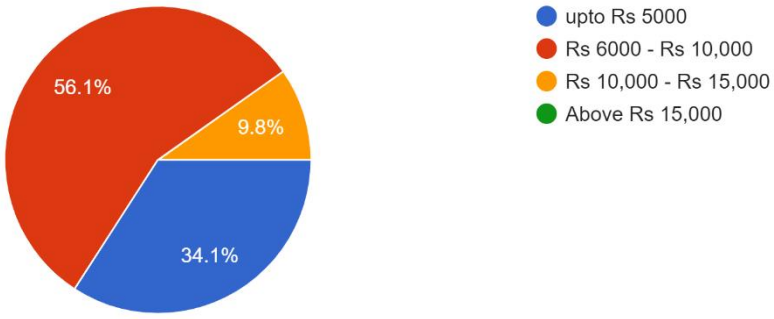
1. Type of residential property you are looking for

41 responses



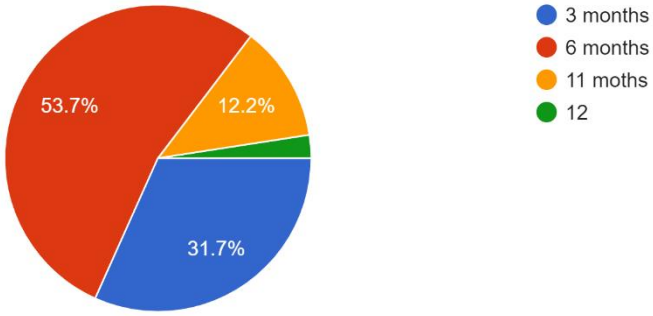
2. What's your ideal budget

41 responses



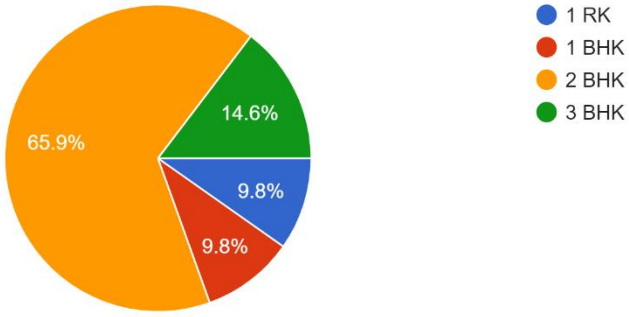
3. With which lease term are you comfortable with

41 responses



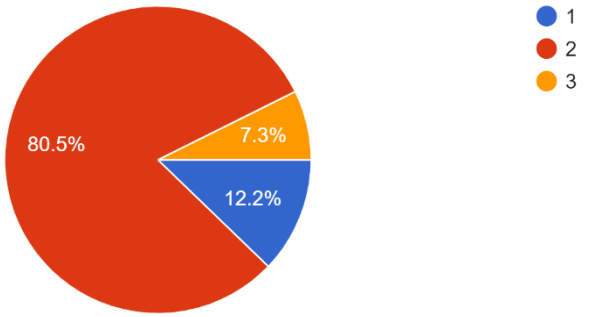
4. House/Apartment type you prefer

41 responses



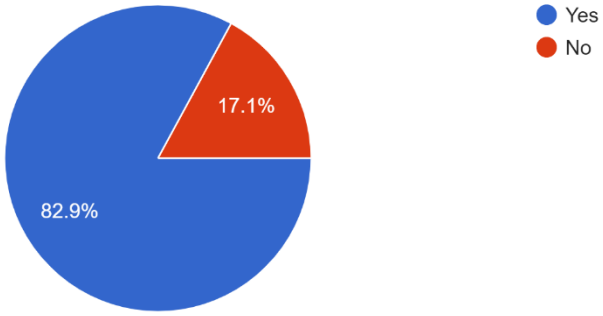
5. Number of bathrooms/washrooms preferred

41 responses



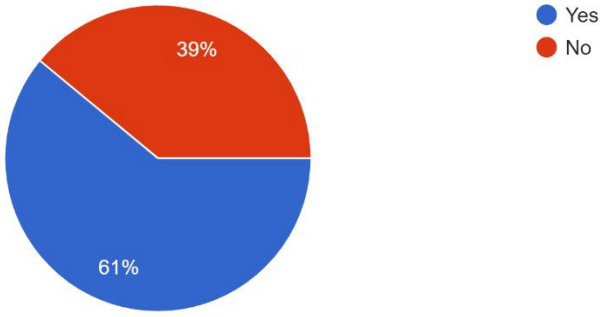
6. Can you offer one month's rent and security deposit in advance

41 responses



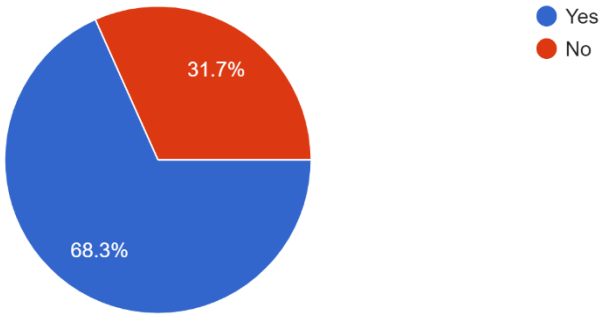
7. Do you require parking space or not

41 responses



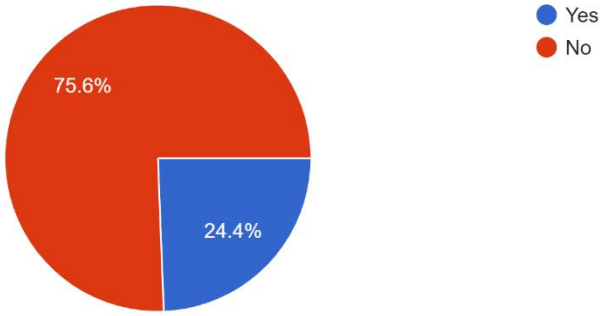
8. Are you willing to pay for utilities (water, gas and electricity)

41 responses



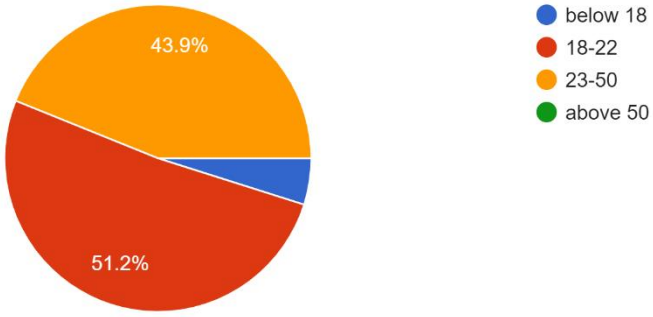
9. Have you ever been evicted ?

41 responses



10. What age category do you fall in ?

41 responses



Observation

We spent a few days understanding how the company and the department works.

They listen to their colleagues through various forums. Through these forums they measure the engagement level of colleagues, understand their strengths, and identify areas of improvement in people/business processes and practices.

Colleagues to interact with leading members in an open gathering through anonymous questions. This helps them in working out problems.

The database department has a great working environment and the senior management is friendly and supportive.

Work is very challenging and employees do extra work. But at the same time, you will get to learn many things and it expands your imagination.

FACT FINDING CHART

Objective	Technique	Subject(s)	Time Commitment
To get the background of the company and how it works	Background Reading	Company reports, Trade journals	0.5 day
To understand the problems and requirements regarding current databases.	Interview	Chief technical Officer	30 minutes
To understand how data of users is collected and processed	Interview	Sr Data Analyst	45 minutes
To understand how the security of data stored is ensured.	Interview	Operations And data services Head	30 minutes
To follow up development of database system	Observation	2 software engineers	1 day
To determine how current computer system is used and its functionality	Interview	Computer Manager	1 hour
To understand the needs of the users and to collect information and requirements within short time period.	Questionnaire	General public	1 day
To understand the working culture of company and the department	Observation	1 Data analyst 2 Software Engineers	2 days

Requirements

Functional Requirements:

1. Should allow end users to register themselves using details provided by them such as name, phone number, email etc.
2. The users should be able to login using email/user name and password.
3. Should allow the owner of house to advertise the house, upload images of the house and adding other information about the house.
4. Should allow tenants to search for house they want and nearby houses.
5. This database should be able to provide the information about properties based on filters [price, size, area etc.]
6. The system should be able to include and manage history and renting properties details of clients.
7. The administrator should be able to:
    - Edit data in the database
    - Delete data in the database
    - Update data in the database

Nonfunctional Requirements:

1. Performance Requirements: The system should respond within short period of time.
2. Security Requirements: All data must be stored, protected, and encrypted. The system should only permit the user whose usernames and passwords match the ones saved in the database from logging into the system. Unauthorized person cannot log in or access the system.
3. Safety Requirement: The data should be backed up in every interval of time (say 1 hour). Under failure in some circumstances, system should be able to come back to normal operation under an hour.
4. Software System Attributes:
    4.1 System should be flexible enough to provide space to add new features and to handle those features conveniently.
    4.2 The system should be maintainable.
    4.3 The system should be able to be tested to confirm the performance and client's specifications.

User Categories

1. Native users (End users) − In home renting system, the end users are tenants, property owners and the agents. The end users need not be aware of the presence of the database system. No prior knowledge or skills are expected from the end user. End user should have basic technical knowledge.
2. Database Administrator (DBA) − People designing the database for a specific purpose. The DBA designs schemas, provide security, restores the system after a failure, and periodically tunes the database to meet the user needs. An administrator of the system should have more knowledge of internal modules of the system and are able to rectify small problems that may arise due to power failures and other catastrophes.

3. Application programmers develop applications that use DBMS functionality to access and manipulate data.
4. System Analyst: System Analyst will analyze the requirements of end users. They will check whether all the requirements of end users are satisfied.

Privileges

1. End Users: They can just avail the services, modify their data, and will only be allowed to look into their own data.
2. Administrator**:** His role includes capacity planning, installation of configuration, database design, data recovery etc. These are exclusive tasks and are only to be performed by the administrator.

Assumptions

1. We assume the users of the application will be well versed with English. The application supports only English language.
2. The users of the application should have some basic knowledge of uploading images and location.
3. The code of the database should not have any compilation or syntax errors.
4. The database department of a house renting company has all the available hardware required to support the intended user load.

Business Constraints

1. Because the database will be a large system, proper maintenance and management of the system will be required.
2. If the percentage of users is exceptionally very vast, the expandability of the database maybe of concern.
3. Large-scale systems should have dedicated customers and departments responsible for system management.
4. Budget Limitation: The cost of development of the database and its maintenance should be properly managed.

NOUN ANALYSIS

TOTAL NOUNS

House	Rent	Papers
Issue	Management	Information
Parts	Issues	Project
Society	Data	Lots
First name	Growth	Reviews
City	Contract	Owner ID
Ratings	Contact	Concept
People	Age	Computing
Hand	Activity	Field
House	Landlord	Project
Owner	Property	Objective
Rent	Agent ID	System
Last name	Start Date	End Date
Time	Password	Amount
Building	Agent	Property
Money	Customer	Rent
Admin	Data	City
Management	Email	Contract ID
Landlord	Security	First name
Sum	AdminID	Contact
Date of birth	Data	Tenant
Tenant	Books	Details

Database	electricity	size
facilities	Gas	sort
History	Store	properties
Property	Tenant ID	price
Bedroom	Housing	System
City	feature	Country
Client	clients	data
Move	State	Product
House	location	User’s
Details	Occupation	registrations
Database	Furnished	properties
Property	Modes	details
Rent	changes	editing
Detail	face	details
Location	problem	parking
Bathroom	Pincode	filters
Number	Property Type	location
Rooms	Payment ID	price
Availability	search	Functions
utilities	Administrator	System
water	Price	Business

ACCEPTED NOUNS

Candidate Entity Set	Candidate Attribute	Candidate Relationship Set
Admin	First Name Last name Admin ID password Email	Agent Owner Tenant Property
Agent	First Name Last name Agent ID password Email contact Gender DOB	Tenant Admin Property
Owner	First Name Last name Owner ID password Email contact Gender DOB	Tenant Admin Property
Tenant	First Name Last name Tenant ID password Email contact Gender DOB Occupation	Agent Owner payment Admin
Property	PropertyID Rent Bedrooms Bathrooms Furnished Parking Property Type	Agent Owner Admin Reviews Contract

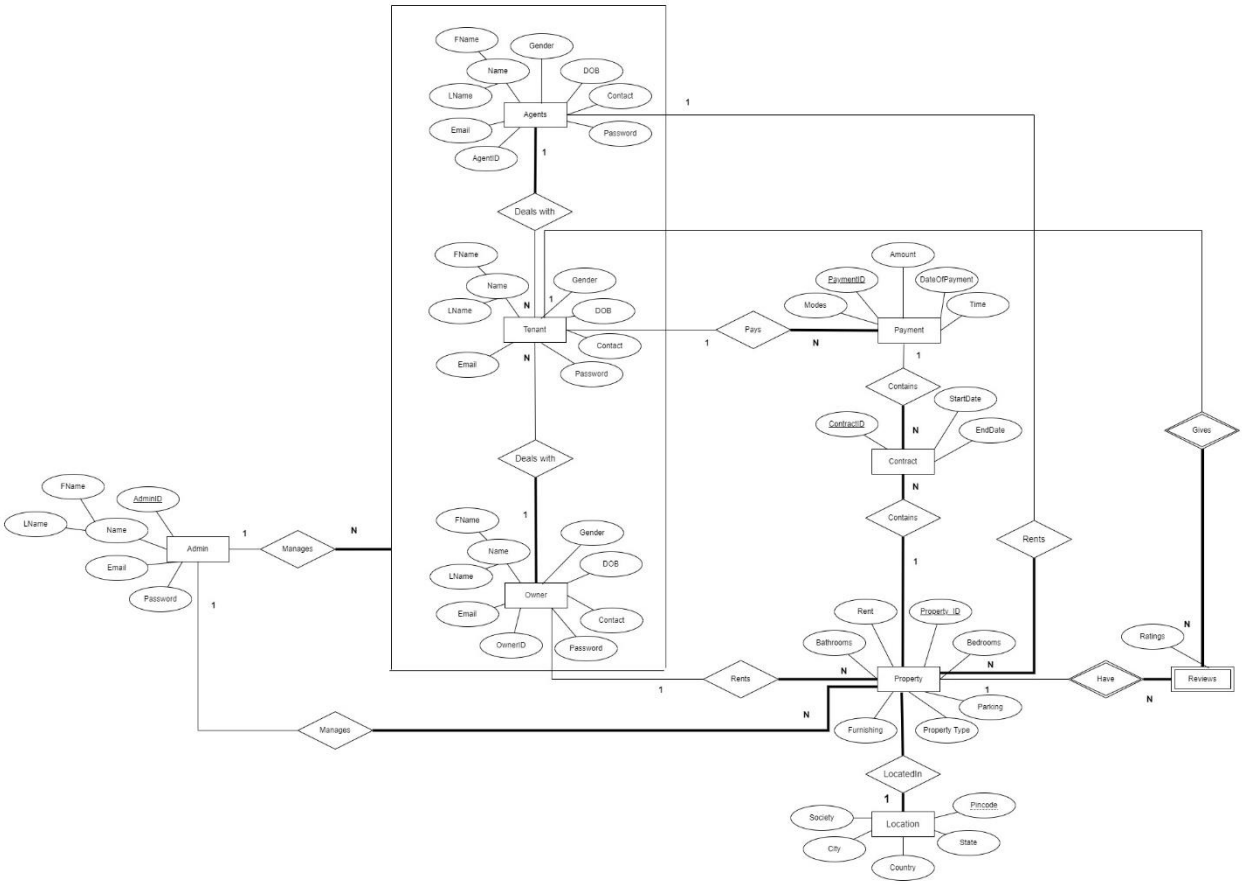
Payment	Payment id Date of payment Amount Modes Time	Tenant Contract
Contract	Contract ID Start date End date	Payment Property
Location	Pincode City State country	Property
Reviews	Ratings	Property Tenant

Rejected Nouns

Rejected Noun	Reason for Rejecting it
Issue	Duplicate
Hand	Irrelevant
Parts	Irrelevant
House	Duplicate
Management	Duplicate
Sum	Vague
Database	Duplicate
End	Vague
Move	Vague
Details	Duplicate , Vague
Area	Duplicate
Type	Duplicate
Clients	Duplicate
Feature	Duplicate , Vague
Property	Duplicate
Application	Irrelevant
Search	Duplicate
Price	Duplicate
Editing	Duplicate
Location	Duplicate
Monitoring	Vague
System	Vague
Reasons	Vague
Paper	Vague
Things	Vague

City	Duplicate
Rent	Duplicate
Search	Duplicate
Landlord	Duplicate
Tenant	Duplicate
Data	Duplicate
Way	Vague
Books	Vague
Information	Duplicate
Project	Vague
Business	Vague
Concept	Irrelevant
Computing	Irrelevant
User	Duplicate
Contact	Duplicate
Programmer	Irrelevant
Place	Duplicate
Face	Vague
Problem	Irrelevant
Web	Irrelevant
Solution	Irrelevant
Monitoring	Vague
Method	Vague
Paying	Association

ER Diagram


FName
Gender
Name
DOB
LName
Contact
Agents
Email
Password
AgentID
1
Deals with
FName
Name
Gender
LName
DOB
Tenant
Contact
Email
Password
N
Amount
PaymentID
DateOfPayment
Modes
Time
Pays
Payment
Contains
StartDate
ContractID
EndDate
Contract
Gives
Deals with
FName
AdminID
LName
Name
Admin
Email
Password
Manages
FName
Gender
Name
DOB
LName
Owner
Email
Contact
OwnerID
Password
Rents
Rent
Property_ID
Bathrooms
Bedrooms
Property
Parking
Furnishing
Property Type
Ratings
Have
Reviews
Manages
LocatedIn
Society
Pincode
Location
City
State
Country

ER to Relational Mapping

Table 1: Admin

AdminId	FName	LName	Email	Password

Table 2: Agent

AgentID	AdminID	FName	LName	Gender	DOB	Email	Password	Contact

Table 3: Owner

OwnerID	AdminID	FName	LName	Gender	DOB	Email	Password	Contact

Table 4: Tenant

Tenant ID	Admin ID	FName	LName	Occup ation	Contact	Gender	DOB	Email	Pass word	Agen tID	Own erID

Table 5: Property

Property ID	Admin Id	Agent ID	Owner ID	Rent	Bed rooms	Bath rooms	Furnished	Parking	Property Type

Table 6: Location

PropertyId	City	State	Pincode	Country

Table 7: Payment

PaymentID	TenantID	DateOfPayment	Time	Modes	Amount

Table 8: Contract

ContractId	PropertyID	PaymentID	StartDate	EndDate

Table 9: Reviews

TenantID	PropertyID	Ratings

Final List of relations with attributes and constraints

Admin

Field Name	Data type	Description	Constraints
Admin_ID	Bigint	Id of the admin	PRIMARY KEY
Fname	Character Varying	First name of the admin	NOT NULL
Lname	Character Varying	Last name of the admin	NOT NULL
Email	Character Varying	Email id of admin	UNIQUE
Password	Character Varying	Password of admin	NOT NULL

Agent

Field Name	Data type	Description	Constraints
Agent_ID	Bignt	Id of the agent	PRIMARY KEY
Admin_ID	Bigint	Id of the admin	FOREIGN KEY
Fname	Character Varying	First name of the agent	NOT NULL
Lname	Character Varying	Last name of the agent	NOT NULL
Gender	Character Varying	Gender of agent	NOT NULL
DOB	Date	Date of birth	NOT NULL
Email	Character Varying	Email id of Agent	UNIQUE
Password	Character Varying	Password	NOT NULL
Contact	Bigint	Phone number of agent	UNIQUE

