

Marketing Database Management & Visualization

Market Analysis on Real Estate in Sacramento, CA

DATA MANIPULATION & ANALYSIS ON SQL
AND VISUALIZATION ON TABLEAU

Phuong (Lucy) Doan

Table of Contents

I.	Problem Description.....	3
1.1.	Business Objective.....	3
1.2.	Data Mining Objective	3
II.	Analytical Questions.....	4
2.1.	Data Preparation.....	4
	• Loading datasets	4
	• Converting data type, handling data errors and creating new calculated columns	4
	• Handling geospatial data	7
	• Designing data schema	7
2.2.	Analytics Tools.....	9
	• Tool for data querying, manipulation and analysis.....	9
	• Tool for data visualization	10
2.3.	Business Analysis.....	11
	• Average Sales Price of homes for each Zip Code.....	11
	• The Ratio of Average Home Price to Median Household Income	11
	• The customers within 20-mile radius of our Real Estate Office.....	12
III.	Managerial Implications	14
3.1.	Conclusions.....	14
3.2.	Recommendations.....	14
	APPENDICES	15
	Appendix 1. Importing the csv data file, SacRealEstate, into SQL Server	15
	Appendix 2. Handling (unmeaningful) values of unconvertible data type	17
	Appendix 3. Process of investigating and removing duplicate rows	18
	Appendix 4. A guide on question topic – report section – SQL file matching	20

I. Problem Description

1.1. Business Objective

Like many other counties in California, Sacramento is one of the most competitive real estate markets of the United States. According to statistics of Redfin (2020), a leading real estate agent in the US, the average home price in Sacramento, US is \$360,000 dollars, up by almost 7% compared to last year and equivalent to 125% national average price. More impressively, its sales price per square foot reaches \$252, increasing by 12% since last year. All of these prove the potential of this market; In fact, ABCD Real Estate Co. has faced with harsh competition in Sacramento. In order to optimize our resources, the company should target right customer segments and deliver customized messages to them. With that in mind, we compile this report garnering profile of ABCD's customers in Sacramento county, including their demographics and transaction details, to derive meaningful insights driving actionable recommendations and strategic plans in next phase.

1.2. Data Mining Objective

This report attempts to analyze customer profiles and transaction data to obtain better understanding about ABCD's customers, thus providing input to tailor marketing messages. The raw data is stored in 2 datasets, and is queried, manipulated and analyzed using SQL. The first dataset, SacRealEstate, contains 986 records which are the past transactions conducted in 2008, with information about house address (street, city, zip, geographic location), house information (number of beds, number of baths, square feet, type) and transaction details (sales date, price). The second dataset, HHIncome, contains median and mean household income of 32,634 zip codes in the US during the 2006-2010 period. Our assumption is the data, if not null or having data type error, are all accurate. A detailed description of the data is as following.

Table Name	Variable Name	Description	Type
SacRealEstate	street	Street address of the house	Varchar (50)
	city	City where the house is	Varchar (50)
	zip	Zip code of the house location	Varchar (50)
	state	State where the house is	Varchar (50)
	beds	Number of beds in the house	Varchar (50)
	baths	Number of baths in the house	Varchar (50)
	sq__ft	Square feet of the house	Varchar (50)
	type	Type of the house	Varchar (50)
	sale date	The date the house was sold	Varchar (50)
	price	Sale price of the house	Varchar (50)
	latitude	Geographic location of the house, y_coordinate	Varchar (50)
	longitude	Geographic location of the house, x_coordinate	Varchar (50)
HHIncome	Zip	Zip code in the United States	Varchar (50)
	Median	Median household income in the zip code	Varchar (50)
	Mean	Mean household income in the zip code	Varchar (50)
	Pop	Total population in the zip code	Varchar (50)

II. Analytical Questions

2.1. Data Preparation

- **Loading datasets**

Both datasets are imported into SQL Server using SQL Server Import and Export Wizard. The main steps are as following:

- Connecting to server from SQL Server Management Studio (COB-MKTSQL)
- Expand **Databases**, right-click on the database where imported data will be stored in (“Student_FS2020”), point to **Tasks**, then click **Import Data...**
- Choose corresponding data source, browse for its path, go through customization steps (optional), destination location, then finish the process.

Although the first and second datasets are of different types of file, csv and Microsoft Excel respectively, their data importing processes are similar for the most part.

See Appendix 1 for illustration on importing the first dataset, SacRealEstate, a csv file.

- **Converting data type, handling data errors and creating new calculated columns**

- Changing data types and dealing with related data errors

Changing data types always comes with handling erroneous data, such as unmeaningful values. For example, a string of “abc” in a numeric field of counting values is an example of such data errors.

Originally both datasets are loaded in varchar(50) data type; therefore, we convert some fields into desired data type based on its domain meaning.

```
----Change Data Type of second dataset

ALTER TABLE Student_FS2020.[UM-AD\ptd9pk].HHIncome
ALTER COLUMN Zip VARCHAR(50) NOT NULL

ALTER TABLE Student_FS2020.[UM-AD\ptd9pk].HHIncome
ALTER COLUMN Median INTEGER

ALTER TABLE Student_FS2020.[UM-AD\ptd9pk].HHIncome
ALTER COLUMN Mean INTEGER

ALTER TABLE Student_FS2020.[UM-AD\ptd9pk].HHIncome
ALTER COLUMN Pop INTEGER
```

For the first dataset, we convert latitude and longitude fields into float type because they are in decimal form, while converting baths, sq_ft, price, and beds fields into integer type.

----Change Data Type of first dataset

```
ALTER TABLE Student_FS2020.[UM-AD\ptd9pk].SacRealEstate
ALTER COLUMN street VARCHAR(50) NOT NULL --Convert the desired PK column "street" from
nullable to not null, so that we can assign primary key to this field later on

ALTER TABLE Student_FS2020.[UM-AD\ptd9pk].SacRealEstate
ALTER COLUMN latitude FLOAT

ALTER TABLE Student_FS2020.[UM-AD\ptd9pk].SacRealEstate
ALTER COLUMN longitude FLOAT

ALTER TABLE Student_FS2020.[UM-AD\ptd9pk].SacRealEstate
ALTER COLUMN baths INTEGER

ALTER TABLE Student_FS2020.[UM-AD\ptd9pk].SacRealEstate
ALTER COLUMN sq__ft INTEGER

ALTER TABLE Student_FS2020.[UM-AD\ptd9pk].SacRealEstate
ALTER COLUMN price INTEGER

ALTER TABLE Student_FS2020.[UM-AD\ptd9pk].SacRealEstate
ALTER COLUMN beds INTEGER --We cannot convert data type of "beds" field to integer
because there is one varchar value in this field
```

The data type conversion for beds field cannot be implemented because there is an “F” value in this column. See Appendix 2 on how to fix this issue.

