UtahRealEstate.com

Project Summary Report

Group 5

Kleyton Polzonoff

Apoorva Dharadhar

Faraaz Mohammed

Joshua Dunn

Table of Contents

Executive Summary

In recent years, online real estate companies have transformed the way people buy, sell, and rent properties. These platforms, including Utah Real Estate, have become necessary tools for both buyers and sellers. These tools provide a wide array of services, from property listings and virtual tours to mortgage calculators and neighborhood insights, empowering users with comprehensive information to make informed decisions. These platforms facilitate seamless communication between buyers, sellers, and agents, fostering a transparent and streamlined transaction process.

In addition to its user-focused approach, Utah Real Estate leverages advanced technologies to ensure a seamless and secure experience. Robust search algorithms and filters enable users to narrow down their options efficiently, saving time and effort. The platform offers valuable resources and information to empower both buyers and sellers. Detailed guides, market insights, and expert advice are readily available, equipping users with the knowledge they need to make informed decisions in the ever-changing real estate market. Whether someone is a first-time homebuyer or an experienced investor, Utah Real Estate provides the necessary tools to navigate the complexities of real estate transactions successfully.

Our team aimed to explore the capabilities of Utahrealestate.com in linking prospective homeowners with reliable realtors, precise information, and suitable properties.

In this report, we provided data generated by Mockaroo - Random Data Generator and API Mocking Tool | JSON / CSV / SQL / Excel. In doing so, we could simulate what it would be like if we were actual database engineers for Utah Real Estate.

General Description

History

Utah Real Estate has been helping customers and sellers alike since 1994. URE enables the distribution of real estate listings to more than 18,500 real estate experts through their multiple listing services. Each year, it is estimated that they facilitate information for over 50,000 real estate transactions.

Vision & Objectives

Utah Real Estate is dedicated to assisting buyers and renters by offering precise and reliable information about properties in their areas of interest. For sellers and landlords, the platform provides an opportunity to showcase their homes to potentially interested parties by reaching a wide audience of millions of viewers.

Products and Services

Utah Real Estate provides its website to anyone who is curious about the housing market in Utah, as well as buyers and sellers. URE also provides RapidStats to its members. Rapid Stats is a user-friendly platform that helps real estate professionals analyze market data easily and communicate insights effectively to their clients.

Transactional Databases

Utah Real Estate uses transactional databases to create users for both buyers and sellers, store key details about houses, manage communication between buyers and sellers, and keep track of transaction history.

In creating our database, we focused on the customer, the realtor, the property, and everything in between.

Prioritized Requirements Summary

Given the company's primary objective and its business model, the company aims to provide users with fast, clear, and up-to-date information about real estate properties while facilitating an affordable and reliable connection with real estate agents. To achieve this, its platform offers various features, with the key ones being:

- The Advanced Search Tool - simplifies the process of finding properties of interest.
- Find a REALTOR® - offers a sense of security for interested parties, bringing reliable information about the realtors and their skills.
- Mortgage Calculator - assists buyers in estimating their payments for the property they're interested in.

Based on these tools, we have decided to create the following tables:

- Brokerage
- Realtor
- Realtor contact details
- Customer
- Customer contact details
- Customer term agreement
- Customer terms of use
- Scheduler
- Property
- Property owner
- Property owner details
- Property class
- Property type
- Property features
- Mortgage calculator

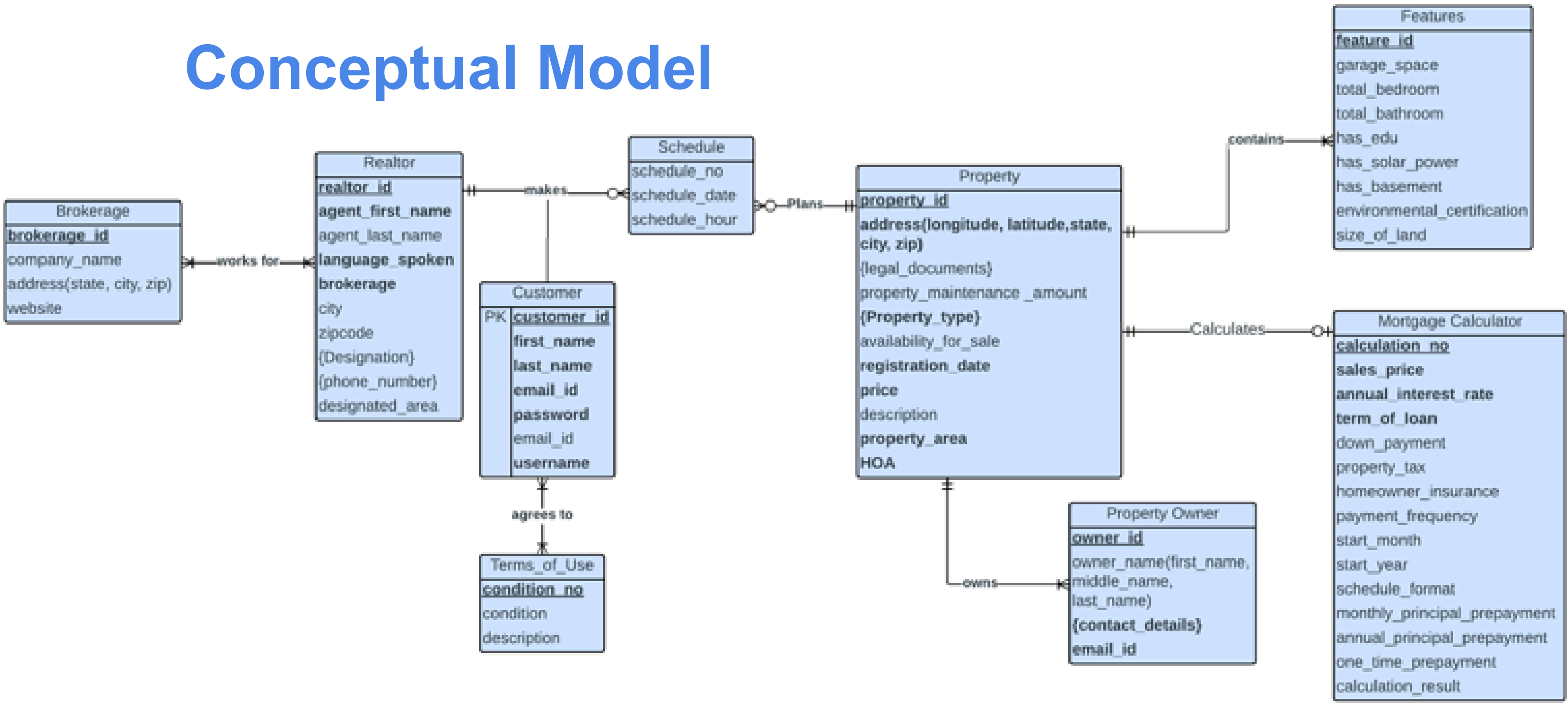
These tables enable us to establish the fundamental structure for the process operation among the parties involved, allowing for the insertion, storage, and maintenance of the necessary information. Using SQL, we have successfully developed this structure and inserted simulated data for 20 properties, clients, and brokers.