Owner

Field Name	Data type	Description	Constraints
Owner_ID	Bignt	Id of the owner	PRIMARY KEY
Admin_ID	Bigint	Id of the admin	FOREIGN KEY
Fname	Character Varying	First name of the owner	NOT NULL
Lname	Character Varying	Last name of the owner	NOT NULL
Gender	Character Varying	Gender of owner	NOT NULL
DOB	Date	Date of birth	NOT NULL
Email	Character Varying	Email id of owner	UNIQUE
Password	Character Varying	Password	NOT NULL
Contact	Bigint	Phone number of owner	UNIQUE

Tenant

Field Name	Data type	Description	Constraints
Tenant_ID	Bignt	Id of the Tenant	PRIMARY KEY
Admin_ID	Bigint	Id of the admin	FOREIGN KEY
Fname	Character Varying	First name of the tenant	NOT NULL
Lname	Character Varying	Last name of the tenant	NOT NULL
Occupation	Character Varying	Occupation of tenant	NOT NULL
Contact	Bigint	Phone number of tenant	UNIQUE
Gender	Character Varying	Gender of tenant	NOT NULL
DOB	Date	Date of birth	NOT NULL
Email	Character Varying	Email id of tenant	UNIQUE
Password	Character Varying	Password	NOT NULL
AgentID	Bignt	Id of the agent	FOREIGN KEY
OwnerID	Bignt	Id of the owner	FOREIGN KEY

Property

Field Name	Data type	Description	Constraints
PropertyID	Bigint	Id of property	PRIMARY KEY
AdminID	Bigint	ID of admin	FOREIGN KEY
AgentID	Bigint	ID of agent	FOREIGN KEY
OwnerID	Bigint	ID of owner	FOREIGN KEY
Rent	Bigint	Rent of house	NOT NULL
Bedrooms	Bigint	Number of bedrooms	NOT NULL
Bathrooms	Bigint	Number of bathrooms	NOT NULL
Furnished	Boolean	Furnished or unfurnished	NOT NULL
Parking	Boolean	Parking available or unavailable	NOT NULL
PropertyType	Character varying	Type of property	NOT NULL

Location

Field Name	Data type	Description	Constraints
PropertyID	Bigint	Id of property	FOREIGN KEY
City	Character varying	City in which property is located	NOT NULL
State	Character varying	City in which property is located	NOT NULL
Pincode	Bigint	Pincode of location	NOT NULL
Country	Character varying	Country in which property is located	NOT NULL

Payment

Field Name	Data type	Description	Constraints
PaymentID	Bigint	Id of payment	PRIMARY KEY
TenantID	Bigint	Id of tenant	FOREIGN KEY
DateOfPayment	Date	Date at which payment is done	NOT NULL
Time	time	Time at which payment is done	NOT NULL
Modes	Character varying	Mode of payment	NOT NULL
Amount	Bigint	Amount paid	NOT NULL

Contract

Field Name	Data type	Description	Constraints
ContractD	Bigint	Id of contract	PRIMARY KEY
PropertyID	Bigint	Id of property	FOREIGN KEY
PaymentID	Bigint	Id of payment	FOREIGN KEY
StartDate	Date	Date at which contract begins	NOT NULL
EndDate	Date	Date at which contract ends	NOT NULL

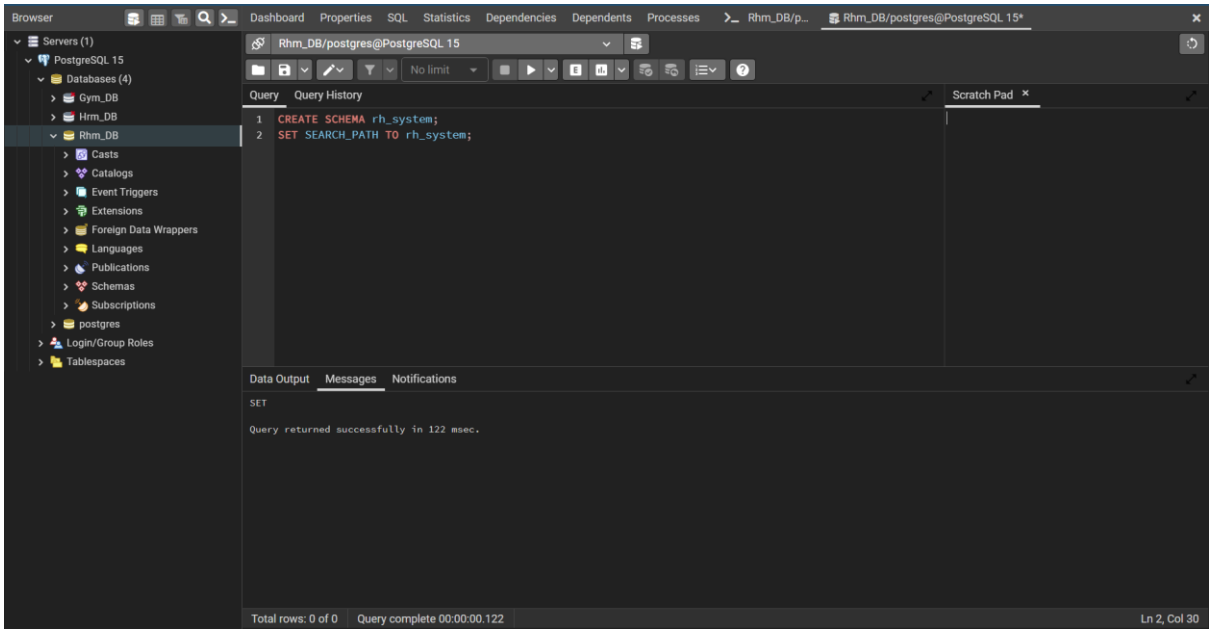
Reviews

Field Name	Data type	Description	Constraints
TenantID	Bigint	Id of tenant	FOREIGN KEY
PropertyID	Bigint	Id of property	FOREIGN KEY
Ratings	Bigint	Ratings given to properrty	-

DDL Commands for creating tables

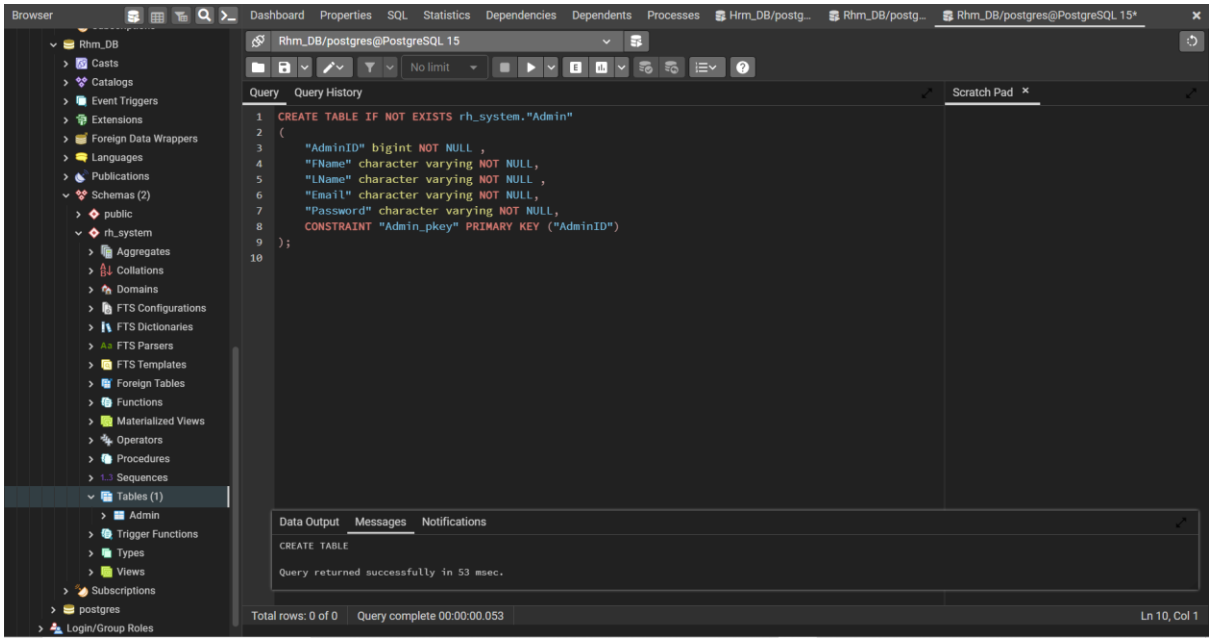
1. **To create schema**

```
CREATE SCHEMA rh_system;
SET SEARCH_PATH TO rh_system;
```



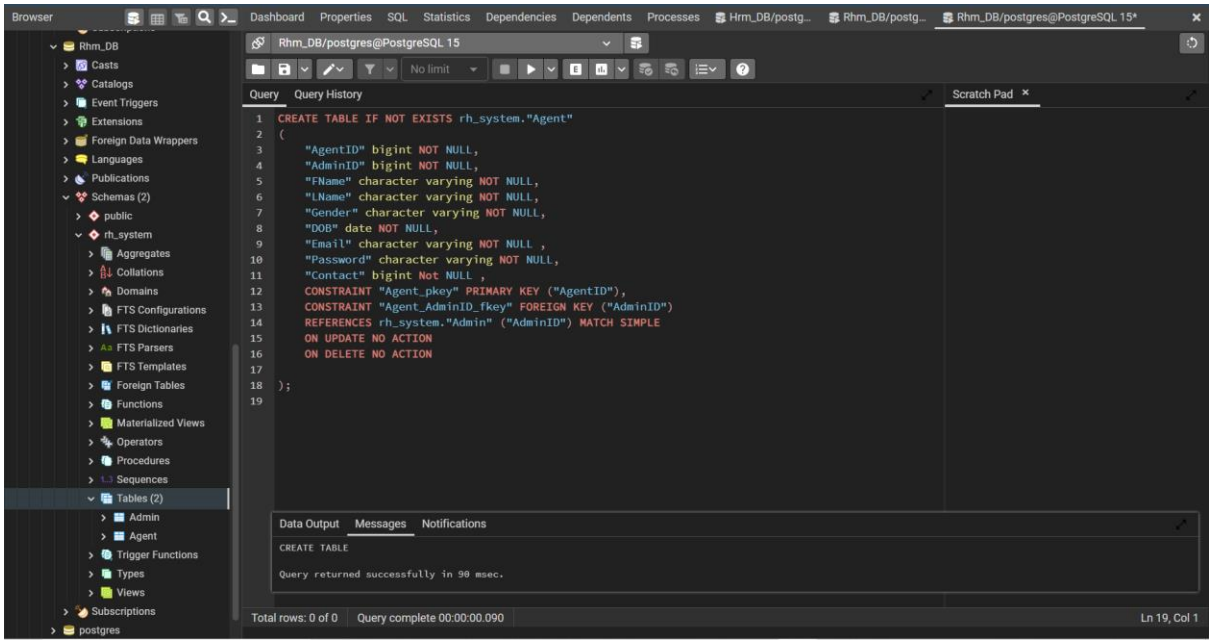
2. To create table admin

```
CREATE TABLE IF NOT EXISTS rh_system."Admin"
(
    "AdminID" bigint NOT NULL ,
    "FName" character varying NOT NULL,
    "LName" character varying NOT NULL ,
    "Email" character varying NOT NULL,
    "Password" character varying NOT NULL,
    CONSTRAINT "Admin_pkey" PRIMARY KEY ("AdminID")
);
```



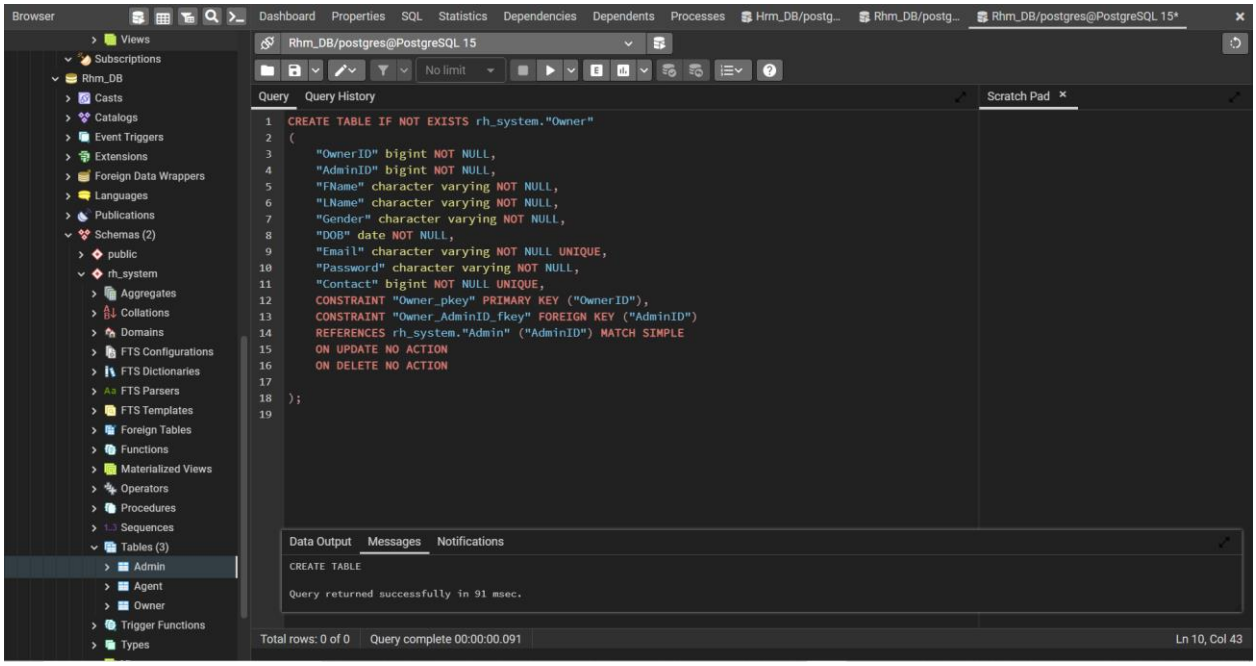
3. To create table agent

```
CREATE TABLE IF NOT EXISTS rh_system."Agent"
(
    "AgentID" bigint NOT NULL,
    "AdminID" bigint NOT NULL,
    "FName" character varying NOT NULL,
    "LName" character varying NOT NULL,
    "Gender" character varying NOT NULL,
    "DOB" date NOT NULL,
    "Email" character varying NOT NULL UNIQUE ,
    "Password" character varying NOT NULL,
    "Contact" bigint Not NULL UNIQUE ,
    CONSTRAINT "Agent_pkey" PRIMARY KEY ("AgentID"),
    CONSTRAINT "Agent_AdminID_fkey" FOREIGN KEY ("AdminID")
    REFERENCES rh_system."Admin" ("AdminID") MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION

);
```



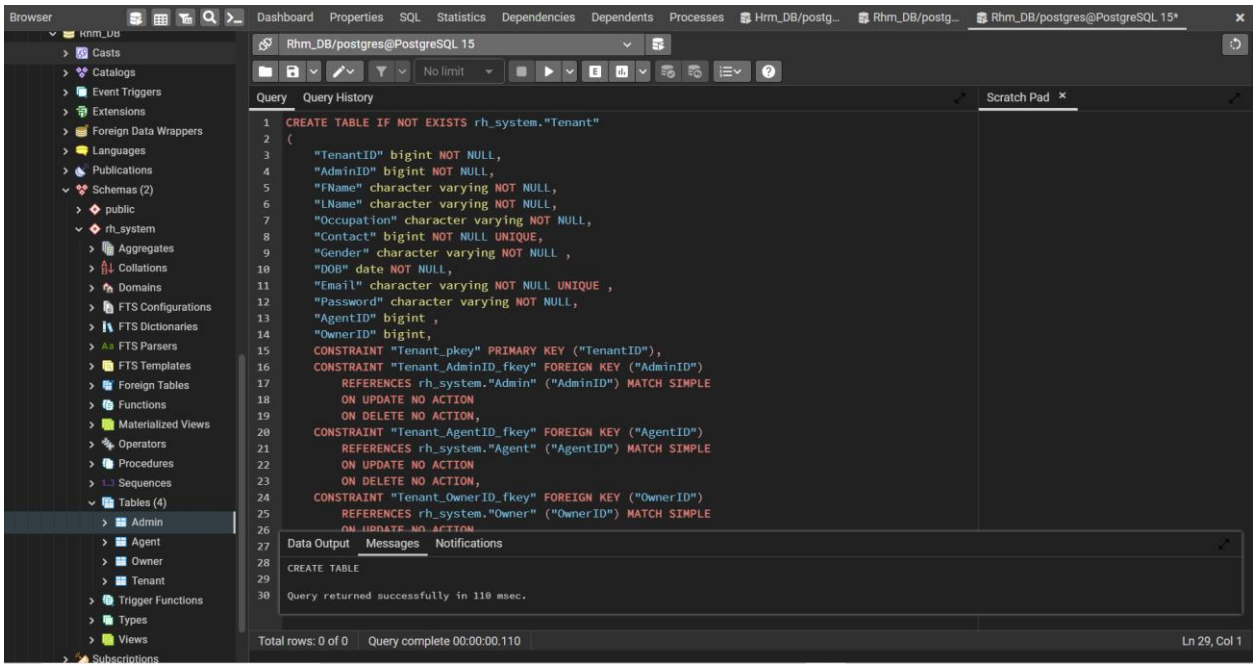
4. To create table owner

```
CREATE TABLE IF NOT EXISTS rh_system."Owner"
(
    "OwnerID" bigint NOT NULL,
    "AdminID" bigint NOT NULL,
    "FName" character varying NOT NULL,
    "LName" character varying NOT NULL,
    "Gender" character varying NOT NULL,
    "DOB" date NOT NULL,
    "Email" character varying NOT NULL UNIQUE,
    "Password" character varying NOT NULL,
    "Contact" bigint NOT NULL UNIQUE,
    CONSTRAINT "Owner_pkey" PRIMARY KEY ("OwnerID"),
     CONSTRAINT "Owner_AdminID_fkey" FOREIGN KEY ("AdminID")
    REFERENCES rh_system."Admin" ("AdminID") MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION

);
```



