A close up of a logo

Description automatically generated

Listing Management on

OrangeHousing.com

IST 659 –Project Implementation Report

Anishka Tahiliani

Prathamesh Datar

Project Summary

OrangeHousing.com has a focal significance to personal satisfaction with extensive monetary, social, and individual importance. Orange Housing is in the rental industry that has become pivotal in Syracuse. People from different social backgrounds, status, age ranging from students to people on vacation look for rental houses in Syracuse. The postings which are listed on the website (OrangeHousing.com) and based on the specifications and predilections to one’s liking. To manage one’s likings, there are filters for e.g. price preference, pet friendly houses, number of bedrooms; there is a vivid range of selection, which gives power to the users to select and match an individual with a room/ apartment choice.

The function of Orange Housing is to connect the people in Syracuse to rent an apartment, give flexibility to the user and the landlord; a 360-degree overview of the property. Orange Housing only operates and facilitates the Syracuse community and is purely a website to maintain integrity. In a similar fashion, the users should have a precise view of the property, which gives an edge for data management of historic data to find the interesting patterns in already sold out apartments.

OrangeHousing.com operates in such a way that it facilitates listings on its websites. Now, whenever a lister removes a listing from his/her side then, it cannot be accessed on the website by the user. If a user spends copious time shortlisting for a house based on multiple criteria’s and abruptly finds the listing not available, then it’s a big problem for the user as he’ll have to redo the entire process of shortlisting a house again. Therefore, by providing a historic database we can solve this issue. Every user is aware to the total visits on a every listing. Secondly, with the way the listings are listed i.e. the filters, for e.g. ‘Bed’ filter has many options; 3BHK, 4BHK, 5BHK, etc. Therefore, enhancing the OrangeHousing.com feature filter just for 3BHK. So, we could design a database that will take care of all possible filter cases for all listings. Thirdly, the site does not that not many filters are available in general. Currently, OrangeHousing offers filters consisting of Beds, Pets, and Furnished Status. We can include many other filters as well which might help the user in shortlisting the house. On the website, under the landlord listing section, one can find several listed landlords but can’t find the listings associated with them. To look for listings associated with landlords one must go through listings first to find the landlord for whom we are looking for listings. But by adding a Landlord filter would surely increase the efficiency of user experience.

The problem of listings not available can be tackled by creating a database that replicates the database of OrangeHousing. Through this historic database, we can know whether a listing is sold out or not. Another main advantage of this solution is that a user can track the Rent Price Trend over the years and decide for themselves whether to rent the house or not. We have created an attribute which will include the number of visits on the site and on what time. Thus, giving an advantage of how popular a particular property is to quickly finalize the house. We have tackled the of lack of filters and inconsistent filters by generating listing consistent filters which further adds features to OrangeHousing. Through landlord listing, we can see all possible listings of a landlord without any hassle. Currently, we can’t create an account on OrangeHousing and therefore can’t save favourite listings. We will create a User ID to facilitate these favourites.

It is important to take into consideration that the number of listings should always be constantly updated for the user and the landlord giving them a lucidity about the range of houses available and the areas based on the price are the hottest in the market.

Entities and Attributes

|  |  |
| --- | --- |
| **Entities and Attributes** | **Description** |
| **Landlord** | Stores the details of the landlord |
| Landlord ID | Unique key given to the landlord |
| Landlord First Name | First name of the landlord |
| Landlord Last Name | Last Name of the landlord |
| Landlord Mobile Number | Mobile Number of the landlord |
| Landlord Email ID | Email address of the landlord |
| **User** | Users who log onto the website |
| User ID | Unique key of the user |
| User Nationality | User nationality to gain additional info of the user |
| User First name | First name of the User |
| User Last Name | Last name of the user |
| User EmailID | Email address of the user |
| User Mobile Number | Mobile Number of the user |
| **Property** | Property posted by landlords on the website |
| Property Apartment No | Unique key to Property for listing the property |
| Property Street | Primary Key to Property for the address of the property |
| Property Zip | Unique key to property table to further justify the address of the property |
| Landlord ID | Foreign Key of the Property table for the landlord to add Property records |
| **FavoriteProperty** | Favorites Added by the users |
| User ID | Primary key and foreign key of FavoriteProperty |
| Property Apartment No | Primary Key & Foreign Key from property table for the address of the property |
| Property Street | Primary Key & Foreign Key from property table for the address of the property |
| Property Zip | Primary Key & Foreign Key from property table for the address of the property |
| FavoriteDate Time | Primary Key of FavoriteProperty |
| **Lease** | The unique lease certificate given to mark the sold identity |
| Lease Number | Unique number for lease to know if the property has been taken |
| Property Apartment No | Primary Key & Foreign Key from property table for the address of the property |
| Property Street | Primary Key & Foreign Key from property table for the address of the property |
| Property Zip | Primary Key & Foreign Key from property table for the address of the property |
| Landlord ID | Primary and foreign key from landlord table |
| User ID | Primary and foreign key from User table |
| Security Deposit | Not a necessary attribute but to gain additional information on the lease |
| Rent per bedroom | The rent acquired per bedroom, to gain information on how much the user is paying for it |
| Lease Issue Date | The date of issue given for the lease to confirm the property has been sold |
| Lease Start Date | The start date of the lease |
| Lease End date | End date given for the lease when issued. To indicate when the house would next be available |
| **PropertyType** | Property posted by landlords on the website |
| Property Apartment No | Primary Key & Foreign Key from property table for the address of the property |
| Property Street | Primary Key & Foreign Key from property table for the address of the property |
| Property Zip | Primary Key & Foreign Key from property table for the address of the property |
| Property TypeID | Unique Primary Key of the PropertyType table |
| Property TypeName | Defines the type of the property eg a row house/apartment |
| TypeDesc | Description of the Property Type |
| **Property Detail** | Detail of the Property |
| Property Apartment Number | Primary Key & Foreign Key from property table for the address of the property |
| Property Street | Primary Key & Foreign Key from property table for the address of the property |
| Property Zip | Primary Key & Foreign Key from property table for the address of the property |
| DetailID | Primary Key & foreign key fetched from detail Table |
| DetailValue | The value of the detail mentioned of the property |
| DetailDesc | Description of the Detail Type |
| **Detail** | Detail of the Property |
| DetailID | Unique Key of Detail Table |
| Detail Name | The name of the detail mentioned |
| **PropertyView** | Views by the user and landlords on webpage |
| User ID | Primary and foreign Key to the hits table |
| Property Apartment Number | Primary Key & Foreign Key from property table for the address of the property |
| Property Street | Primary Key & Foreign Key from property table for the address of the property |
| Property Zip | Primary Key & Foreign Key from property table for the address of the property |
| View Date Time | Primary key to property View to mark the timestamp of the user visiting the property |

Relational Data Model



Data Dictionary

***UserInfo Entity:* This entity captures the user details of the one on the Housing Rental based system**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity Name** | **Attribute Name** | **Field Type** | **NULL or NOT NULL** | **Description** |
| Primary Key | UserID | Varchar(10) | NOT NULL | Unique identifier for each user in the system |
| Other Attributes | UserFirstName  UserLastName  UserEmailID | Varchar(30) | NOT NULL | Information of the user |
| Other Attributes | UserNationality  UserMobile Number | Varchar(20) | NOT NULL | Information of the user |

***Landlord Entity:* This entity captures the landlord details of the one on the Housing Rental based system**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity Name** | **Attribute Name** | **Field Type** | **NULL or NOT NULL** | **Description** |
| Primary Key | LandlordID | Varchar(10) | NOT NULL | Unique identifier for each landlord in the system |
| Other Attributes | LandlordFirstName  LandlordLastName  LandlordEmailID | Varchar(30) | NOT NULL | Information of the landlord |
| Other Attributes | LandlordMobile Number | Varchar(20) | NOT NULL | Information of the user |

***Lease Entity*: This entity captures the leasing details of the Housing Rental based system**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity Name** | **Attribute Name** | **Field Type** | **NULL or NOT NULL** | **Description** |
| Primary Key | Lease Number | Integer | NOT NULL | Unique identifier for each lease created in the system |
| Primary Key and Foreign Key | Property Apartment Number | Varchar(10) | NOT NULL | Foreign Key fetched from Property Table and Unique identifier to Lease |
| Primary Key and Foreign Key | Property Street | Varchar(30) | NOT NULL | Foreign Key fetched from Property Table and Unique identifier to Lease |
| Primary Key and Foreign Key | Property Zip | Varchar(5) | NOT NULL | Foreign Key fetched from Property Table and Unique identifier to Lease |
| Primary Key and Foreign Key | Landlord ID | Varchar(30) | NOT NULL | Foreign Key fetched from Landlord Table and Unique identifier to Lease |
| Primary Key and Foreign Key | UserID | Varchar(10) | NOT NULL | Foreign Key fetched from User Table and Unique identifier to Lease |
| Other Attributes | SecurityDeposit  RentPerBedroom | Integer | NOT NULL | Information added to the Lease table |
| Other Attributes | LeaseIssueDate  LeaseStartDate  LeaseEndDate | DateTime | NOT NULL | Information added to know the date of the lease start and end |

***Property Entity:* This entity captures the listings up for rent on the Housing Rental based system**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity Name** | **Attribute Name** | **Field Type** | **NULL or NOT NULL** | **Description** |
| Primary Key | Property Apartment No | Varchar(30) | NOT NULL | Primary Key to property table |
| Primary Key | Property Street | Varchar(20) | NOT NULL | Primary Key to property table |
| Primary Key | Property Zip | Numeric(5) | NOT NULL | Primary Key to property table |
| Foreign key | Landlord Id | Varchar(10) | NOT NULL | Foreign key fetched from Landlord table to enter/delete properties |

***FavoriteProperty Entity:* This entity captures the users favorite listings details on the Housing Rental based system**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity Name** | **Attribute Name** | **Field Type** | **NULL or NOT NULL** | **Description** |
| Primary Key | FavoriteDateTime | DateTime | NOT NULL | Unique identifier timestamp for each time a user marks a property a favorite |
| Primary Key and Foreign Key | Property Apartment No | Varchar(30) | NOT NULL | Foreign Key fetched from Property Table and Unique identifier to FavoriteProperty |
| Primary Key and Foreign Key | Property Street | Varchar(20) | NOT NULL | Foreign Key fetched from Property Table and Unique identifier to FavoriteProperty |
| Primary Key and Foreign Key | Property Zip | Numeric(5) | NOT NULL | Foreign Key fetched from Property Table and Unique identifier to FavoriteProperty |
| Primary Key and Foreign Key | UserID | Varchar(10) | NOT NULL | Foreign Key fetched from User Table and Unique identifier to FavoriteProperty |

***PropertyView Entity:* This entity captures the users viewed listings details on the Housing Rental based system**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity Name** | **Attribute Name** | **Field Type** | **NULL or NOT NULL** | **Description** |
| Primary Key | ViewDateTime | DateTime | NOT NULL | Unique identifier timestamp for each time a user views a property |
| Primary Key and Foreign Key | Property Apartment No | Varchar(10) | NOT NULL | Foreign Key fetched from Property Table and Unique identifier to PropertyView |
| Primary Key and Foreign Key | Property Street | Varchar(30) | NOT NULL | Foreign Key fetched from Property Table and Unique identifier to PropertyView |
| Primary Key and Foreign Key | Property Zip | Numeric(5) | NOT NULL | Foreign Key fetched from Property Table and Unique identifier to PropertyView |
| Primary Key and Foreign Key | UserID | Varchar(10) | NOT NULL | Foreign Key fetched from User Table and Unique identifier to PropertyView |

***PropertyType Entity:* This entity captures the type and description of property details on the Housing Rental based system**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity Name** | **Attribute Name** | **Field Type** | **NULL or NOT NULL** | **Description** |
| Primary Key | PropertyTypeID | Integer | NOT NULL | Unique identifier timestamp for each type of property |
| Primary Key and Foreign Key | Property Apartment No | Varchar(10) | NOT NULL | Foreign Key fetched from Property Table and Unique identifier to PropertyType |
| Primary Key and Foreign Key | Property Street | Varchar(30) | NOT NULL | Foreign Key fetched from Property Table and Unique identifier to PropertyType |
| Primary Key and Foreign Key | Property Zip | Numeric(5) | NOT NULL | Foreign Key fetched from Property Table and Unique identifier to PropertyType |
| Other Attributes | Property Type Name | Varchar(15) | NOT NULL | Property Name of the property |
| Other Attributes | Property Type Desc | Varchar(150) | NOT NULL | The description of the property mentioned by the landlord |

***Detail Entity:* This entity captures the detail of property on the Housing Rental based system**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity Name** | **Attribute Name** | **Field Type** | **NULL or NOT NULL** | **Description** |
| Primary Key | DetailID | INTEGER | NOT NULL | Unique identifier timestamp for each type of Detail |
| Other Attributes | DetailName | Varchar(15) | NOT NULL | Detail name of a property |

***PropertyDetail Entity: This entity captures the type and description of property details on the Housing Rental based system***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Entity Name** | **Attribute Name** | **Field Type** | **NULL or NOT NULL** | **Description** |
| Primary Key and Foreign Key | Property Apartment No | Varchar(10) | NOT NULL | Foreign Key fetched from Property Table and Unique identifier to PropertyDetail |
| Primary Key and Foreign Key | Property Street | Varchar(30) | NOT NULL | Foreign Key fetched from Property Table and Unique identifier to PropertyDetail |
| Primary Key and Foreign Key | Property Zip | Numeric(5) | NOT NULL | Foreign Key fetched from Property Table and Unique identifier to PropertyDetail |
| Primary Key and Foreign Key | DetailID | Integer | NOT NULL | Foreign Key fetched from Detail Table and Unique identifier to PropertyType |
| Other Attributes | Detail Value | Varchar(10) | NOT NULL | Value of the Property detail |
| Other Attributes | Detail Desc | Varchar(150) | NOT NULL | The description of the property |