New Venture

Following conversations with realtors using Utahrealestate.com, it became apparent that their revenue is solely derived from charging realtors a modest annual fee for site usage. While this approach is acceptable, we propose a strategy to enhance revenue. Utahrealestate.com could introduce a premium package offering limited features. The free version would include basic features such as the search tool that they offer but exclude crucial tools like the mortgage calculator and the Find a Realtor® tool, enticing users to opt into the premium version for access to the complete set of tools and features. We propose that Utahrealestate.com charges a modest monthly fee of approximately $5, ensuring affordability for users while generating additional revenue for URE.

An additional idea to enhance the URE business involves partnering with a bank and assigning a loan officer from the bank to manage a live chat service accessible to clients. Implementing this service would provide instant value to UtahRealEstate.com users. The loan officer could promptly offer clients a preliminary estimate of their monthly payments by evaluating a few essential credentials. Furthermore, having a dedicated loan officer available through live chat not only simplifies the mortgage estimation process but also fosters trust and confidence among clients. This personalized assistance can significantly enhance the overall user experience, making UtahRealEstate.com a go-to platform for potential homebuyers.

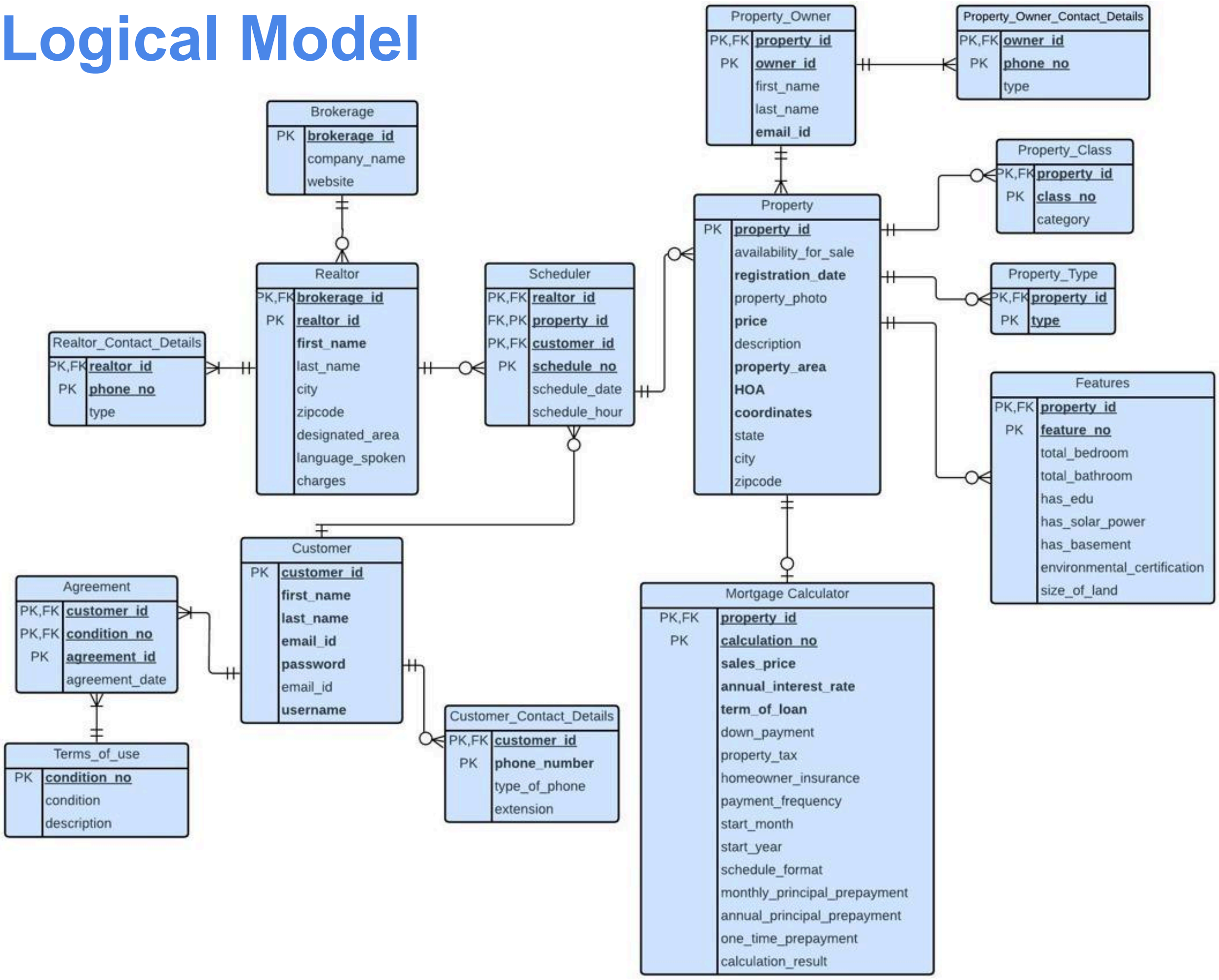
Conceptual Model



A realtor works for the brokerage. Every brokerage will have multiple realtors working for it. A realtor is a primary source of contact for the customer. A customer can schedule an appointment with the realtor to visit the property. We have showcased a ternary relationship between the entity's realtor, customer, and schedule. A customer may or may not schedule an appointment with the realtor to plan the property visit. Therefore, there is a one-to-one to optional many relationship between the realtor, the customer, and the schedule. A user can create an account on UtahRealEstate.com with a username, email, and password. Those fields will be unique to that particular user.