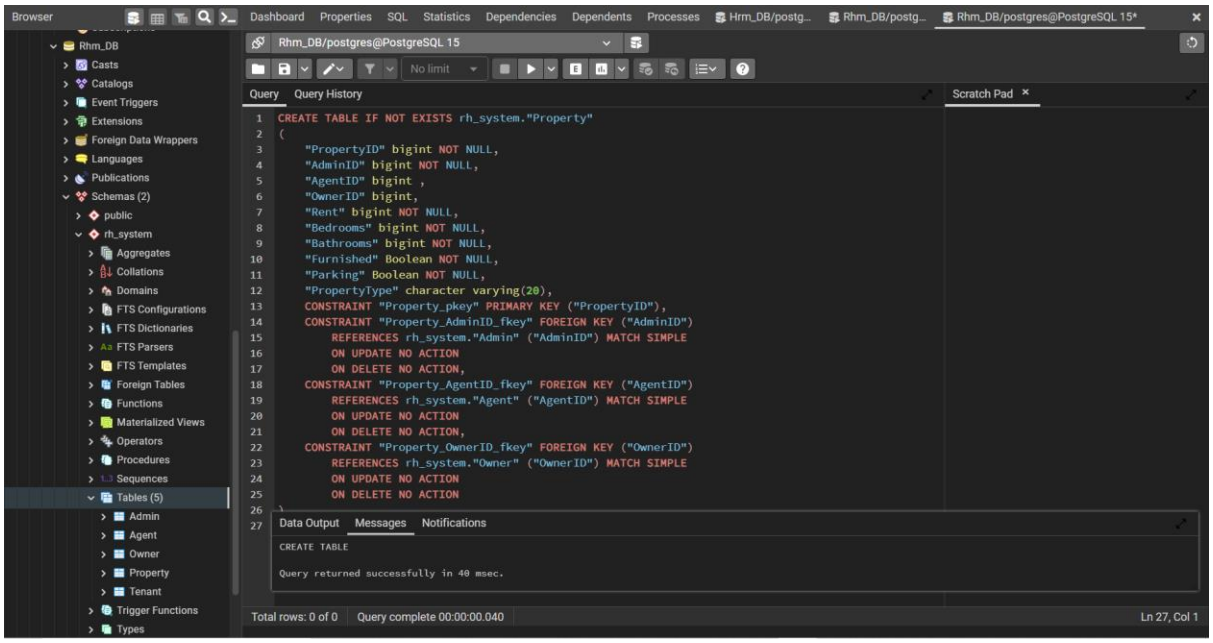
5. To create table Tenant

```
CREATE TABLE IF NOT EXISTS rh_system."Tenant"
(
    "TenantID" bigint NOT NULL,
    "AdminID" bigint NOT NULL,
    "FName" character varying NOT NULL,
    "LName" character varying NOT NULL,
    "Occupation" character varying NOT NULL,
    "Contact" bigint NOT NULL UNIQUE,
    "Gender" character varying NOT NULL ,
    "DOB" date NOT NULL,
    "Email" character varying NOT NULL UNIQUE ,
    "Password" character varying NOT NULL,
    "AgentID" bigint ,
    "OwnerID" bigint,
    CONSTRAINT "Tenant_pkey" PRIMARY KEY ("TenantID"),
    CONSTRAINT "Tenant_AdminID_fkey" FOREIGN KEY ("AdminID")
        REFERENCES rh_system."Admin" ("AdminID") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT "Tenant_AgentID_fkey" FOREIGN KEY ("AgentID")
        REFERENCES rh_system."Agent" ("AgentID") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT "Tenant_OwnerID_fkey" FOREIGN KEY ("OwnerID")
        REFERENCES rh_system."Owner" ("OwnerID") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)
```


Browser
Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
Processes
Rhm_DB/postg...
Rhm_DB/postg...
Rhm_DB/postgres@PostgreSQL 15*
Rhm_DB/postgres@PostgreSQL 15
No limit
Query
Query History
Scratch Pad
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Publications
Schemas (2)
public
rh_system
Aggregates
Collations
Domains
FTS Configurations
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Views
Operators
Procedures
Sequences
Tables (4)
Admin
Agent
Owner
Tenant
Trigger Functions
Types
Views
Subscriptions
CREATE TABLE IF NOT EXISTS rh_system."Tenant"
(
"TenantID" bigint NOT NULL,
"AdminID" bigint NOT NULL,
"FName" character varying NOT NULL,
"LName" character varying NOT NULL,
"Occupation" character varying NOT NULL,
"Contact" bigint NOT NULL UNIQUE,
"Gender" character varying NOT NULL ,
"DOB" date NOT NULL,
"Email" character varying NOT NULL UNIQUE ,
"Password" character varying NOT NULL,
"AgentID" bigint ,
"OwnerID" bigint,
CONSTRAINT "Tenant_pkey" PRIMARY KEY ("TenantID"),
CONSTRAINT "Tenant_AdminID_fkey" FOREIGN KEY ("AdminID")
REFERENCES rh_system."Admin" ("AdminID") MATCH SIMPLE
ON UPDATE NO ACTION
ON DELETE NO ACTION,
CONSTRAINT "Tenant_AgentID_fkey" FOREIGN KEY ("AgentID")
REFERENCES rh_system."Agent" ("AgentID") MATCH SIMPLE
ON UPDATE NO ACTION
ON DELETE NO ACTION,
CONSTRAINT "Tenant_OwnerID_fkey" FOREIGN KEY ("OwnerID")
REFERENCES rh_system."Owner" ("OwnerID") MATCH SIMPLE
Data Output
Messages
Notifications
CREATE TABLE
Query returned successfully in 110 msec.
Total rows: 0 of 0
Query complete 00:00:00.110
Ln 29, Col 1

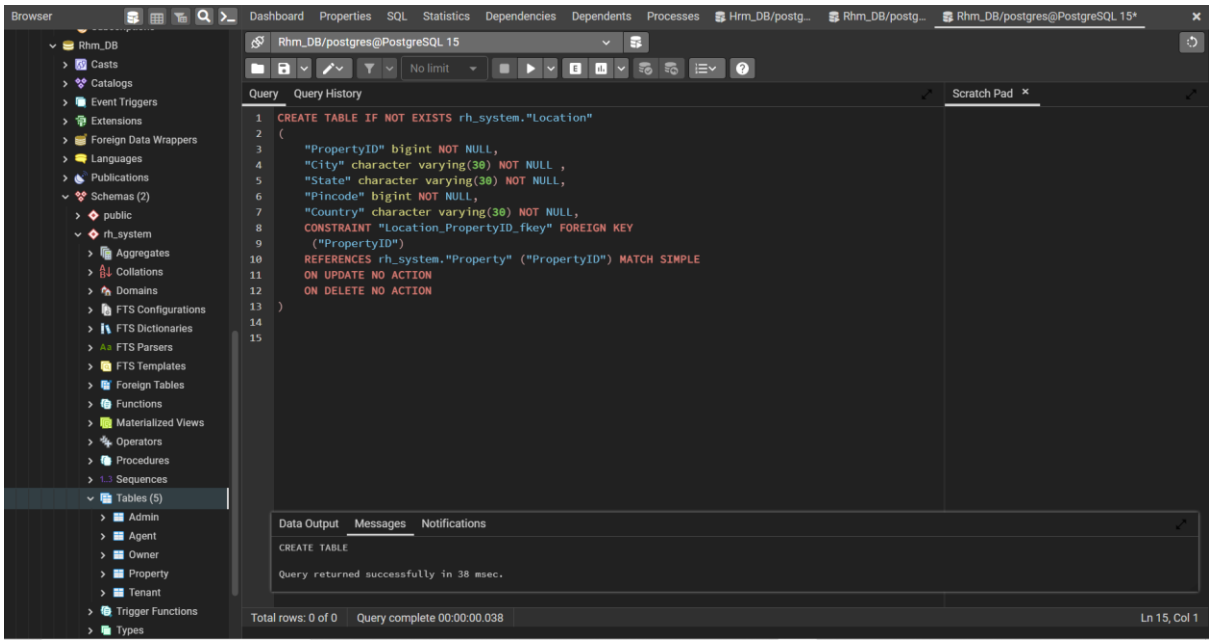
6. To create table Property

```
CREATE TABLE IF NOT EXISTS rh_system."Property"
(
    "PropertyID" bigint NOT NULL,
    "AdminID" bigint NOT NULL,
    "AgentID" bigint ,
    "OwnerID" bigint,
    "Rent" bigint NOT NULL,
    "Bedrooms" bigint NOT NULL,
    "Bathrooms" bigint NOT NULL,
    "Furnished" Boolean NOT NULL,
    "Parking" Boolean NOT NULL,
    "PropertyType" character varying(20),
    CONSTRAINT "Property_pkey" PRIMARY KEY ("PropertyID"),
    CONSTRAINT "Property_AdminID_fkey" FOREIGN KEY ("AdminID")
        REFERENCES rh_system."Admin" ("AdminID") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT "Property_AgentID_fkey" FOREIGN KEY ("AgentID")
        REFERENCES rh_system."Agent" ("AgentID") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT "Property_OwnerID_fkey" FOREIGN KEY ("OwnerID")
        REFERENCES rh_system."Owner" ("OwnerID") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)
```


Browser
Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
Processes
Rhm_DB/postg...
Rhm_DB/postg...
Rhm_DB/postgres@PostgreSQL 15*
Rhm_DB/postgres@PostgreSQL 15
No limit
Query
Query History
Scratch Pad
Rhm_DB
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Publications
Schemas (2)
public
rh_system
Aggregates
Collations
Domains
FTS Configurations
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Views
Operators
Procedures
Sequences
Tables (5)
Admin
Agent
Owner
Property
Tenant
Trigger Functions
Types
CREATE TABLE IF NOT EXISTS rh_system."Property"
(
"PropertyID" bigint NOT NULL,
"AdminID" bigint NOT NULL,
"AgentID" bigint ,
"OwnerID" bigint,
"Rent" bigint NOT NULL,
"Bedrooms" bigint NOT NULL,
"Bathrooms" bigint NOT NULL,
"Furnished" Boolean NOT NULL,
"Parking" Boolean NOT NULL,
"PropertyType" character varying(20),
CONSTRAINT "Property_pkey" PRIMARY KEY ("PropertyID"),
CONSTRAINT "Property_AdminID_fkey" FOREIGN KEY ("AdminID")
REFERENCES rh_system."Admin" ("AdminID") MATCH SIMPLE
ON UPDATE NO ACTION
ON DELETE NO ACTION,
CONSTRAINT "Property_AgentID_fkey" FOREIGN KEY ("AgentID")
REFERENCES rh_system."Agent" ("AgentID") MATCH SIMPLE
ON UPDATE NO ACTION
ON DELETE NO ACTION,
CONSTRAINT "Property_OwnerID_fkey" FOREIGN KEY ("OwnerID")
REFERENCES rh_system."Owner" ("OwnerID") MATCH SIMPLE
ON UPDATE NO ACTION
ON DELETE NO ACTION
)
Data Output
Messages
Notifications
CREATE TABLE
Query returned successfully in 40 msec.
Total rows: 0 of 0
Query complete 00:00:00.040
Ln 27, Col 1

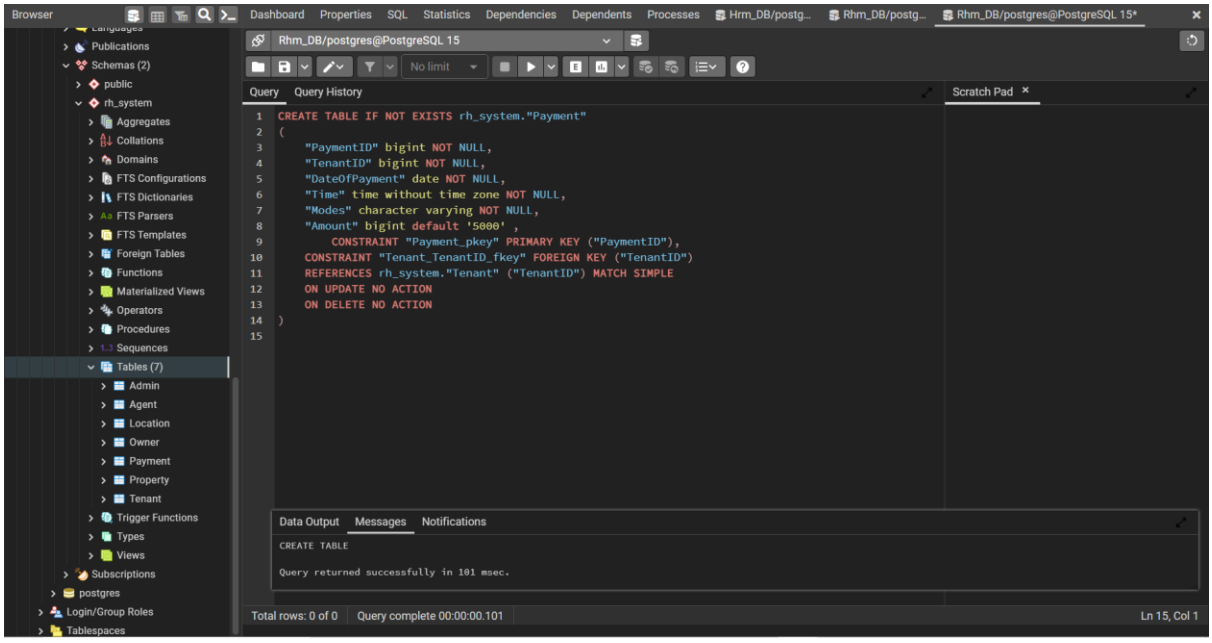
7. To create table Location

```
CREATE TABLE IF NOT EXISTS rh_system."Location"
(
    "PropertyID" bigint NOT NULL,
    "City" character varying(30) NOT NULL ,
    "State" character varying(30) NOT NULL,
    "Pincode" bigint NOT NULL,
    "Country" character varying(30) NOT NULL,
    CONSTRAINT "Location_PropertyID_fkey" FOREIGN KEY
     ("PropertyID")
    REFERENCES rh_system."Property" ("PropertyID") MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
)
```



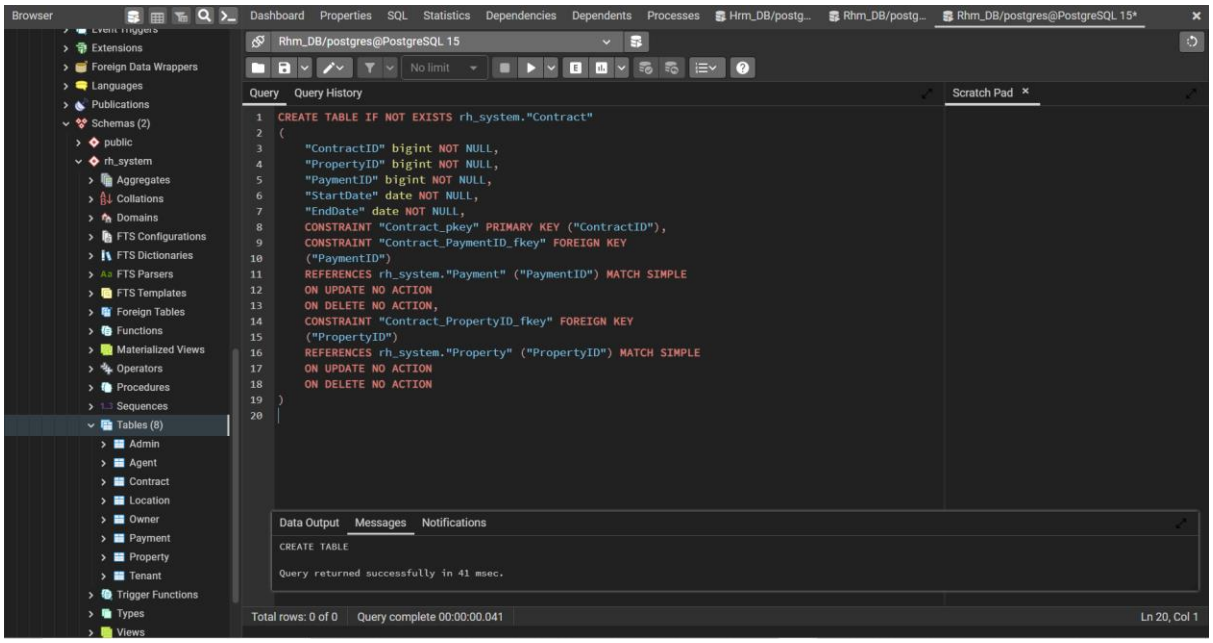
8. To create table payment

```
CREATE TABLE IF NOT EXISTS rh_system."Payment"
(
    "PaymentID" bigint NOT NULL,
    "TenantID" bigint NOT NULL,
    "DateOfPayment" date NOT NULL,
    "Time" time without time zone NOT NULL,
    "Modes" character varying NOT NULL,
    "Amount" bigint default '5000' ,
    CONSTRAINT "Payment_pkey" PRIMARY KEY ("PaymentID"),
    CONSTRAINT "Tenant_TenantID_fkey" FOREIGN KEY ("TenantID")
    REFERENCES rh_system."Tenant" ("TenantID") MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
)
```



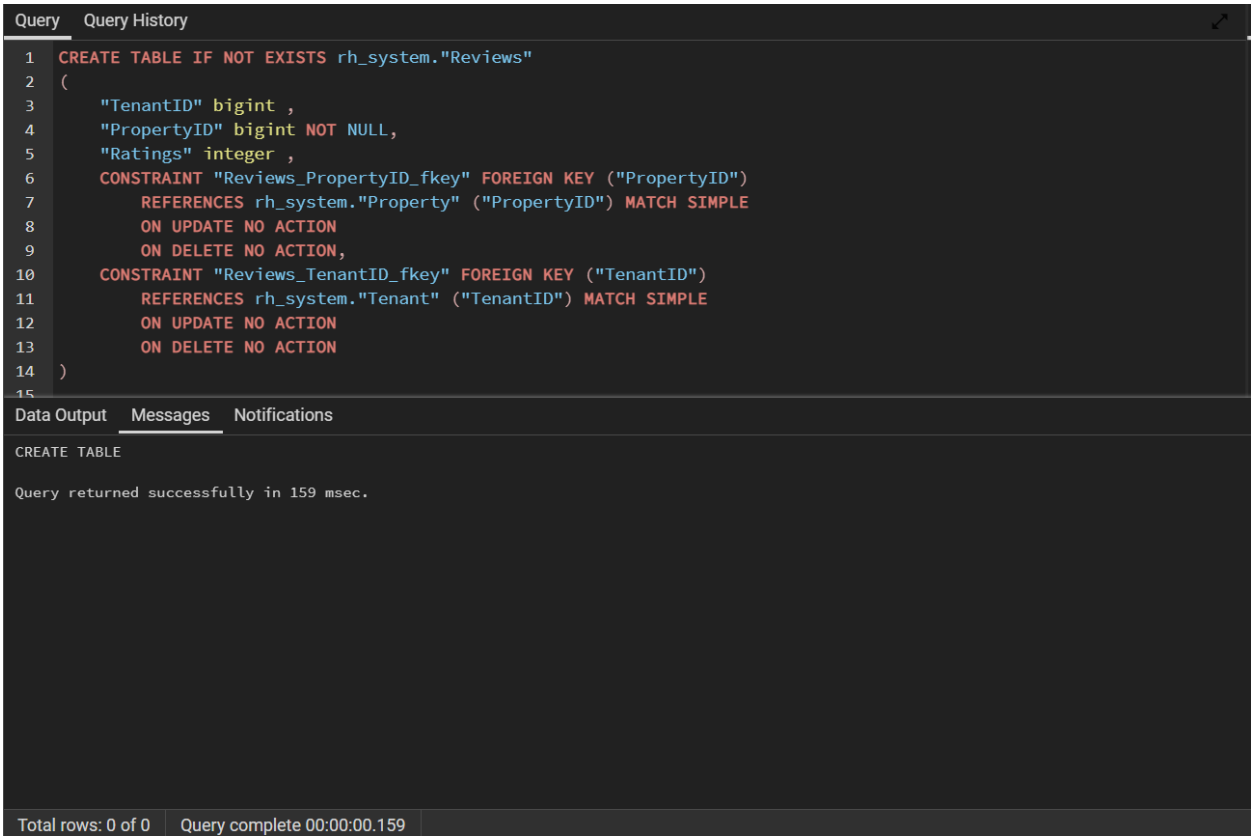
9. To create table Contract

```
CREATE TABLE IF NOT EXISTS rh_system."Contract"
(
    "ContractID" bigint NOT NULL,
    "PropertyID" bigint NOT NULL,
    "PaymentID" bigint NOT NULL,
    "StartDate" date NOT NULL,
    "EndDate" date NOT NULL,
    CONSTRAINT "Contract_pkey" PRIMARY KEY ("ContractID"),
    CONSTRAINT "Contract_PaymentID_fkey" FOREIGN KEY
    ("PaymentID")
    REFERENCES rh_system."Payment" ("PaymentID") MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION,
    CONSTRAINT "Contract_PropertyID_fkey" FOREIGN KEY
    ("PropertyID")
    REFERENCES rh_system."Property" ("PropertyID") MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
)
```



10. To create table Reviews

```
CREATE TABLE IF NOT EXISTS rh_system."Reviews"
(
    "TenantID" bigint ,
    "PropertyID" bigint NOT NULL,
    "Ratings" integer ,
    CONSTRAINT "Reviews_PropertyID_fkey" FOREIGN KEY ("PropertyID")
        REFERENCES rh_system."Property" ("PropertyID") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT "Reviews_TenantID_fkey" FOREIGN KEY ("TenantID")
        REFERENCES rh_system."Tenant" ("TenantID") MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)
```



