|  |  |
| --- | --- |
| Project Title | **Real Estate Price Predictor** |

**1. documentation**

**2. ERD**

**3. Relational schema**

**4. Sample Data in tables**

**5. Queries & output**

**6. Views**

**7. Interface**

**Documentation**

Scenario

***Real Estate (House Price Predictor)***

**New users register on the Real Estate platform, providing their username, password, email, and specifying their user type (Buyer/Seller/Agent).**

**Users log in to the system using their credentials, initiating a session. User authentication is mandatory for all subsequent actions.**

**Users, predominantly buyers, perform property searches based on criteria such as location, budget, and features. The system records each search in the user's search history.**

**Users explore property details, including features, listing prices, and historical transactions. The system displays accurate and up-to-date information.**

**Sellers add properties to the system for selling, providing details such as address, area, bedrooms, bathrooms, features, and the desired selling price. Each property is associated with the respective seller.**

**Users, both buyers and sellers, initiate predictions for selected properties to estimate future house prices. The system uses a Linear Regression model to provide accurate predictions.**

**When buyers decide to purchase a property, they initiate a transaction. The system records the transaction details, including the property, buyer, seller, transaction date, and price.**

**Users can review their search history to track previously explored properties. The system links each entry in the search history to the corresponding user and property.**

**Sellers mark their properties as sold when transactions are completed. The system records the selling details, including the seller, property, selling price, and date.**

**Users log out of the system, ending their session. The system records the logout time in the user session history.**

**The system manages user sessions, allowing users to have multiple sessions and keeping track of login and logout times.**

**Each property is associated with features and types. Features, such as "Swimming Pool" or "Garden," are linked to properties, providing additional information to users. Properties also belong to specific types, such as "Apartment" or "House."**

**This scenario outlines a typical user journey on the Real Estate platform, incorporating actions like property search, selling, prediction, and transaction. The system's entities, attributes, and relationships ensure accurate and transparent information for users, contributing to a reliable real estate experience.**

**Real Estate (House Price Predictor) SQL Database Documentation**

**1. Introduction**

The Real Estate (House Price Predictor) SQL Database is designed to manage and organize data related to properties, users, transactions, predictions, and other entities within the context of a real estate system. The database supports functionalities such as property search, prediction of house prices, user transactions, and more.

***2. Entity-Relationship Diagram (ERD***)

User: Represents registered users of the system**.**

**Property**: Contains information about individual properties listed in the system.

**Transaction**: Records transactions between buyers and sellers.

**Prediction**: Stores predicted prices for properties.

**SearchHistory**: Keeps track of users' property search history.

**SellingProperty**: Records properties that have been sold.

**UserSession**: Manages user login and logout sessions.

**PropertyFeature**: Stores various features associated with properties.

**PropertyType**: Defines different types of properties.

3. **Tables and Attributes**

3.1 **User(**UserID (PK),Username,Password,Email,UserType)

3.2 **Property**(PropertyID (PK),Address.Area,Bedrooms,Bathroor)

3.3 **Transaction (**TransactionID (PK),PropertyID (FK),BuyerID (FK),SellerID (FK)

3.4 **Prediction (**PredictionI (PK),PropertyID(F K),PredictedPrice

PredictionDate)

3.5 **SearchHistory(**SearchID (PK) ,UserID (FK),PropertyID (FK)

SearchDate)

3.6 **SellingProperty(**SellingID (PK),UserID (FK),PropertyID (FK), SellingPrice,SellingDate)

3.7 **UserSession (**SessionID (PK),UserID (FK),LoginTime,LogoutTime)

3.8 **PropertyFeature(**FeatureID (PK),FeatureName)

3.9 **PropertyType Table(**TypeID (PK),TypeName)

3.10 **PropertyPropertyFeature (**PropertyID (FK),FeatureID FK())

3.11 **PropertyPropertyType** (PropertyID (FK),TypeID (FK) )

**4. Relationships**

• User-Property (1:N): One user can own many properties.

• User-Transaction (1:N): One user can be involved in many transactions.

• User-SearchHistory (1:N): One user can have many search history entries.

• User-SellingProperty (1:N): One user can sell many properties.

• User-UserSession (1:N): One user can have many sessions.

• Property-Transaction (1:N): One property can be involved in many transactions.

• Property-Prediction (1:N): One property can have many predictions.

• SearchHistory-Property (1:1): Each search history entry is associated with one property.

• Property-PropertyFeature (1:N): One property can have many features.

• Property-PropertyType (1:1): Each property belongs to one property type.

**Cardinality and Optionality For Each Relationship**

**User-Property (User owns Property):**

Cardinality: One User can own Many Properties (1:N)

Optionality: Mandatory (Each user must own at least one property)

**User-Transaction (User involved in Transaction):**

Cardinality: One User can be involved in Many Transactions (1:N)

Optionality: Optional (A user may not be involved in any transactions)

**User-SearchHistory (User performs property searches):**

Cardinality: One User can have Many Search History entries (1:N)

Optionality: Optional (A user may not have any search history)

**User-SellingProperty (User sells Property):**

Cardinality: One User can sell Many Properties (1:N)

Optionality: Optional (A user may not sell any properties)

**User-UserSession (User has a session):**

Cardinality: One User can have Many User Sessions (1:N)

Optionality: Optional (A user may not have any active sessions)

**Property-Transaction (Property involved in Transaction):**

Cardinality: One Property can be involved in Many Transactions (1:N)

Optionality: Optional (A property may not be involved in any transactions)

**Property-Prediction (Property has Prediction):**

Cardinality: One Property can have Many Predictions (1:N)

Optionality: Optional (A property may not have any predictions)

**SearchHistory-Property (Property is part of a search history):**

Cardinality: One Search History entry can be associated with One Property (1:1)

Optionality: Mandatory (Each search history entry must be associated with a property)

**Property-PropertyFeature (Property has features):**

Cardinality: One Property can have Many Property Features (1:N)

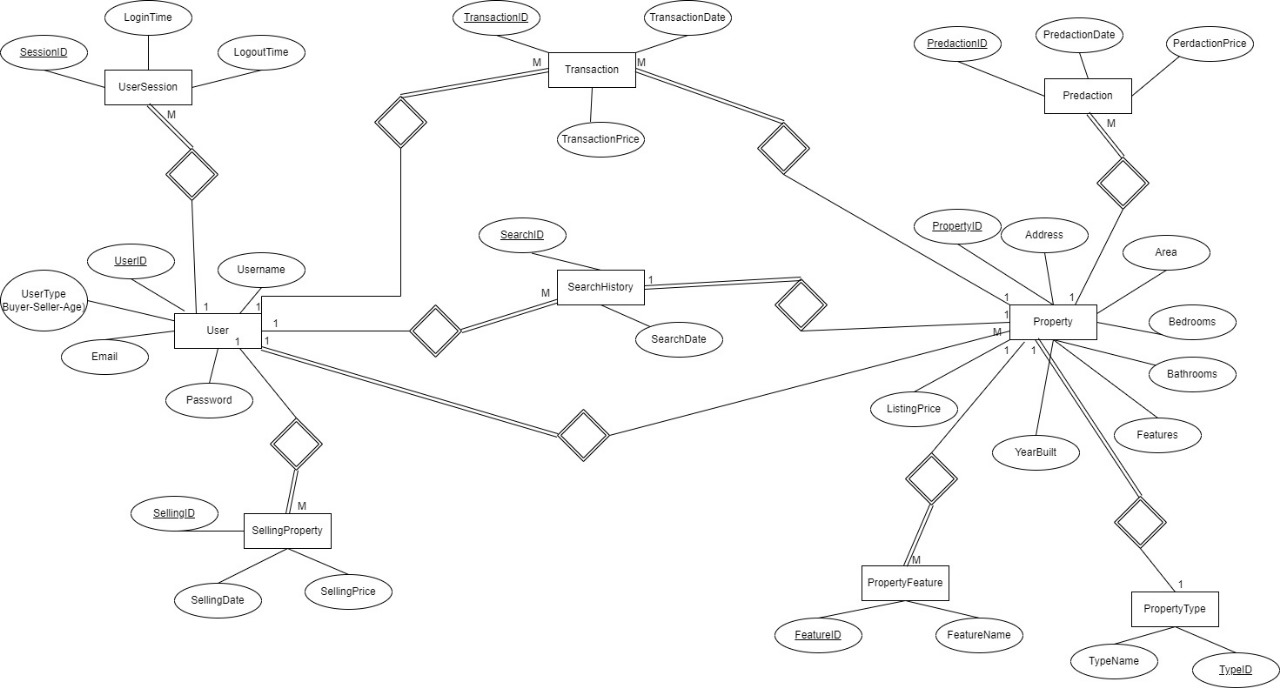
Optionality: Optional (A property may not have any features)