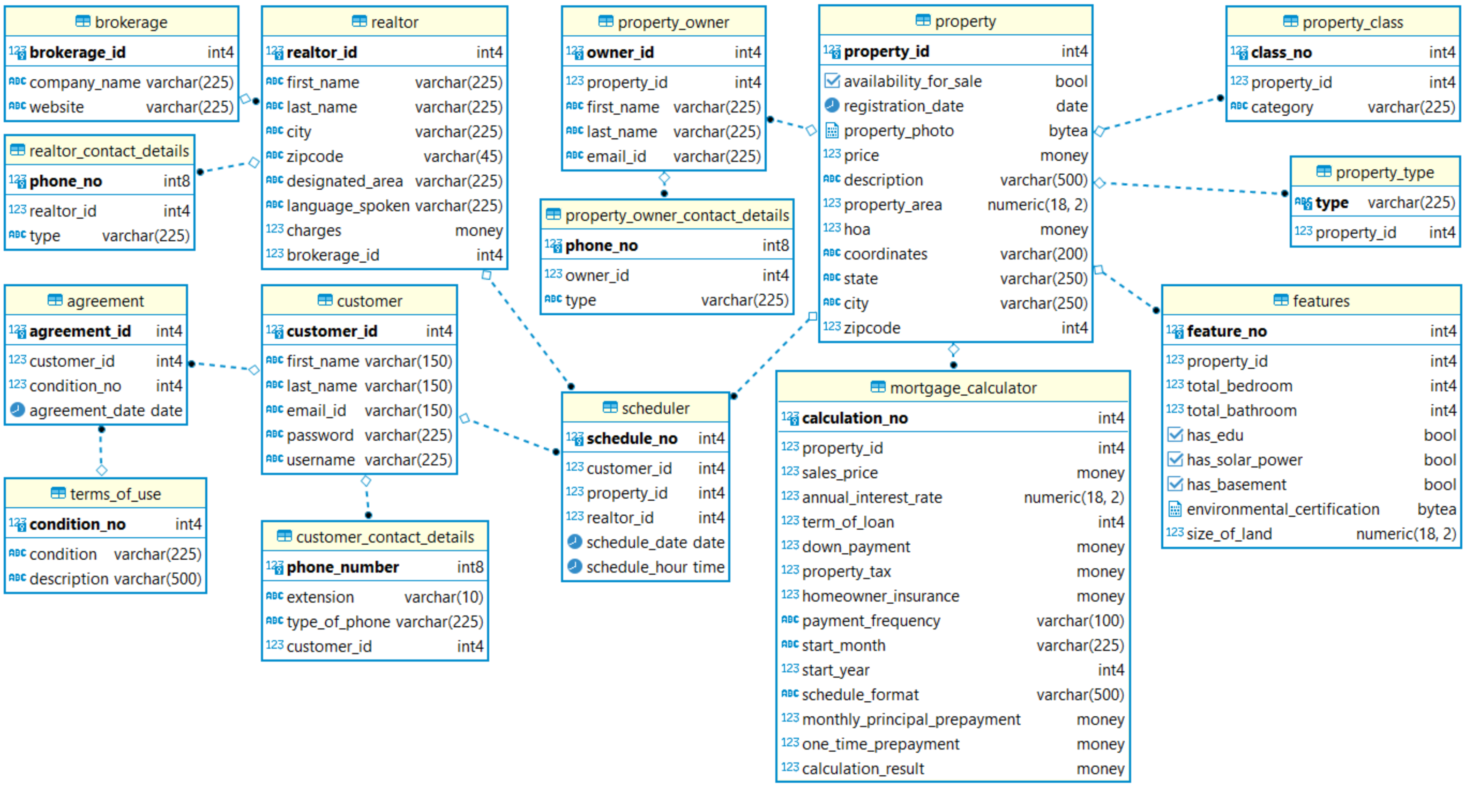
A property can get multiple visits from customers. If a property is new in the market, it may or may not have received any visits from the customers, Therefore, the type of relationship between the property and the schedule is 'one to optional many'. A customer can calculate the Mortgage using a Mortgage Calculator. The annual interest rate, sales price, and term of the loan are some of the mandatory inputs that the system takes to calculate the possible mortgage on the property. You can view the calculated mortgage on the description page of each property for the logged-in customer.

Logical Model



In the conceptual model, we have modeled the multi-valued attributes. The contact details of the realtor are modeled as a separate entity. Every realtor will have a phone number and the type of phone number — for example, a fax number, personal mobile number, etc. As every realtor must have a phone number to contact, the type of relationship between the realtor and the contact details is a 'mandatory one to mandatory many' kind of relationship. A customer agrees to the Terms of Use. Therefore, we have modeled the agreement entity as a bridge between the customer and the Terms of Use. Again, the multi-valued attribute, contact details, is modeled as a separate entity. Similarly, for the other entities such as property owner and property, we modeled the multi-valued attributes into separate individual entities for each. For example, for the property entity, we have modeled property type and property class as separate individual attributes, and for the property owner, contact details is modeled as separate entities. The relationship between the property owner and contact details is of the 'Mandatory One to Mandatory Many' type of relationship. The relationship between property and property class is a mandatory one to optional many relationship. A property can have multiple classes such as 1 bedroom & 1 bathroom, 2 bedrooms & 1 bathroom with a guest house, etc. A property can be of multiple types such as commercial, residential, land, etc. Also, a property may optionally have one mortgage calculator, to help calculate the mortgage on that particular property. A property has mandatory attributes such as coordinates, registration date, the area of the given property, and the HOA.

Physical Model



brokerage	
brokerage_id	int4
company_name	varchar(225)
website	varchar(225)

realtor_contact_details	
phone_no	int8
realtor_id	int4
type	varchar(225)

agreement	
agreement_id	int4
customer_id	int4
condition_no	int4
agreement_date	date

terms_of_use	
condition_no	int4
condition	varchar(225)
description	varchar(500)

realtor	
realtor_id	int4
first_name	varchar(225)
last_name	varchar(225)
city	varchar(225)
zipcode	varchar(45)
designated_area	varchar(225)
language_spoken	varchar(225)
charges	money
brokerage_id	int4

customer	
customer_id	int4
first_name	varchar(150)
last_name	varchar(150)
email_id	varchar(150)
password	varchar(225)
username	varchar(225)

customer_contact_details	
phone_number	int8
extension	varchar(10)
type_of_phone	varchar(225)
customer_id	int4

property_owner	
owner_id	int4
property_id	int4
first_name	varchar(225)
last_name	varchar(225)
email_id	varchar(225)

property_owner_contact_details	
phone_no	int8
owner_id	int4
type	varchar(225)

scheduler	
schedule_no	int4
customer_id	int4
property_id	int4
realtor_id	int4
schedule_date	date
schedule_hour	time

property	
property_id	int4
availability_for_sale	bool
registration_date	date
property_photo	bytea
price	money
description	varchar(500)
property_area	numeric(18, 2)
hoa	money
coordinates	varchar(200)
state	varchar(250)
city	varchar(250)
zipcode	int4

mortgage_calculator	
calculation_no	int4
property_id	int4
sales_price	money
annual_interest_rate	numeric(18, 2)
term_of_loan	int4
down_payment	money
property_tax	money
homeowner_insurance	money
payment_frequency	varchar(100)
start_month	varchar(225)
start_year	int4
schedule_format	varchar(500)
monthly_principal_prepayment	money
one_time_prepayment	money
calculation_result	money

property_class	
class_no	int4
property_id	int4
category	varchar(225)

property_type	
type	varchar(225)
property_id	int4

features	
feature_no	int4
property_id	int4
total_bedroom	int4
total_bathroom	int4
has_edu	bool
has_solar_power	bool
has_basement	bool
environmental_certification	bytea
size_of_land	numeric(18, 2)

Our Approach

When we initially selected UtahRealEstate.com for our research, most of our group members lacked a profound understanding of the online real estate market. However, through extensive research and dedicated efforts, we not only grasped the intricacies of UtahRealEstate.com but also developed a database that correctly mirrors the platform's offerings. Our journey involved a steep learning curve, but with time and persistence, we transformed our initial uncertainties into a deep understanding of URE, resulting in the creation of an accurate and comprehensive database.

"As a collaborative team, we engaged in extensive discussions concerning the incorporation and exclusion of different entities within both the logical and relational databases. There were numerous rounds of discussions and considerations. Ultimately, after a thorough analysis, we are convinced that our final physical model effectively encapsulates the essence of UtahRealEstate.com as a company.

Sample Data

As mentioned above, in this report, we provided data generated by Mockaroo - Random Data Generator and API Mocking Tool | JSON / CSV / SQL / Excel. In doing so, we could simulate what it would be like if we were actual database engineers for Utah Real Estate.