Populating data into tables

Table 1: Admin

```
INSERT INTO rh_system."Admin"(
        "AdminID", "FName", "LName", "Email", "Password")
        VALUES (?, ?, ?, ?, ?);
```

```
SELECT "AdminID", "FName", "LName", "Email", "Password"
    FROM rh_system."Admin";
```

Data Output | Messages | Notifications

	AdminID [PK] bigint	FName character varying	LName character varying	Email character varying	Password character varying
1	1	Osborn	Pache	opache0@arizon...	P6ojyF3L
2	2	Zack	Tellenbroker	ztellenbroker1@e...	VB0pSjlgXa
3	3	Jorry	Cleatherow	jcleatherow2@ne...	ZcxJraEaSDwH
4	4	Don	Cropper	dcropper3@dion....	T5cvjTOtKJj
5	5	Fabio	Wandtke	fwandtke4@free...	jzWS2SPEVsJQ
6	6	Clywd	Menpes	cmenpes5@acqu...	wDk1vM51UaY
7	7	Barrie	Deek	bdeek6@apache....	Fuh0iKK
8	8	Ardath	Macellar	amacellar7@wiki...	CX4FnLk6
9	9	Nesta	Sings	nsings8@wp.com	LwbSHD
10	10	Raven	Leppo	rleppo9@berkele...	l85rY3YX
11	11	Janelle	Bernakiewicz	jbernakiewicza@...	GsUpJlFx9e

Total rows: 80 of 80 | Query complete 00:00:00.283

Table 2: Agent

INSERT INTO rh_system."Agent"(
"AgentID", "AdminID", "FName", "LName", "Gender", "DOB", "Email", "Password", "Contact")
VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?);

```
SELECT "AgentID", "AdminID", "FName", "LName", "Gender", "DOB", "Email", "Password", "Contact"
    FROM rh_system."Agent";
```

Data Output | Messages | Notifications

	AgentID [PK] bigint	AdminID bigint	FName character varying	LName character varying	Gender character varying	DOB date	Email character varying	Password character varying	Contact bigint
1	1	1	Art	MacSween	Male	1965-01-25	amacsween0@o...	6V6bazJ5PbJ	1043764924
2	2	2	Paule	Batting	Female	1967-03-22	pbatting1@mit.e...	KlkplVFo	2563854045
3	3	3	Gilbertine	Pilgram	Female	1988-03-27	gpilgram2@patc...	NKyr1lF7kUg	1503546790
4	4	4	Burtie	Heinzel	Male	1973-03-07	bheinzel3@twitte...	NSWE2c	1603859550
5	5	5	Ruby	Fatharly	Female	1984-07-10	rfatharly4@berke...	AbN0xO	2936329448
6	6	6	Donaugh	Boteman	Male	1981-02-24	dboteman5@yell...	LhUwve7Y3	4456381755
7	7	7	Kermie	Todman	Male	1981-08-18	ktodman6@bing....	2VbLVWN6Z6M	2456140251
8	8	8	Benedicta	Andriss	Female	1966-08-16	bandriss7@ft.com	KuidziS	9516284928
9	9	9	Alysa	Caff	Female	1974-06-05	acaff8@tinyurl.c...	G3we56CRarmD	1896324092
10	10	10	Isaiah	Crow	Male	1983-06-18	icrow9@live.com	PeMN2HivRVsk	7583044435
11	11	11	Lowell	Sherme	Male	1973-05-19	lshermea@weibo...	pG76Zjl	4861543151

Total rows: 100 of 100 Query complete 00:00:00.095

Table 3: Owner

INSERT INTO rh_system."Owner"(
"OwnerID", "AdminID", "FName", "LName", "Gender", "DOB", "Email", "Password", "Contact")
VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?);

```
SELECT "OwnerID", "AdminID", "FName", "LName", "Gender", "DOB", "Email", "Password", "Contact"
    FROM rh_system."Owner";
```

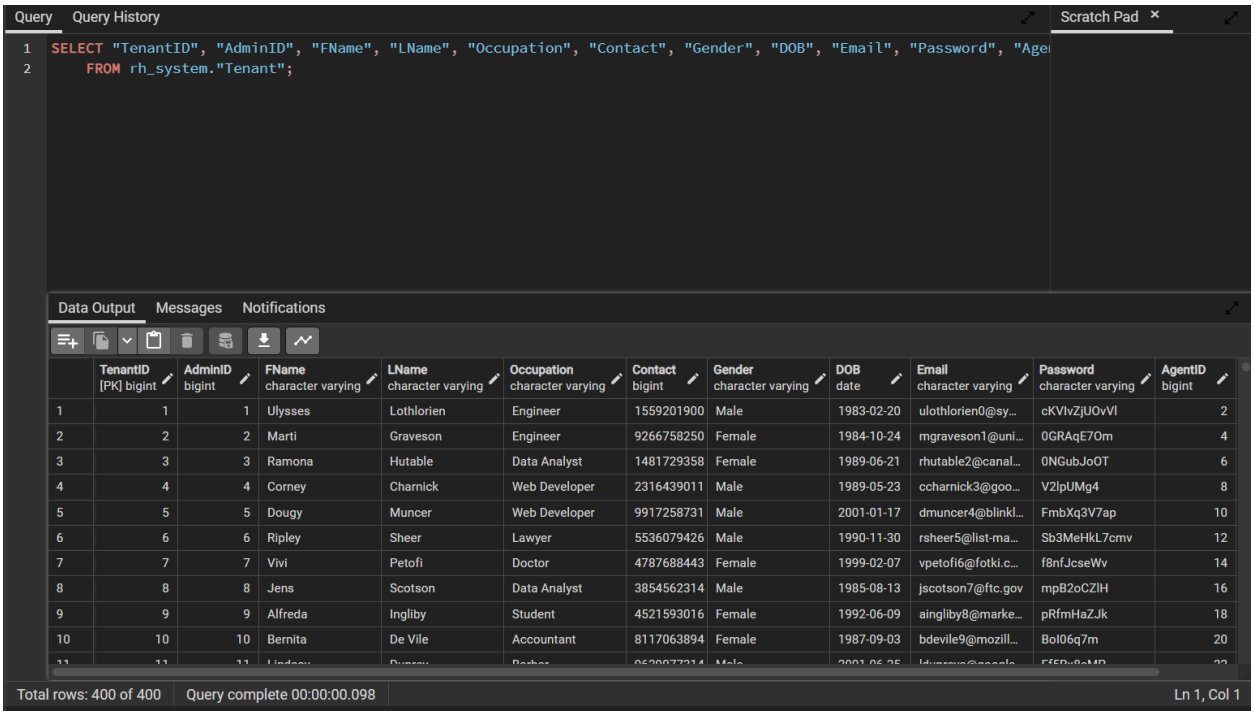
Data Output | Messages | Notifications

	OwnerID [PK] bigint	AdminID bigint	FName character varying	LName character varying	Gender character varying	DOB date	Email character varying	Password character varying	Contact bigint
1	1	19	Massimo	Scogin	Male	1965-02-07	mscogin0@infos...	uKgMx2S	8282236946
2	2	23	Garrott	Harbinson	Male	1968-07-17	gharbinson1@dis...	EQTAviyG	3914507094
3	3	12	Ichabod	Roderham	Male	1968-10-19	iroderham2@hud...	GjgXZZnX	7142483361
4	4	13	Sebastian	Lisle	Male	1983-11-23	slisle3@sun.com	zCgfj76YF	6602422067
5	5	24	Paolina	McReynolds	Female	1972-08-23	pmcreynolds4@y...	xOu8lAVi12pE	2096190233
6	6	23	Kissee	Dougal	Female	1971-10-17	kdougal5@sams...	d1flcCCEtp6l	9992716896
7	7	43	Cointon	Anlay	Male	1979-03-18	canlay6@themef...	ZshLIZ84L	6803981054
8	8	53	Karola	Andrzej	Female	1965-07-13	kandrzej7@tumbl...	h0lO3aTt5HVZ	2245997153
9	9	34	Broderic	Biggadyke	Male	1979-04-26	bbiggadyke8@ho...	XSas5ek	4969762190
10	10	12	Sigismond	Routley	Male	1966-07-11	sroutley9@acquir...	lQTKLXOxGf	4332305319
11	11	53	Huberto	Nekrews	Male	1971-06-25	hnekrewsa@net...	pv93Jzpj	8442455864

Total rows: 100 of 100 Query complete 00:00:00.127

Table 4: Tenant

```
INSERT INTO rh_system."Tenant"(
        "TenantID", "AdminID", "FName", "LName", "Occupation", "Contact", "Gender",
"DOB", "Email", "Password", "AgentID", "OwnerID")
        VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?);
```



Query | Query History | Scratch Pad

```
SELECT "TenantID", "AdminID", "FName", "LName", "Occupation", "Contact", "Gender", "DOB", "Email", "Password", "Age
    FROM rh_system."Tenant";
```

Data Output | Messages | Notifications

	TenantID [PK] bigint	AdminID bigint	FName character varying	LName character varying	Occupation character varying	Contact bigint	Gender character varying	DOB date	Email character varying	Password character varying	AgentID bigint
1	1	1	Ulysses	Lothlorien	Engineer	1559201900	Male	1983-02-20	ulothlorien0@sy...	cKVlvZjUOvVl	2
2	2	2	Marti	Graveson	Engineer	9266758250	Female	1984-10-24	mgraveson1@uni...	0GRAqE7Om	4
3	3	3	Ramona	Hutable	Data Analyst	1481729358	Female	1989-06-21	rhutable2@canal...	0NGubJoOT	6
4	4	4	Corney	Charnick	Web Developer	2316439011	Male	1989-05-23	ccharnick3@goo...	V2lpUMg4	8
5	5	5	Dougy	Muncer	Web Developer	9917258731	Male	2001-01-17	dmuncer4@blinkl...	FmbXq3V7ap	10
6	6	6	Ripley	Sheer	Lawyer	5536079426	Male	1990-11-30	rsheer5@list-ma...	Sb3MeHkL7cmv	12
7	7	7	Vivi	Petofi	Doctor	4787688443	Female	1999-02-07	vpetofi6@fotki.c...	f8nfJcseWv	14
8	8	8	Jens	Scotson	Data Analyst	3854562314	Male	1985-08-13	jscotson7@ftc.gov	mpB2oCZlH	16
9	9	9	Alfreda	Ingliby	Student	4521593016	Female	1992-06-09	aingliby8@marke...	pRfmHaZJk	18
10	10	10	Bernita	De Vile	Accountant	8117063894	Female	1987-09-03	bdevile9@mozill...	BoI06q7m	20

Total rows: 400 of 400 | Query complete 00:00:00.098 | Ln 1, Col 1

Table 5: Property

INSERT INTO rh_system."Property"(
"PropertyID", "AdminID", "AgentID", "OwnerID", "Rent", "Bedrooms", "Bathrooms", "Furnished", "Parking", "PropertyType")
VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?);

```
SELECT "PropertyID", "AdminID", "AgentID", "OwnerID", "Rent", "Bedrooms", "Bathrooms", "Furnished", "Parking", "Pro
    FROM rh_system."Property";
```

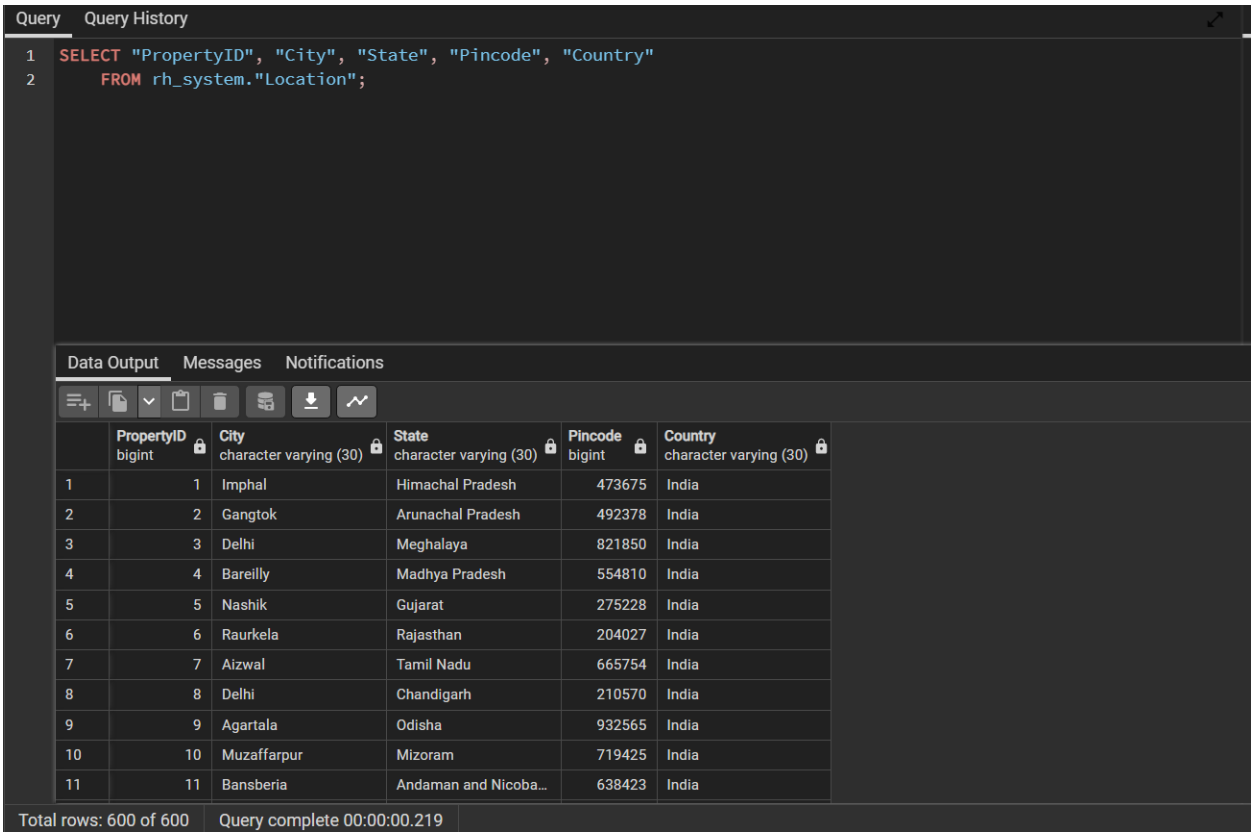
Data Output | Messages | Notifications

	PropertyID [PK] bigint	AdminID bigint	AgentID bigint	OwnerID bigint	Rent bigint	Bedrooms bigint	Bathrooms bigint	Furnished boolean	Parking boolean	PropertyType character varying (20)
1	1	61	[null]	50	14256	3	3	false	true	Flat
2	2	79	13	[null]	13031	2	3	false	false	Bungalow
3	3	21	[null]	89	20080	1	1	true	true	Villa
4	4	69	[null]	26	20927	5	3	false	true	Independent Floor
5	5	25	15	[null]	23106	5	1	true	true	Flat
6	6	66	71	[null]	25015	4	3	false	true	Flat
7	7	56	4	[null]	23858	5	1	false	false	Villa
8	8	13	42	[null]	18547	4	2	true	true	Independent Floor
9	9	74	[null]	44	9371	4	1	true	true	Flat
10	10	61	[null]	14	25506	2	1	true	true	Independent Floor
11	11	13	[null]	30	14887	3	1	true	true	Independent Floor

Total rows: 600 of 600 Query complete 00:00:00.185

Table 6: Location

```
INSERT INTO rh_system."Location"(
        "PropertyID", "City", "State", "Pincode", "Country")
        VALUES (?, ?, ?, ?, ?);
```



Query | Query History

```
SELECT "PropertyID", "City", "State", "Pincode", "Country"
    FROM rh_system."Location";
```

Data Output | Messages | Notifications

	PropertyID bigint	City character varying (30)	State character varying (30)	Pincode bigint	Country character varying (30)
1	1	Imphal	Himachal Pradesh	473675	India
2	2	Gangtok	Arunachal Pradesh	492378	India
3	3	Delhi	Meghalaya	821850	India
4	4	Bareilly	Madhya Pradesh	554810	India
5	5	Nashik	Gujarat	275228	India
6	6	Raurkela	Rajasthan	204027	India
7	7	Aizwal	Tamil Nadu	665754	India
8	8	Delhi	Chandigarh	210570	India
9	9	Agartala	Odisha	932565	India
10	10	Muzaffarpur	Mizoram	719425	India
11	11	Bansberia	Andaman and Nicoba...	638423	India

Total rows: 600 of 600 | Query complete 00:00:00.219

Table 7: Payment

```
INSERT INTO rh_system."Payment"(
        "PaymentID", "TenantID", "DateOfPayment", "Time", "Modes", "Amount")
        VALUES (?, ?, ?, ?, ?, ?);
```

Query | Query History

```
SELECT "PaymentID", "TenantID", "DateOfPayment", "Time", "Modes", "Amount"
    FROM rh_system."Payment";
```

Data Output | Messages | Notifications

	PaymentID [PK] bigint	TenantID bigint	DateOfPayment date	Time time without time zone	Modes character varying	Amount bigint
1	1	237	2022-02-25	22:16:39	UPI	23194
2	2	144	2022-01-22	22:39:06	Net Banking	27710
3	3	62	2022-02-15	20:56:24	Credit Card	28518
4	4	16	2022-01-04	04:17:11	Net banking	19794
5	5	143	2022-03-31	02:04:45	Debit card	18097
6	6	118	2022-02-10	05:25:57	Net Banking	10041
7	7	166	2022-01-21	23:55:56	Net Banking	15824
8	8	278	2022-03-15	19:53:02	UPI	22872
9	9	72	2022-01-07	00:16:07	Net banking	21988
10	10	20	2022-03-20	15:51:13	Net Banking	22420
11	11	243	2022-01-22	13:46:42	Debit card	8755

Total rows: 300 of 300 | Query complete 00:00:00.147

Table 8: Contract

INSERT INTO rh_system."Contract"(
"ContractID", "PropertyID", "PaymentID", "StartDate", "EndDate")
VALUES (?, ?, ?, ?, ?);

```
SELECT "ContractID", "PropertyID", "PaymentID", "StartDate", "EndDate"
    FROM rh_system."Contract";
```