**Property-PropertyType (Property belongs to a type):**

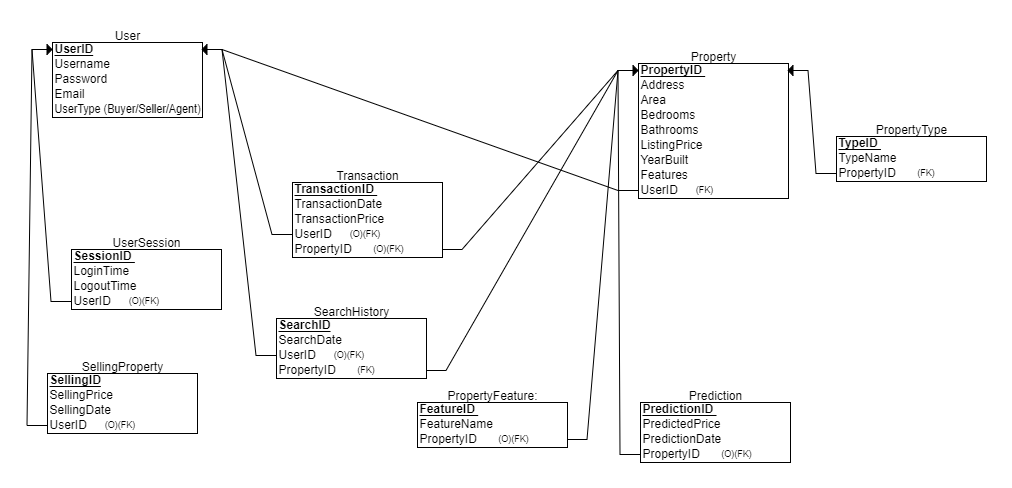
Cardinality: One Property can belong to One Property Type (1:1)

Optionality: Mandatory (Each property must belong to a type)