Our data was broken into the following:

20 Properties

20 Realtors

20 Terms of use

20 Brokerages

20 Property Owners

20 Customers

Keeping the data set small, and all the same number helped us gain a better understanding of the physical model.

Requirements Review

We successfully executed and integrated the prioritized features, rendering them fully functional within our database system. Below is an elaborate breakdown of the specific features that were created:

- Customers and realtors can complete their profiles.
- Properties and their features can be stored and accessed effectively.
- Customers can arrange property visits with realtors.
- The mortgage calculator provides property information for calculations

To implement these features, we've identified the following specific tables to be established:

- Brokerage
- Realtor
- Realtor contact details
- Customer
- Customer contact details
- Customer term agreement
- Customer terms of use
- Scheduler
- Property
- Property owner
- Property owner details
- Property class
- Property type
- Property Features
- Mortgage calculator

At the end of the process, we filled those tables. With the data mentioned above.

Ethical Considerations

Even when using randomly generated data, ethical considerations and security measures are still essential. While the data might not be directly tied to real individuals, it's crucial to maintain ethical standards to ensure the integrity of the research process. To be completely transparent about the data we have used, we mentioned the following in the general description, on page 3 of this report. “In this report, we provided data generated by Mockaroo - Random Data Generator and API Mocking Tool | JSON / CSV / SQL / Excel.”

It is also crucial to prioritize security measures when dealing with generated data. While our project didn't have a formal security system, our team actively ensured that the generated data was limited to those directly involved. Preventing unauthorized access was a collective effort, acknowledging the potential fallout if a breach were to occur. Maintaining the confidentiality of even synthetic data was paramount, as any compromise could lead to significant trust issues within our organization.

Conclusion and Next Steps

In conclusion, it has been a rewarding and educational experience for us to investigate UtahRealEstate.com and then create the database structure through in-depth study and teamwork. We effectively analyzed the platform's offerings by diving into the nuances of the online real estate market and pinpointing important features and functionalities that are essential to the platform's operation.

We created an effective database system by carefully designing and considering the key details, which reflect the core values of UtahRealEstate.com. The foundation of a smooth and effective real estate platform is made up of the tables we created, which include everything from realtor and consumer profiles to property information and mortgage calculators. Our strategy included extensive talks, continuous improvement, and a strong emphasis on user experience to make sure the system meets the various needs of real estate agents, buyers, and sellers.

The effective integration of prioritized features, which allowed users and realtors to fill out profiles, save and retrieve property information, schedule property visits, and use the mortgage calculator to make well-informed decisions, was one of our project's major accomplishments.

All things considered, our path from designing the database architecture to actually putting it into practice has been characterized by commitment, teamwork, and ongoing education. We are certain that the thorough and well-considered database we have created would work in perfect harmony with UtahRealEstate.com's mission to link buyers, sellers, and real estate brokers with dependable information quickly, clearly, and up-to-date.

We are proud of our collective achievements as a team and think that UtahRealEstate.com would be happy and impressed with the solid foundation our effort has laid. With a thorough grasp of the platform and an efficient database system in place, UtahRealEstate.com would be well-positioned to carry on changing the real estate market in Utah, for years to come, empowering consumers and enabling easy and transparent transactions.

References

https://www.utahrealestate.com/index/public.index?gclid=CjwKCAjws9ipBhB1EiwAccEi1IBtIIeIGwECpunHYBMhpoDtEw0-dfMLfCrgigKOS5UVmpa0EOvs9hoCAfUQAvD_BwE

https://why.utahrealestate.com/

https://www.mockaroo.com/

https://chat.openai.com/

https://growjo.com/company/UtahRealEstate.com

https://play.google.com/store/apps/details?id=com.eur android&hl=en_US&gl=US

Appendix

Team member	Hours Spent	Description of Work
Kleyton Polzonoff	25	• Company understanding: 2 hours • Group meetings: 10 hours • Conceptual & Logical Models: 4 Hours • Database and SQL: 3 Hour • Written report: 2 hours • Presentation preparation: 3 hours
Apoorva Dharadhar	27	Group Meetings:10 Hours Company Research: 4 Hours Conceptual & Logical Modeling: 8 Hours SQL Coding: 3 Hours Written Report: 2 hours
Faraaz Mohammed	25	Group Meetings:10 hours Conceptual & Logical Modeling: 3 Hours SQL Coding: 10 Hours Written Report: 2
Joshua Dunn	26	Group Meetings:10 Company Research: 2 Slide Deck: 4 Written Report: 10

Create table statements

```
--Customer Table
drop table if exists Customer;
create table Customer(
            customer_id int primary key,
            first_name varchar(150) not null,
            last_name varchar(150) NOT null,
            email_id varchar(150) not null,
            password varchar(225) not null,
            username varchar(225) not NULL
);

--Terms_of_Use table
drop table if exists Terms_of_Use;
create table Terms_of_Use(
            condition_no int primary key,
            condition varchar(225),
            description varchar(500)
);
--Agreement Table
drop table if exists Agreement;
create table Agreement(
            customer_id int,
            condition_no int ,
            agreement_id int primary key,
            agreement_date date,
            constraint fk_customer_agreement foreign key (customer_id)
            references Customer(customer_id),
            constraint fk_termsofuse_agreement foreign key (condition_no)
            references Terms_of_Use(condition_no)
);

--Table Customer_Contact_Details
drop table if exists Customer_Contact_Details;
create table Customer_Contact_Details(
            phone_number bigint primary key,
            extension varchar(10),
            type_of_phone varchar(225),
            customer_id int,
            constraint fk_customer_customercontactdetails
            foreign key(customer_id) references Customer(customer_id)

);

--Table Brokerage
drop table if exists Brokerage;
create table Brokerage(
            brokerage_id int primary key,
            company_name varchar(225),
            website varchar(225)
);

--Table Realtor
```