Finally, we convert sale_date field to datetime type. Originally the values in sale_date field is not in the format that is readily to be converted to datetime type. Therefore, now we create a new calculated field "sell_date" from "sale_date" field with better organized date and time order

--Convert the sale_date field to datetime type

```
ALTER TABLE Student_FS2020.[UM-AD\ptd9pk].SacRealEstate
ADD sell_date AS (CONVERT(datetime, SUBSTRING(sale_date, 5, 6) + ' '
+ RIGHT(sale_date, 4) + ' ' + SUBSTRING(sale_date, 12, 8)))

ALTER TABLE Student_FS2020.[UM-AD\ptd9pk].SacRealEstate
ADD saledate datetime

UPDATE Student_FS2020.[UM-AD\ptd9pk].SacRealEstate
SET saledate = sell_date

ALTER TABLE Student_FS2020.[UM-AD\ptd9pk].SacRealEstate
DROP COLUMN sell_date

ALTER TABLE Student_FS2020.[UM-AD\ptd9pk].SacRealEstate
DROP COLUMN sale_date
```

```
USE Student_FS2020;
GO
EXEC sp_rename '[UM-AD\ptd9pk].SacRealEstate.saledate', 'sale_date', 'COLUMN';
GO
```

- Assigning primary keys and handling duplicate values

Primary keys cannot be assigned to a field if that field is nullable, contains null values or duplicates values.

The second dataset, HHIncome, is about median and mean household income for each zip code in the US; therefore, zip code should be primary key.

```
--Assign Primary Key to Zip field in second dataset, HHIncome

ALTER TABLE Student_FS2020.[UM-AD\ptd9pk].HHIncome
ADD PRIMARY KEY (Zip)
```

In the first dataset, SacRealEstate, since each record is a customer with unique house address, we should assign primary key to street field.

```
----Assign Primary Key to street field in first dataset, SacRealEstate

ALTER TABLE Student_FS2020.[UM-AD\ptd9pk].SacRealEstate
ADD PRIMARY KEY (street)  --Cannot assign PK to this column because it has duplicate
values
```

The primary key cannot be assigned to street field because this field has duplicate values. See Appendix 3 for the steps of investigating and removing duplicate rows properly.

- Handling unmeaningful values

A value is unmeaningful when there is no such a thing in real life. In the first dataset, “0” in sq__ft field is such a case. The solution to fix this issue should depend on how many % such unmeaningful values contained in sq__ft field. There are 170 out of 981 “0” values on sq__ft field, equivalent to approximately 17%. This is way too high, so we will not discard these 170 rows to prevent a major loss of information in other fields. A safe and temporary solution is to change the name of this column to sq__ft_deleted.

```
--Change the name of "sq__ft" field to "sq__ft_deleted"
USE Student_FS2020;
GO
EXEC sp_rename '[UM-AD\ptd9pk].SacRealEstate.sq__ft', 'sq__ft_deleted', 'COLUMN';
GO
```

- **Handling geospatial data**

```
--Create a geography field from "latitude" and "longitude" geometry field

ALTER TABLE Student_FS2020.[UM-AD\ptd9pk].SacRealEstate
ADD GeoLocation GEOGRAPHY

UPDATE Student_FS2020.[UM-AD\ptd9pk].SacRealEstate
SET GeoLocation = geography::STPointFromText('POINT(' + CAST(longitude AS VARCHAR(20))
+ ' ' + CAST(latitude AS VARCHAR(20)) + ')', 4326)
```

Now our first dataset, SacRealEstate, has 981 rows with desired data type as following.

	TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	COLUMN_NAME	ORDINAL_POSITION	COLUMN_DEFAULT	IS_NULLABLE	DATA_TYPE	CHARACTER_MAXIMUM_LENGTH
1	Student_FS2020	UM-AD\ptd9pk	SacRealEstate	street	1	NULL	NO	varchar	50
2	Student_FS2020	UM-AD\ptd9pk	SacRealEstate	city	2	NULL	YES	varchar	50
3	Student_FS2020	UM-AD\ptd9pk	SacRealEstate	zip	3	NULL	YES	varchar	50
4	Student_FS2020	UM-AD\ptd9pk	SacRealEstate	state	4	NULL	YES	varchar	50
5	Student_FS2020	UM-AD\ptd9pk	SacRealEstate	beds	5	NULL	YES	int	NULL
6	Student_FS2020	UM-AD\ptd9pk	SacRealEstate	baths	6	NULL	YES	int	NULL
7	Student_FS2020	UM-AD\ptd9pk	SacRealEstate	sq_ft	7	NULL	YES	int	NULL
8	Student_FS2020	UM-AD\ptd9pk	SacRealEstate	type	8	NULL	YES	varchar	50
9	Student_FS2020	UM-AD\ptd9pk	SacRealEstate	sale_date	9	NULL	YES	datetime	NULL
10	Student_FS2020	UM-AD\ptd9pk	SacRealEstate	price	10	NULL	YES	int	NULL
11	Student_FS2020	UM-AD\ptd9pk	SacRealEstate	latitude	11	NULL	YES	float	NULL
12	Student_FS2020	UM-AD\ptd9pk	SacRealEstate	longitude	12	NULL	YES	float	NULL
13	Student_FS2020	UM-AD\ptd9pk	SacRealEstate	GeoLocation	13	NULL	YES	geography	-1

Query executed successfully. COB-MKTSQL (13.0 SP1) UM-AD\ptd9pk (56) Student_FS2020 00:00:00 13 rows

- **Designing data schema**

For the purpose of joining two datasets to obtain median household income in each zip code, we need to assign the pair of foreign key – primary key through the paired columns of zip code. Before that, let's make sure the data type of all fields are appropriate and primary keys are not nullable.