Business Rules

1. The landlord must be over 18 years of age and citizen of United States
2. User must search for house listings only in Syracuse
3. House listings should be made available only when the lease expires
4. User/Renter must have a valid ID i.e. Driver’s license or Passport
5. One bedroom should only have one user
6. Only one posting must be of one landlord corresponding to one house

SQL Script for Creation of Table

/\* DROPPING ALL TABLES \*/

DROP TABLE Lease

DROP TABLE PropertyView

DROP TABLE FavoriteProperty

DROP TABLE Property

DROP TABLE UserInfo

DROP TABLE Landlord

DROP TABLE Detail

DROP TABLE PropertyDetail

DROP TABLE PropertyType

/\* Create UserInfo table \*/

CREATE TABLE UserInfo (

UserID VARCHAR(10) NOT NULL,

UserFirstName VARCHAR(30) NOT NULL,

UserLastName VARCHAR(30) NOT NULL,

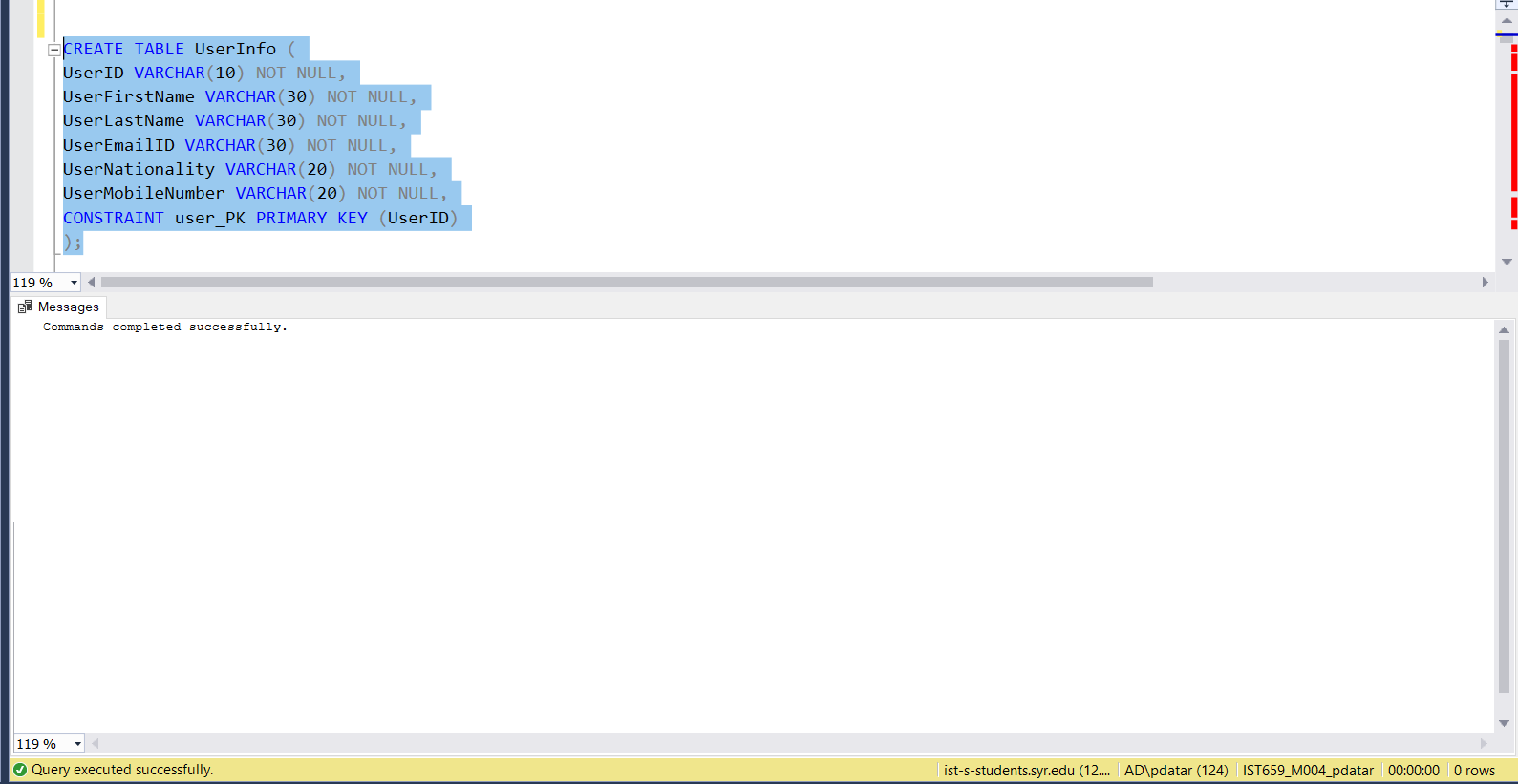
UserEmailID VARCHAR(30) NOT NULL,

UserNationality VARCHAR(20) NOT NULL,

UserMobileNumber VARCHAR(20) NOT NULL,

CONSTRAINT user\_PK PRIMARY KEY (UserID)

);



/\* Create Landlord table \*/

CREATE TABLE Property (

PropertyApartmentNo VARCHAR(10) NOT NULL,

PropertyStreet VARCHAR(30) NOT NULL,

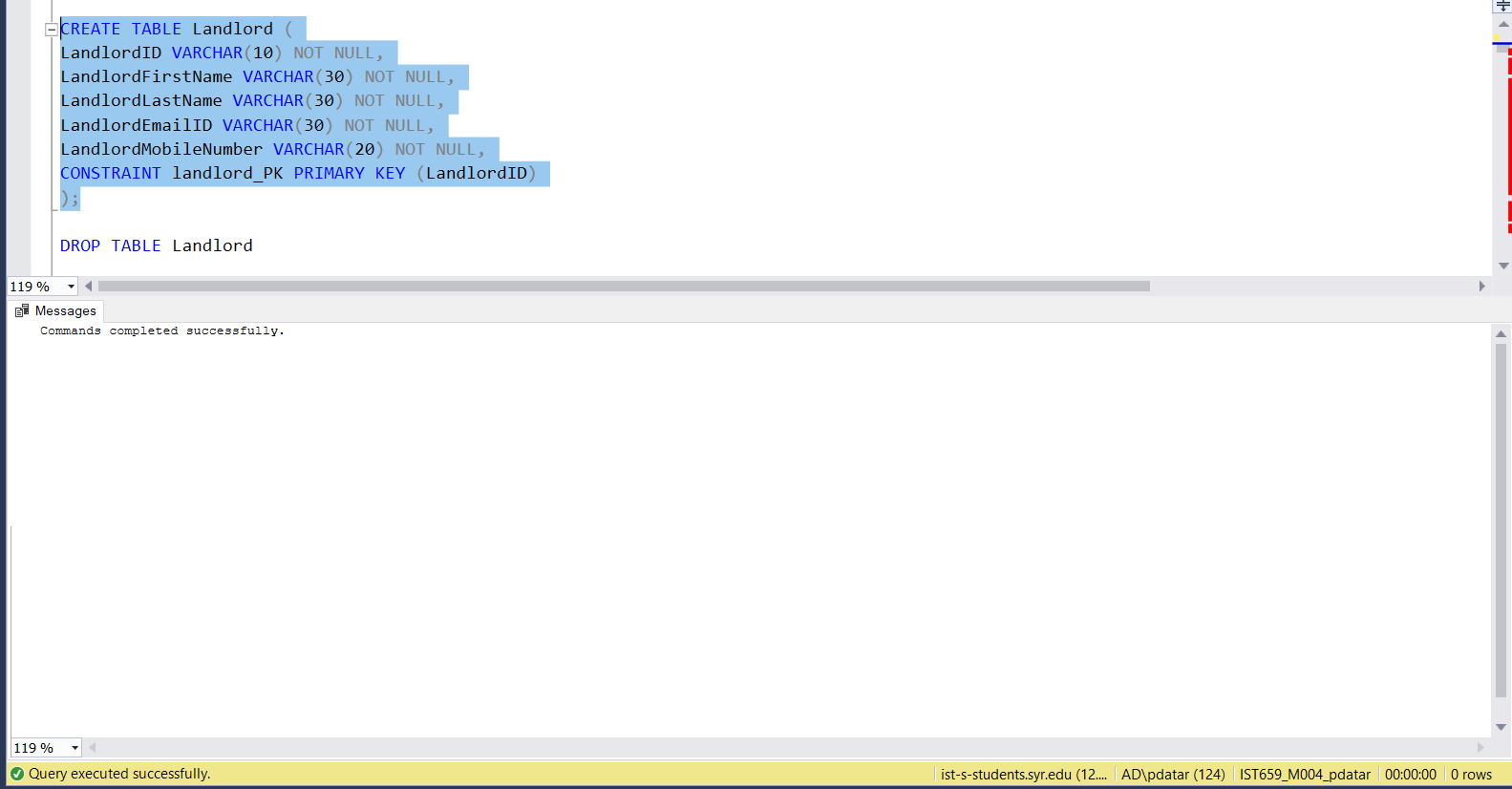
PropertyZip NUMERIC(5) NOT NULL,

LandlordID VARCHAR(10) NOT NULL,

CONSTRAINT property\_PK PRIMARY KEY (PropertyApartmentNo,PropertyStreet,PropertyZip),

CONSTRAINT property\_FK FOREIGN KEY(LandlordID) REFERENCES Landlord(LandlordID)

);



/\* Create Property table \*/

CREATE TABLE Property (

PropertyApartmentNo VARCHAR(10) NOT NULL,

PropertyStreet VARCHAR(30) NOT NULL,

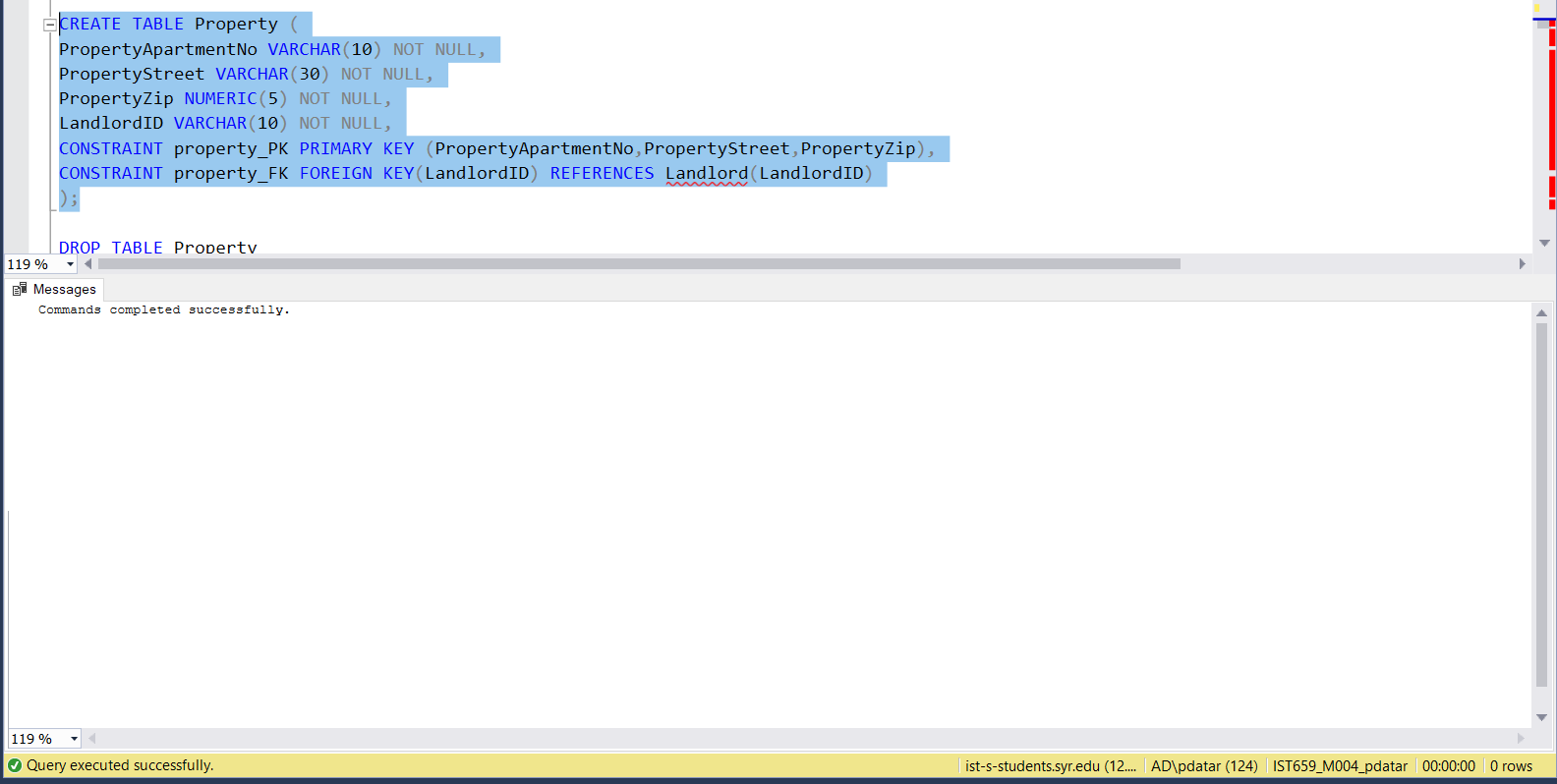
PropertyZip NUMERIC(5) NOT NULL,

LandlordID VARCHAR(10) NOT NULL,

CONSTRAINT property\_PK PRIMARY KEY (PropertyApartmentNo,PropertyStreet,PropertyZip),

CONSTRAINT property\_FK FOREIGN KEY(LandlordID) REFERENCES Landlord(LandlordID)

);