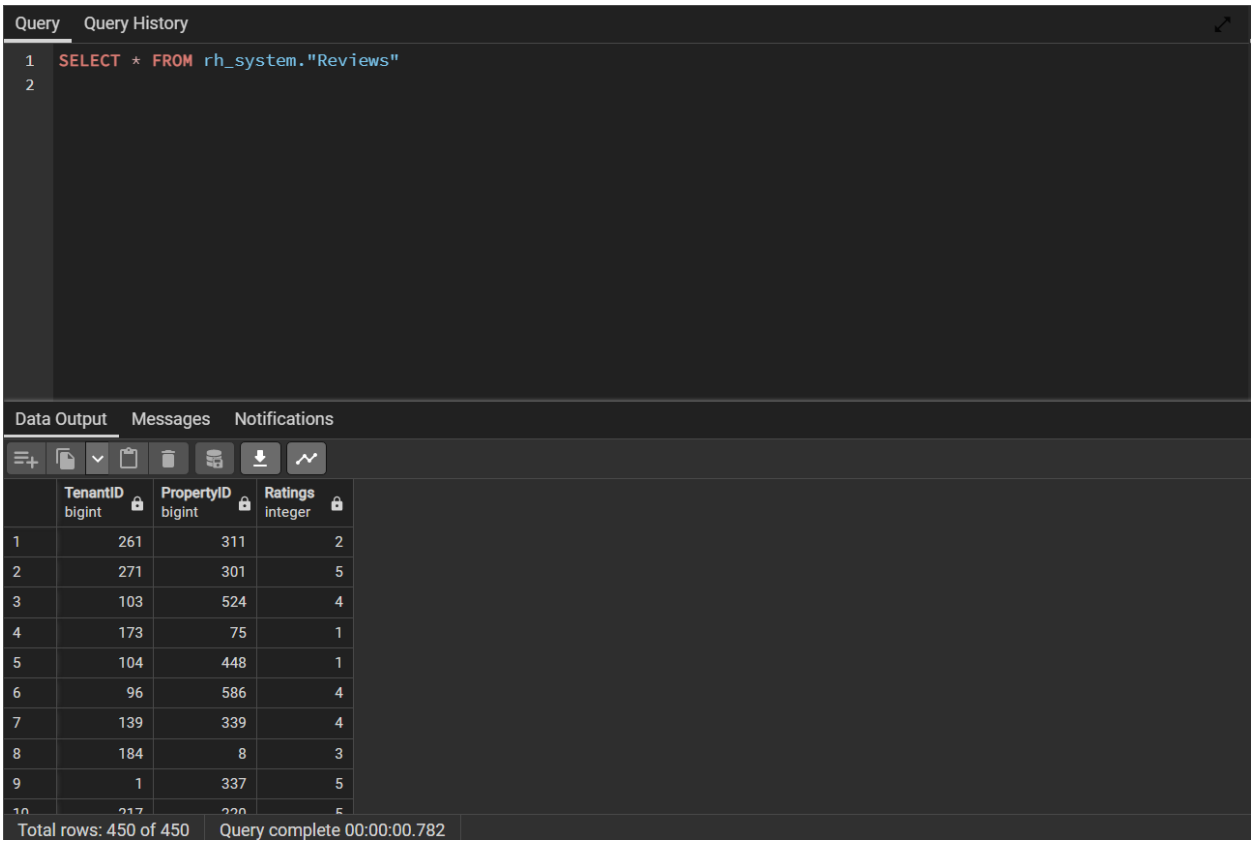
Data Output | Messages | Notifications

	ContractID [PK] bigint	PropertyID bigint	PaymentID bigint	StartDate date	EndDate date
1	1	174	258	2022-04-01	2023-10-13
2	2	128	255	2022-04-15	2023-11-09
3	3	218	118	2022-03-28	2024-03-10
4	4	177	45	2022-04-05	2023-07-21
5	5	58	160	2022-03-24	2024-01-24
6	6	274	208	2022-03-06	2023-11-21
7	7	52	208	2022-04-05	2023-08-24
8	8	260	36	2022-02-26	2023-07-13
9	9	54	153	2022-01-20	2024-02-25
10	10	80	259	2022-01-06	2023-09-15
11	11	28	70	2022-02-03	2024-01-18

Total rows: 300 of 300 Query complete 00:00:00.121

Table 9: Reviews

```
INSERT INTO rh_system."Reviews"(
        "TenantID", "PropertyID", "Ratings")
        VALUES (?, ?, ?);
```



Queries for the database

English Queries

1. Show average ratings of each property type given by customer.

2. Show first and last name of all the female agents.

3. Number of properties where rent is between 10000 and 20000.

4. Details of male tenant.

5. Dhow tenant id, first name, last name, and date of birth of tenants born after 1980.

6. Details of payments that happened between 9 months ago to 6 months ago from the current date.

7. Number of ways to pay.

8. Number of Properties in each state of India.

9. Number of properties where which are not furnished but parking is available.

10. Average rent of houses which are bungalow.

11. Print property type and its maximum rent.

12. Show names of tenants whose first name begin with ‘K’.

13. Agents whose name is 3 characters long.

14. Show PropertyID and their rent for those properties which have more than 2 bedrooms, is a flat and has parking space available.

15. Find the last name of Tenants having “ab” in them and their last name is 5 characters long.

16. Find all the last names of tenants and owners together alphabetically.

17. Show the OwnerIDs, date of birth and their age till now in ascending order of their age.

18. Show PropertyId of those properties which are in Uttar Pradesh and are furnished.

19. IDs of Agents and tenants with same first name.

20. Users who have used Net Banking as mode of Payment

21. Find Agent ids, Agents names and the number of properties they have.

22. Display Owner ID and number of properties they have where rent is greater than 15000 and where parking space is available.

23. Find second highest rent and show its respective property id.

24. Find maximum rent in each state and order the data according to state (count union territories in state).

25. Display IDs, first name and payment time of those tenants who made the payments from 9 am in morning to 9 am at night.

26. Reverse the details of furnished and parking where furnished property is not available but parking is available.

27. Display any one trainer's name who is paying the maximum rent.

28. Print ContractID, PropertyID and rent of those properties which is less than the average rent.

29. Display the Agent id who have more than 3 properties on lease.

30. Display the TenantID , First name , last name of female tenants who have started living between '2022-04-05' and '2022-04-09'.

31. Number of tenants whose property rent is greater than the average rent.

32. Display ID, first name and last name of Tenants who started living after 24 January 2022 up to 2 months.

33. Display the Owner ids and rent of properties they own where rent is between 10000 and 20000.

34. Print Id, first name and last name and number of properties of agents where age is greater than 50 years 4 months 12 days or last name ends with e.

35. Number of tenants who are paying more than the average rent

36. Display ID, first name and last name of tenant who signed the contract of more than 2 years.

37. Print ID, First and last name,and number of properties information that each admin is managing.

38. Create a view of the "Admin" table and add three rows to that view and check if there is

any change in the “Admin” table.

39. Update the view created in above question. Set Fname to ‘Hritik, and LName to ‘MacNeilley’ and then print all the data of admins having ‘MacNeilley’ as last name.

40. Find tenant ids of those tenants who are living in Villa.

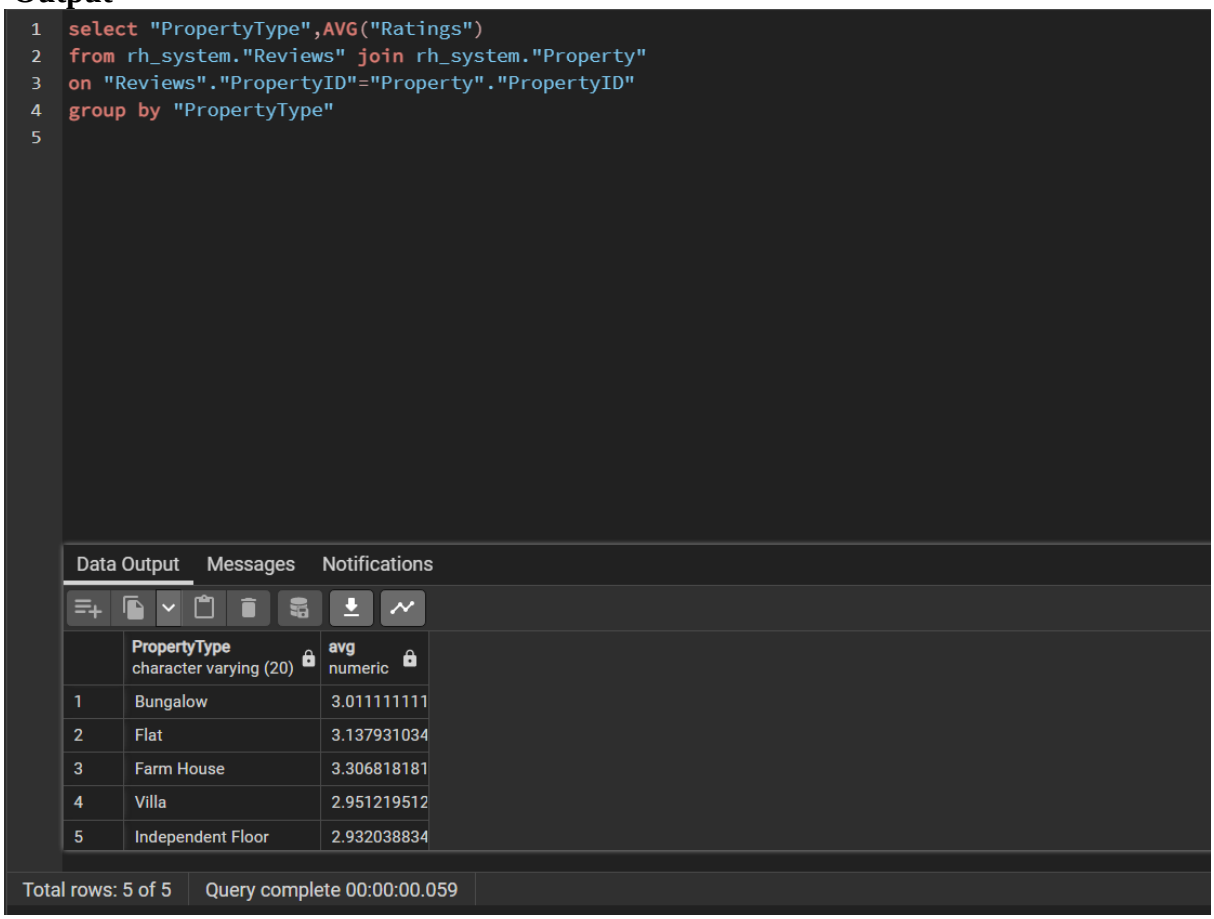
SQL Queries

1. **Show average ratings of each property type given by customer.**

Query

```
select "PropertyType",AVG("Ratings")
from rh_system."Reviews" join rh_system."Property"
on "Reviews"."PropertyID"="Property"."PropertyID"
group by "PropertyType"
```

Output



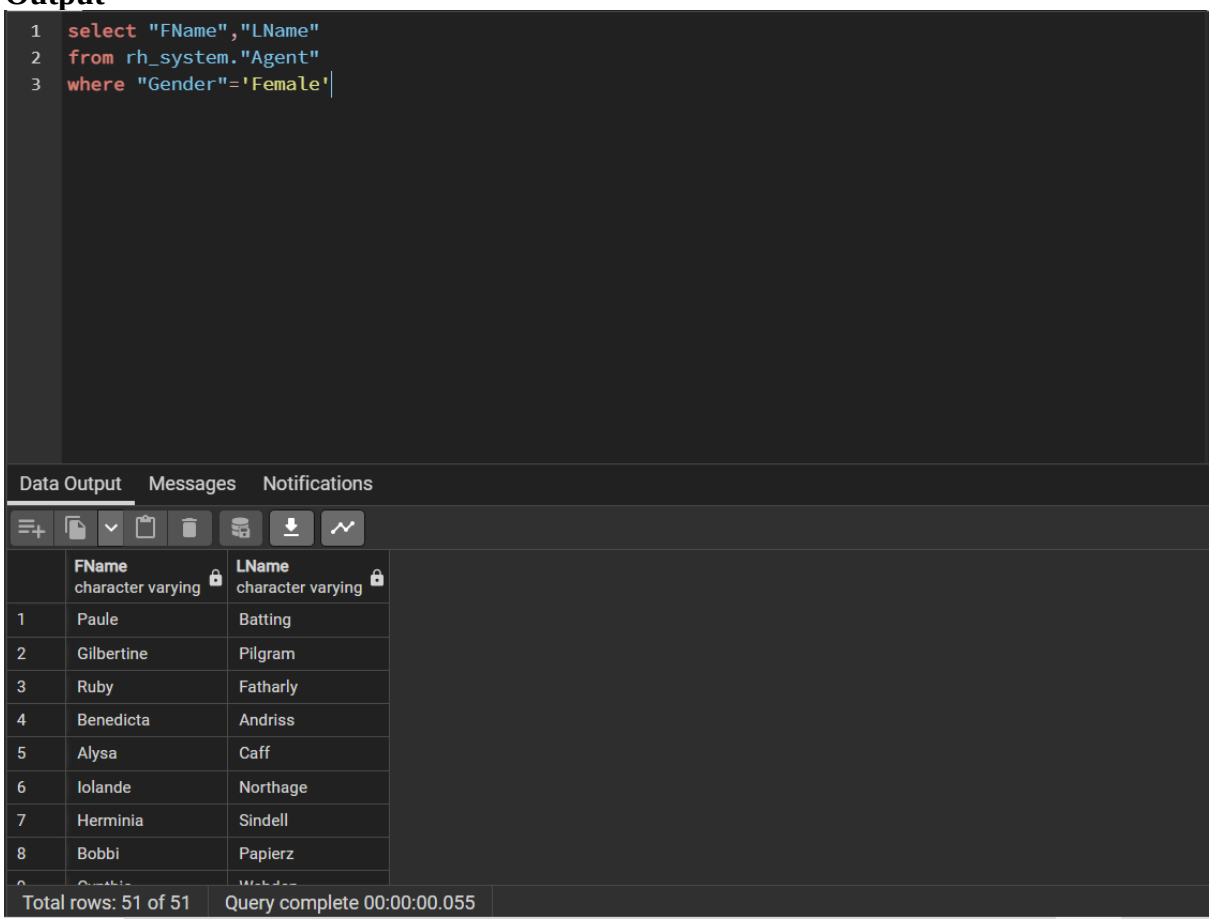
Number of tuples: 5

2. **Show first and last name of all the female agents.**

Query

```
select "FName","LName"
from rh_system."Agent"
where "Gender"='Female'
```

Output



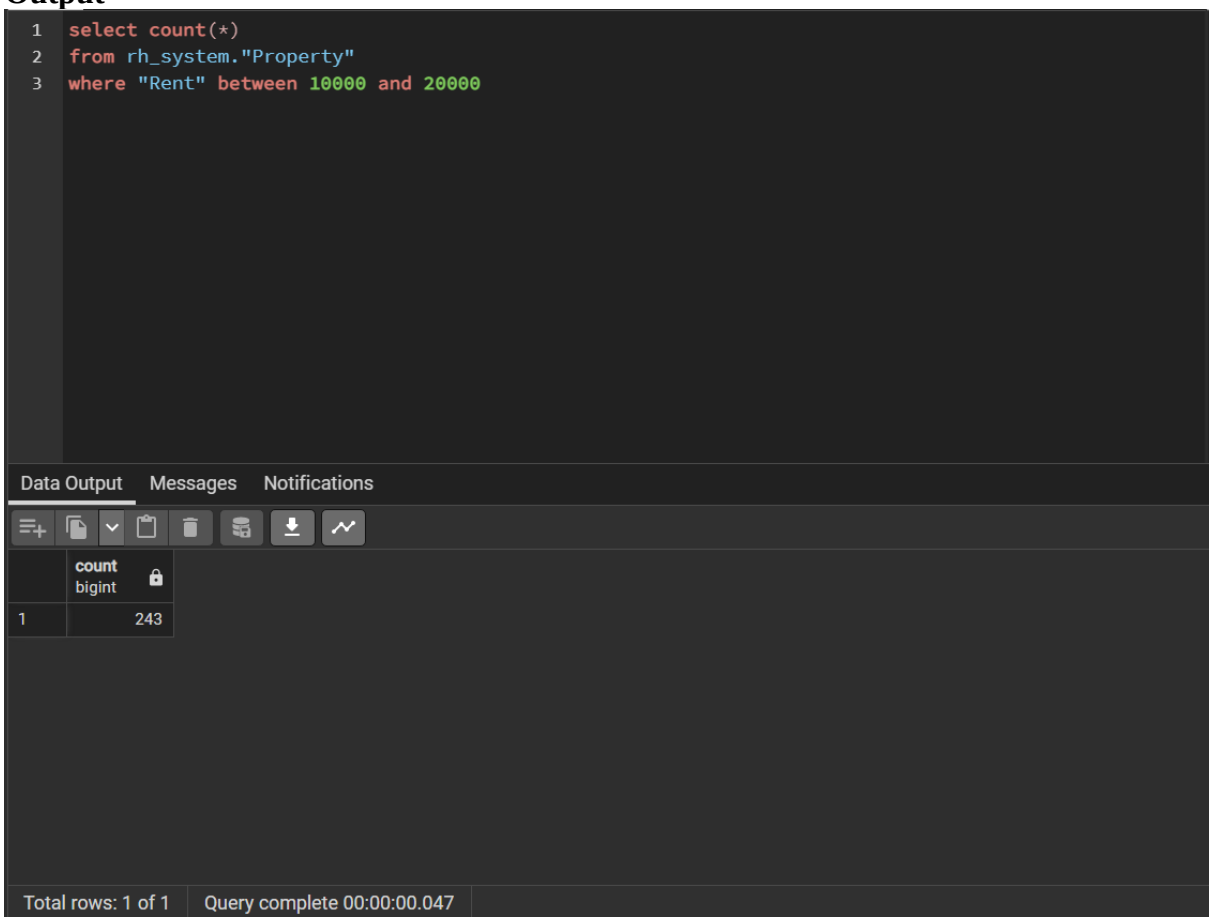
Number of tuples: 51

3. **Number of properties where rent is between 10000 and 20000.**

Query

```
select count(*)
from rh_system."Property"
where "Rent" between 10000 and 20000
```

Output



Number of tuples: 243

4. **Details of male tenant.**

Query

```
select *
from rh_system."Tenant"
where "Gender"='Male'
```

Output

```
select *
from rh_system."Tenant"
where "Gender"='Male'
```

Data Output | Messages | Notifications

	TenantID [PK] bigint	AdminID bigint	FName character varying	LName character varying	Occupation character varying	Contact bigint	Gender character varying	DOB date	Email character varying	Password character varying
1	1	2	Ulysses	Lothlorien	Engineer	1559201900	Male	1983-02-20	ulothlorien0@sy...	cKVIvZjUOvVl
2	4	54	Corney	Charnick	Web Developer	2316439011	Male	1989-05-23	ccharnick3@goo...	V2lpUMg4
3	5	21	Dougy	Muncer	Web Developer	9917258731	Male	2001-01-17	dmuncer4@blinkl...	FmbXq3V7ap
4	6	32	Ripley	Sheer	Lawyer	5536079426	Male	1990-11-30	rsheer5@list-ma...	Sb3MeHkL7cmv
5	8	11	Jens	Scotson	Data Analyst	3854562314	Male	1985-08-13	jscotson7@ftc.gov	mpB2oCZlH
6	11	70	Lindsay	Duprey	Barber	9629977314	Male	2001-06-25	ldupreya@google...	Ef5Rx8eMP
7	13	34	Cornell	Kimpton	Barber	6519118484	Male	1984-09-11	ckimptonc@unic...	UY905gpy7
8	14	25	Jareb	O' Donohue	Web Developer	9477706643	Male	1998-12-20	jodonohued@bus...	vBnWrERKV

Total rows: 161 of 161 Query complete 00:00:00.056

Number of tuples: 161

5. **Dhow tenant id, first name , last name and date of birth of tenants born after 1980.**

Query

```
select "TenantID","FName","LName","DOB"
from rh_system."Tenant"
where "DOB" > '12/31/1980'
```

Output

```
1  select "TenantID","FName","LName","DOB"
2  from rh_system."Tenant"
3  where "DOB" > '12/31/1980'
```

Data Output | Messages | Notifications

	TenantID [PK] bigint	FName character varying	LName character varying	DOB date
1	1	Ulysses	Lothlorien	1983-02-20
2	2	Marti	Graveson	1984-10-24
3	3	Ramona	Hutable	1989-06-21
4	4	Corney	Charnick	1989-05-23
5	5	Dougy	Muncer	2001-01-17
6	6	Ripley	Sheer	1990-11-30
7	7	Vivi	Petofi	1999-02-07
8	8	Jens	Scotson	1985-08-13

Total rows: 288 of 288 | Query complete 00:00:00.048

Number of tuples: 288

6. **Details of payments that happened between 9 months ago to 6 months ago from the current date.**

Query

```
select *
from rh_system."Payment"
where "DateOfPayment" between
(CURRENT_DATE- Interval '9
month') and (CURRENT_DATE- Interval '6 month')
```

Output

```
select *
from rh_system."Payment"
where "DateOfPayment" between
(CURRENT_DATE- Interval '9
month') and (CURRENT_DATE- Interval '6 month')
```