For the first dataset, SacRealEstate, procedure of modifying data type is already described in previous section. The only thing left is to make sure the foreign key in first dataset, zip field, has no Null/Blank/Space values. Sorting the field will help identify Null/Blank/Space values if any.

```
SELECT *
FROM Student_FS2020.[UM-AD\ptd9pk].SacRealEstate
ORDER BY zip
```

```
--The result show there is one blank value in zip field.
--My solution is extracting the full address (street + city + state),
--then search for its zip code accordingly.
--Google Maps shows this address has zip code of '95828'
```

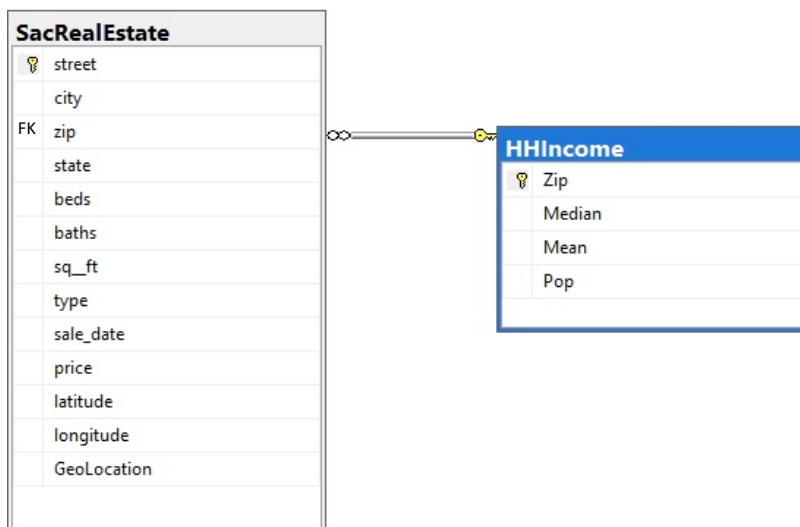
```
UPDATE Student_FS2020.[UM-AD\ptd9pk].SacRealEstate
SET zip = '95828'
WHERE street = '7342 DAVE ST'
```

```
--Link Zip field in HHIncome table as a Foreign Key with zip field in SacRealEstate table
```

```
ALTER TABLE Student_FS2020.[UM-AD\ptd9pk].SacRealEstate WITH NOCHECK
ADD CONSTRAINT FK_RealEstate_HHIncome
FOREIGN KEY (zip) REFERENCES Student_FS2020.[UM-AD\ptd9pk].HHIncome(Zip);
```

On a side note, “WITH NOCHECK” is added in the query in order to avoid the error due to not all zip codes in HHIncome dataset match perfectly with zip codes in SacRealEstate dataset. In fact, the former dataset contains zip codes all over the US, which are way more than the zip codes within Sacramento, CA in the latter dataset.

This is the final Data Model for analysis.



2.2. Analytics Tools

- **Tool for data querying, manipulation and analysis**

- SQL Management Server Studio (SSMS)

I choose SQL Server, a relational database management system of Microsoft, to work on the real estate data. Both functions, DDL (data definition language) and DML (data manipulation language) are utilized to create the data schema, with the commands CREATE, ALTER, DROP, etc. and to manipulate and populate data, with the commands SELECT, UPDATE, INSERT, DELETE, etc. Considering the data and analysis nature of this real estate report, SQL Server is relatively a good option. On the one hand, SQL can handle 3V's of data (volume, velocity, variety) than other drag-and-drop tools of Microsoft such as Excel, Access. On the other hand, our analysis is pretty attainable and not so complicated, e.g. regression modeling; therefore, SQL Server is more popular and appropriate than such high-level programming languages as Python or R. This means, using SQL with the queries stored, it will be easier for business leaders to retrieve and perform further analysis for decision-making purpose.

Regarding platform, SQL Management Server Studio (SSMS) is chosen to access and work on the data. Compared to cloud-based database services, such as MS Azure SQL or Amazon Aurora, SSMS has both advantages and disadvantages. The comparison among these three platforms can reveal about SSMS' pros and cons.

Criteria	SSMS	MS Azure SQL	Amazon Aurora
Nature	On-premises software	Software as a Service (SaaS) on cloud	SaaS on cloud
Database management system	SQL Server	SQL Server	MySQL, PostgreSQL
Clustered index	Optional	Required	Optional
DB-Engines ranking (2020)	#3	#23	#46

The biggest shortcoming of SSMS might be it requires users to have the software installed in their computer. Therefore, other on-cloud SaaS platforms are more convenient to work on, especially for Mac users. In my case, although I am using Mac, I am still able to connect with SSMS thanks to VPN software and Microsoft Remote Desktop to connect to Cornell's Lab of University of Missouri.

However, SSMS should be more popular than Amazon Aurora because SSMS runs SQL Server language. Also, SSMS proves a huge advantage when ranking at #3, dominating MS Azure and Amazon Aurora ranking at #23 and #46, respectively, based on DB-Engines ranking (2020). The ranking is determined based on a variety of factors, e.g. number of mentions and technical discussions on the Internet, number of searches on Google, number of times it is mentioned on job offers, etc. For these reasons, I decided to pick SSMS.

- **Tool for data visualization**

- Tableau software

Tableau is a data visualization software that allows users to visualize a large volume of quantitative and qualitative information with interactive features, nice graphics and very user-friendly. The visualization part in this report is delivered on Tableau platform.

When thinking of data visualization with decent volume of raw data, the top-of-mind options of an analysts are usually Microsoft Excel, Tableau, and Power BI.