/\* Create Lease table \*/

CREATE TABLE Lease (

LeaseNumber INTEGER NOT NULL,

PropertyApartmentNo VARCHAR(10) NOT NULL,

PropertyStreet VARCHAR(30) NOT NULL,

PropertyZip NUMERIC(5) NOT NULL,

LandlordID VARCHAR(10) NOT NULL,

UserID VARCHAR(10) NOT NULL,

SecurityDeposit INTEGER NOT NULL,

RentPerBedroom INTEGER NOT NULL,

LeaseIssueDate DATETIME NOT NULL,

LeaseStartDate DATETIME NOT NULL,

LeaseEndDate DATETIME NOT NULL,

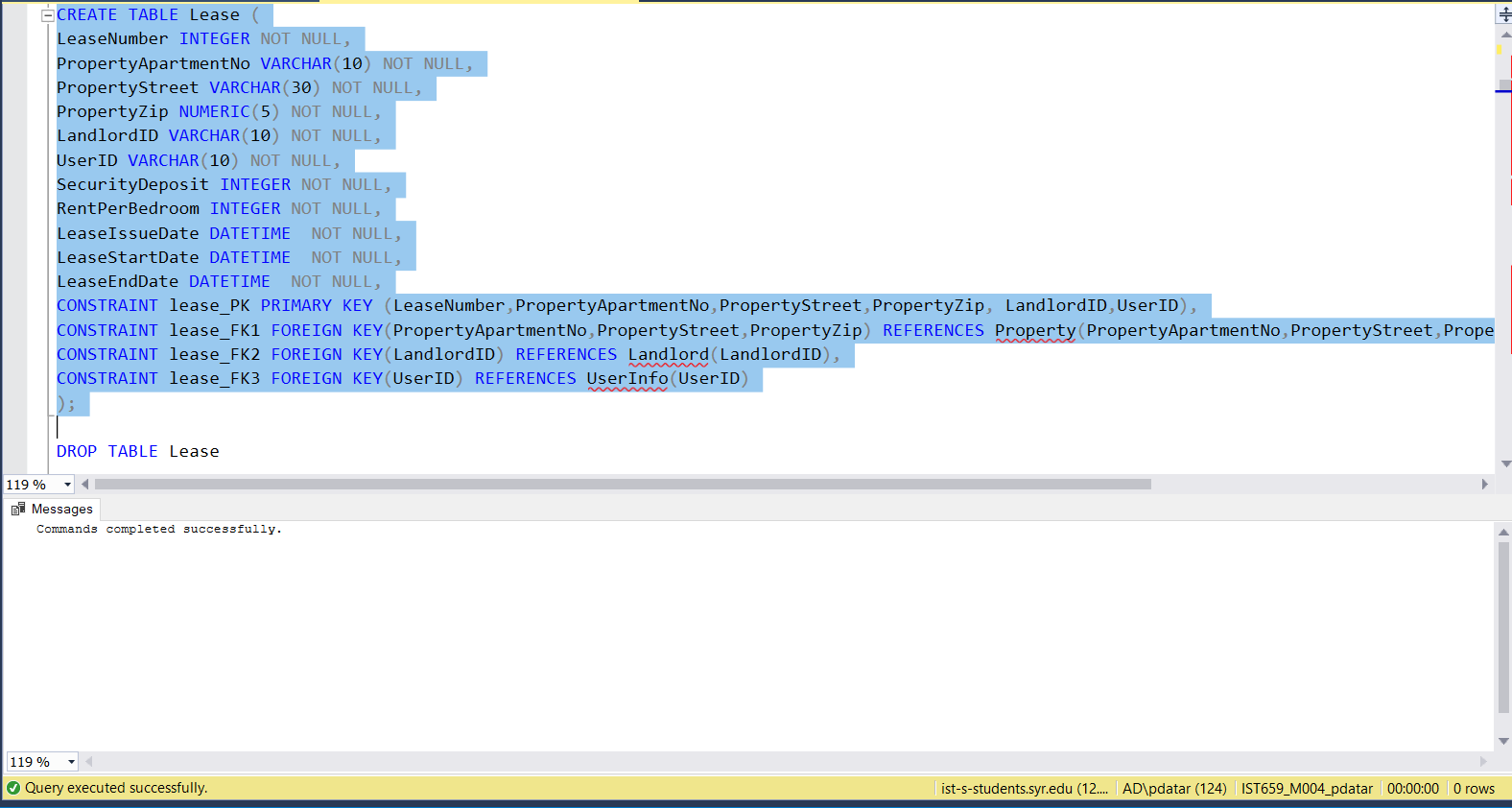
CONSTRAINT lease\_PK PRIMARY KEY (LeaseNumber,PropertyApartmentNo,PropertyStreet,PropertyZip, LandlordID,UserID),

CONSTRAINT lease\_FK1 FOREIGN KEY(PropertyApartmentNo,PropertyStreet,PropertyZip) REFERENCES Property(PropertyApartmentNo,PropertyStreet,PropertyZip),

CONSTRAINT lease\_FK2 FOREIGN KEY(LandlordID) REFERENCES Landlord(LandlordID),

CONSTRAINT lease\_FK3 FOREIGN KEY(UserID) REFERENCES UserInfo(UserID)

);



/\* Create FavoriteProperty table \*/

CREATE TABLE FavoriteProperty (

UserID VARCHAR(10) NOT NULL,

PropertyApartmentNo VARCHAR(10) NOT NULL,

PropertyStreet VARCHAR(30) NOT NULL,

PropertyZip NUMERIC(5) NOT NULL,

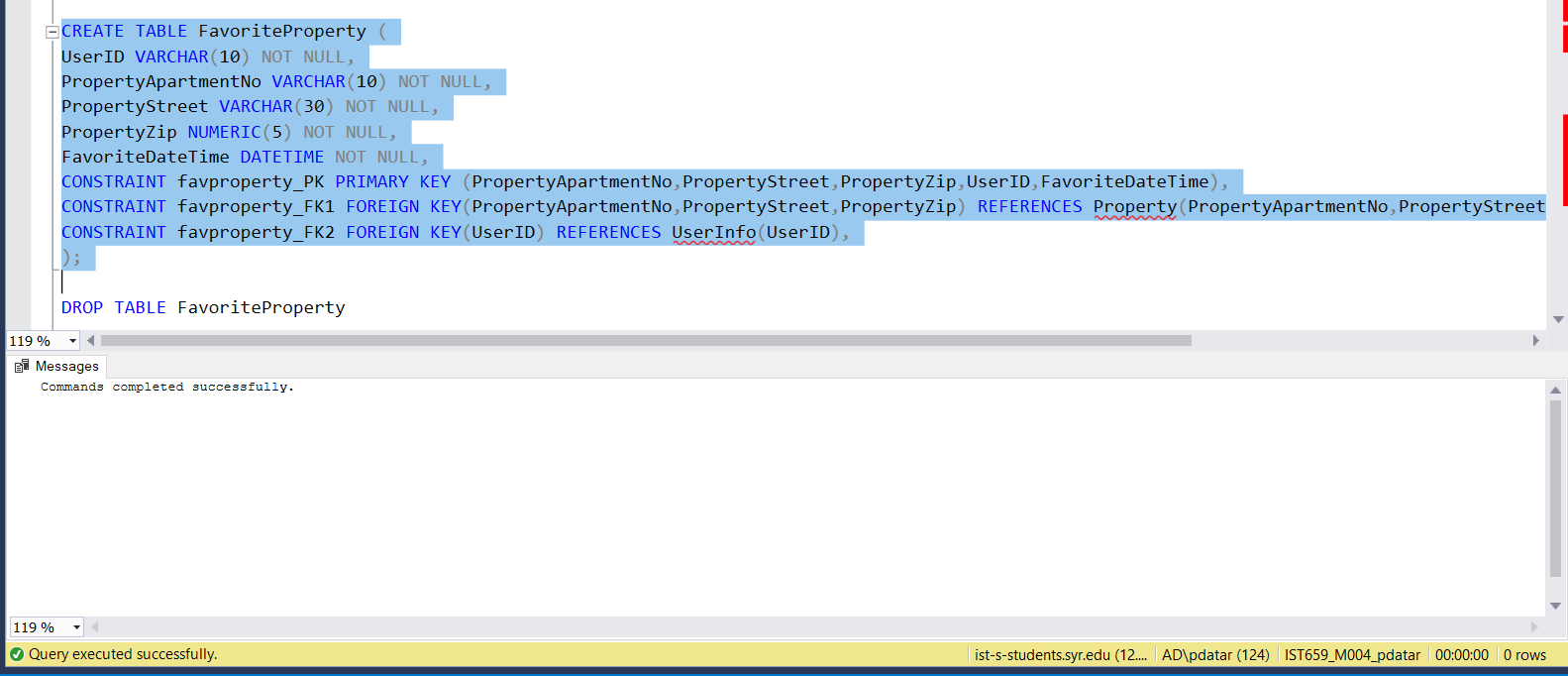
FavoriteDateTime DATETIME NOT NULL,

CONSTRAINT favproperty\_PK PRIMARY KEY (PropertyApartmentNo,PropertyStreet,PropertyZip,UserID,FavoriteDateTime),

CONSTRAINT favproperty\_FK1 FOREIGN KEY(PropertyApartmentNo,PropertyStreet,PropertyZip) REFERENCES Property(PropertyApartmentNo,PropertyStreet,PropertyZip),

CONSTRAINT favproperty\_FK2 FOREIGN KEY(UserID) REFERENCES UserInfo(UserID),

);



/\*Create PropertyView table \*/

CREATE TABLE PropertyView (

UserID VARCHAR(10) NOT NULL,

PropertyApartmentNo VARCHAR(10) NOT NULL,

PropertyStreet VARCHAR(30) NOT NULL,

PropertyZip NUMERIC(5) NOT NULL,

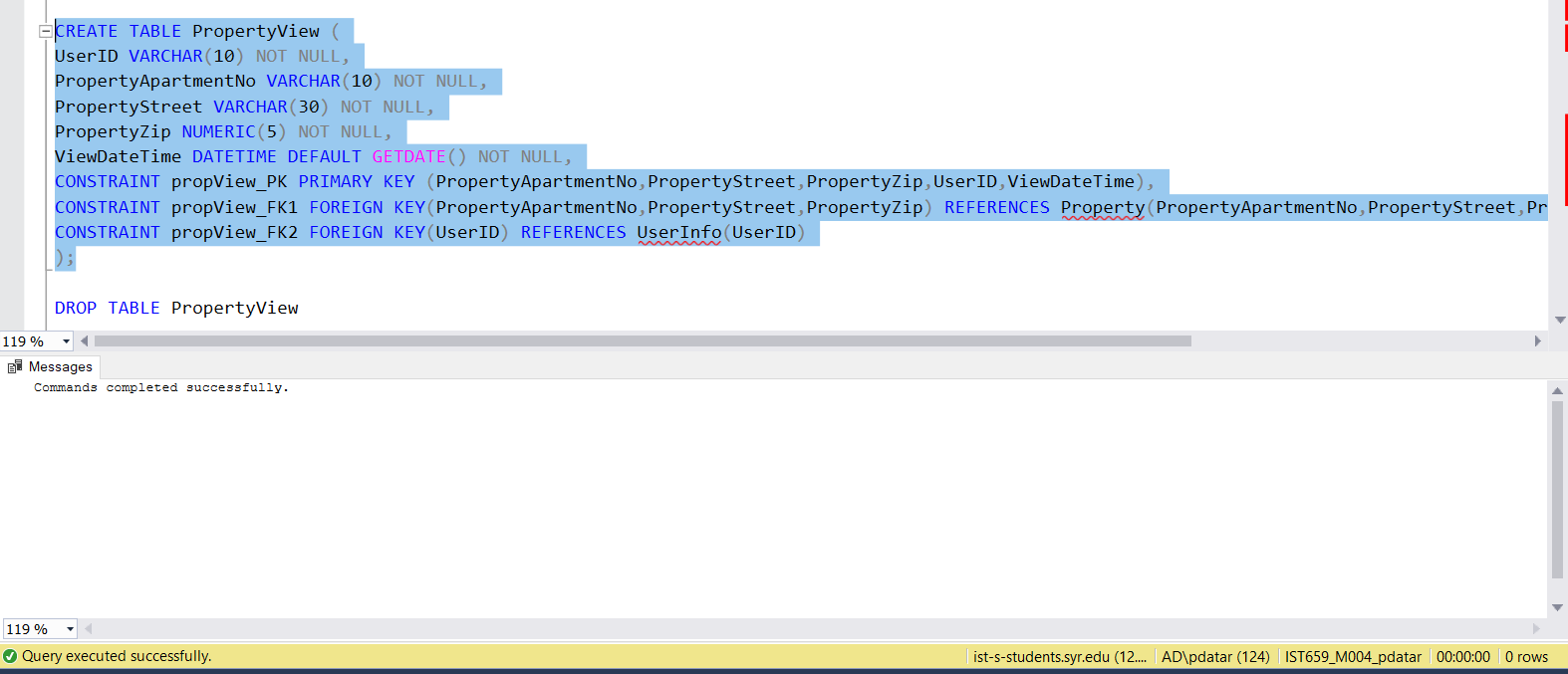
ViewDateTime DATETIME DEFAULT GETDATE() NOT NULL,

CONSTRAINT propView\_PK PRIMARY KEY (PropertyApartmentNo,PropertyStreet,PropertyZip,UserID,ViewDateTime),

CONSTRAINT propView\_FK1 FOREIGN KEY(PropertyApartmentNo,PropertyStreet,PropertyZip) REFERENCES Property(PropertyApartmentNo,PropertyStreet,PropertyZip),

CONSTRAINT propView\_FK2 FOREIGN KEY(UserID) REFERENCES UserInfo(UserID)

);