Data Output | Messages | Notifications

	PaymentID [PK] bigint	TenantID bigint	DateOfPayment date	Time time without time zone	Modes character varying
1	1	237	2022-02-25	22:16:39	UPI
2	5	143	2022-03-31	02:04:45	Debit card
3	8	278	2022-03-15	19:53:02	UPI
4	10	20	2022-03-20	15:51:13	Net Banking
5	12	150	2022-03-16	01:28:54	Credit Card
6	16	146	2022-03-07	19:09:32	UPI
7	19	281	2022-03-09	16:40:06	UPI
8	22	31	2022-03-25	14:23:25	Debit card

Total rows: 108 of 108 | Query complete 00:00:00.065

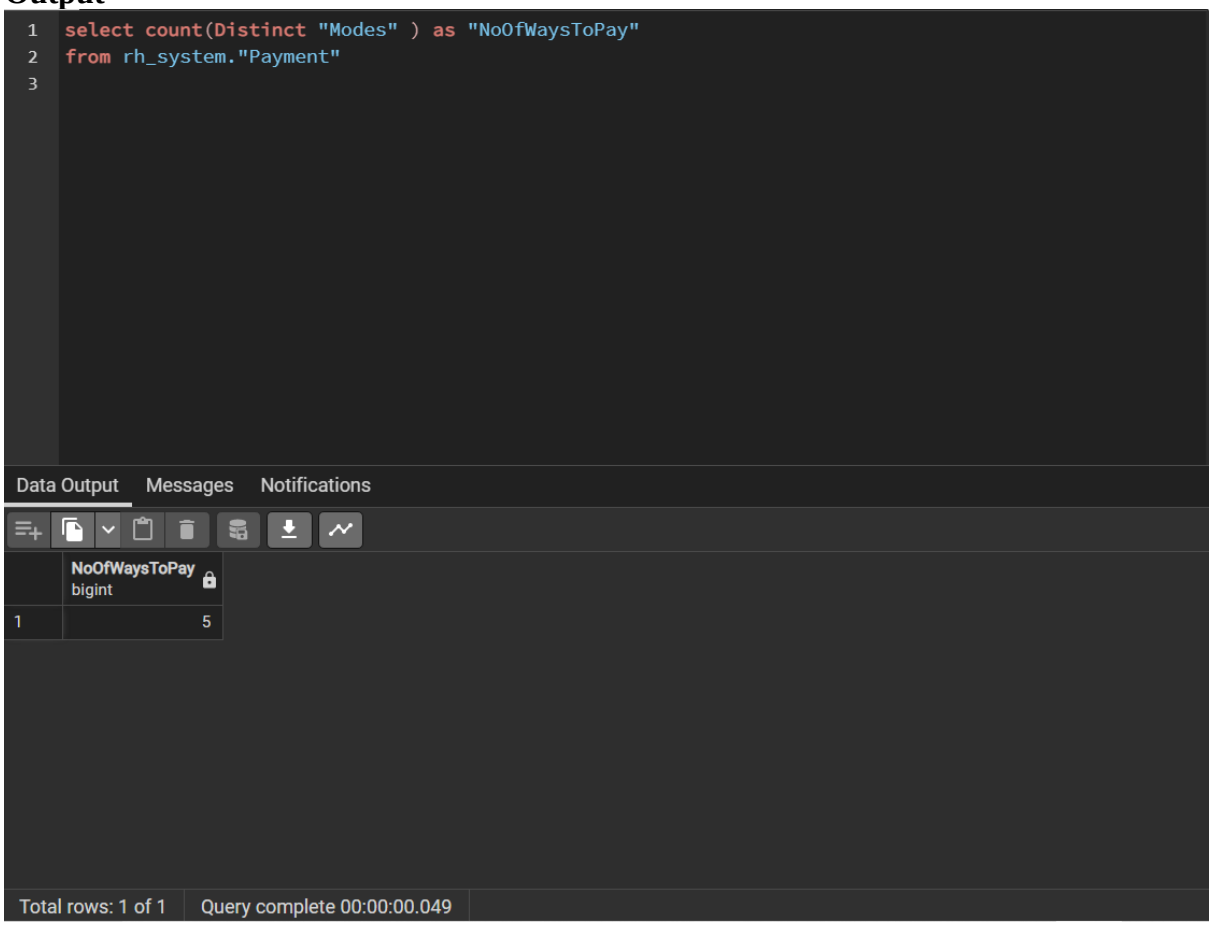
Number of tuples: 108

7. **Number of ways to pay.**

Query

```
select count(Distinct "Modes" ) as "NoOfWaysToPay"
from rh_system."Payment"
```

Output



Number of tuples: 1

8. **Number of Properties in each state of India (consider union territories in state).**

Query

```
select "State",count(*) as "NoOfProp"
from rh_system."Location"
group by "State"
order by "State"
```

Output

```
1  select "State",count(*) as "NoOfProp"
2  from rh_system."Location"
3  group by "State"
4  order by "State"
5
```

Data Output | Messages | Notifications

	State character varying (30)	NoOfProp bigint
1	Andaman and Nicobar Islan...	12
2	Andhra Pradesh	14
3	Arunachal Pradesh	12
4	Assam	20
5	Bihar	12
6	Chandigarh	21
7	Chhattisgarh	10
8	Dadra and Nagar Haveli	15

Total rows: 35 of 35 | Query complete 00:00:00.051

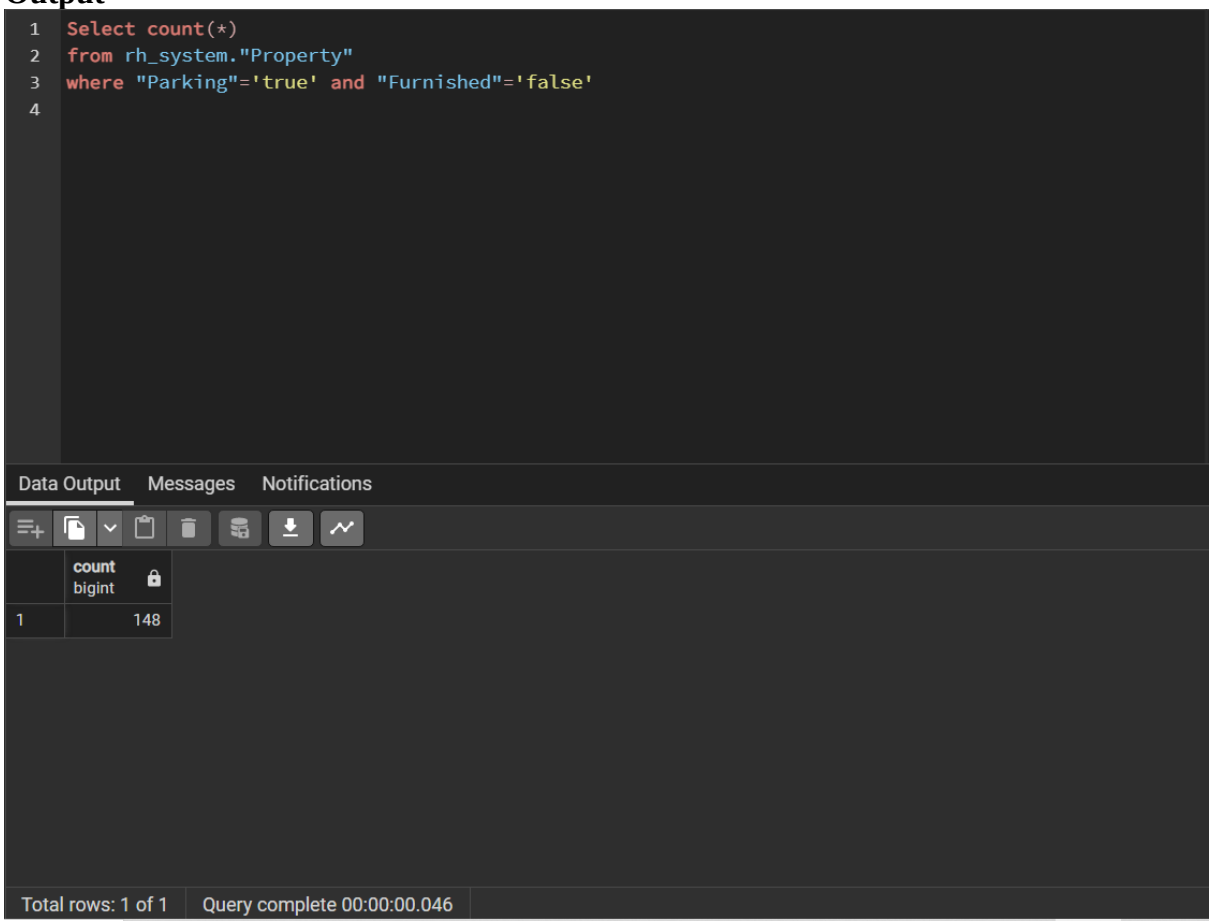
Number of tuples: 35

9. Number of properties where which are not furnished but parking is available.

Query

```
Select count(*)
from rh_system."Property"
where "Parking"='true' and "Furnished"='false'
```

Output



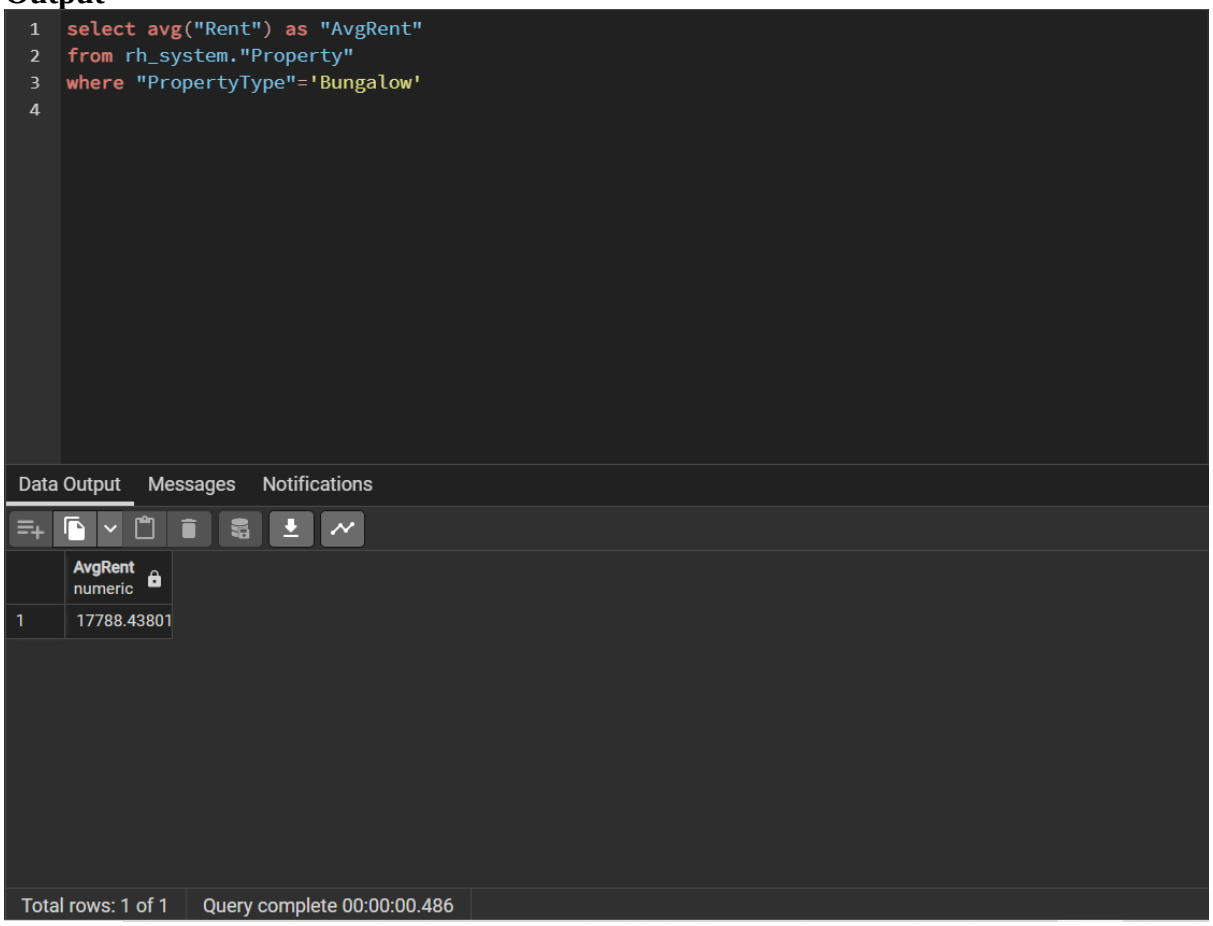
Number of tuples: 1

10. Average rent of houses which are bungalow.

Query

```
select avg("Rent") as "AvgRent"
from rh_system."Property"
where "PropertyType"='Bungalow'
```

Output



Number of tuples: 1

11. Print property type and its maximum rent.

Query

```
select Distinct "PropertyType",Max("Rent")
from rh_system."Property"
group by "PropertyType"
```

Output

```
1  select Distinct "PropertyType",Max("Rent")
2  from rh_system."Property"
3  group by "PropertyType"
```

Data Output | Messages | Notifications

	PropertyType character varying (20)	max bigint
1	Bungalow	29790
2	Farm House	29970
3	Flat	29506
4	Independent Floor	29708
5	Villa	29851

Total rows: 5 of 5 | Query complete 00:00:00.139

Number of tuples: 5

12. Show names of tenants whose first name begin with ‘K’.

Query

```
select "FName","LName"
from rh_system."Tenant"
where "Tenant"."FName" like 'K%'
```

Output

```
1  select "FName","LName"
2  from rh_system."Tenant"
3  where "Tenant"."FName" like 'K%'
```

Data Output | Messages | Notifications

	FName character varying	LName character varying
1	Kristel	Laughlin
2	Kenna	Grugerr
3	Karlyn	Heinecke
4	Karel	Kagan
5	Karim	O'Doherty
6	Katharyn	Leavold
7	Kimberly	Schwand
8	Kort	De Zuani

Total rows: 10 of 10 | Query complete 00:00:00.095

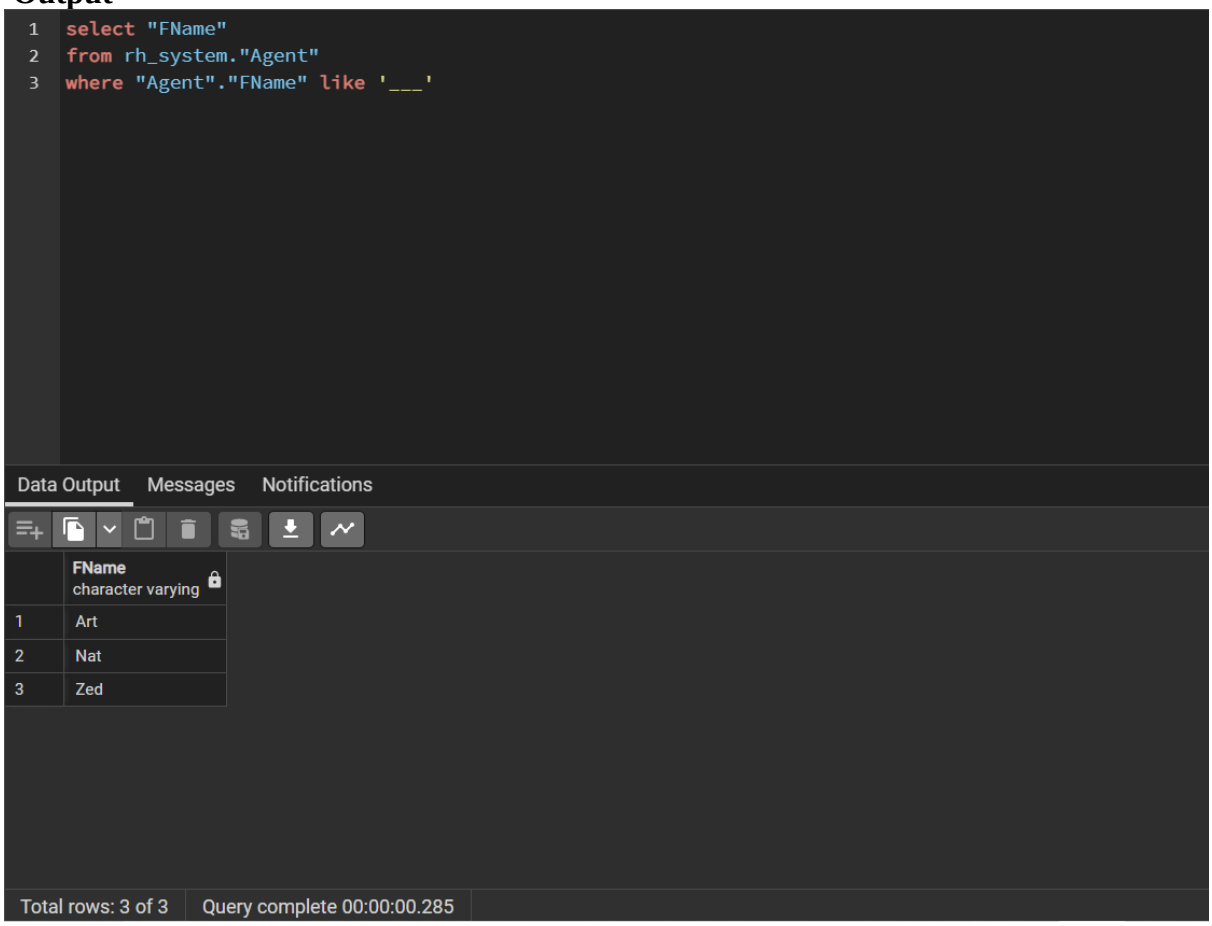
Number of tuples: 10

13. **Agents whose name is 3 characters long.**

Query

```
select "FName"
from rh_system."Agent"
where "Agent"."FName" like '___'
```

Output



Number of tuples: 3

14. **Show PropertyID and their rent for those properties which have more than 2 bedrooms, is a flat and has parking space available.**

Query

```
select "Property"."PropertyID","Property"."Rent"
from rh_system."Property"
where "Bedrooms" >2 and "Parking"='true' and "PropertyType"='Flat'
```

Output

```
1  select "Property"."PropertyID","Property"."Rent"
2  from rh_system."Property"
3  where "Bedrooms" >2 and "Parking"='true' and "PropertyType"='Flat'
```

Data Output | Messages | Notifications

	PropertyID [PK] bigint	Rent bigint
1	1	14256
2	5	23106
3	6	25015
4	9	9371
5	13	26113
6	37	14153
7	43	8758
8	65	28602

Total rows: 42 of 42 | Query complete 00:00:00.175

Number of tuples: 42

15. Find the last name of Tenants having “ab” in them and their last name is 5 characters long .

Query

```
select "FName", "LName"
from rh_system."Tenant"
where "FName" like '%ab%'
    and "LName" like '_____'
```

Output

```
select "FName", "LName"
from rh_system."Tenant"
where "FName" like '%ab%'
      and "LName" like '_____'
```

Data Output | Messages | Notifications

	FName character varying	LName character varying
1	Ichabod	Crann
2	Ichabod	Buggs
3	Gabriello	Helis

Total rows: 3 of 3 | Query complete 00:00:00.470

Number of tuples: 3

16. Find all the last names of tenants and owners together alphabetically.

Query

```
SELECT  "LName"
from rh_system."Tenant"
UNION
SELECT  "LName"
from rh_system."Owner"
ORDER BY "LName"
```

Output

```
SELECT  "LName"
from rh_system."Tenant"
UNION
SELECT  "LName"
from rh_system."Owner"
ORDER BY "LName"
```

Data Output | Messages | Notifications

	LName character varying
1	Agass
2	Alder
3	Alelsandrovich
4	Allenson
5	Allibon
6	Andreini
7	Andrzej
8	Anlay