Criteria	Tableau	Power BI	MS Excel
Visualization capabilities	Perfect graphics	Easy-to-use	Less attractive visuals
Data analysis features	Diverse built-in features, even for complicated calculations	Diverse built-in features, even for complicated calculations	No built-in features for complicated calculations
Connect with SQL Server data?	Yes, very fast and allowing Custom SQL Query	Yes, very handy	Yes, but through MS Query
Building a map?	Yes. Within a couple of clicks	Yes. Within a couple of clicks	Yes, but less user-friendly
Support community	Big community all over the world and increasing over time	Fewer discussion topics than Tableau	Gradually less support since users are leaving (for data viz purpose)
Pricing	Free 1-year license for students, 14 days for others	60 days trial for Power BI Pro	Free

Although it might not be as popular with students and office workers as Excel, Tableau provides various benefits to users. First of all, Tableau is well-known for its fantastic graphics and visual effects. Tableau users have a wide range of choices regarding chart types, formats, and interact with dashboards using multiple filter options. Similarly, data analysis capability is much easier on Tableau with the feature to create new calculated fields and many types of table calculations. Furthermore, since we need to build map later on in the report, Tableau is a good choice. Last but not least, Tableau offers an entire one year of freemium for students and has a huge support community, thus helping users access this tool very early in their career and convenient with available support.

2.3.Business Analysis

- **Average Sales Price of homes for each Zip Code**

Although home price in Sacramento, CA is 25% higher than national average, there is still large discrepancy among each house. Therefore, let's see how each zip code differs from each other in terms of average home's sale price.

```
--<A> Average Sales Price of homes for each Zip Code?
```

```
SELECT zip, AVG(price) AS Avg_SalesPrice_PerZip
FROM Student_FS2020.[UM-AD\ptd9pk].SacRealEstate
GROUP BY zip
```



The screenshot shows a SQL Server query results window with two tabs: 'Results' and 'Messages'. The 'Results' tab is active, displaying a table with two columns: 'zip' and 'Avg_SalesPrice_PerZip'. The table contains 10 rows of data. The status bar at the bottom indicates 'Query executed successfully.' and provides details about the connection and execution time.

	zip	Avg_SalesPrice_PerZip
1	95742	350009
2	95822	157677
3	95838	149293
4	95757	338334
5	95667	363863
6	95661	360903
7	95603	405890
8	95635	395000
9	95842	143281
10	95630	414960

Query executed successfully. | COB-MKTSQL (13.0 SP1) | UM-AD\ptd9pk (51) | Student_FS2020 | 00:00:00 | 68 rows

As expected, some zip codes have really high average house's sale price, such as the zip code 95630 at \$414,960 per house, whereas the zip code 95842 at roughly a third of that, \$143,281 per house.

- **The Ratio of Average Home Price to Median Household Income**

The ratio of (Average Home's Sales Price) / (Household Median Income) can tell us the affordability of ABCD's customers. When this ratio is too high, it signals financial risk and legal risk, because such customers are likely to default on their mortgage loans.

To find the zip code with highest ratio, we first join the subquery used above (average sales price by zip code) with the second dataset (median household income by zip codes in the US), with zip code as mutual column. The resulting joined table has 68 rows, the same as above query; this means all zip codes in first dataset are found in the second dataset. Then, we create a new calculated field by dividing average sales price per zip by median household income per zip, sort by the ratio column in descending order, and choose top 1.

```
--<B> 10 Zip Codes with Highest Ratio of Avg. Home's Price to Household Median Income
```

```
SELECT TOP 10 Avg_HomePrice.zip, Avg_SalesPrice_PerZip,
              Median AS Median_HHIncome_PerZip,
              (Avg_SalesPrice_PerZip/Median) AS Ratio
FROM (
    SELECT zip, AVG(price) AS Avg_SalesPrice_PerZip
    FROM Student_FS2020.[UM-AD\ptd9pk].SacRealEstate
    GROUP BY zip
) AS Avg_HomePrice
INNER JOIN Student_FS2020.[UM-AD\ptd9pk].HHIncome AS Median_HHIncome
ON Avg_HomePrice.zip = Median_HHIncome.Zip
ORDER BY Ratio DESC
```

	zip	Avg_SalesPrice_PerZip	Median_HHIncome_PerZip	Ratio
1	95811	403474	28473	14.1700000000000
2	95814	367728	26668	13.7900000000000
3	95633	490000	61035	8.0300000000000
4	95690	380000	50723	7.4900000000000
5	95819	465750	67740	6.8800000000000
6	95816	348750	50669	6.8800000000000
7	95693	617508	91431	6.7500000000000
8	95603	405890	60642	6.6900000000000
9	95650	567000	85269	6.6500000000000
10	95635	395000	61880	6.3800000000000

Top the list is zip code 95811 with the ratio at 14. It means that average home price in this area equals 14 times median household income of its residents. This is too high of a ratio, compared to 3 or 4 times which is the national average ratio (Longtermrends, 2020). We should investigate such absurd cases regarding underlying reasons and may adjust our policies accordingly to prevent any potential risk.

- **The customers within 20-mile radius of our Real Estate Office**

The location of houses sold can imply several meaningful insights, one of which was about the effectiveness of the company's salesforce by location.

```
--<C> Count all the places within 20 miles from the Real Estate Office
```

```
--First, calculate distance from each place to the Real Estate Company's address
```

```
---(1315 10TH ST, SACRAMENTO, CA, 95814)
```

```
--The Real Estate Company's geometry coordinates are X: -121.494995, Y: 38.576763
```