**ERD**

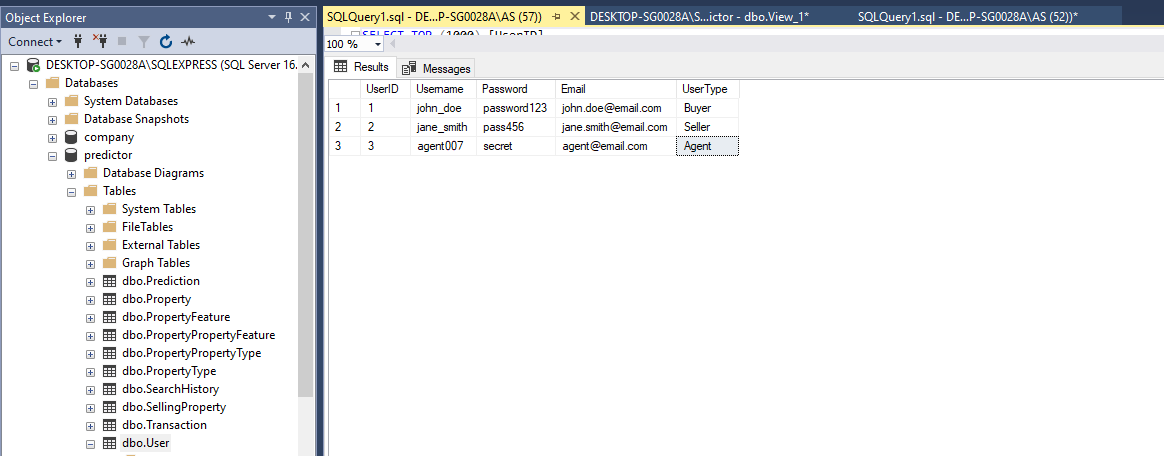
****

**Relational schema**

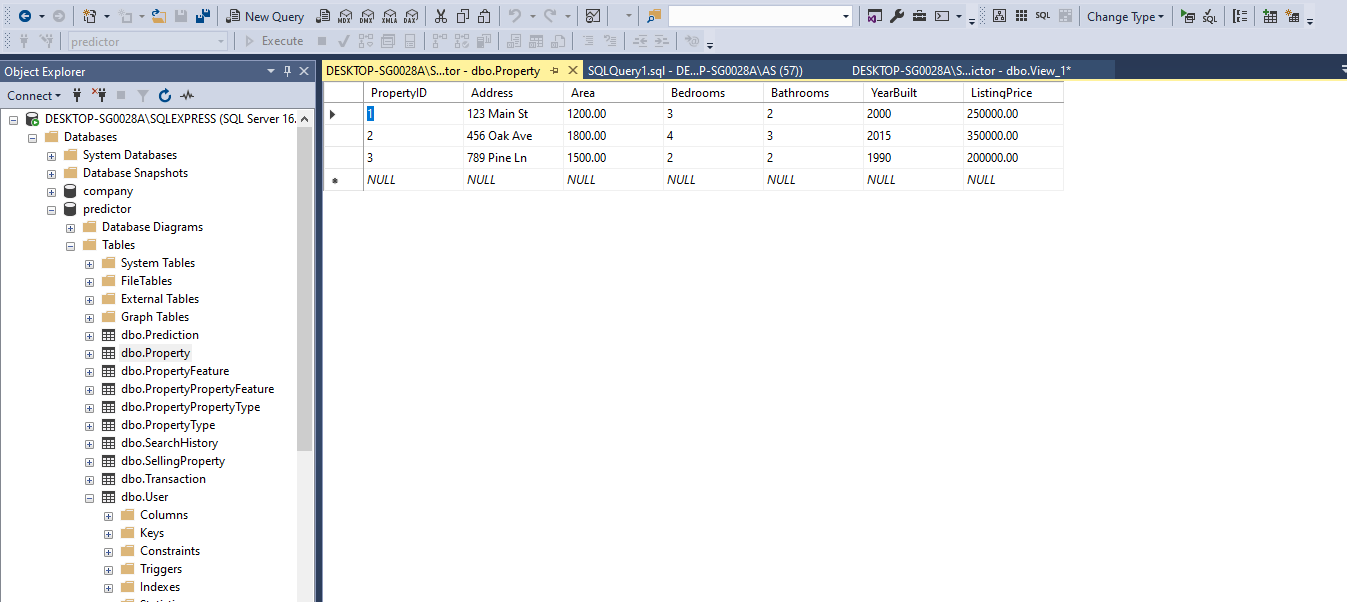
****

**Sample Data in tables**

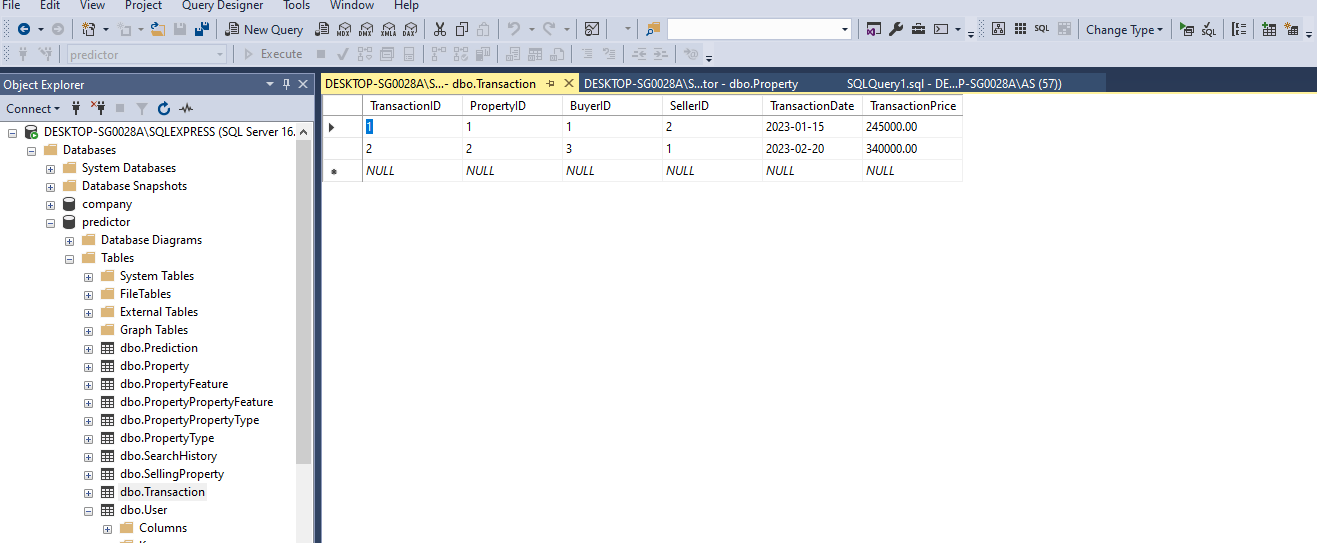
**User table**



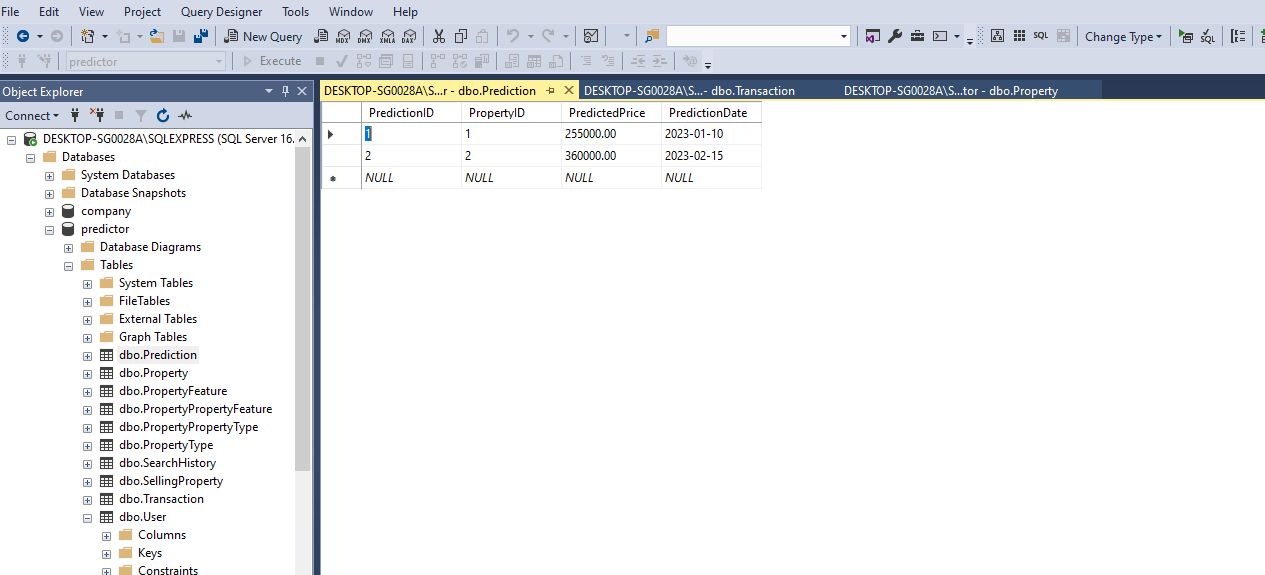