Total rows: 399 of 399 Query complete 00:00:00.046

Number of tuples: 399

17. Show the OwnerIDs,date of birth and their age till now in ascending order of their age.

Query

```
SELECT "OwnerID","DOB", AGE(current_date,"DOB")
from rh_system."Owner"
order by AGE(current_date,"DOB")
```

Output

```
1  SELECT "OwnerID","DOB", AGE(current_date,"DOB")
2  from rh_system."Owner"
3  order by AGE(current_date,"DOB")
```

Data Output | Messages | Notifications

	OwnerID [PK] bigint	DOB date	age interval
1	13	1989-04-18	33 years 7 mons 7 days
2	44	1989-02-15	33 years 9 mons 10 days
3	34	1988-11-28	33 years 11 mons 27 da...
4	22	1988-06-17	34 years 5 mons 8 days
5	38	1988-04-18	34 years 7 mons 7 days
6	76	1988-02-26	34 years 8 mons 28 days
7	90	1988-02-18	34 years 9 mons 7 days
8	20	1987-08-30	35 years 2 mons 26 days

Total rows: 100 of 100 Query complete 00:00:00.147

Number of tuples: 100

18. Show PropertyId of those properties which are in Uttar Pradesh and are furnished.

Query

```
select "Location"."PropertyID"
from rh_system."Location"  join rh_system."Property"
on "Location"."PropertyID"="Property"."PropertyID"
where "State"='Uttar Pradesh' and "Furnished"='true'
```

Output

```
1  select "Location"."PropertyID"
2  from rh_system."Location"  join rh_system."Property"
3  on "Location"."PropertyID"="Property"."PropertyID"
4  where "State"='Uttar Pradesh' and "Furnished"='true'
```

Data Output | Messages | Notifications

	PropertyID bigint
1	23
2	59
3	160
4	169
5	244
6	335
7	501
8	503

Total rows: 9 of 9 | Query complete 00:00:00.127

Number of tuples: 9

19. IDs of Agents and tenants with same first name .

Query

```
select "AgentID","TenantID"
from rh_system."Agent" cross join rh_system."Tenant"
where "Agent"."FName"="Tenant"."FName"
```

Output

```
1  select "AgentID","TenantID"
2  from rh_system."Agent" cross join rh_system."Tenant"
3  where "Agent"."FName"="Tenant"."FName"
```

Data Output | Messages | Notifications

	AgentID bigint	TenantID bigint
1	71	7
2	36	96
3	87	124
4	71	174
5	68	194
6	18	295
7	57	299

Total rows: 7 of 7 | Query complete 00:00:00.135

Number of tuples: 7

20. Users who have used Net Banking as mode of Payment.

Query

```
Select "FName","LName"
from rh_system."Tenant" join rh_system."Payment"
on "Tenant"."TenantID"="Payment"."TenantID"
where "Payment"."Modes"='Net Banking'
```

Output

```
Select "FName","LName"
from rh_system."Tenant" join rh_system."Payment"
on "Tenant"."TenantID"="Payment"."TenantID"
where "Payment"."Modes"='Net Banking'
```

Data Output | Messages | Notifications

	FName character varying	LName character varying
1	Dougy	Muncer
2	Jens	Scotson
3	Pepi	Moore
4	Franky	Defraine
5	Conroy	Aubrun
6	Jehu	Bennett
7	Parry	Riley
8	Jo	McTrusty

Total rows: 70 of 70 | Query complete 00:00:00.048

Number of tuples: 70

21. Find Agent ids ,Agents names and the number of properties they have.

Query

```
select "Agent"."AgentID","FName","LName",count(*) as "NoOfprop"
from rh_system."Agent" join rh_system."Property"
on "Agent"."AgentID"="Property"."AgentID"
group by "Agent"."AgentID"
order by count(*)
```

Output

```
select "Agent"."AgentID","FName","LName",count(*) as "NoOfprop"
from rh_system."Agent" join rh_system."Property"
on "Agent"."AgentID"="Property"."AgentID"
group by "Agent"."AgentID"
order by count(*)
```

Data Output | Messages | Notifications

	AgentID [PK] bigint	FName character varying	LName character varying	NoOfprop bigint
1	99	Walsh	Grimolbie	1
2	80	Brnaba	Luckings	1
3	76	Weidar	Maylor	1
4	95	Atlanta	Beetles	1
5	78	Micaela	Lucas	1
6	98	Carie	Sharvill	1
7	90	Dennis	Bloxsome	1
8	97	Netta	Glowacha	1

Total rows: 100 of 100 Query complete 00:00:00.153

Number of tuples: 100

22. Display Owner ID and number of properties they have where rent is greater than 15000 and where parking space is available.

Query

```
select "Owner"."OwnerID",count(*)
from rh_system."Owner" join rh_system."Property"
on "Owner"."OwnerID"="Property"."OwnerID"
where "Rent" >15000 and "Parking"='true'
group by "Owner"."OwnerID"
order by "OwnerID"
```

Output

```
select "Owner"."OwnerID",count(*)
from rh_system."Owner" join rh_system."Property"
on "Owner"."OwnerID"="Property"."OwnerID"
where "Rent" >15000 and "Parking"='true'
group by "Owner"."OwnerID"
order by "OwnerID"
```

Data Output | Messages | Notifications

	OwnerID [PK] bigint	count bigint
1	1	1
2	2	1
3	3	2
4	4	1
5	7	1
6	9	2
7	11	1
8	12	2

Total rows: 67 of 67 Query complete 00:00:00.657

Number of tuples: 67

23. Find second highest rent and show its respective property id.

Query

```
Select  distinct "Rent" ,"PropertyID"
from rh_system."Property" p1
where 2=(Select count(distinct "Rent")
            from rh_system."Property" p2
            where p1."Rent"<p2."Rent")
```

Output

```
1  Select  distinct "Rent" ,"PropertyID"
2  from rh_system."Property" p1
3  where 2=(Select count(distinct "Rent")
4           from rh_system."Property" p2
5           where p1."Rent"<p2."Rent")
6
7
8
```

Data Output | Messages | Notifications

	Rent bigint	PropertyID [PK] bigint
1	29790	291

Total rows: 1 of 1 | Query complete 00:00:00.105

Number of tuples: 1

24. Find maximum rent in each state and order the data according to state (count union territories in state).

Query

```
select distinct (max("Rent")),"State"
from rh_system."Property" join rh_system."Location"
on "Property"."PropertyID"="Location"."PropertyID"
group by "State"
order by "State"
```

Output

```
select distinct (max("Rent")),"State"
from rh_system."Property" join rh_system."Location"
on "Property"."PropertyID"="Location"."PropertyID"
group by "State"
order by "State"
```

Data Output | Messages | Notifications

	max bigint	State character varying (30)
1	29157	Andaman and Nicobar Islan...
2	29790	Andhra Pradesh
3	29851	Arunachal Pradesh
4	29405	Assam
5	26494	Bihar
6	28159	Chandigarh
7	28367	Chhattisgarh
8	29075	Dadra and Nagar Haveli

Total rows: 35 of 35 | Query complete 00:00:00.111

Number of tuples: 35

25. Display IDs, first name and payment time of those tenants who made the payments from 9 am in morning to 9 am at night.

Query

```
select "Tenant"."TenantID","FName","Time"
from rh_system."Payment" join rh_system."Tenant"
on "Payment"."TenantID"="Tenant"."TenantID"
where "Time" between '09:00:00' and '21:00:00'
order by "Time"
```

Output

```
select "Tenant"."TenantID","FName","Time"
from rh_system."Payment" join rh_system."Tenant"
on "Payment"."TenantID"="Tenant"."TenantID"
where "Time" between '09:00:00' and '21:00:00'
order by "Time"
```

Data Output | Messages | Notifications

	TenantID bigint	FName character varying	Time time without time zone
1	138	Andreas	09:12:17
2	104	Dari	09:13:34
3	119	Lacy	09:18:12
4	152	Aluino	09:18:22
5	137	Bentlee	09:32:37
6	17	Ashia	09:34:34
7	189	Colette	09:37:00
8	82	Kenna	09:54:21

Total rows: 155 of 155 Query complete 00:00:00.078

Number of tuples: 155

26. Reverse the details of furnished and parking where furnished property is not available but parking is available.

Query

```
select Count(*)
from rh_system."Property"
where "Furnished"='false' and "Parking"='true'


UPDATE rh_system."Property"
SET "Furnished"='true' ,"Parking"='false'
WHERE "Furnished"='false' and "Parking"='True';


select Count(*)
from rh_system."Property"
where "Furnished"='false' and "Parking"='true'
```

Output

```
1  select Count(*)
2  from rh_system."Property"
3  where "Furnished"='false' and "Parking"='true'
4
5
6  UPDATE rh_system."Property"
7  SET "Furnished"='true' ,"Parking"='false'
8  WHERE "Furnished"='false' and "Parking"='True';
9
10
11 select Count(*)
12 from rh_system."Property"
13 where "Furnished"='false' and "Parking"='true'
14
```

Data Output | Messages | Notifications

	count bigint
1	440

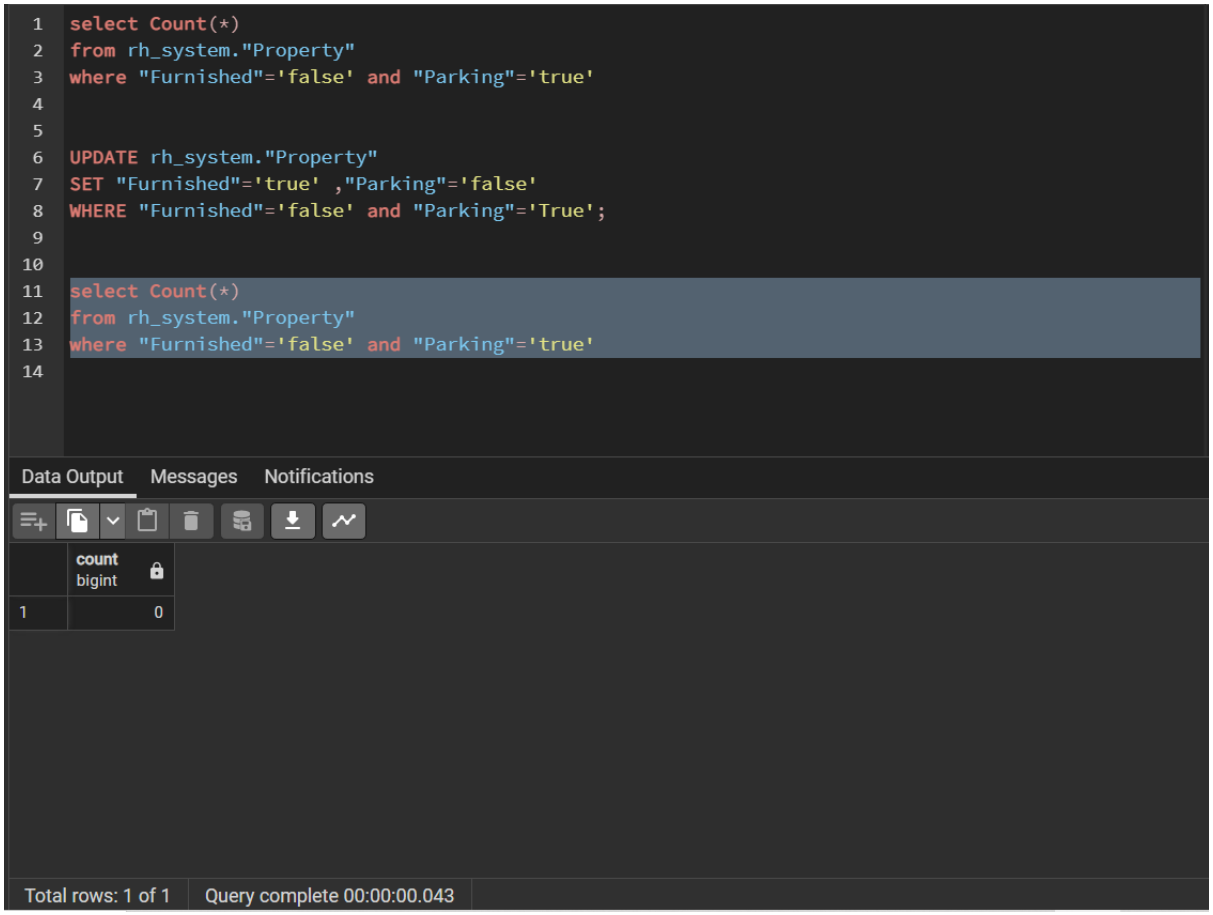
Total rows: 1 of 1 | Query complete 00:00:00.046

```sql
select Count(*)
from rh_system."Property"
where "Furnished"='false' and "Parking"='true'


UPDATE rh_system."Property"
SET "Furnished"='true' ,"Parking"='false'
WHERE "Furnished"='false' and "Parking"='True';


select Count(*)
from rh_system."Property"
where "Furnished"='false' and "Parking"='true'
```

Data Output | Messages | Notifications

```
UPDATE 440

Query returned successfully in 169 msec.
```

Total rows: 1 of 1 | Query complete 00:00:00.169

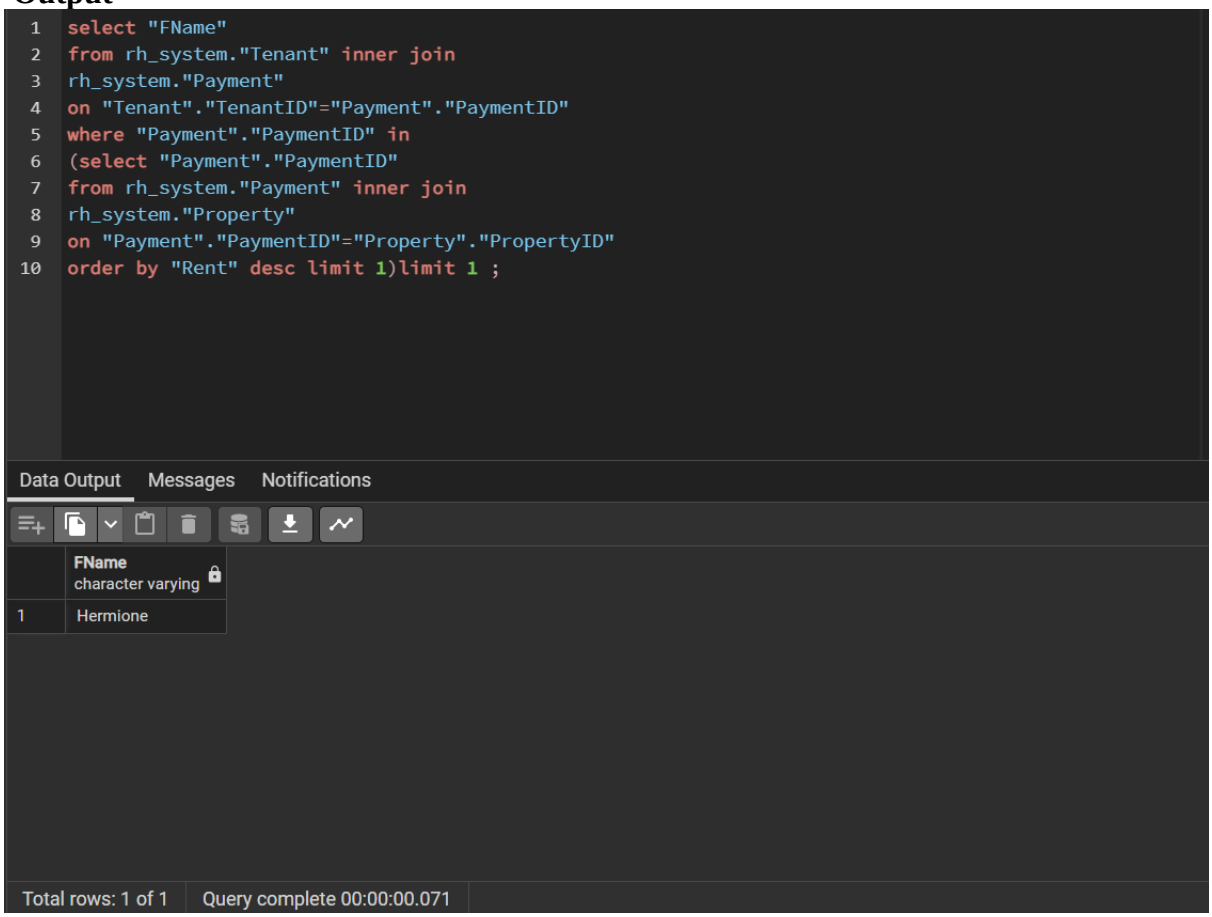


27. Display any one trainer's name who is paying the maximum rent.

Query

```
select "FName"
from rh_system."Tenant" inner join
rh_system."Payment"
on "Tenant"."TenantID"="Payment"."PaymentID"
where "Payment"."PaymentID" in
(select "Payment"."PaymentID"
from rh_system."Payment" inner join
rh_system."Property"
on "Payment"."PaymentID"="Property"."PropertyID"
order by "Rent" desc limit 1)limit 1 ;
```

Output



Number of tuples: 1

28. Print ContractID ,PropertyID and rent of those properties which is less than the average rent.

Query

```
select "Property"."PropertyID","ContractID","Property"."Rent"
from rh_system."Contract" inner join
rh_system."Property"
ON "Contract"."PropertyID" = "Property"."PropertyID"
where "Property"."Rent" < (select avg("Rent")
from rh_system."Property")
```

Output

```
select "Property"."PropertyID","ContractID","Property"."Rent"
from rh_system."Contract" inner join
rh_system."Property"
ON "Contract"."PropertyID" = "Property"."PropertyID"
where "Property"."Rent" < (select avg("Rent")
from rh_system."Property")
```

Data Output | Messages | Notifications

	PropertyID bigint	ContractID bigint	Rent bigint
101	298	213	16248
102	230	215	11575
103	9	216	9371
104	289	217	9932
105	46	219	11592
106	71	220	15196
107	83	221	9714
108	94	222	16401

Total rows: 149 of 149 | Query complete 00:00:00.081

Number of tuples: 149

29. Display the Agent id who have more than 3 properties on lease.

Query

```
select "Agent"."AgentID",Count("PropertyID")
from rh_system."Agent" inner join
rh_system."Property"
ON "Agent"."AgentID" = "Property"."AgentID"
group by "Agent"."AgentID"
having Count("PropertyID")>3
```

Output

```
select "Agent"."AgentID",Count("PropertyID")
from rh_system."Agent" inner join
rh_system."Property"
ON "Agent"."AgentID" = "Property"."AgentID"
group by "Agent"."AgentID"
having Count("PropertyID")>3
```

Data Output | Messages | Notifications

	AgentID [PK] bigint	count bigint
1	27	4
2	23	4
3	29	4
4	34	4
5	35	5
6	45	5
7	39	5
8	36	4

Total rows: 33 of 33 | Query complete 00:00:00.129

Number of tuples: 33

30. Display the TenantID , First name , last name of female tenants who have started living between '2022-04-05' and '2022-04-09'.