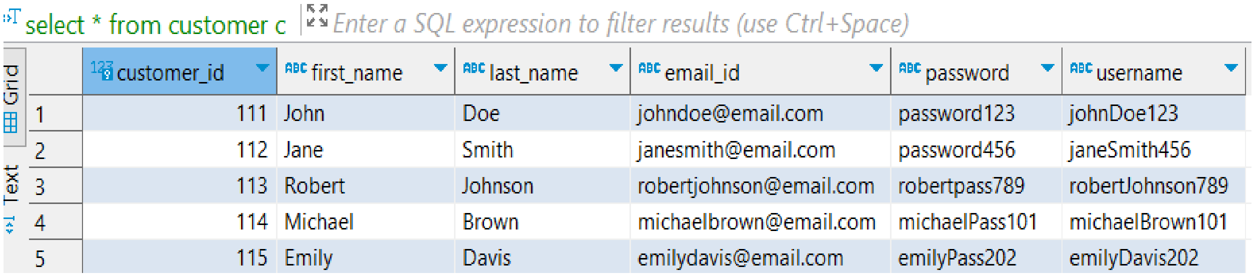
```
drop table if exists Realtor;
create table Realtor(
            realtor_id int primary key,
            first_name varchar(225) not null,
            last_name varchar(225), city varchar(225),
            zip code varchar(45), designated_area varchar(225),
            language_spoken varchar(225), charges money,
            brokerage_id int,
            constraint fk_brokerage_realtor_realtorid
            foreign key(brokerage_id) references Brokerage(brokerage_id)

);

--Table Realtor_Contact_Details
drop table if exists Realtor_Contact_Details;
create table Realtor_Contact_Details(
            phone_no bigint primary key,
            realtor_id int,
            type varchar(225),
            constraint fk_realtor_contactdetails_brokerageid
            foreign key(realtor_id) references Realtor(realtor_id)
);

--Table Property
drop table if exists Property;
create table Property(
            property_id int primary key,
            availability_for_sale boolean,
            registration_date date not null,
            property_photo bytea,
            price money not null,
            description varchar(500),
            property_area decimal(18,2) not null,
            HOA money not null,
            coordinates varchar(200) not null,
            state varchar(250),
            city varchar(250),
            zip code int
```

```
);



--Table Property_Owner
drop table if exists Property_Owner;
create table Property_Owner(
            property_id int,
            owner_id int primary key,
            first_name varchar(225),
            last_name varchar(225),
            email_id varchar(225) not null,
            constraint fk_property_owner_propertyid foreign key(property_id)
            references Property(property_id)
);

--Table Property_Owner_Contact_Details
drop table if exists Property_Owner_Contact_Details;
create table Property_Owner_Contact_Details(
            phone_no bigint primary key,
            owner_id int,
            type varchar(225),
            constraint fk_owner_contactdetails_ownerid foreign key(owner_id)
            references Property_Owner(owner_id)
);

--Table Property_Type
drop table if exists Property_Type;
create table Property_Type(
            type varchar(225) primary key,
            property_id int,
      constraint fk_property_type_propertyid
      foreign key(property_id)references Property(property_id)
);

--Table Features
drop table if exists Features;
create table Features(
            feature_no int primary key,
            property_id int,
            total_bedroom int,
            total_bathroom int,
            has_edu boolean,
            has_solar_power boolean,
            has_basement boolean,
            environmental_certification bytea,
            size_of_land decimal(18,2),
      CONSTRAINT fk_features_property_id
      FOREIGN KEY (property_id) REFERENCES Property(property_id)
);

--Table Mortgage_Calculator
drop table if exists Mortgage_Calculator;
create table Mortgage_Calculator(
            property_id int,
            calculation_no int primary key,
            sales_price money not null,
            annual_interest_rate decimal(18,2) not null,
            term_of_loan int not null,
            down_payment money,
            property_tax money,
            homeowner_insurance money,
            payment_frequency varchar(100),
            start_month varchar(225),
            start_year int,
            schedule_format varchar(500),
            monthly_principal_prepayment money,
            one_time_prepayment money,
            calculation_result money,
      CONSTRAINT fk_mortgage_calculator_property_id
      FOREIGN KEY (property_id) REFERENCES Property(property_id)
);
```

```
--Table Scheduler
drop table if exists Scheduler;
create table Scheduler(
            schedule_no int primary key,
            customer_id int,
            property_id int,
            realtor_id int,
            schedule_date date,
            schedule_hour time,
constraint fk_realtor_scheduler_realtorid
      foreign key(realtor_id)references Realtor(realtor_id),
constraint fk_customer_scheduler_customerid
      foreign key(customer_id)references Customer(customer_id),
constraint fk_property_scheduler_schedulerid
      foreign key(property_id)references Property(property_id)
);


--Table Property_Class
drop table if exists Property_Class;
create table Property_Class(
            class_no int primary key,
            property_id int,
            category varchar(225),
            constraint fk_propertyclass_property_id foreign key(property_id)
            references Property(property_id)
);
```

Data:

Table 1: Customer

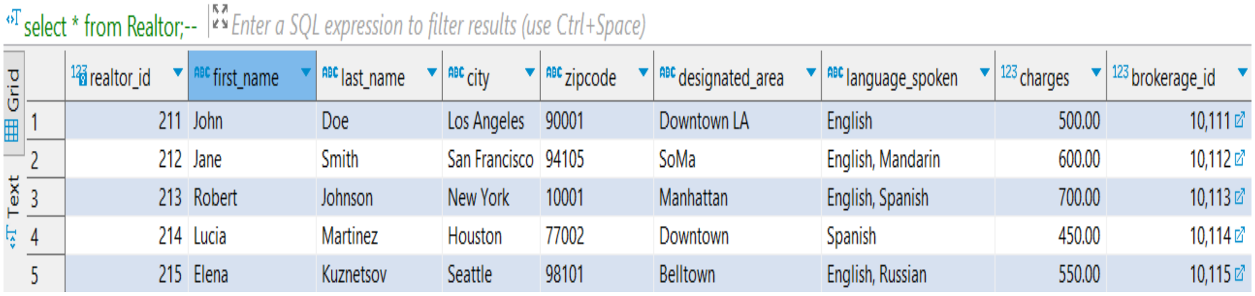


select * from customer c

Enter a SQL expression to filter results (use Ctrl+Space)

	customer_id	first_name	last_name	email_id	password	username
1	111	John	Doe	johndoe@email.com	password123	johnDoe123
2	112	Jane	Smith	janesmith@email.com	password456	janeSmith456
3	113	Robert	Johnson	robertjohnson@email.com	robertpass789	robertJohnson789
4	114	Michael	Brown	michaelbrown@email.com	michaelPass101	michaelBrown101
5	115	Emily	Davis	emilydavis@email.com	emilyPass202	emilyDavis202

Table 2: Realtor

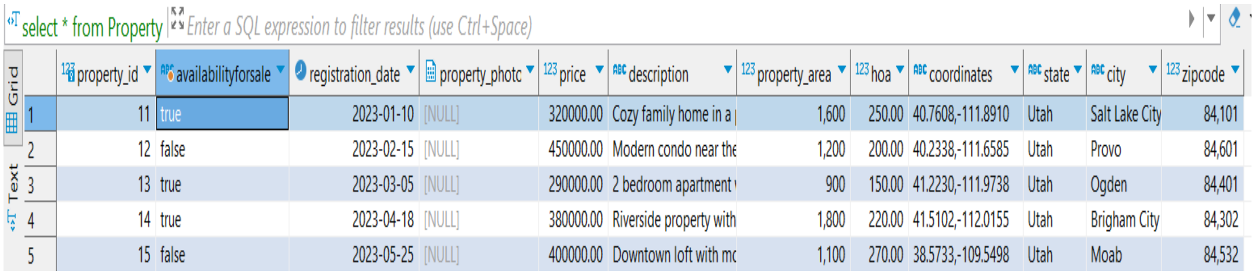


select * from Realtor;--

Enter a SQL expression to filter results (use Ctrl+Space)

	realtor_id	first_name	last_name	city	zipcode	designated_area	language_spoken	charges	brokerage_id
1	211	John	Doe	Los Angeles	90001	Downtown LA	English	500.00	10,111
2	212	Jane	Smith	San Francisco	94105	SoMa	English, Mandarin	600.00	10,112
3	213	Robert	Johnson	New York	10001	Manhattan	English, Spanish	700.00	10,113
4	214	Lucia	Martinez	Houston	77002	Downtown	Spanish	450.00	10,114
5	215	Elena	Kuznetsov	Seattle	98101	Belltown	English, Russian	550.00	10,115