**Property table**



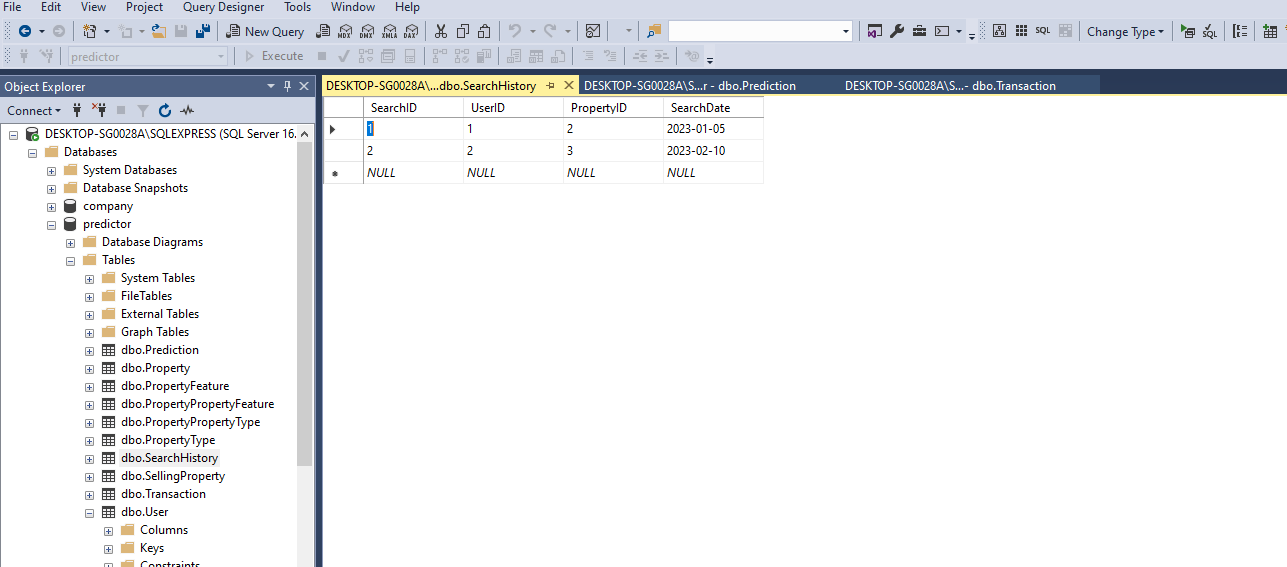
**Transaction table**



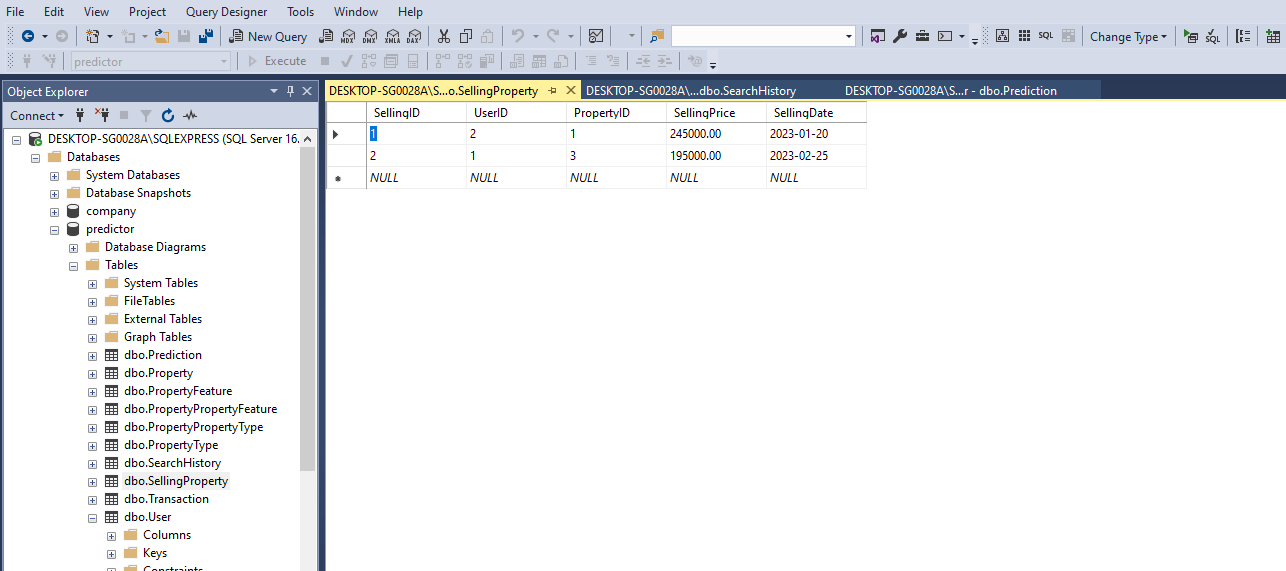
**Prediction table**



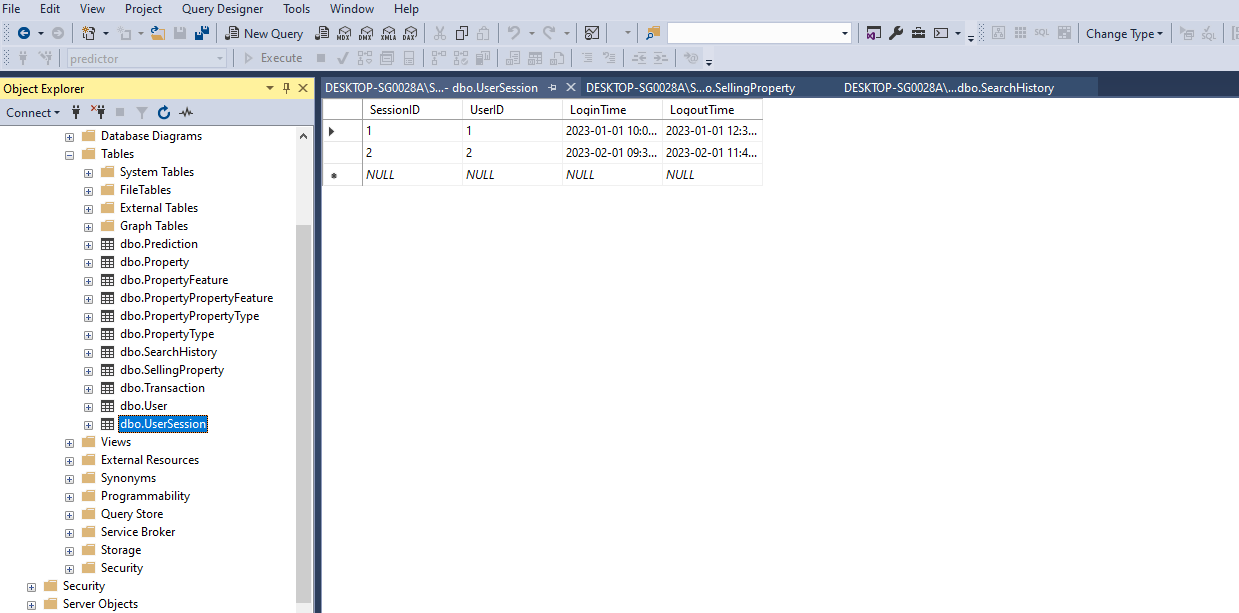
**SearchHistory table**



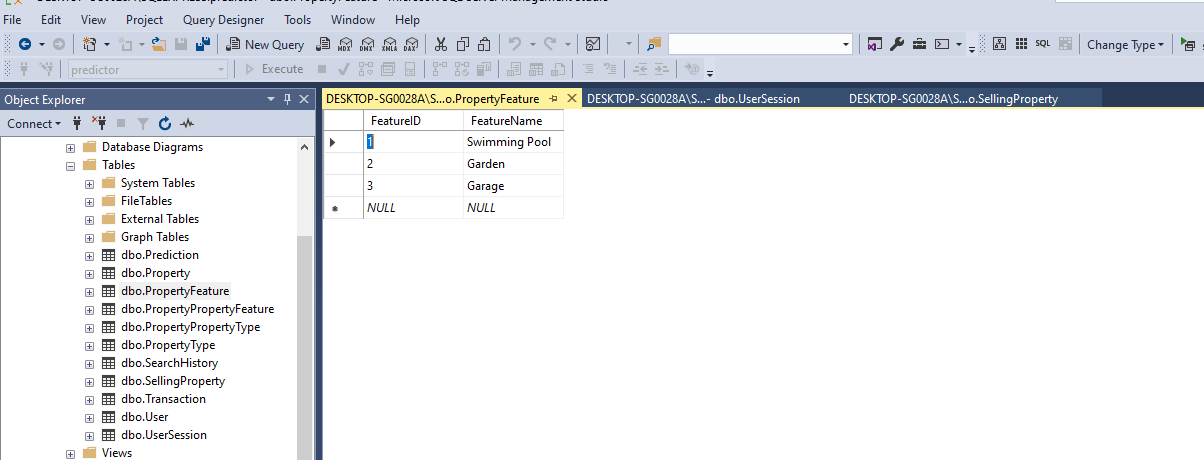
**SellingProperty table**



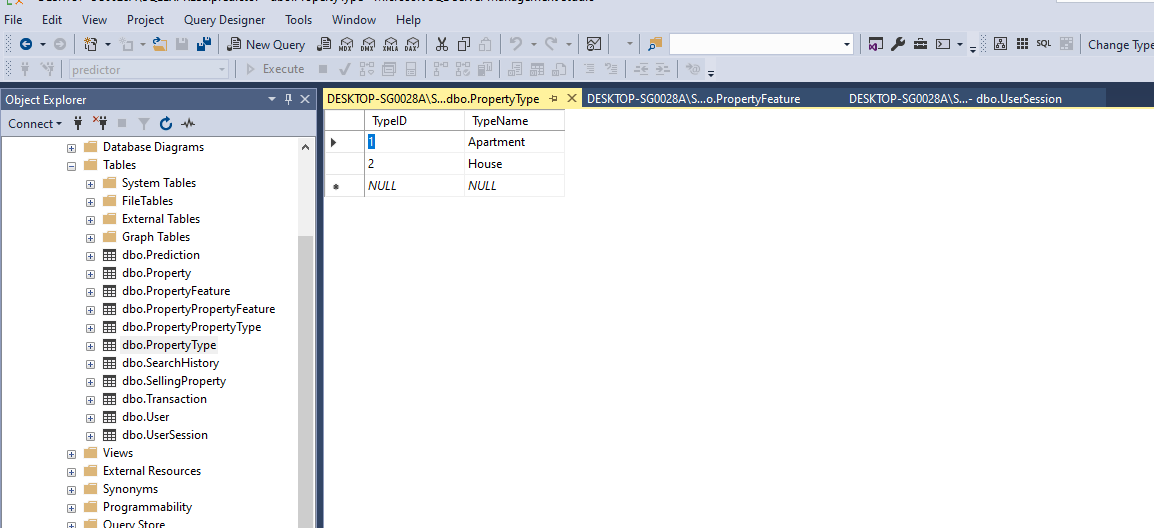