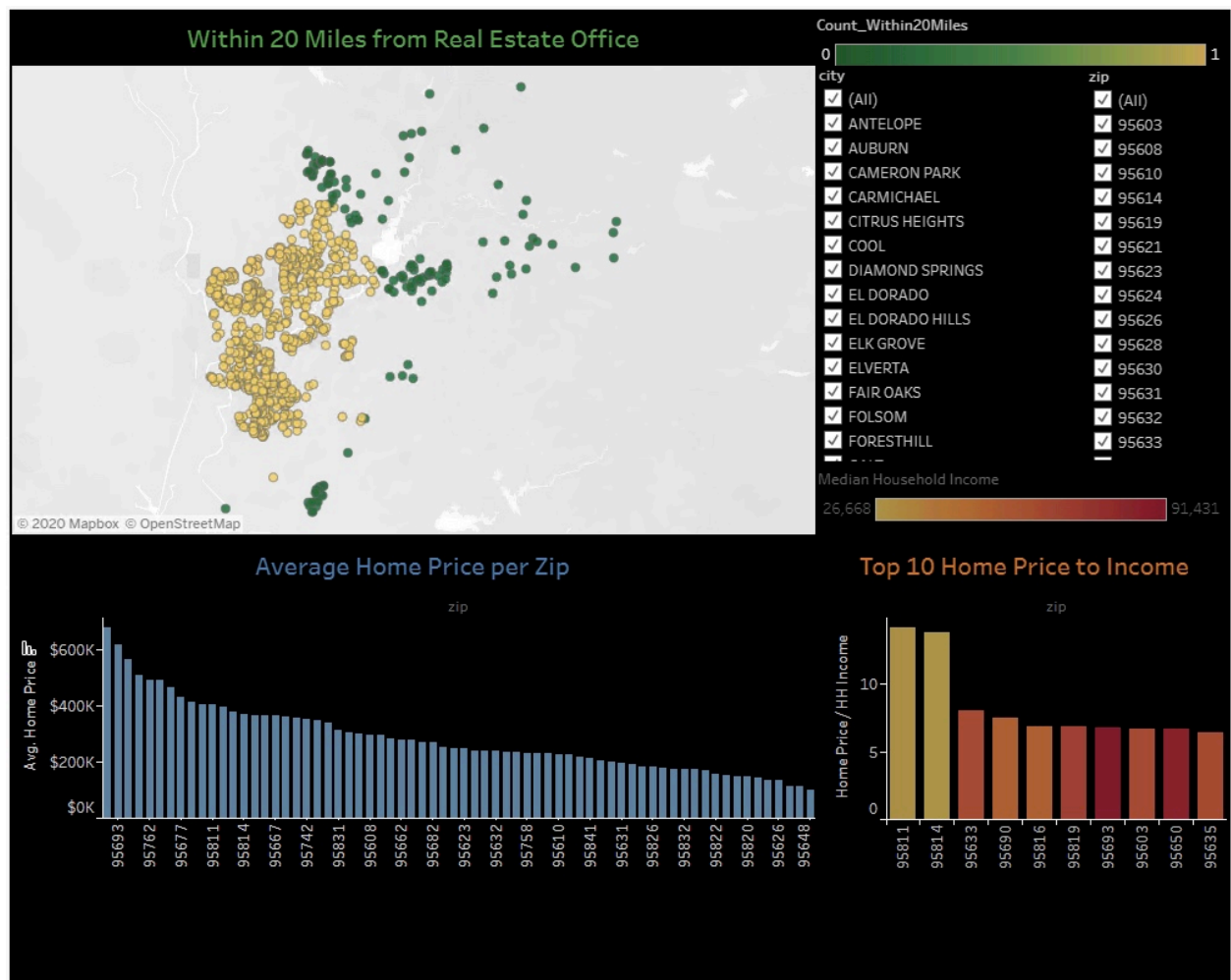
```
SELECT SUM (Count_Within20Miles) AS Count_20Miles_FromOffice
FROM (
    SELECT *, CASE WHEN Distance_Miles <= 20 THEN '1' ELSE 0 END
    AS Count_Within20Miles
    FROM (
        SELECT *,
            GeoLocation.STDistance(geography::STGeomFromText('POINT
            (-121.494995 38.576763)', 4326)) *0.000621371 AS Distance_Miles
        FROM Student_FS2020.[UM-AD\ptd9pk].SacRealEstate
        ) AS distance
    ) AS within20Miles
```

Results	Messages
Count_20Miles_FromOffice	
1	796

Query executed successfully. COB-MKTSQL (13.0 SP1) UM-AD\ptd9pk (51) Student_FS2020 00:00:00 1 rows

Recent years, ABCD Real Estate Co. has been gradually allocating more sales rep to areas outside of 20-mile radius of our office, while this past data tells us that up to 80% of the homes sold, 796 out of 981, are within 20 miles from the office. However, it is still too soon to say that the sales reallocation did not live up to expectations.

The insights related to 3 above question topics can be found in the following dashboard. On a side note, beside the attached Tableau workbook, the dashboard is also available for review on my Tableau Public page here:
https://public.tableau.com/views/SQLFinalProjectSacramentoRealEstate/Dashboard?:display_count=y&:origin=viz_share_link



III. Managerial Implications

3.1. Conclusions

As expected, the average home's sale prices per Zip code in Sacramento are very high; some even reach more than \$400,000 per house. Besides, it is worth noting about the big gap between the top home's price and the bottom price. This reflects a discrepancy in consumer's needs that we need to pay attention for market segmentation.

Another remarkable point is the crazily high ratio of home price to household median income in some areas. As seen on the dashboard, the highest ratio per zip code is 14.17, in the zip code 95811. Looking at the color band, we can tell these top ratios are not due to too high home price, but rather, due to too low median household income. As mentioned above, this implies loan default risks that can affect negatively the company's revenue.

Finally, by counting the home sales within 20 miles from ABCD Real Estate Co.'s office, we will have an indicator to evaluate salesforce's performance. Roughly 80% of the company's customers are surrounding the office, this could be either a good news or a bad news. On the bright side, with nearby customers, it will be easier for ABCD's customer services to reach out and deliver post-sales services. However, since the company has started to focus more on further areas in recent years, this somehow indicates the ineffectiveness of the salesmen assigned to those further areas.

3.2. Recommendations

Based on the insights collected above, we can realize that ABCD's market is hugely fragmented in terms of home sales price and ratio of home price to median household income. The company should take into account these metrics to make sure it targets the customers who are capable of buying its homes and only accepts the applicants that have low risks of mortgage loan default.