/\*Create PropertyType table \*/

CREATE TABLE PropertyType (

PropertyTypeID INTEGER NOT NULL,

PropertyApartmentNo VARCHAR(10) NOT NULL,

PropertyStreet VARCHAR(30) NOT NULL,

PropertyZip NUMERIC(5) NOT NULL,

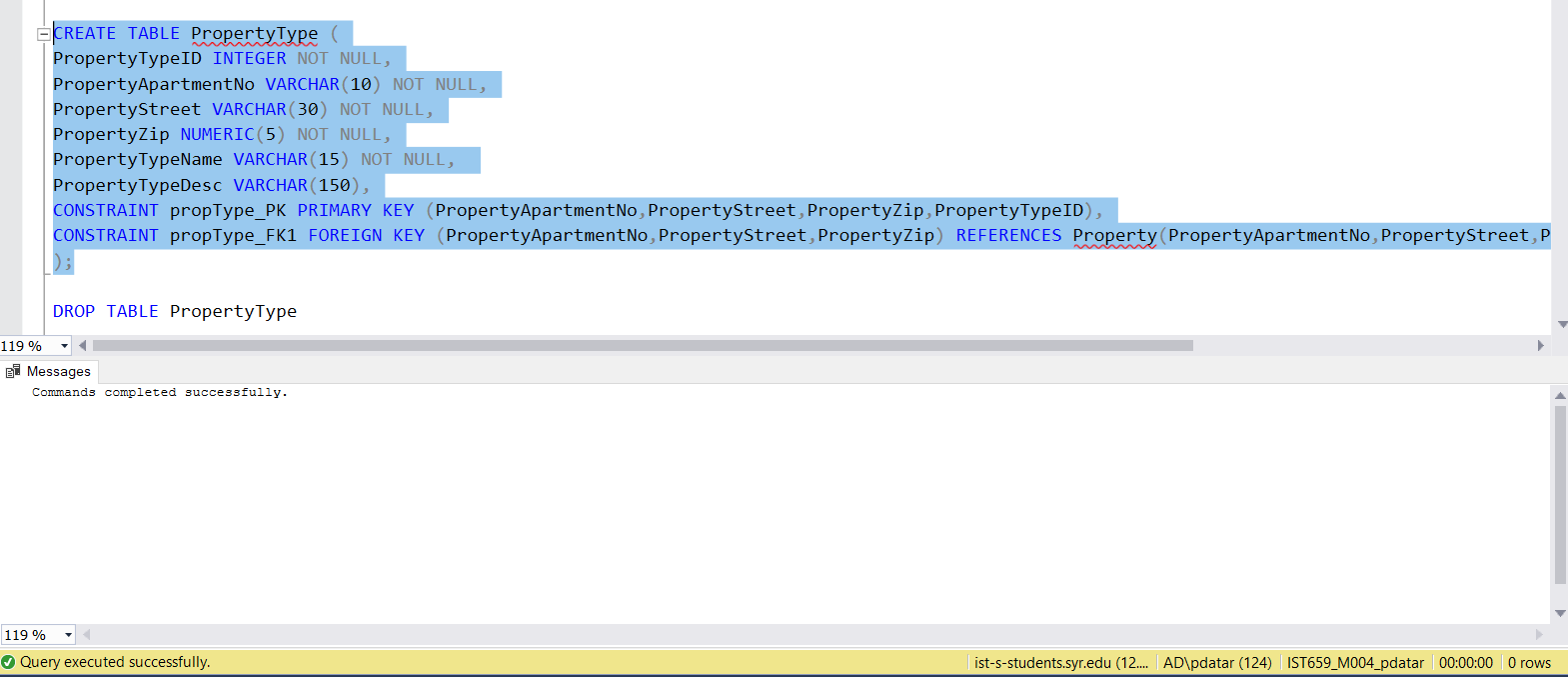
PropertyTypeName VARCHAR(15) NOT NULL,

PropertyTypeDesc VARCHAR(150),

CONSTRAINT propType\_PK PRIMARY KEY (PropertyApartmentNo,PropertyStreet,PropertyZip,PropertyTypeID),

CONSTRAINT propType\_FK1 FOREIGN KEY (PropertyApartmentNo,PropertyStreet,PropertyZip) REFERENCES Property(PropertyApartmentNo,PropertyStreet,PropertyZip),

);



/\*Create Detail table \*/

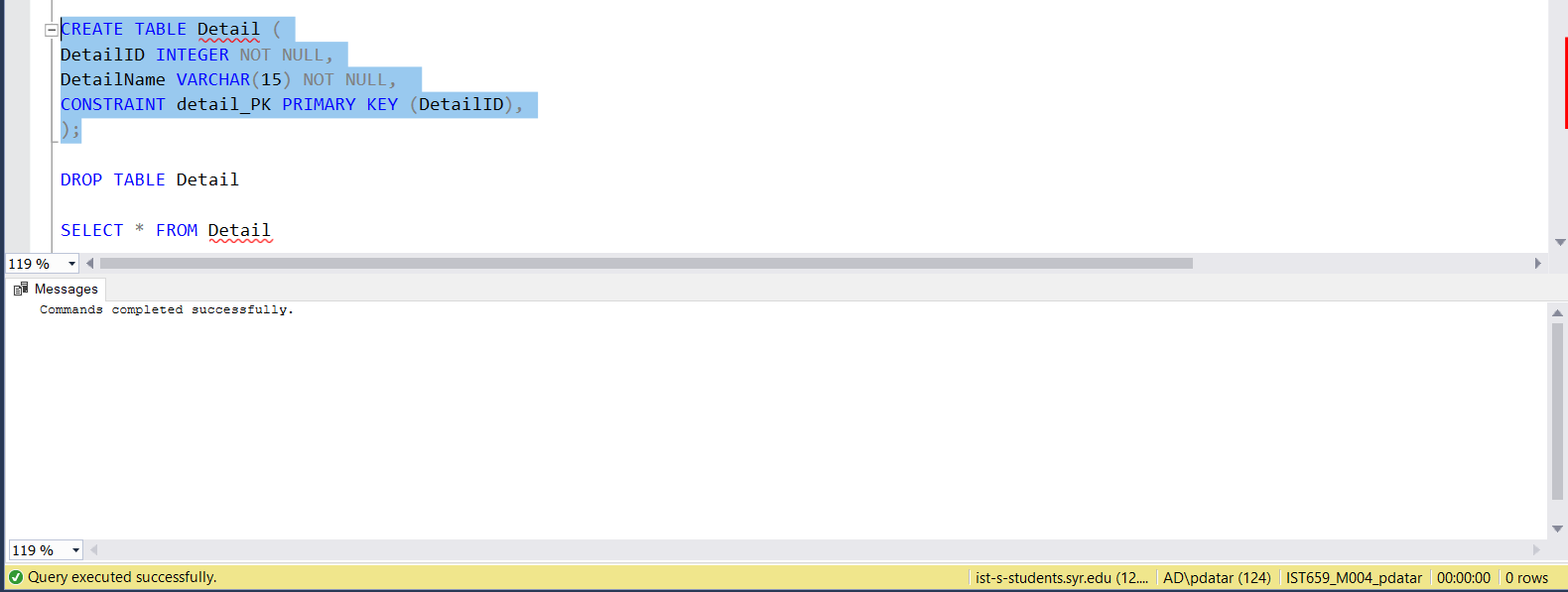
CREATE TABLE Detail (

DetailID INTEGER NOT NULL,

DetailName VARCHAR(15) NOT NULL,

CONSTRAINT detail\_PK PRIMARY KEY (DetailID),

);



/\*Create PropertyDetail table \*/

CREATE TABLE PropertyDetail (

DetailID INTEGER NOT NULL,

DetailValue VARCHAR(10) NOT NULL,

PropertyDetailDesc VARCHAR(150),

PropertyApartmentNo VARCHAR(10) NOT NULL,

PropertyStreet VARCHAR(30) NOT NULL,

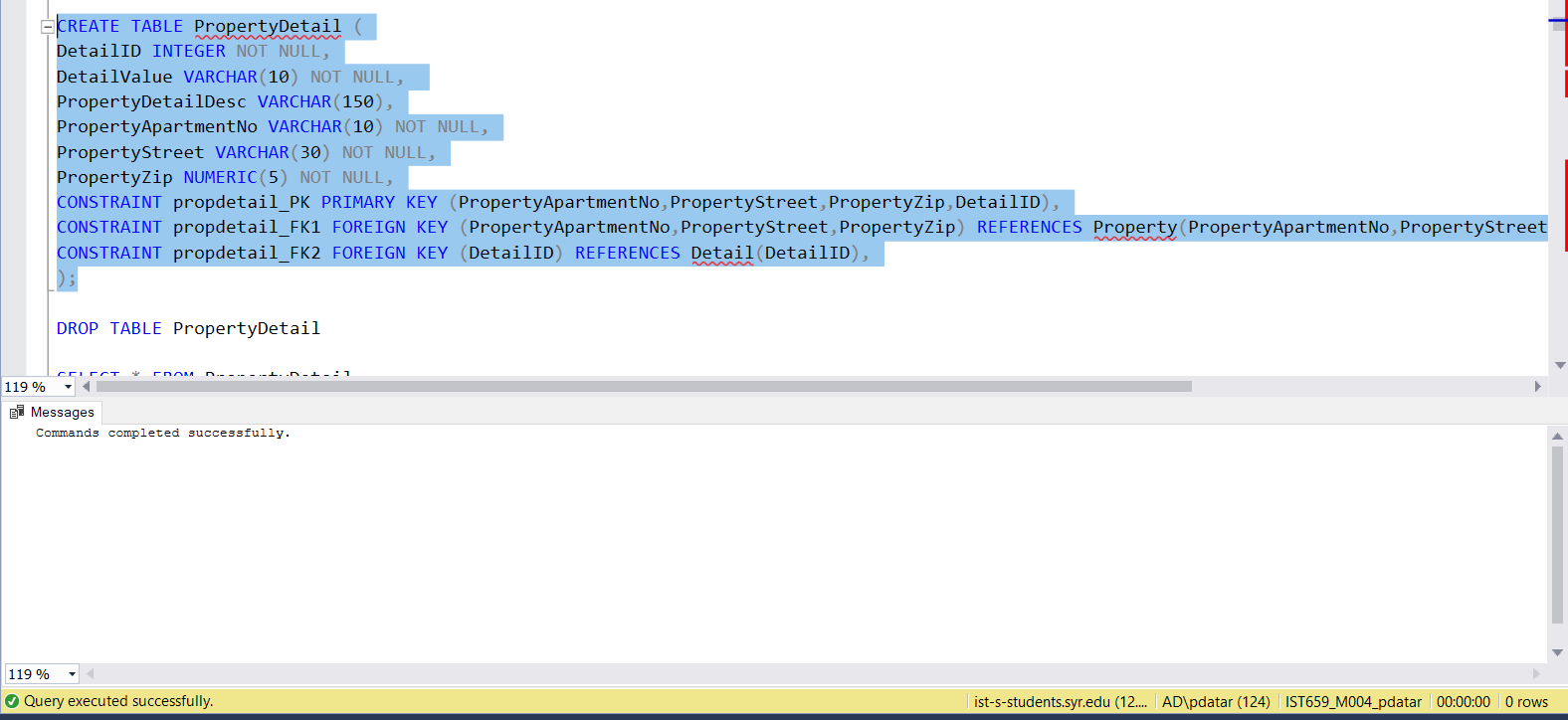
PropertyZip NUMERIC(5) NOT NULL,

CONSTRAINT propdetail\_PK PRIMARY KEY (PropertyApartmentNo,PropertyStreet,PropertyZip,DetailID),

CONSTRAINT propdetail\_FK1 FOREIGN KEY (PropertyApartmentNo,PropertyStreet,PropertyZip) REFERENCES Property(PropertyApartmentNo,PropertyStreet,PropertyZip),

CONSTRAINT propdetail\_FK2 FOREIGN KEY (DetailID) REFERENCES Detail(DetailID),

);



SQL Script for Insertion in Table

/\*Insert Data in UserInfo table \*/

INSERT INTO UserInfo values

('nathan9', 'Nathan', 'Kerr','nathan.kerr@syr.edu','USA','315-555-5555'),

('seb2', 'Sebastian', 'Chapman','s.chapman@syr.edu','USA','315-555-6666'),

('heath77', 'Heather', 'Cameron', 'heather.c@syr.edu', 'USA','315-555-7777'),

('oli421', 'Olivia', 'Wallace', 'oliviawallace@syr.edu', 'England','315-555-8888'),

('lil99', 'Lily', 'Turner', 'lily.turner@syr.edu', 'Ireland','315-555-9999');