**UserSession table**



**PropertyFeature Table**

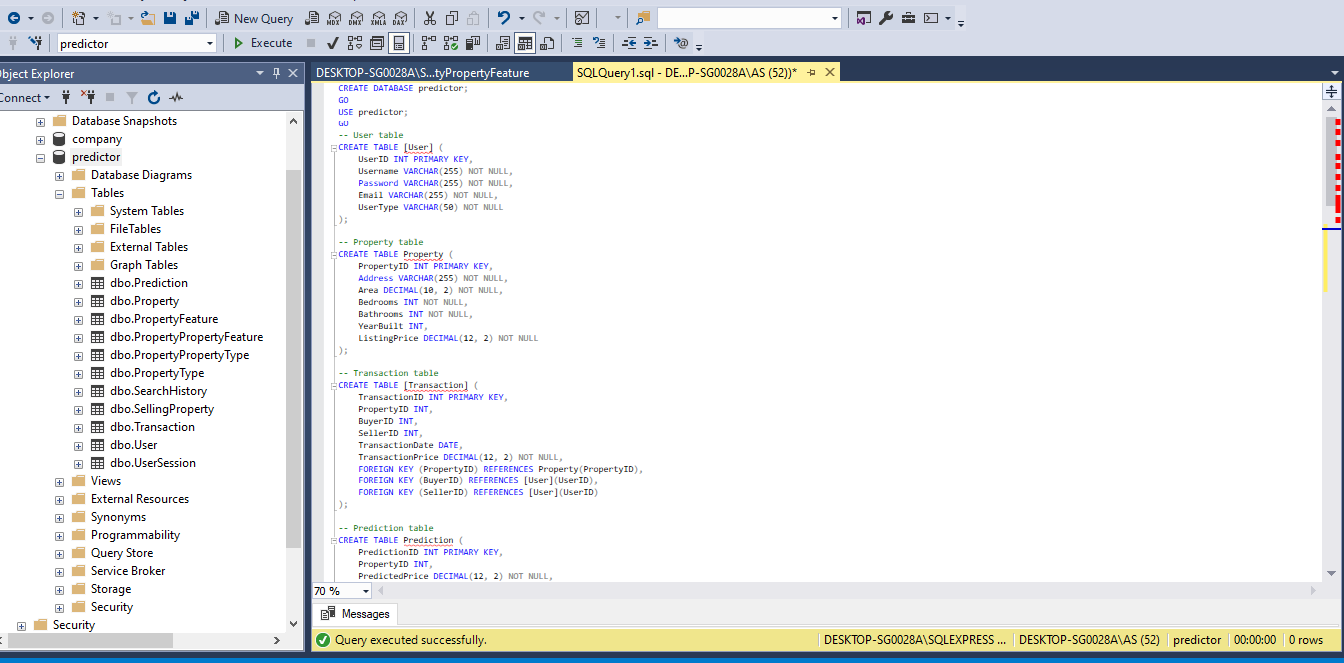


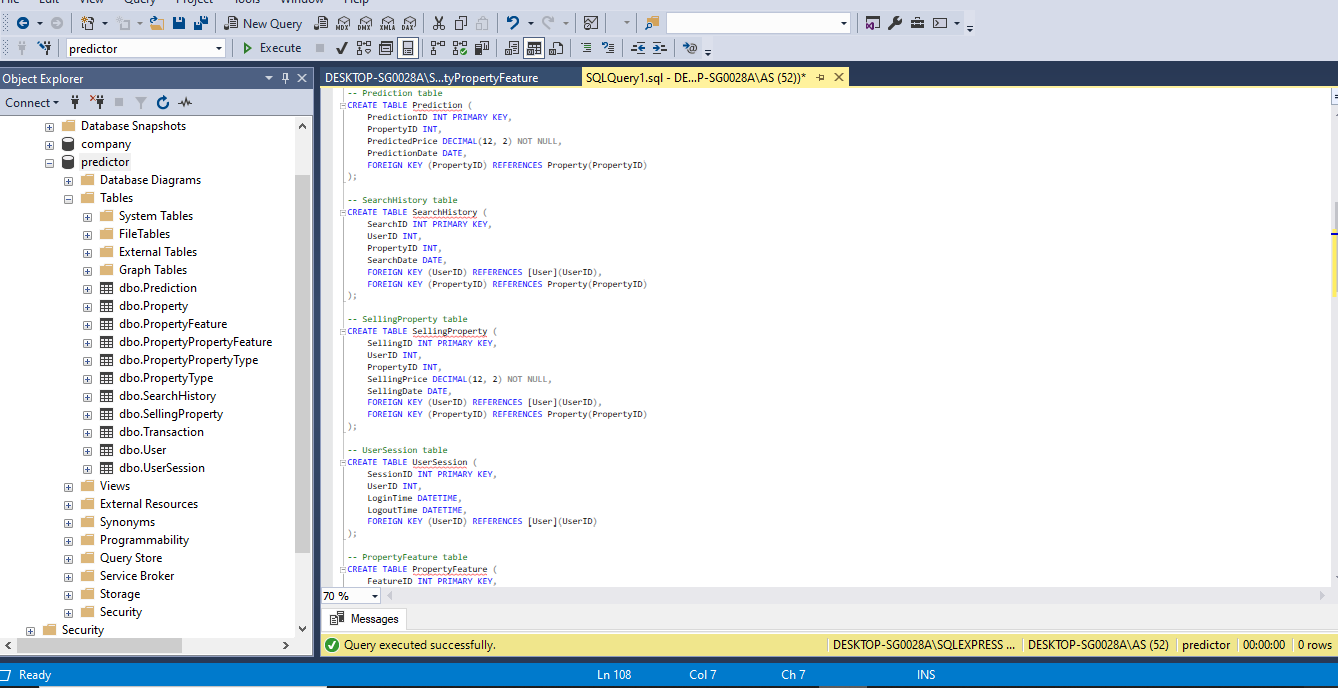
**PropertyType Table**

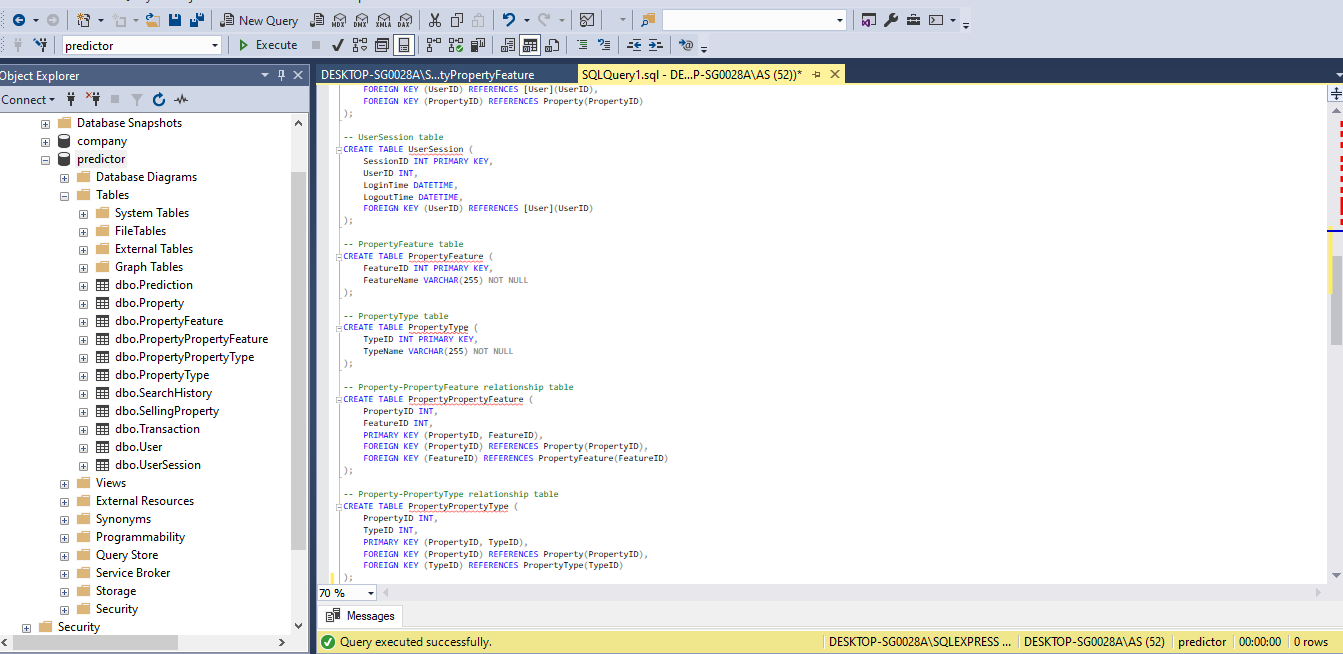


**Queries & output**

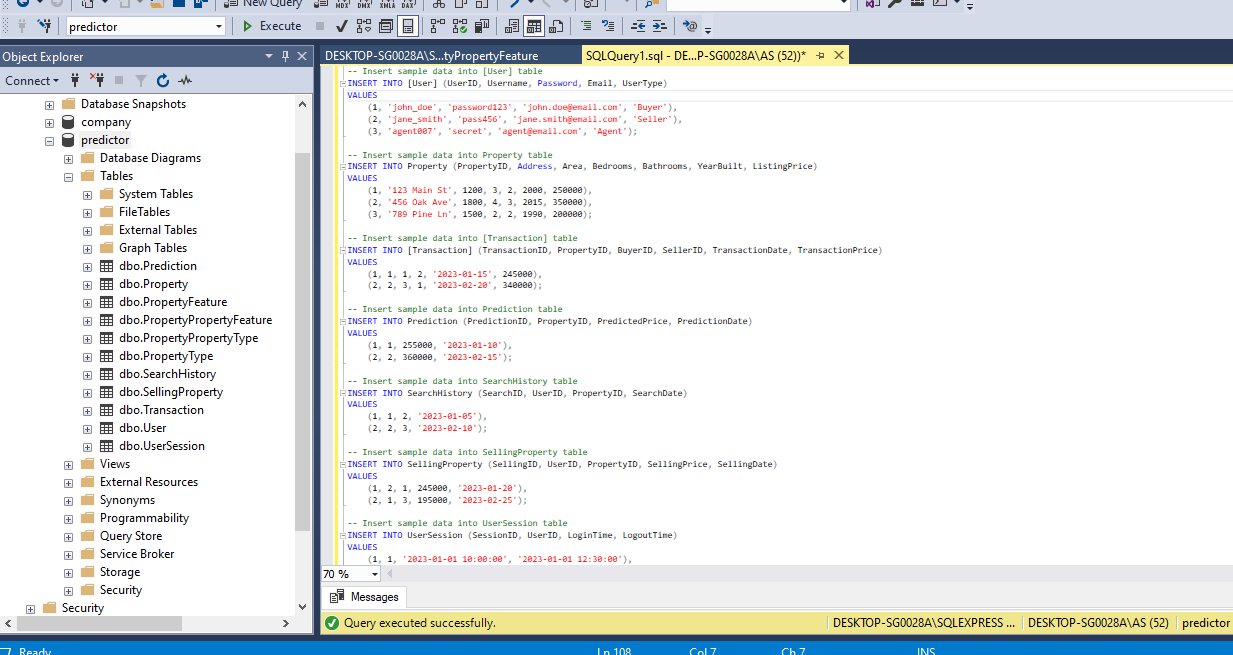