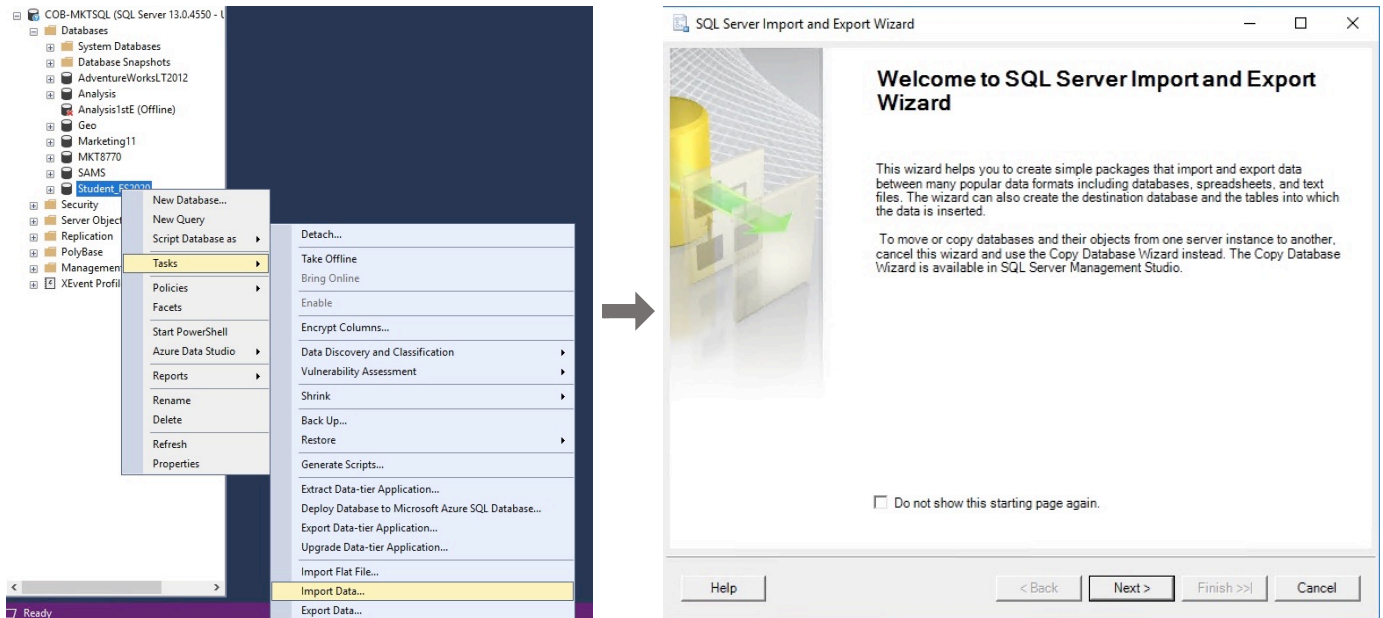
In addition to the above metrics, the company would need lots of other indicators to provide input for customer segmentation efforts. For example, it needs to start recording information about customer's financial health, such as credit score, individual and household incomes, customer's ratings and feedback. On another note, the "square feet" is among the most important data; however, it is almost useless now in ABCD's database because of too many unmeaningful values in this field. All of the information is essential for the company to segment the market, target the right customers and tailor their marketing messages matching customer's needs.

Not only consumer insights, ABCD Real Estate Co. also needs to have comprehensive understanding about the properties it is selling and follow-up information about the sales. Such key performance indicators (KPIs) are purchase price, gross income, operating expenses, vacancy rate, etc. It helps the company have a better understanding about their portfolio and offer the "right products" at "right price" to the "right customers".