Table 3: Property

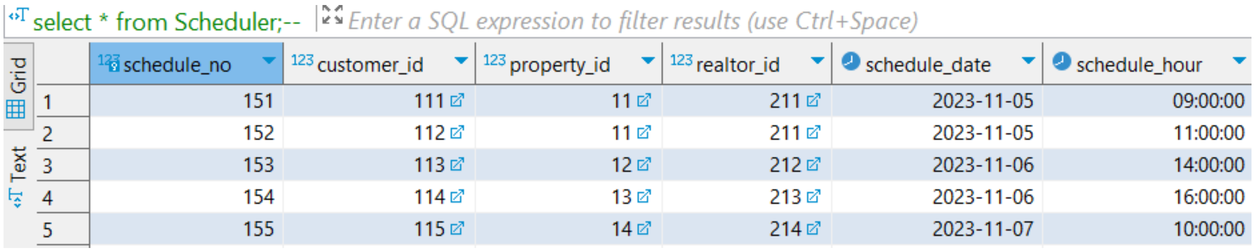


select * from Property

Enter a SQL expression to filter results (use Ctrl+Space)

	property_id	availabilityforsale	registration_date	property_photc	price	description	property_area	hoa	coordinates	state	city	zipcode
1	11	true	2023-01-10	[NULL]	320000.00	Cozy family home in a	1,600	250.00	40.7608,-111.8910	Utah	Salt Lake City	84,101
2	12	false	2023-02-15	[NULL]	450000.00	Modern condo near the	1,200	200.00	40.2338,-111.6585	Utah	Provo	84,601
3	13	true	2023-03-05	[NULL]	290000.00	2 bedroom apartment	900	150.00	41.2230,-111.9738	Utah	Ogden	84,401
4	14	true	2023-04-18	[NULL]	380000.00	Riverside property with	1,800	220.00	41.5102,-112.0155	Utah	Brigham City	84,302
5	15	false	2023-05-25	[NULL]	400000.00	Downtown loft with mc	1,100	270.00	38.5733,-109.5498	Utah	Moab	84,532

Table 4: Scheduler

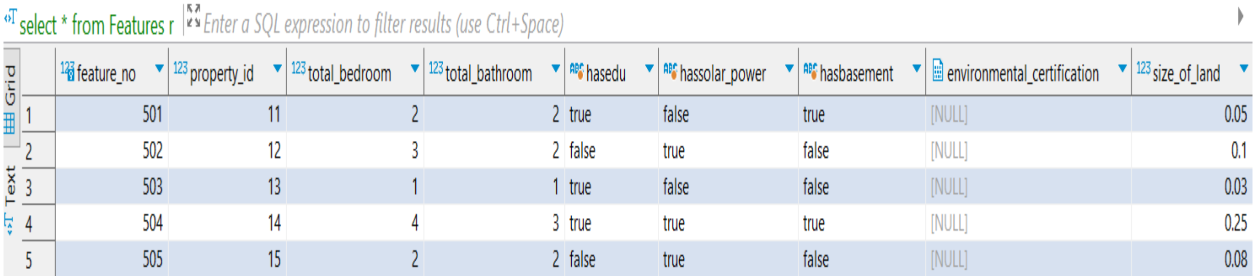


select * from Scheduler;--

Enter a SQL expression to filter results (use Ctrl+Space)

	schedule_no	customer_id	property_id	realtor_id	schedule_date	schedule_hour
1	151	111	11	211	2023-11-05	09:00:00
2	152	112	11	211	2023-11-05	11:00:00
3	153	113	12	212	2023-11-06	14:00:00
4	154	114	13	213	2023-11-06	16:00:00
5	155	115	14	214	2023-11-07	10:00:00

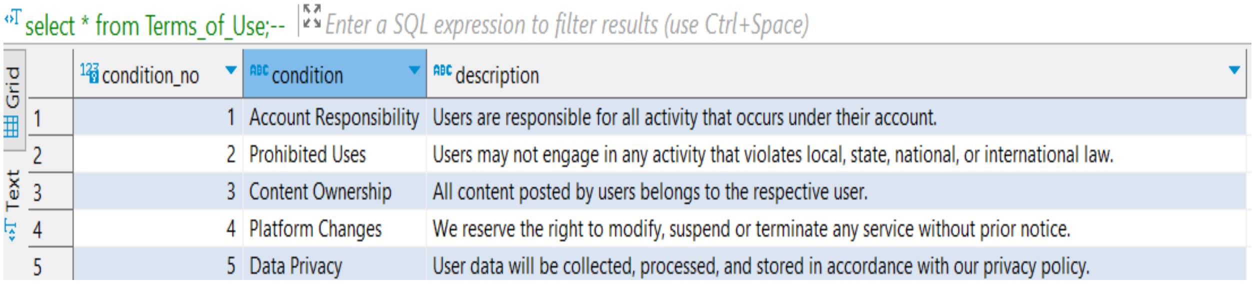
Table 5: Features



select * from Features r

	feature_no	property_id	total_bedroom	total_bathroom	hasedu	hassolar_power	hasbasement	environmental_certification	size_of_land
1	501	11	2	2	true	false	true	[NULL]	0.05
2	502	12	3	2	false	true	false	[NULL]	0.1
3	503	13	1	1	true	false	false	[NULL]	0.03
4	504	14	4	3	true	true	true	[NULL]	0.25
5	505	15	2	2	false	true	false	[NULL]	0.08

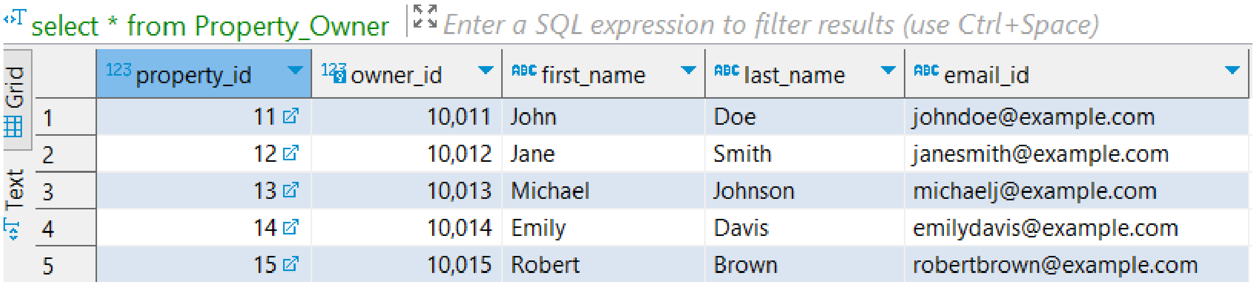
Table: Terms of Use



select * from Terms_of_Use;--

	condition_no	condition	description
1	1	Account Responsibility	Users are responsible for all activity that occurs under their account.
2	2	Prohibited Uses	Users may not engage in any activity that violates local, state, national, or international law.
3	3	Content Ownership	All content posted by users belongs to the respective user.
4	4	Platform Changes	We reserve the right to modify, suspend or terminate any service without prior notice.
5	5	Data Privacy	User data will be collected, processed, and stored in accordance with our privacy policy.