**DDL:**

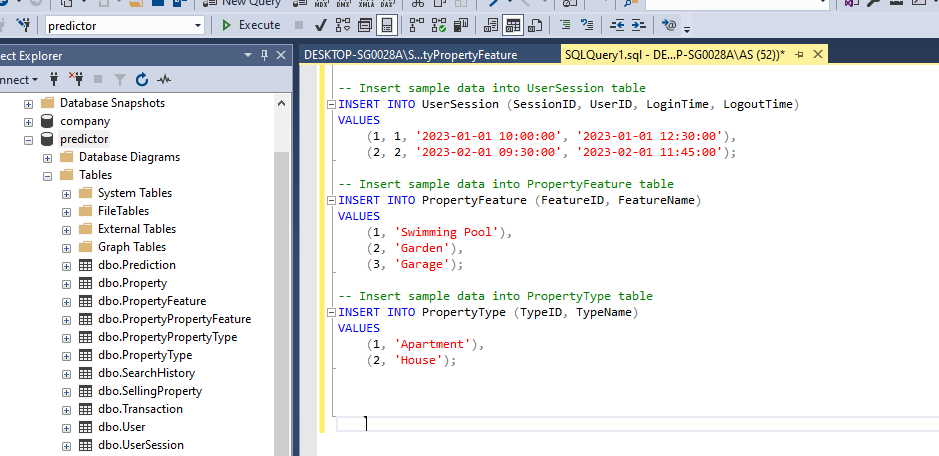




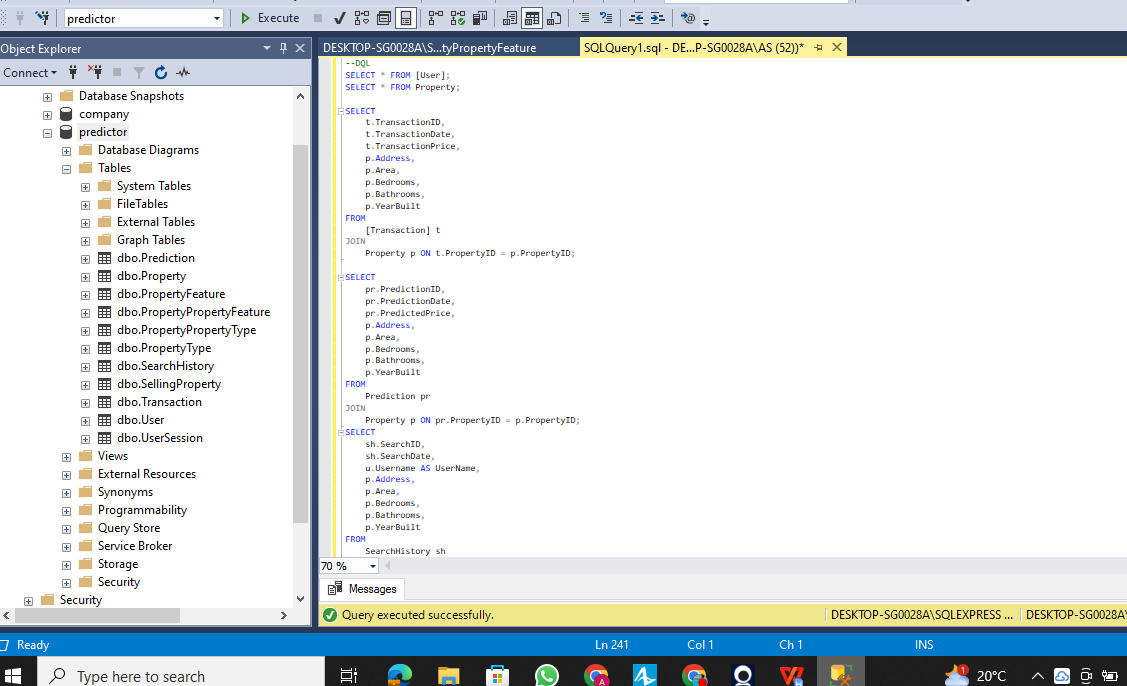


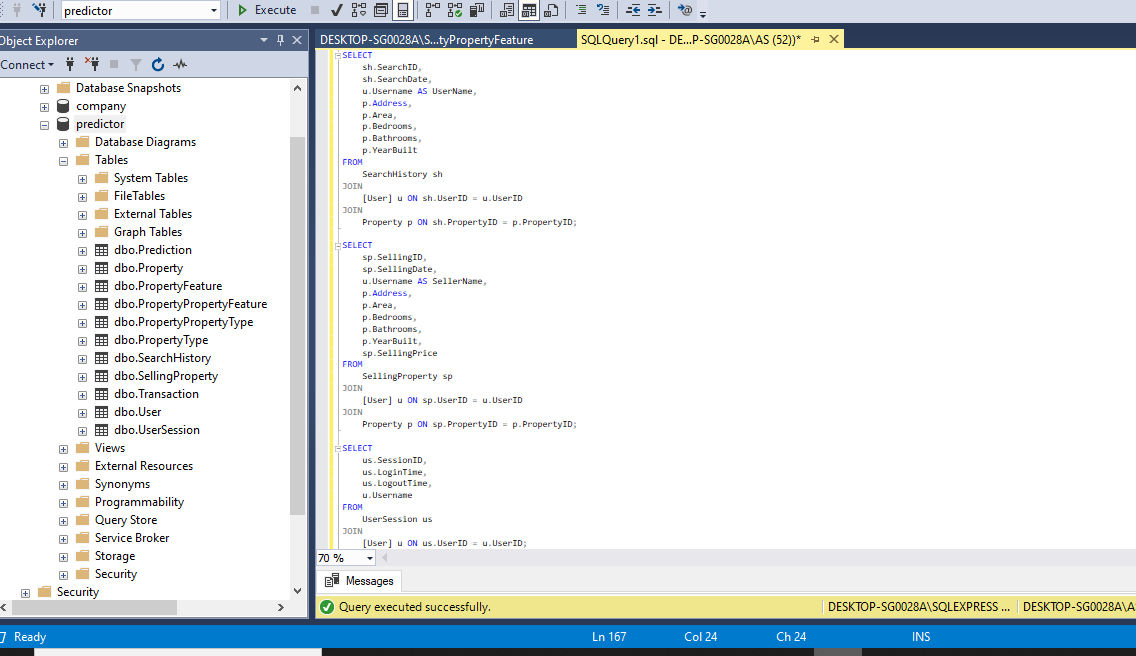
**DML:**



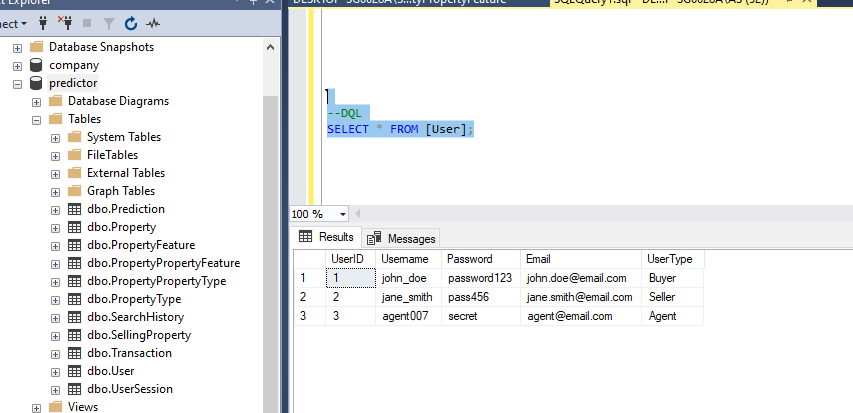


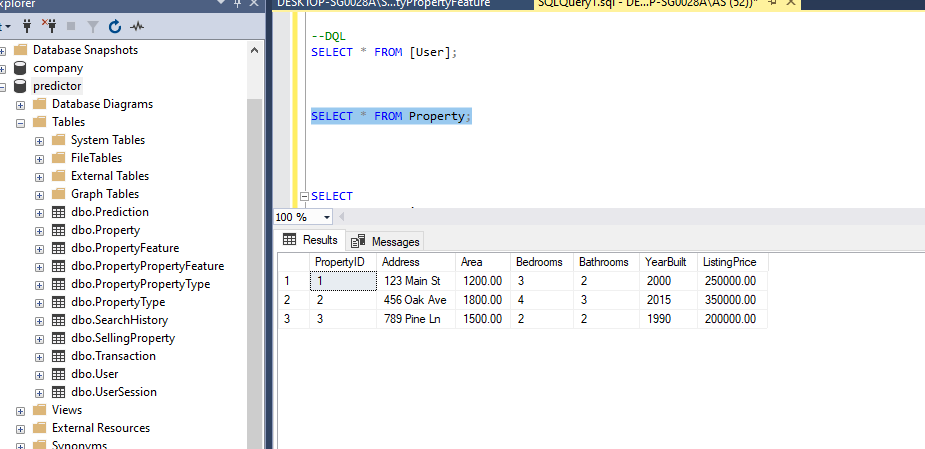
**DQL:**

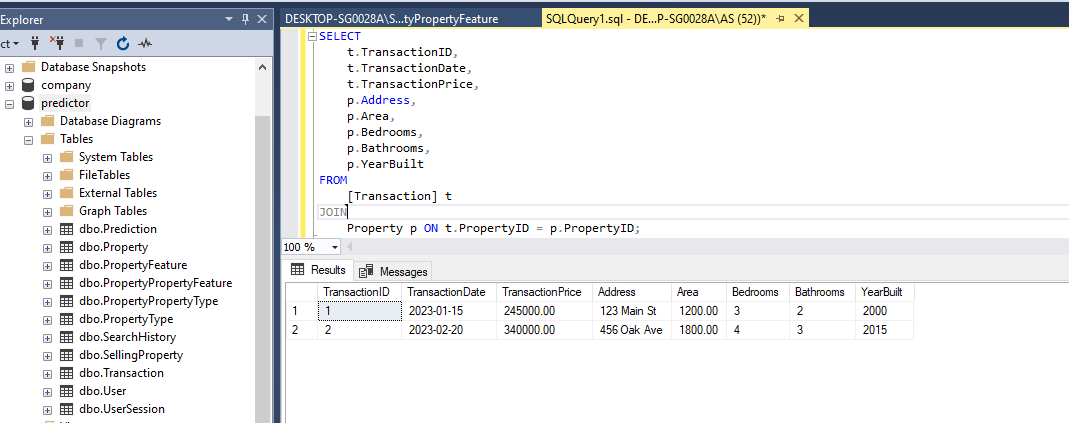