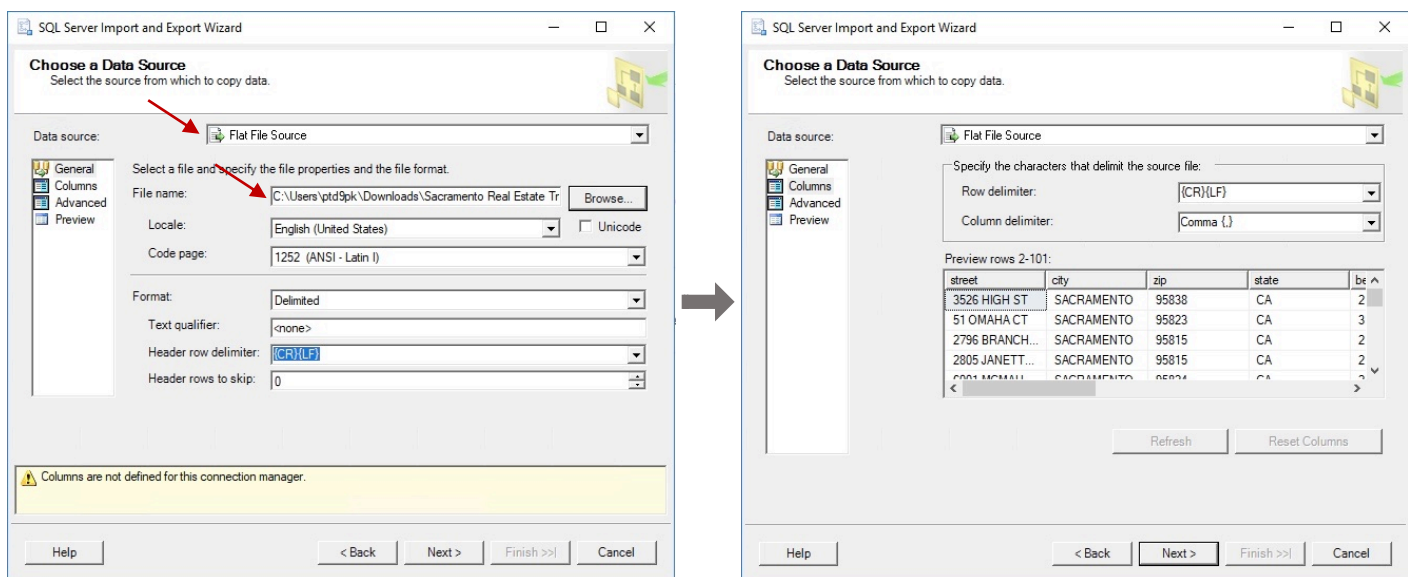
APPENDICES

Appendix 1. Importing the csv data file, SacRealEstate, into SQL Server

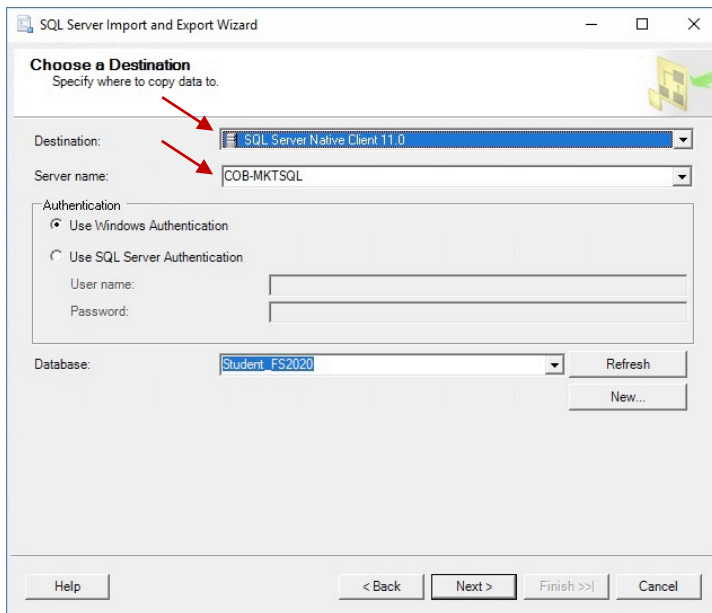
Step 1: Right click on database desired to store the imported data, point to Tasks, and click Import Data... Then, click Next to open SQL Server Import and Export Wizard



Step 2: Choose the type (Flat File Source for CSV files) and browse location where the file is currently stored. Click through Columns or Advanced options (on the left of the pane) for any customization



Step 3: Choose destination (SQL Server Native Client 11.0), Server name, and Database. Click Next, then click “Edit Mapping...” if there is a need to modify how the imported data show up in SQL (column name, data type, nullable/ not nullable). After that, run package and complete the wizard



SQL Server Import and Export Wizard

Choose a Destination
Specify where to copy data to.

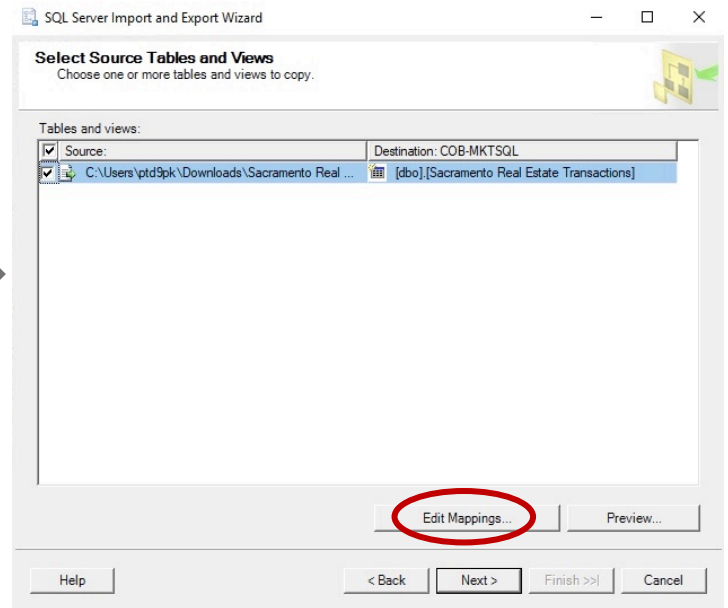
Destination: **SQL Server Native Client 11.0**

Server name: **COB-MKTSQL**

Authentication:
☒ Use Windows Authentication
☐ Use SQL Server Authentication
 User name:
 Password:

Database: **Student_FS2020** Refresh New...

Help < Back Next > Finish >> Cancel



SQL Server Import and Export Wizard

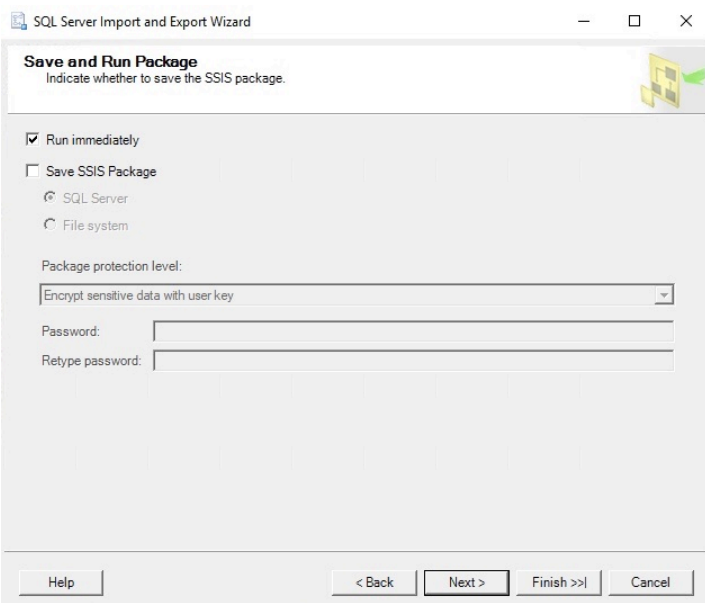
Select Source Tables and Views
Choose one or more tables and views to copy.

Tables and views:

Source	Destination
<input checked="" type="checkbox"/> C:\Users\ptd9pk\Downloads\Sacramento Real ...	<input checked="" type="checkbox"/> [dbo].[Sacramento Real Estate Transactions]

Edit Mappings... Preview...

Help < Back Next > Finish >> Cancel



SQL Server Import and Export Wizard

Save and Run Package
Indicate whether to save the SSIS package.