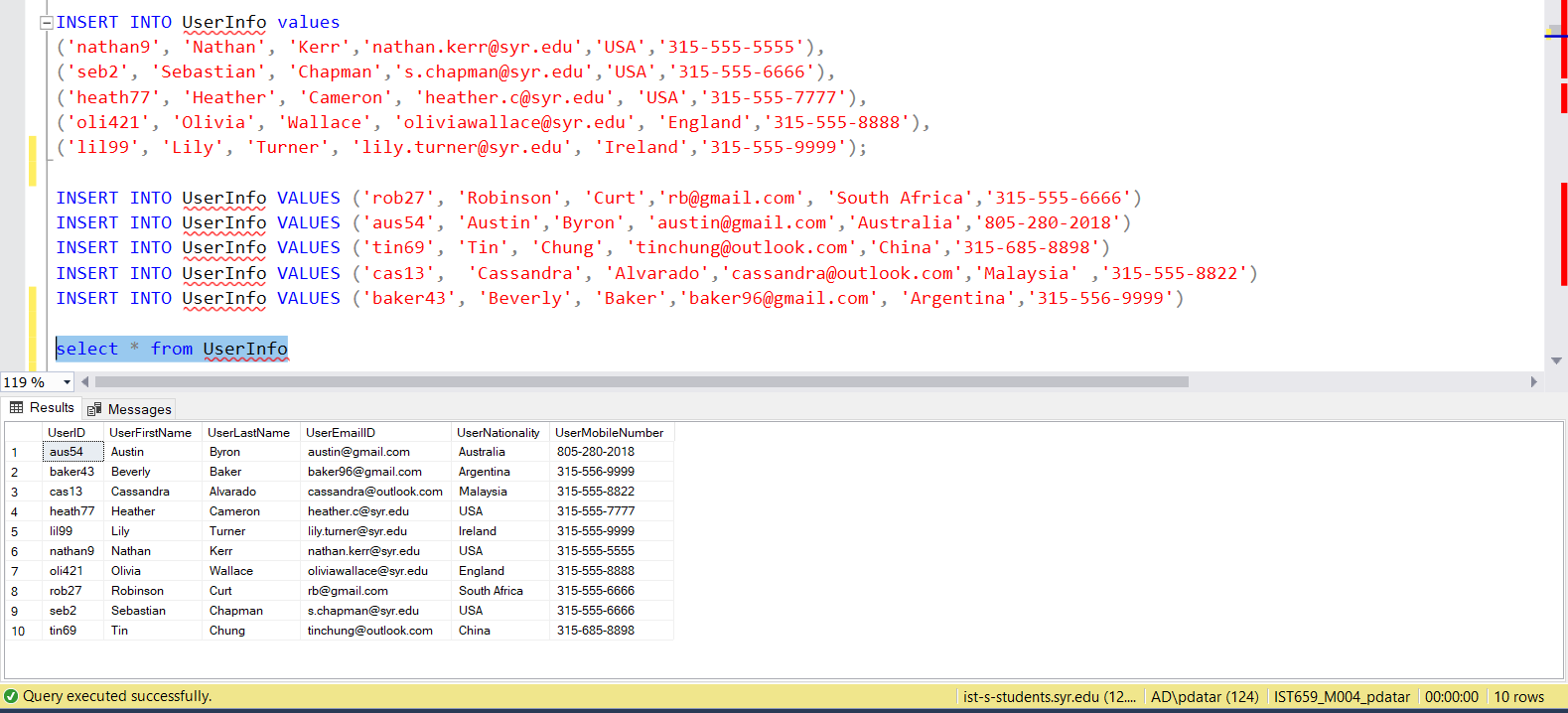
INSERT INTO UserInfo VALUES ('rob27', 'Robinson', 'Curt','rb@gmail.com', 'South Africa','315-555-6666')

INSERT INTO UserInfo VALUES ('aus54', 'Austin','Byron', 'austin@gmail.com','Australia','805-280-2018')

INSERT INTO UserInfo VALUES ('tin69', 'Tin', 'Chung', 'tinchung@outlook.com','China','315-685-8898')

INSERT INTO UserInfo VALUES ('cas13', 'Cassandra', 'Alvarado','cassandra@outlook.com','Malaysia' ,'315-555-8822')

INSERT INTO UserInfo VALUES ('baker43', 'Beverly', 'Baker','baker96@gmail.com', 'Argentina','315-556-9999')



/\*Insert Data in Landlord table \*/

INSERT INTO Landlord values

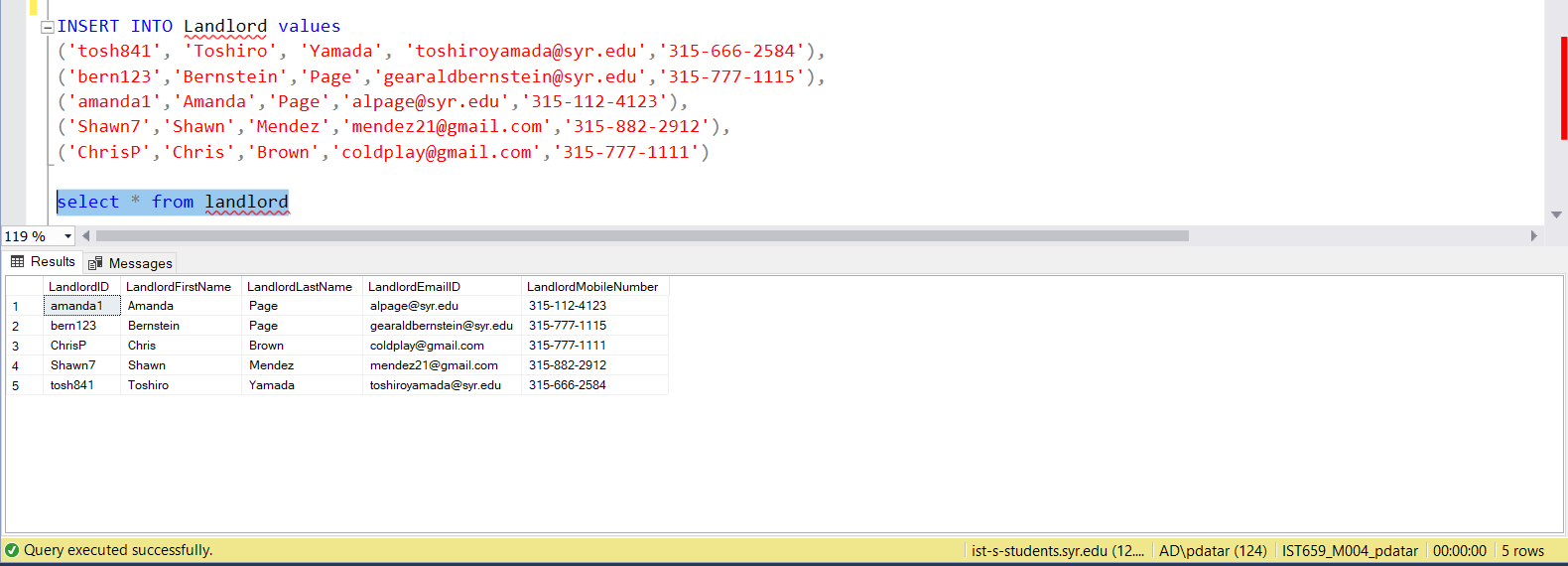
('tosh841', 'Toshiro', 'Yamada', 'toshiroyamada@syr.edu','315-666-2584'),

('bern123','Bernstein','Page','gearaldbernstein@syr.edu','315-777-1115'),

('amanda1','Amanda','Page','alpage@syr.edu','315-112-4123'),

('Shawn7','Shawn','Mendez','mendez21@gmail.com','315-882-2912'),

('ChrisP','Chris','Brown','coldplay@gmail.com','315-777-1111')



/\*Insert Data in Property table \*/

INSERT INTO Property VALUES

('966','Lancaster Avenue','13205','bern123'),

('738','Lancaster Avenue','13204','tosh841'),

('1021','WestCott Street', '14234','amanda1'),

('703-3','Westcott Street','12343','ChrisP'),

('112','Avondale Street','13124','Shawn7'),

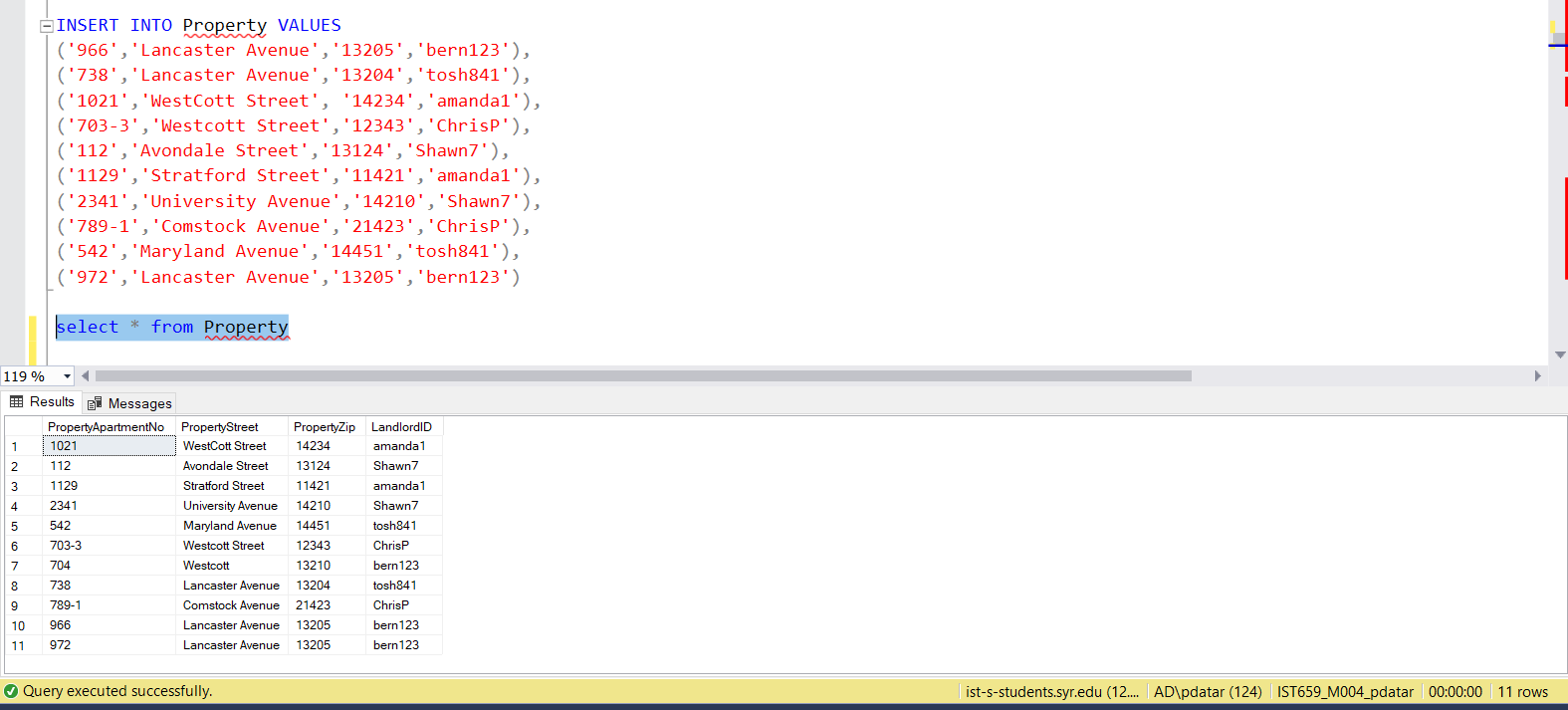
('1129','Stratford Street','11421','amanda1'),

('2341','University Avenue','14210','Shawn7'),

('789-1','Comstock Avenue','21423','ChrisP'),

('542','Maryland Avenue','14451','tosh841'),

('972','Lancaster Avenue','13205','bern123')



/\*Insert Data in Lease table \*/

INSERT INTO Lease VALUES

(4792,'1021','WestCott Street', '14234','amanda1','cas13', 500, 465, '2013-05-07 00:00:00.000','2013-08-01 00:00:00.000','2014-07-31 00:00:00.000'),

(2331,'972','Lancaster Avenue','13205','bern123','cas13',500, 430, '2014-05-07 00:00:00.000','2014-08-01 00:00:00.000','2015-07-31 00:00:00.000'),

(1231,'738','Lancaster Avenue','13204','tosh841','nathan9',600, 550,'2014-06-01 00:00:00.000','2014-08-05 00:00:00.000','2015-08-04 00:00:00.000'),

(1231,'738','Lancaster Avenue','13204','tosh841','tin69',600, 550,'2014-06-01 00:00:00.000','2014-08-05 00:00:00.000','2015-08-04 00:00:00.000'),

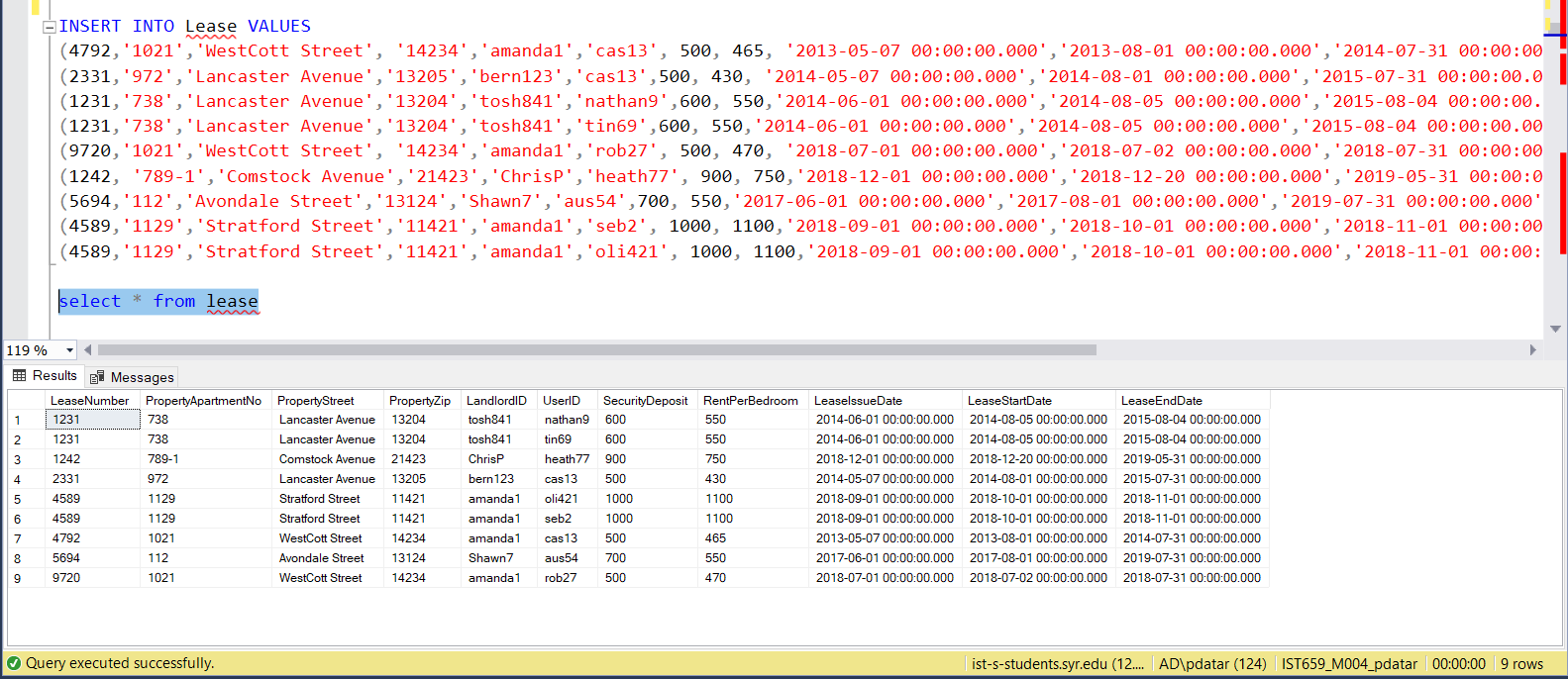
(9720,'1021','WestCott Street', '14234','amanda1','rob27', 500, 470, '2018-07-01 00:00:00.000','2018-07-02 00:00:00.000','2018-07-31 00:00:00.000'),