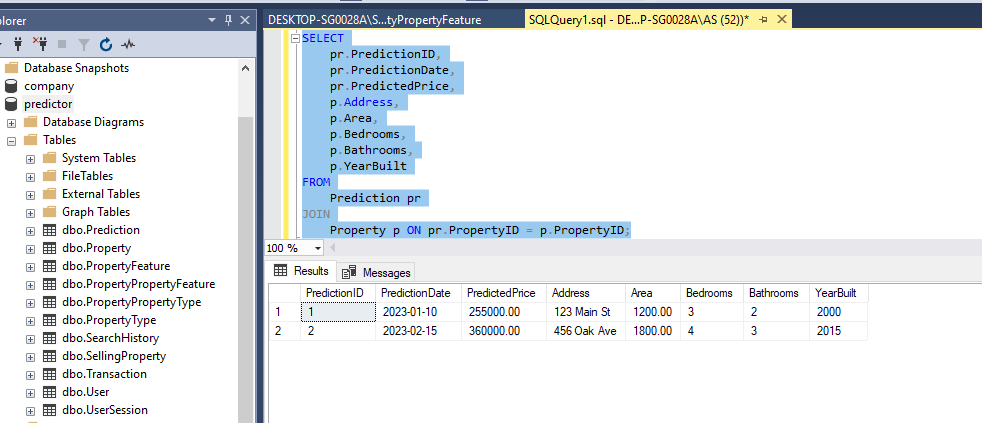


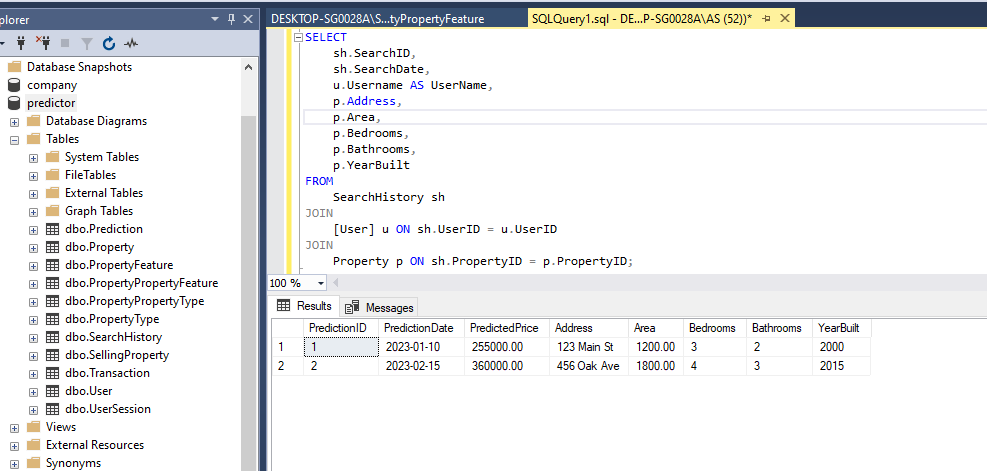
**Output**

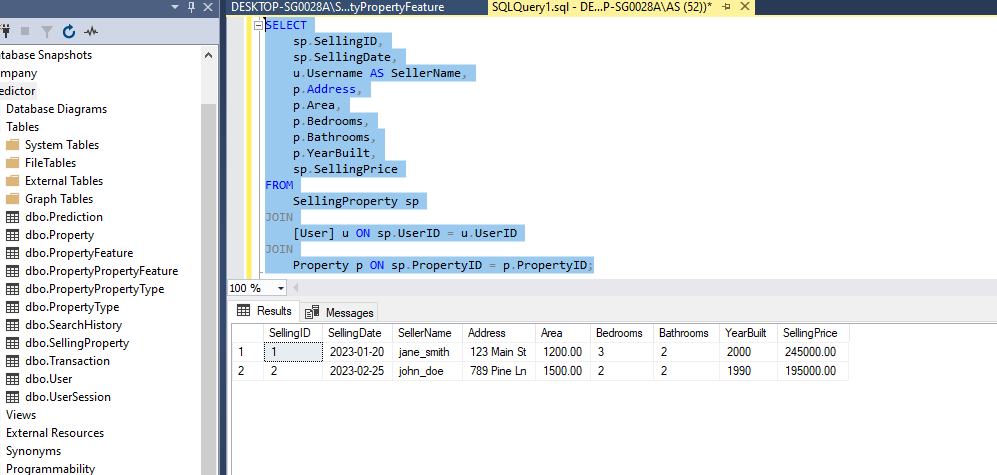


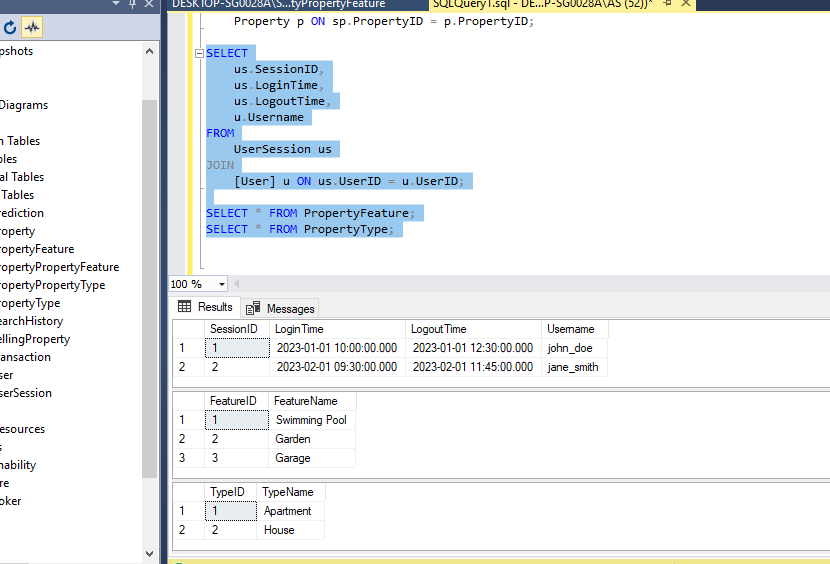






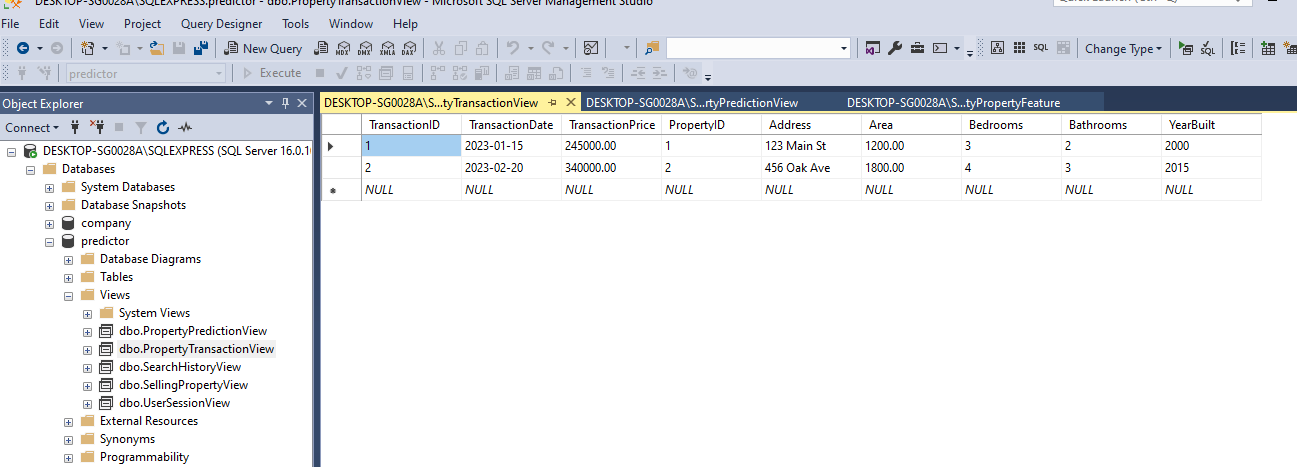




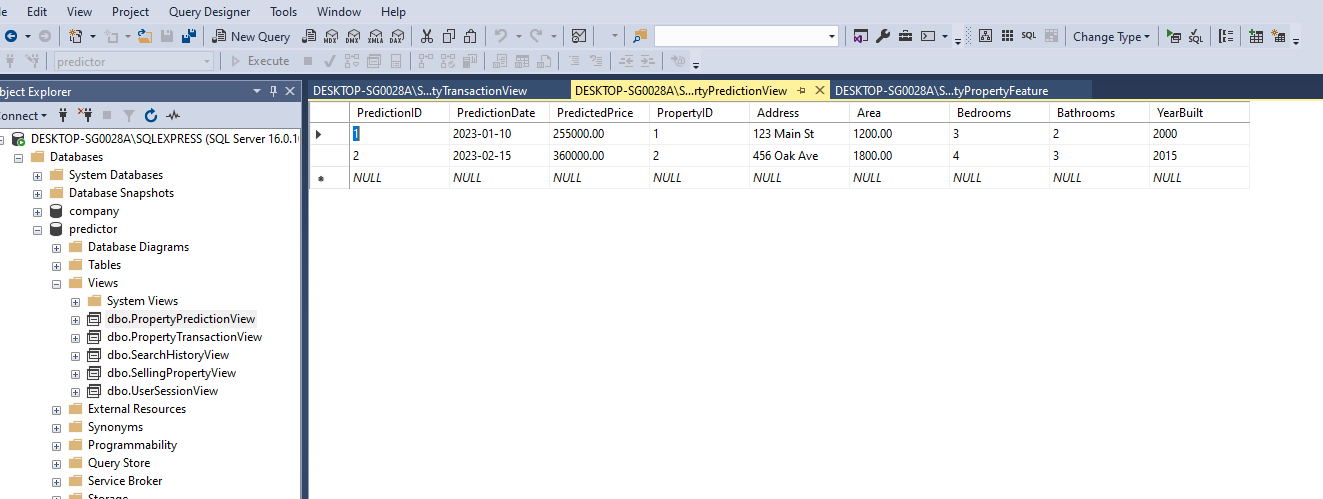