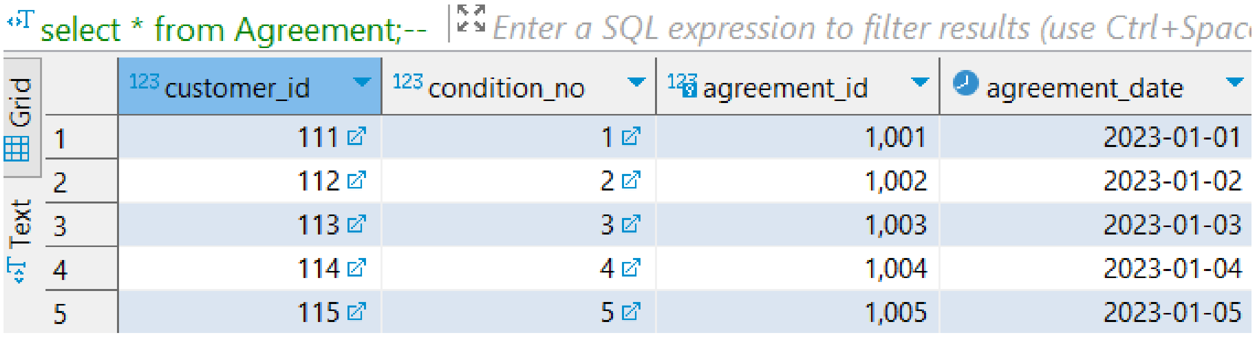
Table: Property owner contact details



select * from Property_Owner

	property_id	owner_id	first_name	last_name	email_id
1	11	10,011	John	Doe	johndoe@example.com
2	12	10,012	Jane	Smith	janesmith@example.com
3	13	10,013	Michael	Johnson	michaelj@example.com
4	14	10,014	Emily	Davis	emilydavis@example.com
5	15	10,015	Robert	Brown	robertbrown@example.com

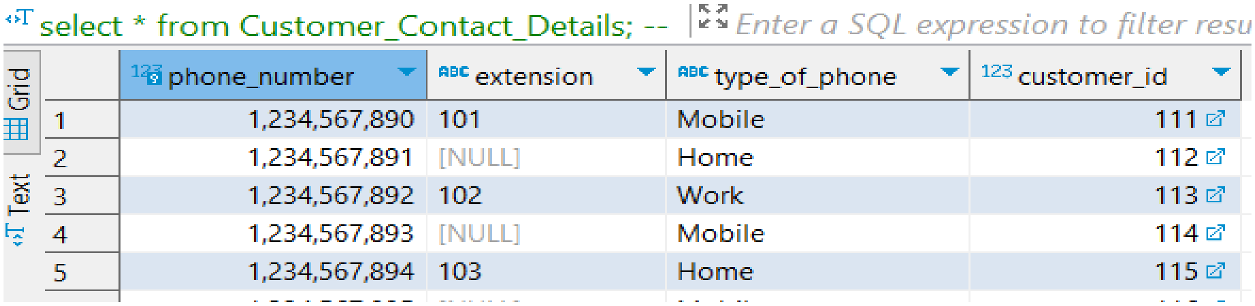
Table: Agreement



select * from Agreement;--

	customer_id	condition_no	agreement_id	agreement_date
1	111	1	1,001	2023-01-01
2	112	2	1,002	2023-01-02
3	113	3	1,003	2023-01-03
4	114	4	1,004	2023-01-04
5	115	5	1,005	2023-01-05

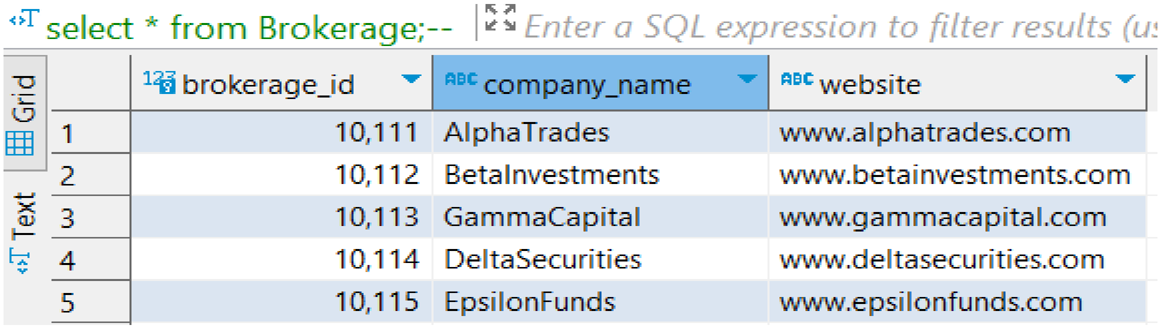
Table: Customer contact details



```
select * from Customer_Contact_Details; --
```

	phone_number	extension	type_of_phone	customer_id
1	1,234,567,890	101	Mobile	111
2	1,234,567,891	[NULL]	Home	112
3	1,234,567,892	102	Work	113
4	1,234,567,893	[NULL]	Mobile	114
5	1,234,567,894	103	Home	115

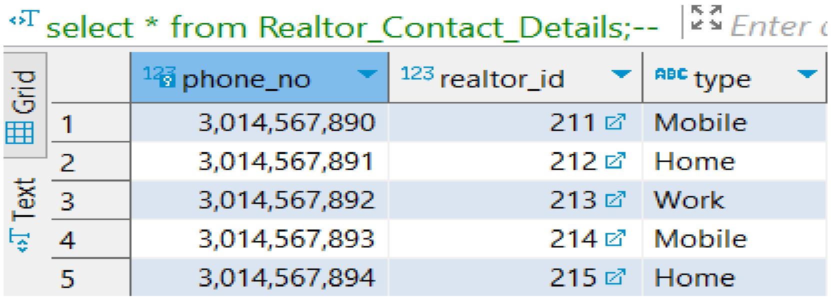
Table: Brokerage



```
select * from Brokerage;--
```

	brokerage_id	company_name	website
1	10,111	AlphaTrades	www.alphatrades.com
2	10,112	BetaInvestments	www.betainvestments.com
3	10,113	GammaCapital	www.gammacapital.com
4	10,114	DeltaSecurities	www.deltasecurities.com
5	10,115	EpsilonFunds	www.epsilonfunds.com

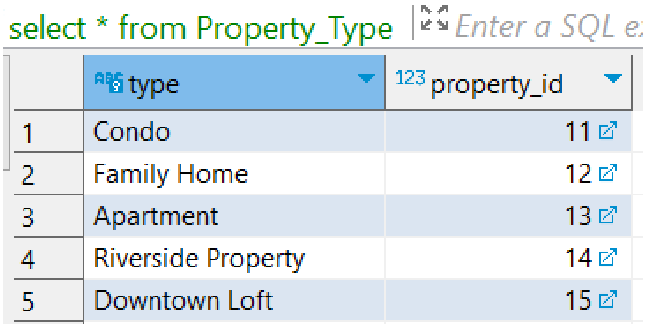
Table: Realtor contact details



```
select * from Realtor_Contact_Details;--
```

	phone_no	realtor_id	type
1	3,014,567,890	211	Mobile
2	3,014,567,891	212	Home
3	3,014,567,892	213	Work
4	3,014,567,893	214	Mobile
5	3,014,567,894	215	Home