(1242, '789-1','Comstock Avenue','21423','ChrisP','heath77', 900, 750,'2018-12-01 00:00:00.000','2018-12-20 00:00:00.000','2019-05-31 00:00:00.000'),

(5694,'112','Avondale Street','13124','Shawn7','aus54',700, 550,'2017-06-01 00:00:00.000','2017-08-01 00:00:00.000','2019-07-31 00:00:00.000'),

(4589,'1129','Stratford Street','11421','amanda1','seb2', 1000, 1100,'2018-09-01 00:00:00.000','2018-10-01 00:00:00.000','2018-11-01 00:00:00.000'),

(4589,'1129','Stratford Street','11421','amanda1','oli421', 1000, 1100,'2018-09-01 00:00:00.000','2018-10-01 00:00:00.000','2018-11-01 00:00:00.000')



/\*Insert Data in FavoritePropert table \*/

INSERT INTO FavoriteProperty VALUES

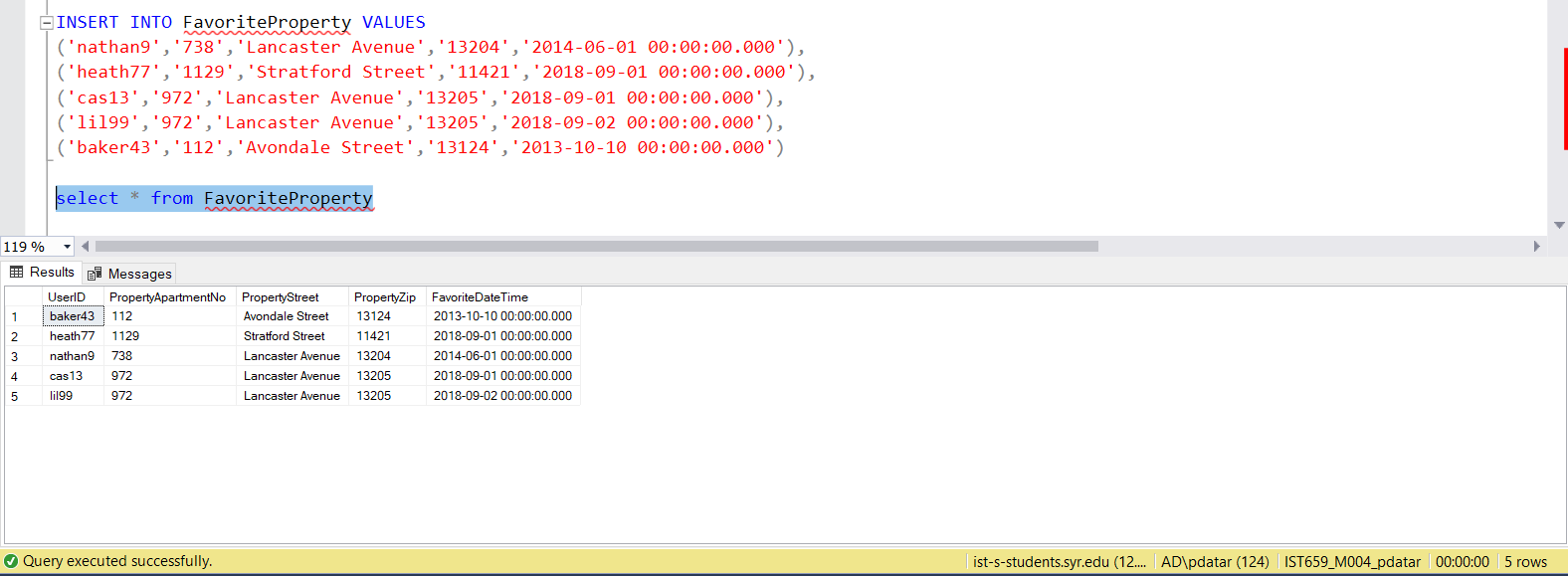
('nathan9','738','Lancaster Avenue','13204','2014-06-01 00:00:00.000'),

('heath77','1129','Stratford Street','11421','2018-09-01 00:00:00.000'),

('cas13','972','Lancaster Avenue','13205','2018-09-01 00:00:00.000'),

('lil99','972','Lancaster Avenue','13205','2018-09-02 00:00:00.000'),

('baker43','112','Avondale Street','13124','2013-10-10 00:00:00.000')



/\*Insert Data in PropertyView table \*/

INSERT INTO PropertyView VALUES

('nathan9','738','Lancaster Avenue','13204',''),

('nathan9','738','Lancaster Avenue','13204','2013-10-10 00:00:00.000')

INSERT INTO PropertyView VALUES

('nathan9','738','Lancaster Avenue','13204','2013-10-11 00:00:00.000'),

('nathan9','738','Lancaster Avenue','13204','2013-10-12 00:00:00.000'),

('heath77','1129','Stratford Street','11421','2018-09-01 00:00:00.000'),

('cas13','972','Lancaster Avenue','13205','2018-09-01 00:00:00.000'),

('lil99','972','Lancaster Avenue','13205','2018-09-02 00:00:00.000'),

('baker43','112','Avondale Street','13124','2013-10-10 00:00:00.000'),

('cas13','1021','WestCott Street', '14234','2018-11-10 00:00:00.000'),

('cas13','972','Lancaster Avenue','13205','2019-02-20 00:00:00.000'),

('nathan9','738','Lancaster Avenue','13204','2013-12-01 00:00:00.000'),

('tin69','738','Lancaster Avenue','13204',''),

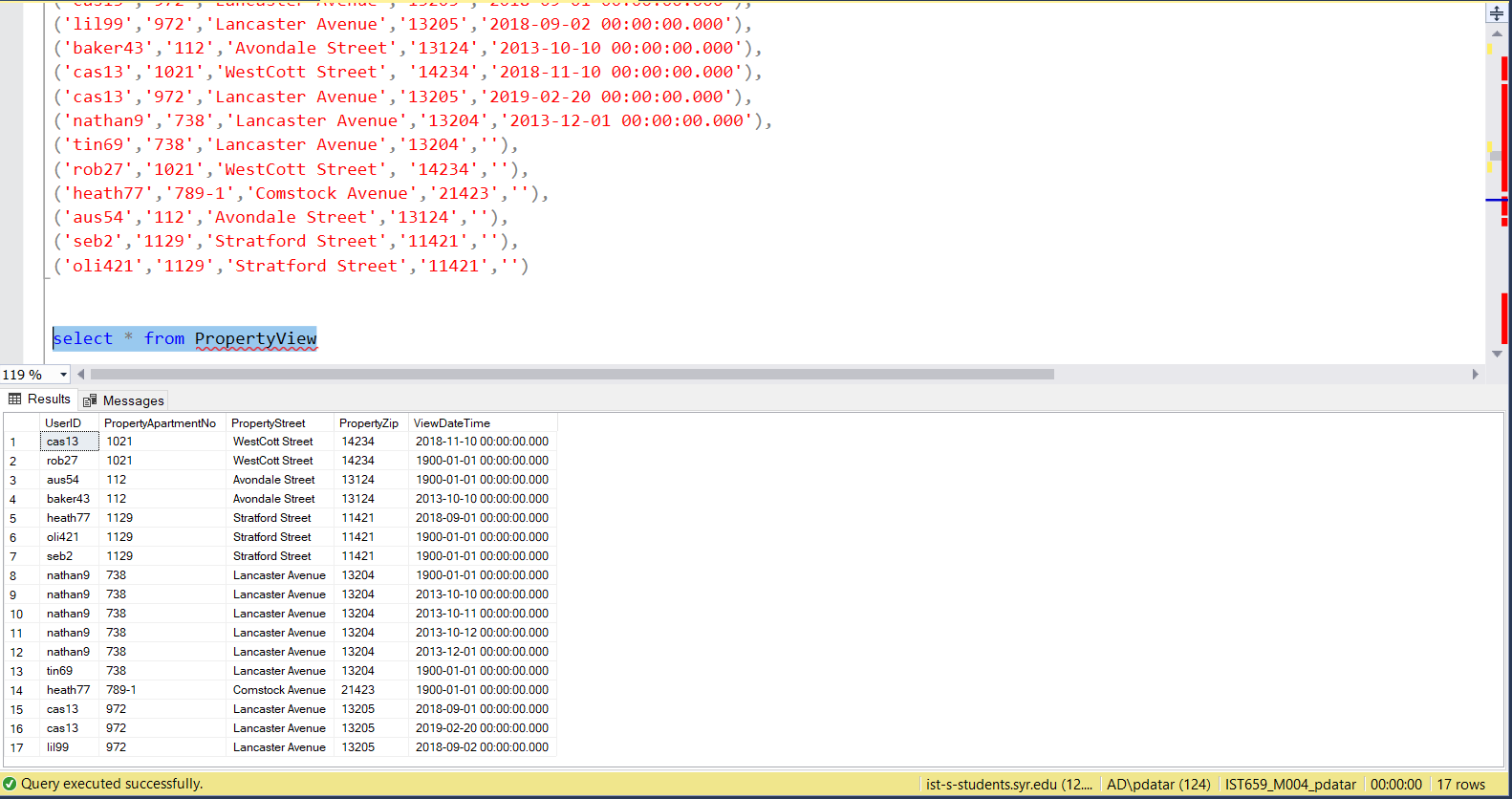
('rob27','1021','WestCott Street', '14234',''),

('heath77','789-1','Comstock Avenue','21423',''),

('aus54','112','Avondale Street','13124',''),

('seb2','1129','Stratford Street','11421',''),

('oli421','1129','Stratford Street','11421','')



/\*Insert Data in PropertyType table \*/

INSERT INTO PropertyType VALUES

(1,'966','Lancaster Avenue','13205','house','')

INSERT INTO PropertyType VALUES

(1,'738','Lancaster Avenue','13204','house','Safe Neighbourhood'),

(1,'1021','WestCott Street', '14234','house','Large Rooms'),

(1,'703-3','Westcott Street','12343','house','Big Spacious House'),

(2,'112','Avondale Street','13124','apartment','Close to University'),

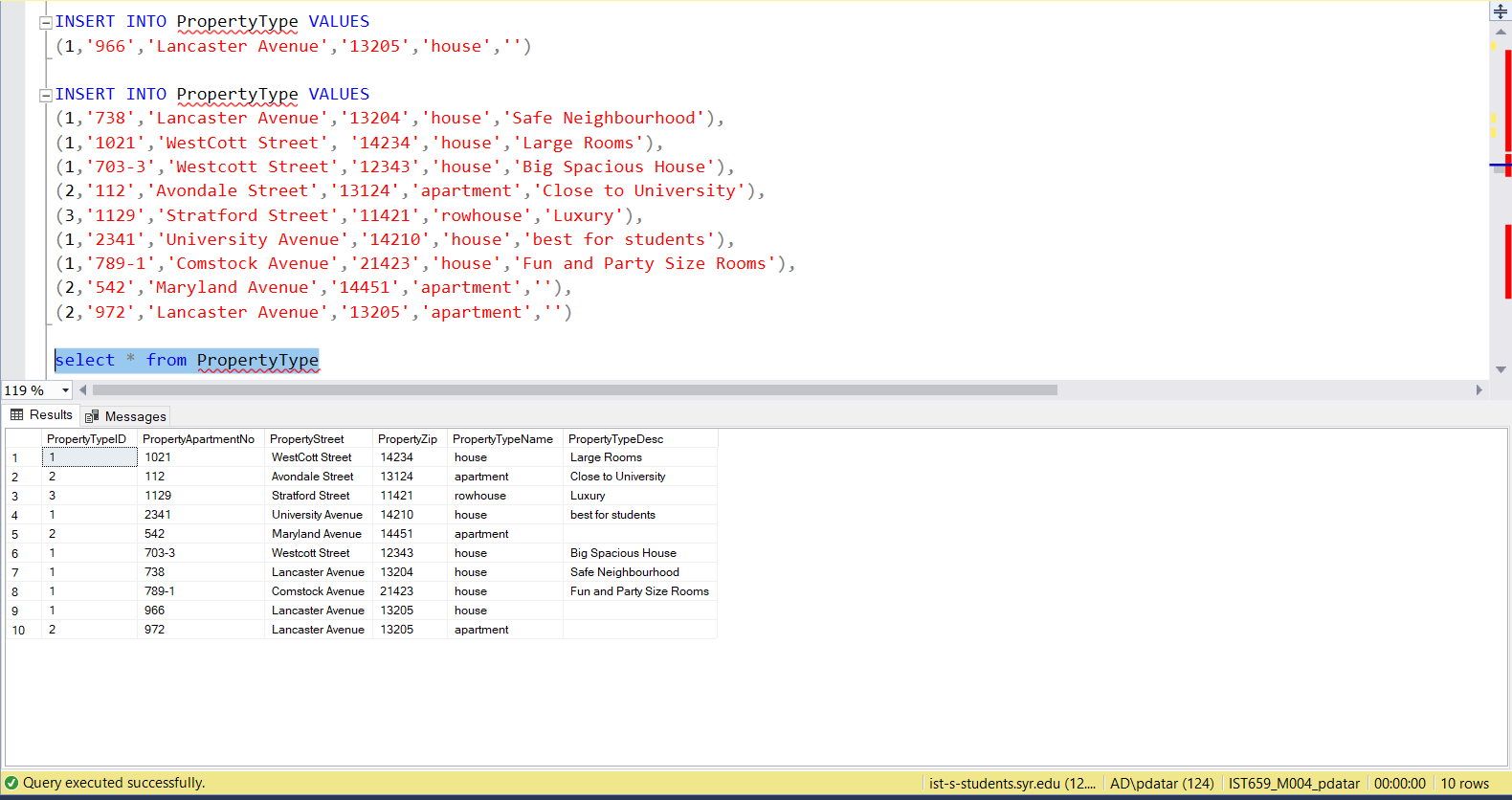
(3,'1129','Stratford Street','11421','rowhouse','Luxury'),