**Views**

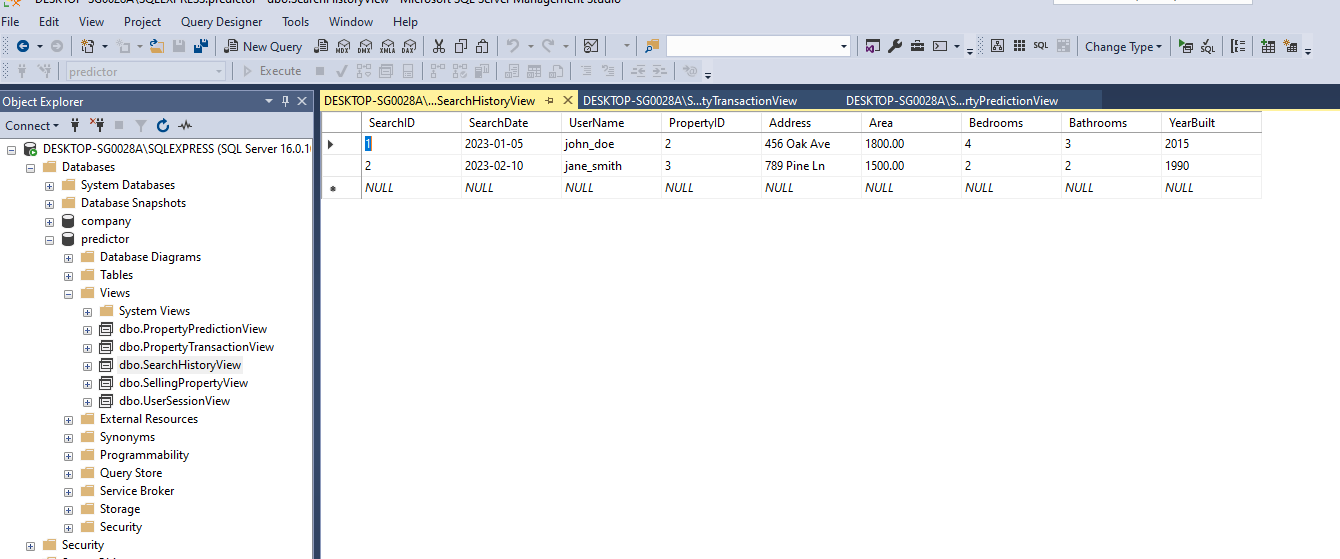
PropertyTransactionView



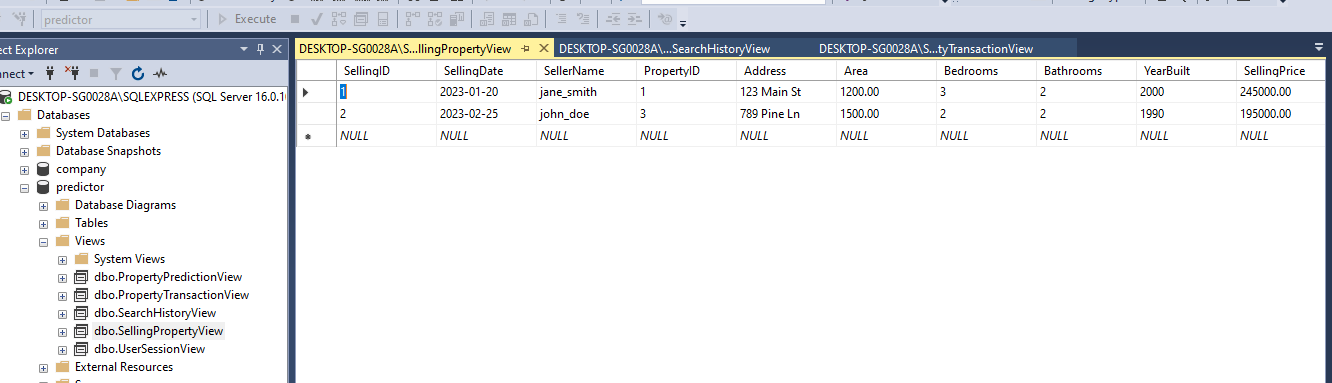
**PropertyPredictionView**



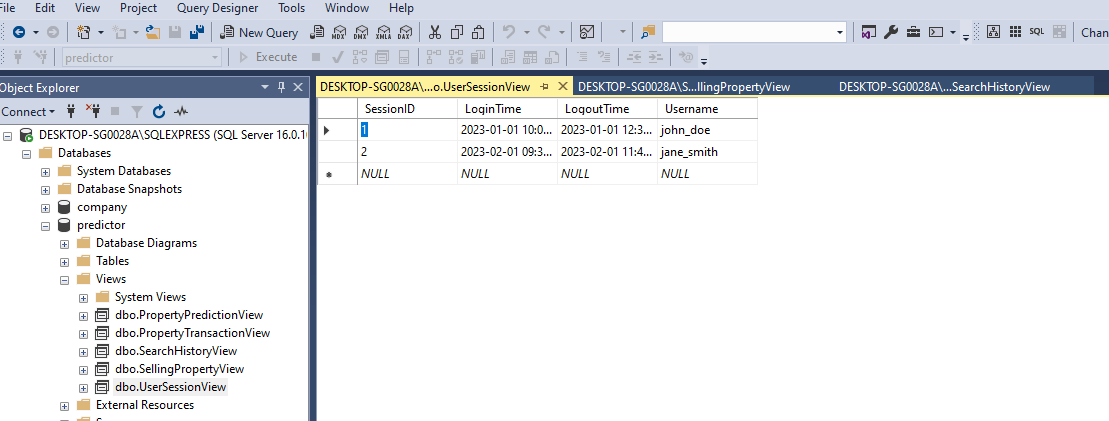
SearchHistoryView



SellingPropertyView



UserSessionView



**Interface**