Table: Property Type

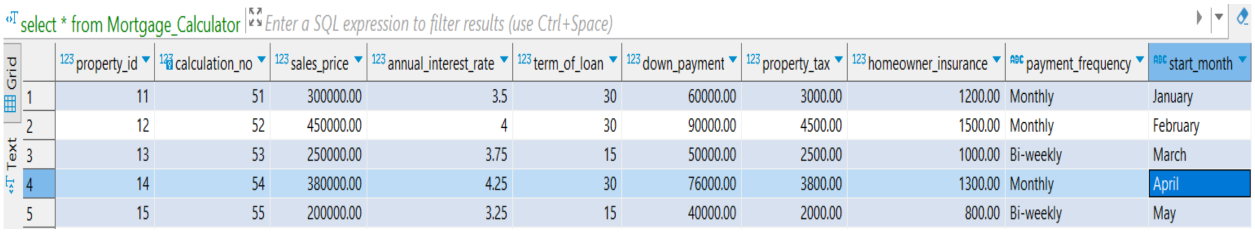


```
select * from Property_Type
```

	type	property_id
1	Condo	11
2	Family Home	12
3	Apartment	13
4	Riverside Property	14
5	Downtown Loft	15

Table: Mortgage calculator

Part_1:

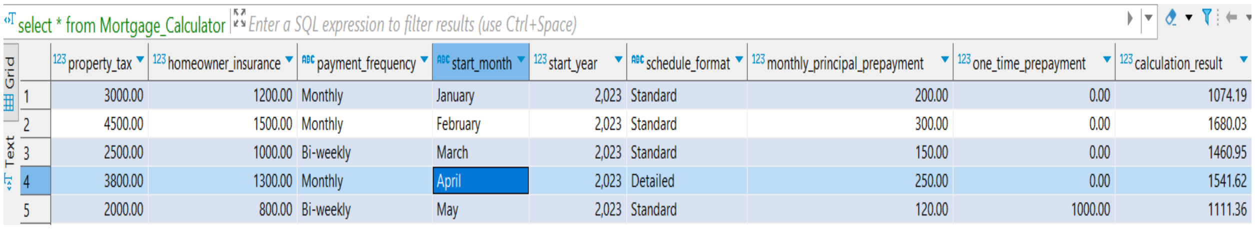


select * from Mortgage_Calculator

Enter a SQL expression to filter results (use Ctrl+Space)

	property_id	calculation_no	sales_price	annual_interest_rate	term_of_loan	down_payment	property_tax	homeowner_insurance	payment_frequency	start_month
1	11	51	300000.00	3.5	30	60000.00	3000.00	1200.00	Monthly	January
2	12	52	450000.00	4	30	90000.00	4500.00	1500.00	Monthly	February
3	13	53	250000.00	3.75	15	50000.00	2500.00	1000.00	Bi-weekly	March
4	14	54	380000.00	4.25	30	76000.00	3800.00	1300.00	Monthly	April
5	15	55	200000.00	3.25	15	40000.00	2000.00	800.00	Bi-weekly	May

Part 2:



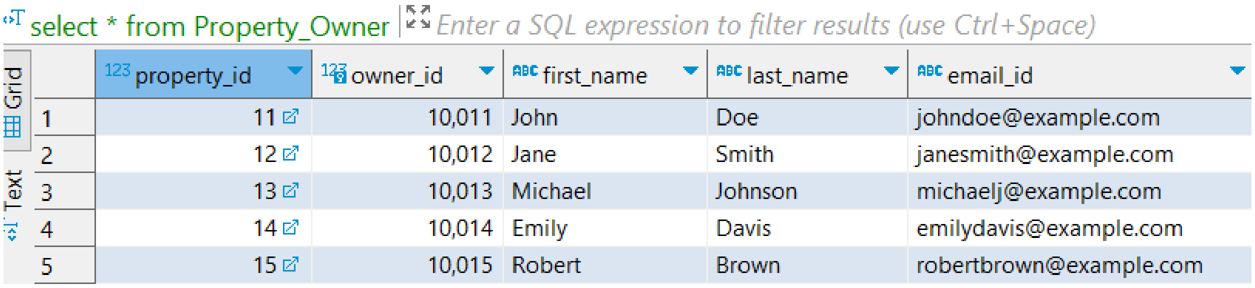
select * from Mortgage_Calculator

Enter a SQL expression to filter results (use Ctrl+Space)

	property_tax	homeowner_insurance	payment_frequency	start_month	start_year	schedule_format	monthly_principal_prepayment	one_time_prepayment	calculation_result
1	3000.00	1200.00	Monthly	January	2,023	Standard	200.00	0.00	1074.19
2	4500.00	1500.00	Monthly	February	2,023	Standard	300.00	0.00	1680.03
3	2500.00	1000.00	Bi-weekly	March	2,023	Standard	150.00	0.00	1460.95
4	3800.00	1300.00	Monthly	April	2,023	Detailed	250.00	0.00	1541.62
5	2000.00	800.00	Bi-weekly	May	2,023	Standard	120.00	1000.00	1111.36

We have split the diagram into two screenshots as all the columns could not be captured in one screenshot.

Table: Property owner

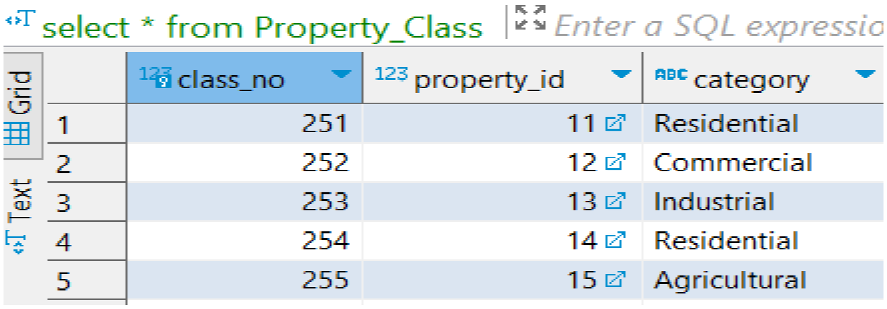


select * from Property_Owner

Enter a SQL expression to filter results (use Ctrl+Space)

	property_id	owner_id	first_name	last_name	email_id
1	11	10,011	John	Doe	johndoe@example.com
2	12	10,012	Jane	Smith	janesmith@example.com
3	13	10,013	Michael	Johnson	michaelj@example.com
4	14	10,014	Emily	Davis	emilydavis@example.com
5	15	10,015	Robert	Brown	robertbrown@example.com

Table: Property class



select * from Property_Class

Enter a SQL expressio

	class_no	property_id	category
1	251	11	Residential
2	252	12	Commercial
3	253	13	Industrial
4	254	14	Residential
5	255	15	Agricultural