(1,'2341','University Avenue','14210','house','best for students'),

(1,'789-1','Comstock Avenue','21423','house','Fun and Party Size Rooms'),

(2,'542','Maryland Avenue','14451','apartment',''),

(2,'972','Lancaster Avenue','13205','apartment','')



/\*Insert Data in Detail table \*/

INSERT INTO Detail VALUES

(1, 'No.of Beds'),

(2, 'No.of Baths'),

(3, 'Pets Allowed'),

(4, 'Furnished'),

(5, 'Fireplace'),

(6, 'Dishwasher'),

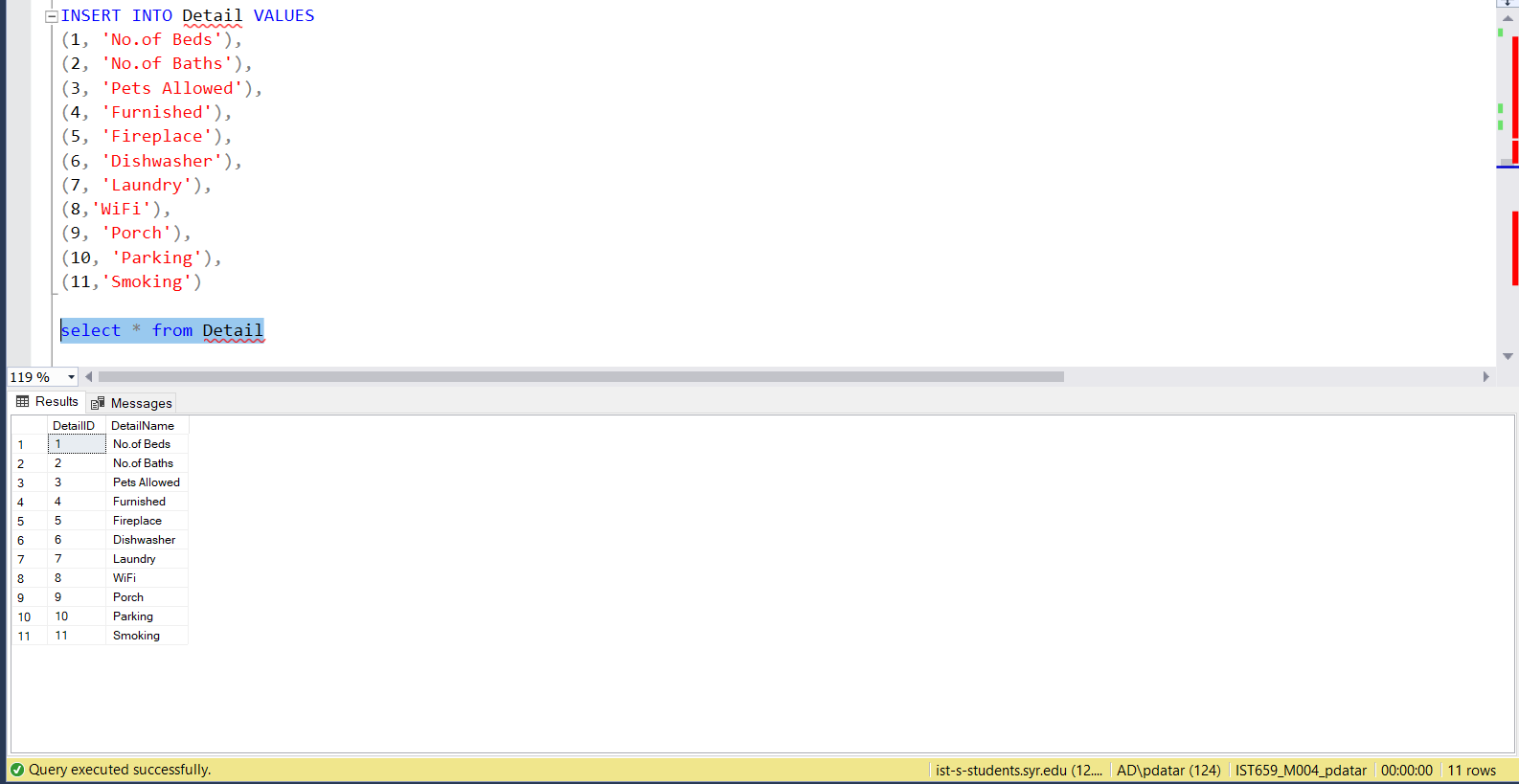
(7, 'Laundry'),

(8,'WiFi'),

(9, 'Porch'),

(10, 'Parking'),

(11,'Smoking')



/\*Insert Data in PropertyDetail table \*/

INSERT INTO PropertyDetail VALUES

(1,'4','','966','Lancaster Avenue','13205'),

(2,'4','','966','Lancaster Avenue','13205'),

(3,'Yes','Only Dogs','966','Lancaster Avenue','13205'),

(4,'Yes','All','966','Lancaster Avenue','13205'),

(5,'No','','966','Lancaster Avenue','13205'),

(6,'Yes','','966','Lancaster Avenue','13205'),

(7,'Yes','Coin','966','Lancaster Avenue','13205'),

(8,'No','','966','Lancaster Avenue','13205'),

(9,'Yes','','966','Lancaster Avenue','13205'),

(10,'No','','966','Lancaster Avenue','13205'),

(11,'No','','966','Lancaster Avenue','13205'),

(1,'2','','738','Lancaster Avenue','13204'),

(2,'1','','738','Lancaster Avenue','13204'),

(3,'Yes','','738','Lancaster Avenue','13204'),

(4,'Yes','','738','Lancaster Avenue','13204'),

(5,'Yes','','738','Lancaster Avenue','13204'),

(6,'Yes','','738','Lancaster Avenue','13204'),

(7,'Yes','','738','Lancaster Avenue','13204'),

(8,'Yes','','738','Lancaster Avenue','13204'),

(9,'Yes','','738','Lancaster Avenue','13204'),

(10,'Yes','','738','Lancaster Avenue','13204'),

(11,'Yes','','738','Lancaster Avenue','13204'),

(1,'5','','1021','WestCott Street', '14234'),

(2,'3','','1021','WestCott Street', '14234'),

(3,'Yes','','1021','WestCott Street', '14234'),

(4,'Yes','','1021','WestCott Street', '14234'),

(5,'No','','1021','WestCott Street', '14234'),

(6,'Yes','','1021','WestCott Street', '14234'),

(7,'Yes','Coin','1021','WestCott Street', '14234'),

(8,'Yes','','1021','WestCott Street', '14234'),

(9,'Yes','','1021','WestCott Street', '14234'),

(10,'No','','1021','WestCott Street', '14234'),

(11,'No','','1021','WestCott Street', '14234'),

(1,'4','','703-3','Westcott Street','12343'),

(2,'1','','703-3','Westcott Street','12343'),

(3,'Yes','Cats Only','703-3','Westcott Street','12343'),

(4,'No','','703-3','Westcott Street','12343'),

(5,'Yes','','703-3','Westcott Street','12343'),

(6,'Yes','','703-3','Westcott Street','12343'),

(7,'Yes','','703-3','Westcott Street','12343'),

(8,'No','','703-3','Westcott Street','12343'),

(9,'No','','703-3','Westcott Street','12343'),

(10,'Yes','','703-3','Westcott Street','12343'),

(11,'Yes','','703-3','Westcott Street','12343'),

(1,'6','','112','Avondale Street','13124'),

(2,'1','','112','Avondale Street','13124'),

(3,'Yes','','112','Avondale Street','13124'),

(4,'Yes','','112','Avondale Street','13124'),

(5,'No','','112','Avondale Street','13124'),

(6,'No','','112','Avondale Street','13124'),

(7,'Yes','','112','Avondale Street','13124'),

(8,'Yes','','112','Avondale Street','13124'),

(9,'Yes','','112','Avondale Street','13124'),

(10,'No','','112','Avondale Street','13124'),

(11,'No','','112','Avondale Street','13124'),

(1,'3','','1129','Stratford Street','11421'),

(2,'2','','1129','Stratford Street','11421'),

(3,'Yes','','1129','Stratford Street','11421'),

(4,'Yes','','1129','Stratford Street','11421'),

(5,'Yes','','1129','Stratford Street','11421'),

(6,'Yes','','1129','Stratford Street','11421'),

(7,'Yes','','1129','Stratford Street','11421'),

(8,'Yes','','1129','Stratford Street','11421'),

(9,'Yes','','1129','Stratford Street','11421'),

(10,'Yes','','1129','Stratford Street','11421'),

(11,'Yes','','1129','Stratford Street','11421'),

(1,'2','','2341','University Avenue','14210'),

(2,'2','','2341','University Avenue','14210'),

(3,'No','','2341','University Avenue','14210'),

(4,'Yes','Wifi Excluded','2341','University Avenue','14210'),

(5,'Yes','','2341','University Avenue','14210'),

(6,'Yes','','2341','University Avenue','14210'),

(7,'Yes','','2341','University Avenue','14210'),

(8,'No','','2341','University Avenue','14210'),

(9,'Yes','','2341','University Avenue','14210'),

(10,'Yes','','2341','University Avenue','14210'),

(11,'No','','2341','University Avenue','14210'),

(1,'6','','789-1','Comstock Avenue','21423'),

(2,'6','','789-1','Comstock Avenue','21423'),

(3,'Yes','Only Fish','789-1','Comstock Avenue','21423'),

(4,'Yes','','789-1','Comstock Avenue','21423'),

(5,'Yes','','789-1','Comstock Avenue','21423'),

(6,'Yes','','789-1','Comstock Avenue','21423'),

(7,'Yes','','789-1','Comstock Avenue','21423'),

(8,'Yes','','789-1','Comstock Avenue','21423'),

(9,'No','','789-1','Comstock Avenue','21423'),

(10,'No','','789-1','Comstock Avenue','21423'),

(11,'No','','789-1','Comstock Avenue','21423'),

(1,'3','','542','Maryland Avenue','14451'),

(2,'1','','542','Maryland Avenue','14451'),

(3,'No','','542','Maryland Avenue','14451'),

(4,'No','','542','Maryland Avenue','14451'),

(5,'No','','542','Maryland Avenue','14451'),

(6,'No','','542','Maryland Avenue','14451'),

(7,'No','','542','Maryland Avenue','14451'),

(8,'No','','542','Maryland Avenue','14451'),

(9,'Yes','','542','Maryland Avenue','14451'),

(10,'No','','542','Maryland Avenue','14451'),

(11,'Yes','Only in Porch','542','Maryland Avenue','14451'),

(1,'3','','972','Lancaster Avenue','13205'),

(2,'2','','972','Lancaster Avenue','13205'),

(3,'Yes','Dogs/Cats','972','Lancaster Avenue','13205'),

(4,'Yes','','972','Lancaster Avenue','13205'),

(5,'Yes','','972','Lancaster Avenue','13205'),

(6,'Yes','','972','Lancaster Avenue','13205'),

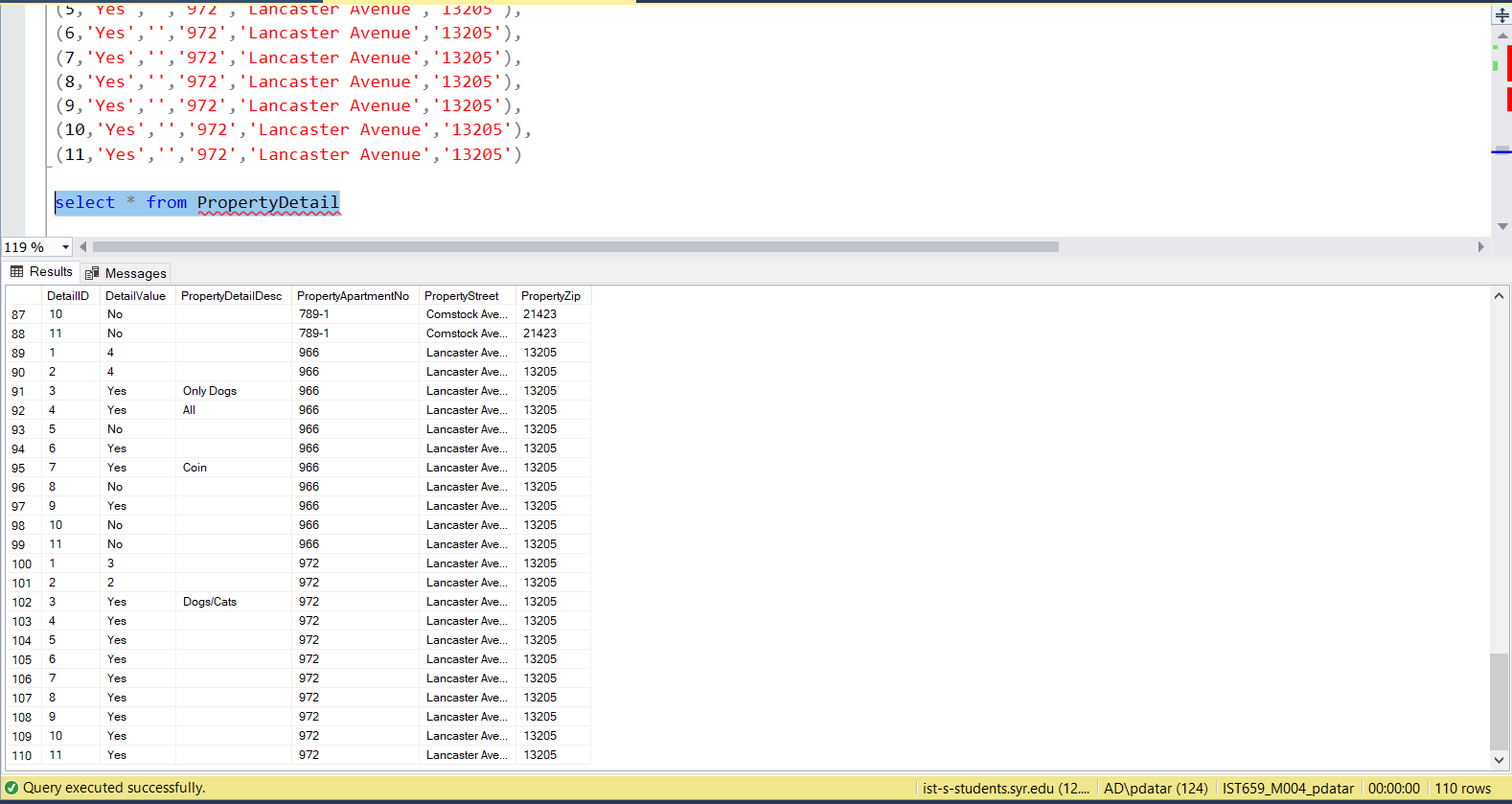
(7,'Yes','','972','Lancaster Avenue','13205'),