Query

```
select "Tenant"."TenantID", "FName","LName","Occupation"
from rh_system."Tenant" join rh_system."Payment"
on "Payment"."TenantID" = "Tenant"."TenantID"
where "PaymentID" in (select "Contract"."PaymentID"
                      from rh_system."Contract" left join rh_system."Payment"
                      on "Payment"."TenantID" = "Tenant"."TenantID"
                      where "StartDate" between '2022-04-05' and '2022-04-09' )
                      and "Gender"='Female'
```

Output

```
select "Tenant"."TenantID", "FName","LName","Occupation"
from rh_system."Tenant" join rh_system."Payment"
on "Payment"."TenantID" = "Tenant"."TenantID"
where "PaymentID" in (select "Contract"."PaymentID"
                      from rh_system."Contract" left join rh_system."Payment"
                      on "Payment"."TenantID" = "Tenant"."TenantID"
                      where "StartDate" between '2022-04-05' and '2022-04-09' )
     and "Gender"='Female'
```

Data Output | Messages | Notifications

	TenantID [PK] bigint	FName character varying	LName character varying	Occupation character varying
1	275	Ardella	Meenan	Lawyer
2	55	Stacia	Endrizzi	Engineer
3	32	Dode	McIlenna	Data Entry Operat...
4	2	Marti	Graveson	Engineer
5	125	Dayle	Fehely	Engineer
6	50	Carlotta	Grain	Engineer
7	142	Henrieta	Porch	Data Analyst
8	246	Georgeta	MacAlister	Engineer

Total rows: 8 of 8 | Query complete 00:00:00.395

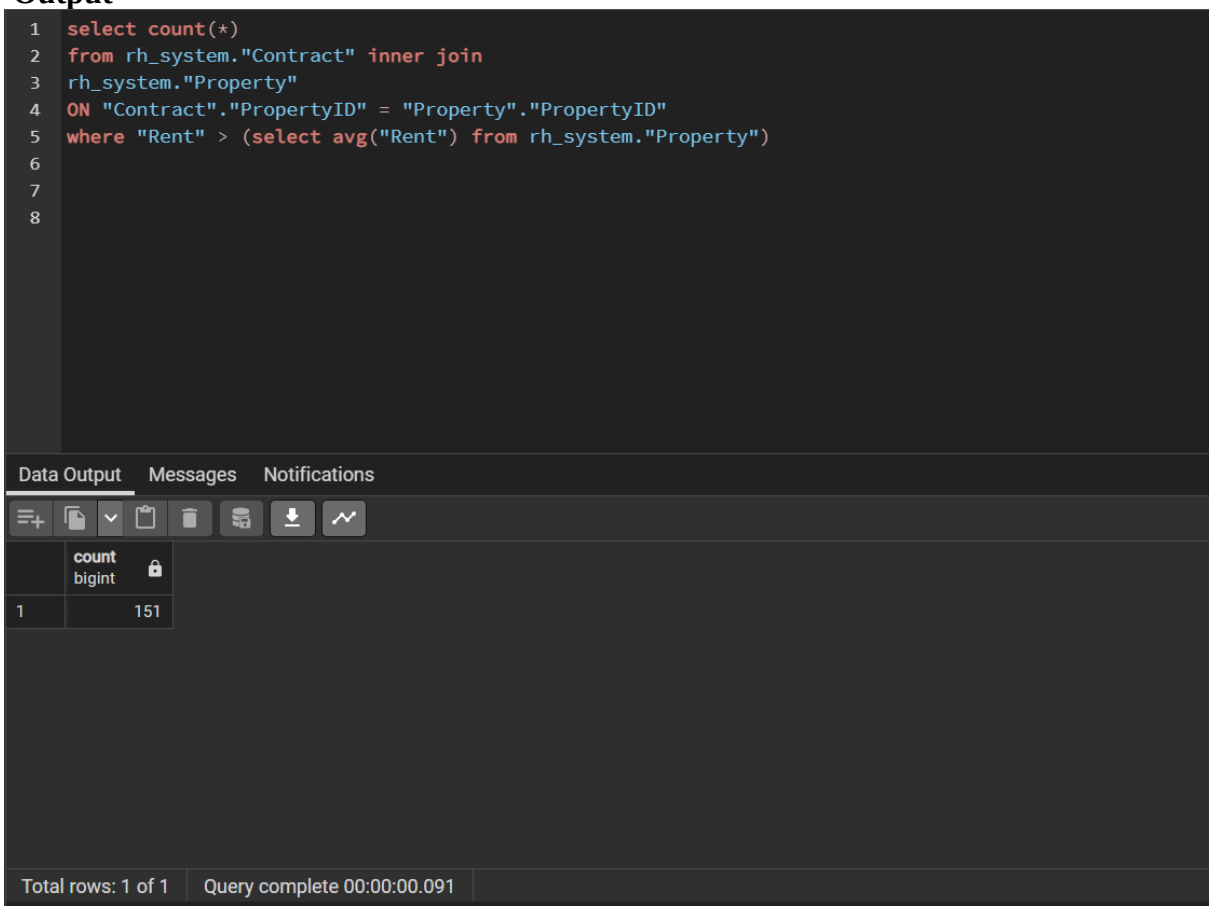
Number of tuples: 8

31. Number of tenants whose property rent is greater than the average rent.

Query

```
select count(*)
from rh_system."Contract" inner join
rh_system."Property"
ON "Contract"."PropertyID" = "Property"."PropertyID"
where "Rent" > (select avg("Rent") from rh_system."Property")
```

Output



Number of tuples: 1

32. Display ID, first name and last name of Tenants who started living after 24 January 2022 up to 2 months.

Query

```
select "Tenant"."TenantID","FName","LName"
from rh_system."Tenant" inner join
rh_system."Payment"
ON "Tenant"."TenantID" = "Payment"."TenantID"
where "PaymentID" in
(select "Payment"."PaymentID"
 from rh_system."Contract" inner join
rh_system."Payment"
ON "Contract"."PaymentID" = "Payment"."PaymentID"
 where "StartDate" between '2022/01/24'
and (select date '2022/01/24' + (Interval '2 month')))
```

Output

```
select "Tenant"."TenantID","FName","LName"
from rh_system."Tenant" inner join
rh_system."Payment"
ON "Tenant"."TenantID" = "Payment"."TenantID"
where "PaymentID" in
(select "Payment"."PaymentID"
 from rh_system."Contract" inner join
rh_system."Payment"
ON "Contract"."PaymentID" = "Payment"."PaymentID"
 where "StartDate" between '2022/01/24'
and (select date '2022/01/24' + (Interval '2 month')))
```

Data Output | Messages | Notifications

	TenantID [PK] bigint	FName character varying	LName character varying
1	62	Richmound	Jancso
2	16	Lindsay	Laterza
3	143	Rick	Peace
4	166	Antonina	Webbe
5	278	Annamarie	Scarth
6	20	Pepi	Moore
7	275	Ardella	Meenan
8	214	Juliet	Wonfor

Total rows: 120 of 120 | Query complete 00:00:00.099

Number of tuples: 120

33. Display the Owner ids and rent of properties they own where rent is between 10000 and 20000.

Query

```
select "Owner"."OwnerID","Rent"
from rh_system."Owner" inner join
rh_system."Property"
ON "Owner"."OwnerID" = "Property"."OwnerID"
where "Rent" between 10000 and 20000
order by "Owner"."OwnerID"
```

Output

```
select "Owner"."OwnerID","Rent"
from rh_system."Owner" inner join
rh_system."Property"
ON "Owner"."OwnerID" = "Property"."OwnerID"
where "Rent" between 10000 and 20000
order by "Owner"."OwnerID"
```

Data Output | Messages | Notifications

	OwnerID bigint	Rent bigint
1	1	15824
2	1	15718
3	3	17527
4	4	18350
5	5	16248
6	5	18960
7	6	13296
8	6	13854

Total rows: 121 of 121 | Query complete 00:00:00.217

Number of tuples: 121

34. Print Id, first name and last name and number of properties of agents where age is greater than 50 years 4 months 12 days or last name ends with e.

Query

```
SELECT "Agent"."AgentID","FName","LName",
AGE(current_date,"DOB"),count("PropertyID")
from rh_system."Agent" join rh_system."Property"
on "Property"."AgentID" = "Agent"."AgentID"
where AGE(current_date,"DOB") > '50 years 4 months 12 days'
        or "LName" like '%e'
group by "Agent"."AgentID"
```

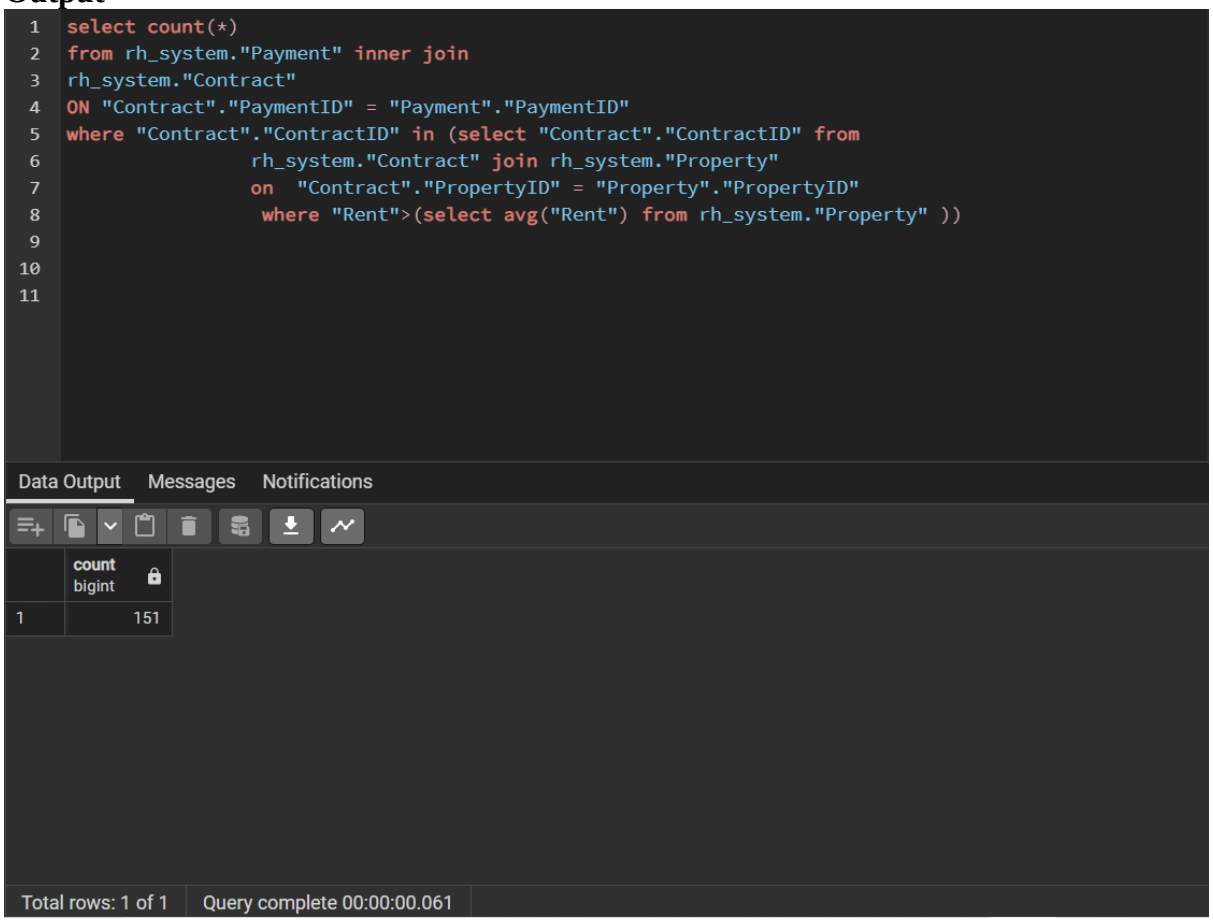
35. Number of tenants who are paying more than the average rent.

Query

```
select count(*)
from rh_system."Payment" inner join
```

```
rh_system."Contract"
ON "Contract"."PaymentID" = "Payment"."PaymentID"
where "Contract"."ContractID" in (select "Contract"."ContractID" from
                                    rh_system."Contract" join rh_system."Property"
                                on  "Contract"."PropertyID" = "Property"."PropertyID"
                                 where "Rent">(select avg("Rent") from rh_system."Property" ))
```

Output



Number of tuples: 1

36. Display ID, first name and last name of tenant who signed the contract of more than 2 years.

Query

```
select "Tenant"."TenantID","FName","LName"
from rh_system."Tenant" inner join
rh_system."Payment"
ON "Tenant"."TenantID" = "Payment"."TenantID"
where "PaymentID" in  (select "Payment"."PaymentID"
                        from rh_system."Contract" inner join
                        rh_system."Payment"
```

```
ON "Contract"."PaymentID" = "Payment"."PaymentID"
 where  AGE("EndDate","StartDate") > '2 year 0 months 0 days ')
```

Output

```
1  select "Tenant"."TenantID","FName","LName"
2  from rh_system."Tenant" inner join
3  rh_system."Payment"
4  ON "Tenant"."TenantID" = "Payment"."TenantID"
5  where "PaymentID" in
6  (select "Payment"."PaymentID"
7   from rh_system."Contract" inner join
8  rh_system."Payment"
9  ON "Contract"."PaymentID" = "Payment"."PaymentID"
10  where  AGE("EndDate","StartDate") > '2 year 0 months 0 days ')
11
12
13
14
```

Data Output | Messages | Notifications

	TenantID [PK] bigint	FName character varying	LName character varying
1	16	Lindsay	Laterza
2	19	Frasco	Pringuer
3	20	Pepi	Moore
4	28	Maryanna	Renshell
5	32	Dode	Mcllenna
6	37	Marie-jeanne	Donnell
7	38	Darin	Ducker
8	47	Brendin	Strood

Total rows: 41 of 41 | Query complete 00:00:00.493

Number of tuples: 41

37. Print ID, First and last name,and number of properties information that each admin is managing .

Query

```
select "Admin"."AdminID","FName","LName",count(*)
from rh_system."Admin" join rh_system."Property"
on "Admin"."AdminID"="Property"."AdminID"
group by "Admin"."AdminID"
```

Output

```
1  select "Admin"."AdminID","FName","LName",count(*)
2  from rh_system."Admin" join rh_system."Property"
3  on "Admin"."AdminID"="Property"."AdminID"
4  group by "Admin"."AdminID"
```

Data Output | Messages | Notifications

	AdminID [PK] bigint	FName character varying	LName character varying	count bigint
1	74	Wain	Whipple	7
2	29	Nikolai	Cluney	10
3	54	Mikel	Irnys	4
4	71	Suzann	Clurow	10
5	68	Melitta	Tanman	7
6	4	Don	Cropper	5
7	34	Zebulon	Bradborne	6
8	51	Anett	de Lloyd	8

Total rows: 80 of 80 | Query complete 00:00:00.058

Number of tuples: 80

38. Create a view of the “Admin” table and add three rows to that view and check if there is any change in the “Admin” table.

Query

```
create view rh_system."NewAdmin" as
select *
from rh_system."Admin";

insert into rh_system."NewAdmin"
Values
(81,'Rithik','Singh','rithiksingh@outlook.com','qdbkjbcvsn1'),
(82,'Shreya','Jha','jhashreya@aoautlook.com','zxadsxcbcvsn1'),
(83,'Dhairya','Kapoor','kapoordhairya@gmail.com','sdfscbcvsn1')

Select * from rh_system."Admin"
```

Output

```
1  create view rh_system."NewAdmin" as
2  select *
3  from rh_system."Admin";
4
5  insert into rh_system."NewAdmin"
6  Values
7  (81,'Rithik','Singh','rithiksingh@outlook.com','qdbkjbcvsn1'),
8  (82,'Shreya','Jha','jhashreya@aoautlook.com','zxadsxcbcvsn1'),
9  (83,'Dhairya','Kapoor','kapoordhairya@gmail.com','sdfscbcvsn1')
10
11 Select * from rh_system."Admin"
12
13
```

Data Output | Messages | Notifications

```
CREATE VIEW

Query returned successfully in 1 secs 77 msec.
```

```sql
create view rh_system."NewAdmin" as
select *
from rh_system."Admin";

insert into rh_system."NewAdmin"
Values
(81,'Rithik','Singh','rithiksingh@outlook.com','qdbkjbcvsn1'),
(82,'Shreya','Jha','jhashreya@aoautlook.com','zxadsxcbcvsn1'),
(83,'Dhairya','Kapoor','kapoordhairya@gmail.com','sdfscbcvsn1')

Select * from rh_system."Admin"
```

Data Output | Messages | Notifications

```
INSERT 0 3

Query returned successfully in 535 msec.
```

```
create view rh_system."NewAdmin" as
select *
from rh_system."Admin";

insert into rh_system."NewAdmin"
Values
(81,'Rithik','Singh','rithiksingh@outlook.com','qdbkjbcvsn1'),
(82,'Shreya','Jha','jhashreya@aoautlook.com','zxadsxcbcvsn1'),
(83,'Dhairya','Kapoor','kapoordhairya@gmail.com','sdfscbcvsn1')

Select * from rh_system."Admin"
```

Data Output | Messages | Notifications

	AdminID [PK] bigint	FName character varying	LName character varying	Email character varying	Password character varying
76	76	Louisette	Comettoi	lcomettoi23@wei...	R6v6zZ7iQ
77	77	Ralf	Klimov	rklimov24@usne...	A0M49xWaQAkA
78	78	Lukas	Calow	lcalow25@imdb....	7jmLnLvoR
79	79	Clemence	MacNeilley	cmacneilley26@...	kPfGhFlOuB
80	80	Cherilyn	Janatka	cjanatka27@unic...	bj3pDnoK
81	81	Rithik	Singh	rithiksingh@outl...	qdbkjbcvsn1
82	82	Shreya	Jha	jhashreya@aoaut...	zxadsxcbcvsn1
83	83	Dhairya	Kapoor	kapoordhairya@...	sdfscbcvsn1

Total rows: 83 of 83 Query complete 00:00:00.144

Number of tuples: 83

39. Update the view created in above question. Set Fname to ‘Hritik, and LName to ‘MacNeilley’ and then print all the data of admins having ‘MacNeilley’ as last name.

Query

```
Update rh_system."Admin"
set "FName"='Hritik',"LName"='MacNeilley'
where "AdminID"=81;

Select *
from rh_system."NewAdmin"
where "LName"='MacNeilley'
```

Output

```
1  Update rh_system."Admin"
2  set "FName"='Hritik',"LName"='MacNeilley'
3  where "AdminID"=81;
4
5  Select *
6  from rh_system."NewAdmin"
7  where "LName"='MacNeilley'
```

Data Output | Messages | Notifications

	AdminID bigint	FName character varying	LName character varying	Email character varying	Password character varying
1	79	Clemence	MacNeilley	cmacneilley26@...	kPfGhFlOuB
2	81	Hritik	MacNeilley	rithiksingh@outl...	qdbkjbcvsn1

Total rows: 2 of 2 | Query complete 00:00:00.153

Number of tuples: 2

40. Find tenant ids of those tenants who are living in Villa.

Query

```
select "Tenant"."TenantID"
from rh_system."Tenant" join rh_system."Payment"
on "Payment"."TenantID" = "Tenant"."TenantID"
where "PaymentID" in (select "Contract"."PaymentID"
            from rh_system."Contract" join rh_system."Payment"
            on "Payment"."PaymentID" = "Contract"."PaymentID"
            where "Contract"."PropertyID" in
              (select "Contract"."PropertyID"
               from rh_system."Contract"  left join rh_system."Property"
               on "Property"."PropertyID" = "Contract"."PropertyID"
               where "PropertyType"='Villa'))
```

Output

```
select "Tenant"."TenantID"
from rh_system."Tenant" join rh_system."Payment"
on "Payment"."TenantID" = "Tenant"."TenantID"
where "PaymentID" in (select "Contract"."PaymentID"
                      from rh_system."Contract" join rh_system."Payment"
                      on "Payment"."PaymentID" = "Contract"."PaymentID"
                      where "Contract"."PropertyID" in
                                          (select "Contract"."PropertyID"
                                    from rh_system."Contract"  left join rh_system."Property"
                                    on "Property"."PropertyID" = "Contract"."PropertyID"
                                    where "PropertyType"='Villa'))
```

Data Output | Messages | Notifications

	TenantID [PK] bigint
1	16
2	143
3	166
4	20
5	6
6	255
7	171
8	224

Total rows: 59 of 59 | Query complete 00:00:00.153

Number of tuples: 59