(8,'Yes','','972','Lancaster Avenue','13205'),

(9,'Yes','','972','Lancaster Avenue','13205'),

(10,'Yes','','972','Lancaster Avenue','13205'),

(11,'Yes','','972','Lancaster Avenue','13205')



Major Data Questions

1. **Which is the most viewed Property?**

SELECT TOP 1 WITH TIES PropertyApartmentNo, PropertyStreet, PropertyZip, COUNT(\*) AS 'UserPropertyViewCount'

-- from respective tables

FROM PropertyView

-- GROUP BY PropertyApartmentNo, PropertyStreet, PropertyZip

GROUP BY PropertyApartmentNo, PropertyStreet, PropertyZip

-- using ORDER BY to sort the result of groupby in descending order

ORDER BY UserPropertyViewCount DESC;

A screenshot of a social media post

Description automatically generated

1. **What is the Average Lease Price By Landlord per Year?**

SELECT La.LandlordFirstName, La.LandlordLastName, YEAR(L.LeaseIssueDate) AS 'YearSigned', AVG(L.RentPerBedroom) AS 'AVG\_Rent'

-- from respective tables

FROM Landlord La, Lease L

-- where condition to filter rows which are not needed in view

WHERE La.LandlordID = L.LandlordID

-- GROUP by La.LandlordID, La.LandlordFirstName, La.LandlordLastName, IssueYear

GROUP BY La.LandlordID, La.LandlordFirstName, La.LandlordLastName,YEAR(L.LeaseIssueDate)

ORDER BY La.LandlordFirstName

A screenshot of a social media post

Description automatically generated

1. **What are the Number of Lease Signed By Landlord per Year?**

SELECT La.LandlordFirstName, La.LandlordLastName, YEAR(L.LeaseIssueDate) AS 'YearSigned', COUNT(\*) AS 'Lease\_signed'

-- from respective tables

FROM Landlord La, Lease L

-- where condition to filter rows which are not needed in view

WHERE La.LandlordID = L.LandlordID

-- GROUP by La.LandlordID, La.LandlordFirstName, La.LandlordLastName

GROUP BY La.LandlordID, La.LandlordFirstName, La.LandlordLastName,YEAR(L.LeaseIssueDate)

ORDER BY La.LandlordFirstName

A screenshot of a social media post

Description automatically generated

1. **Which is the Most Expensive Lease Signed in History?**

SELECT TOP 1 WITH TIES PropertyApartmentNo, PropertyStreet, PropertyZip, MAX(RentPerBedroom) AS 'MaxRentListing'

-- from respective tables

FROM Lease, Landlord

-- GROUP BY PropertyApartmentNo, PropertyStreet, PropertyZip

GROUP BY PropertyApartmentNo, PropertyStreet, PropertyZip

-- using ORDER BY to sort the result of groupby in descending order

ORDER BY MaxRentListing DESC;

A screenshot of a social media post

Description automatically generated

1. **Which type of Property is rented the most?**

SELECT TOP 1 WITH TIES PropertyTypeName, COUNT(\*) AS 'No\_ofLease'

-- from respective tables

FROM PropertyType P, Lease L

-- where condition to filter rows which are not needed in view

WHERE P.PropertyApartmentNo = L.PropertyApartmentNo AND P.PropertyStreet = L.PropertyStreet AND P.PropertyZip = L.PropertyZip

-- GROUP BY PropertyTypeName

GROUP BY PropertyTypeName

-- using ORDER BY to sort the result of groupby in descending order

ORDER BY No\_ofLease DESC;

**A screenshot of a social media post

Description automatically generated**

1. **Which month most of the properties are rented?**

SELECT TOP 1 WITH TIES MONTH(LeaseIssueDate) AS 'Month', Count(\*) AS 'No\_LeaseSigned'

FROM Lease

GROUP BY MONTH(LeaseIssueDate)

ORDER BY No\_LeaseSigned DESC;

A screenshot of a social media post

Description automatically generated

1. **Which landlord has the most lease signings?**

SELECT TOP 1 WITH TIES La.LandlordFirstName, La.LandlordLastName, COUNT(\*) AS 'LeaseSigned'

-- from respective tables

FROM Landlord La, Lease L

-- where condition to filter rows which are not needed in view

WHERE La.LandlordID = L.LandlordID

-- GROUP by La.LandlordID, La.LandlordFirstName, La.LandlordLastName

GROUP BY La.LandlordID, La.LandlordFirstName, La.LandlordLastName

-- using ORDER BY to sort the result of groupby in Descending order

ORDER BY LeaseSigned DESC;

A screenshot of a social media post

Description automatically generated

1. **What is the geographic area wise Average Price of houses per year?**

SELECT PropertyStreet,PropertyZip, YEAR(LeaseIssueDate) AS 'YearSigned', AVG(RentPerBedroom) AS 'AVG\_Price'

-- from respective tables

FROM Lease

-- GROUP by La.LandlordID, La.LandlordFirstName, La.LandlordLastName

GROUP BY PropertyStreet, PropertyZip, YEAR(LeaseIssueDate)

ORDER BY PropertyStreet

A screenshot of a social media post

Description automatically generated

1. **What are the Pet friendly listings?**

SELECT P.PropertyApartmentNo, P.PropertyStreet, P.PropertyZip

FROM Property P

JOIN PropertyDetail PD

ON P.PropertyApartmentNo = PD.PropertyApartmentNo AND P.PropertyStreet = PD.PropertyStreet AND P.PropertyZip = PD.PropertyZip

JOIN detail D

ON PD.DetailID = D.DetailID

WHERE PD.DetailID = (SELECT DetailID FROM Detail WHERE DetailName = 'Pets Allowed') AND PD.DetailValue = 'Yes'

A screenshot of a social media post

Description automatically generated

1. **What are the houses with 5 Bedrooms and 3 Baths?**

SELECT A.PropertyApartmentNo, A.PropertyStreet, A.PropertyZip

FROM

(SELECT P.PropertyApartmentNo, P.PropertyStreet, P.PropertyZip

FROM Property P

JOIN PropertyDetail PD

ON P.PropertyApartmentNo = PD.PropertyApartmentNo AND P.PropertyStreet = PD.PropertyStreet AND P.PropertyZip = PD.PropertyZip

JOIN detail D

ON PD.DetailID = D.DetailID

WHERE (PD.DetailID = (SELECT DetailID FROM Detail WHERE DetailName = 'No.of Beds') AND PD.DetailValue = '5')) AS A

JOIN

(SELECT P.PropertyApartmentNo, P.PropertyStreet, P.PropertyZip

FROM Property P

JOIN PropertyDetail PD

ON P.PropertyApartmentNo = PD.PropertyApartmentNo AND P.PropertyStreet = PD.PropertyStreet AND P.PropertyZip = PD.PropertyZip

JOIN detail D

ON PD.DetailID = D.DetailID

WHERE (PD.DetailID = (SELECT DetailID FROM Detail WHERE DetailName = 'No.of Baths') AND PD.DetailValue = '3')) AS B

ON A.PropertyApartmentNo = B.PropertyApartmentNo AND A.PropertyStreet = B.PropertyStreet AND A.PropertyZip = B.PropertyZip

A screenshot of a social media post

Description automatically generated

1. **What is the Number of Lease Signed per Landlord?**

SELECT La.LandlordID, La.LandlordFirstName, La.LandlordLastName, COUNT(\*) AS 'LeaseSigned'

-- from respective tables

FROM Landlord La, Lease L

-- where condition to filter rows which are not needed in view

WHERE La.LandlordID = L.LandlordID

-- GROUP by La.LandlordID, La.LandlordFirstName, La.LandlordLastName

GROUP BY La.LandlordID, La.LandlordFirstName, La.LandlordLastName

A screenshot of a social media post

Description automatically generated

1. **What is the Number of Listings per Landlord?**

SELECT P.LANDLORDID, LandlordFirstName, LandlordLastName, COUNT(\*) AS 'No\_ofListings'

FROM PROPERTY P

JOIN LANDLORD L

ON P.LANDLORDID = L.LANDLORDID

GROUP BY P.LANDLORDID, LandlordFirstName, LandlordLastName

A screenshot of a social media post

Description automatically generated

1. **How many Lease Signed per Year?**

SELECT Year(LeaseIssueDate) AS 'YearSigned', Count(\*) AS 'LeaseSigned'

FROM Lease

GROUP BY Year(LeaseIssueDate)

A screenshot of a computer

Description automatically generated

Interfaces

There are two types of people Visiting Syracuse Listing Management Database System–

1. User

2. Landlord

The Welcome page provides two choices for people which is User or Landlord.

A User or a Landlord can click the choice he wants to see further options.

A screenshot of a social media post

Description automatically generated

The system will ask whether the user is an Existing User, or does he want to Register.

A close up of a logo

Description automatically generated

If the Person Clicks on Landlord, the system will ask whether the person is an Existing Landlord, or does he want to Register.

A picture containing screenshot

Description automatically generated

If a User is an Existing User, then a new Window will Open with all the available options.

A picture containing indoor, living, sofa, screenshot

Description automatically generated

If a Landlord is an Existing Landlord, then a new Window will Open with all the available options.

A screenshot of a social media post

Description automatically generated

Forms and Reports

**User Information Creation Portal**

If the person is a new User and he/she wants to register, then the system will take the new User to the New User Registration Portal.

A screenshot of a cell phone

Description automatically generated

**Landlord Information Creation Portal**

If the person is a new Landlord and he/she wants to register, then the system will take the new User to the New Landlord Registration Portal.

A screenshot of a cell phone

Description automatically generated

For an existing User, following are the options which he can click on:

1. Trending Listings
2. Favorite Property
3. View Property
4. Add/Remove Favorite Property
5. User Favorite Summary
6. *Trending Listings*

Once a User Clicks on Trending Listings, a Report will be generated based on the Properties Users are visiting. The report will show Top 5 Trending Listings

A screenshot of a cell phone

Description automatically generated

1. *Favorite Property*

If the User clicks on Favorite Property then a report will be generated and he/she will be able to see the favorite listings by users. The report has the UserID for users favoriting a listing and along with that, it has the FavoriteDateTime to let the user know when the User has favorited the listing.

A screenshot of a cell phone

Description automatically generated

1. *View Property*

When a User Clicks on View Property, a report will be generated and then he/she will be able to all the available listings on the system. In the report the user will also be able to see the type of the house for e.g., Rowhouse, Apartment, House, etc. Every House has 11 Details associated with it. A user will be able to see all the Details such as whether the Place has a Dishwasher, Wifi, or is the Place a pet friendly place or how many beds or baths does the place has.

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

1. *Add/Remove Favorite Property*

A user can Add a new property as a Favorite property, or he might also remove the property from his favorites tab. For the same, there are two buttons provided which does Adds or Removes Favorite Listing accordingly.

A screenshot of a cell phone

Description automatically generated

1. *User Favorite Summary*

If the User clicks on User Favorite Summary, a report will be generated and he/she will be able to see all the listings which users have favorited along with the properties entire details.

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

For an existing Landlord, following are the options which he can click on:

1. Add/Remove Listings
2. View, Add, Remove Lease
3. Landlord Details
4. Landlord Lease Listing Summary
5. *Add/Remove Listings*

When a landlord has to remove or add listing he can click on this button and a form opens up wherein the landlord can detail of the Property he wants to Add or Remove from the listing.

*A screenshot of a cell phone

Description automatically generated*

1. *View, Add, Remove Lease*

If a landlord wants to Add a new lease or remove the existing lease from the system, he can click on Add lease or Remove Lease respectively. Landlord just needs to enter the lease details.

*A screenshot of a cell phone

Description automatically generated*

1. *Landlord Details*

If a landlord wants to check Landlord Details, he/she can click on this button and a form will open. There is a drop down selection of landlordID and when a landlord selects the respective landlord, he/she will be able to see the details of the same.

A screenshot of a cell phone

Description automatically generated

1. *Landlord Lease Listing Summary*

When the landlord clicks on *Landlord Lease Listing Summary,* a report will be generated which will showcase the landlord summary such as the First Name and the Last Name. Along with it there would be count of Lease Signed and No of Listings for the particular landlord.

*A screenshot of a cell phone

Description automatically generated*