☒ Run immediately

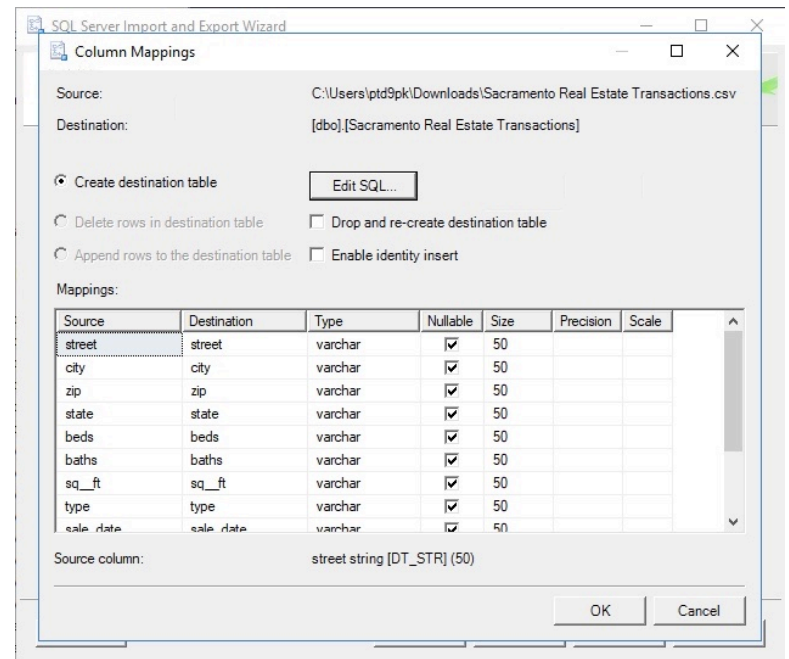
☐ Save SSIS Package

SQL Server
File system

Package protection level:
Encrypt sensitive data with user key

Password:
 Retype password:

Help < Back Next > Finish >> Cancel



SQL Server Import and Export Wizard

Column Mappings

Source: C:\Users\ptd9pk\Downloads\Sacramento Real Estate Transactions.csv
 Destination: [dbo].[Sacramento Real Estate Transactions]

☒ Create destination table Edit SQL...
☐ Delete rows in destination table ☐ Drop and re-create destination table
☐ Append rows to the destination table ☐ Enable identity insert

Mappings:

Source	Destination	Type	Nullable	Size	Precision	Scale
street	street	varchar	<input checked="" type="checkbox"/>	50		
city	city	varchar	<input checked="" type="checkbox"/>	50		
zip	zip	varchar	<input checked="" type="checkbox"/>	50		
state	state	varchar	<input checked="" type="checkbox"/>	50		
beds	beds	varchar	<input checked="" type="checkbox"/>	50		
baths	baths	varchar	<input checked="" type="checkbox"/>	50		
sq_ft	sq_ft	varchar	<input checked="" type="checkbox"/>	50		
type	type	varchar	<input checked="" type="checkbox"/>	50		
sale_date	sale_date	varchar	<input checked="" type="checkbox"/>	50		

Source column: street string [DT_STR] (50)

OK Cancel

Appendix 2. Handling (unmeaningful) values of unconvertible data type

Instead of deleting the entire row of this value, I choose to insert a meaningful value so that we will not lose important information. In order to avoid distorted value due to skewed data, I do not replace it with average (mean), but rather median value.

```
--Calculate median value of beds field

SELECT *, (lowervalue + highervalue)/2 AS Median_beds
FROM
(
    SELECT MAX(beds)*1.0 AS lowervalue
    FROM (
        SELECT TOP 50 PERCENT *
        FROM Student_FS2020.[UM-AD\ptd9pk].SacRealEstate
        WHERE beds <> 'F'
        ORDER BY beds
    ) AS lowerhalf
) AS p1
,
(
    SELECT MIN(beds)*1.0 AS highervalue
    FROM (
        SELECT TOP 50 PERCENT *
        FROM Student_FS2020.[UM-AD\ptd9pk].SacRealEstate
        WHERE beds <> 'F'
        ORDER BY beds DESC
    ) AS upperhalf
) AS p2
```

Results		Messages	
	lowervalue	highervalue	Median_beds
1	3.00	3.00	3.000000

```
--Now replace the "F" value in "beds" field with the median (3)
UPDATE Student_FS2020.[UM-AD\ptd9pk].SacRealEstate
SET beds = 3
WHERE street =
(SELECT street
FROM Student_FS2020.[UM-AD\ptd9pk].SacRealEstate
WHERE beds = 'F')

--Now that all values in "beds" field are integer, let's convert its data type to integer
ALTER TABLE Student_FS2020.[UM-AD\ptd9pk].SacRealEstate
ALTER COLUMN beds INTEGER
```

Appendix 3. Process of investigating and removing duplicate rows

When handling null values in a certain field, a fallacy is simply deleting all rows including these null values based on the assumption that the null values will occur at same rows for the rest of the table. However, that is not necessarily the case for this dataset. To check it, we count all the rows in table, count distinct rows, and count distinct values in street field.

```
--Count all rows in table, distinct rows, and distinct values in street field

SELECT COUNT(street) AS Count_All_Rows, COUNT(DISTINCT street) AS Count_Dist_Street
FROM Student_FS2020.[UM-AD\ptd9pk].SacRealEstate--Count all rows and distinct values in
street column

SELECT COUNT (*) AS Count_Dist_Rows
FROM (
    SELECT DISTINCT *
    FROM Student_FS2020.[UM-AD\ptd9pk].SacRealEstate
) AS dist --Count distinct rows
```

Results		Messages	
	Count_All_Rows	Count_Dist_Street	
1	985	981	

	Count_Dist_Rows	
1	982	

The results show different numbers. There are 985 rows in total, in which there are 982 unique rows (for all fields) and 981 unique values for street field. We can conclude that 3 rows are entirely duplicated for all fields, and 1 row has duplicate value in street field but has different values in other fields. Since the duplicate rows are fewer than duplicate street's values, for conservative approach, we delete the duplicate rows first

```
--Delete duplicate rows

WITH Count_Dup
AS (
    SELECT *, ROW_NUMBER() OVER (PARTITION BY street, city, zip, state, beds, baths,
                                sq_ft, type, sale_date, price, latitude, longitude
                                ORDER BY street, city, zip, state, beds, baths,
                                sq_ft, type, sale_date, price, latitude, longitude)
                                AS Row_Number --Use ROW_NUMBER function to identify the duplicate
rows, which are the rows having row_number > 1
    FROM Student_FS2020.[UM-AD\ptd9pk].SacRealEstate
)
DELETE FROM Count_Dup
WHERE Row_Number > 1
```

Now that we still have one value that is partly duplicated (on street field). Let's look closer into the rows having that duplicate value.

```
--Filter the rows with duplicate values on street field

SELECT p2.*, Count_Dup_street.Row_Number_street
FROM (
    SELECT *, ROW_NUMBER() OVER (PARTITION BY street
                                ORDER BY street, city, zip, state, beds, baths, sq_ft,
                                         type, sale_date, price, latitude, longitude)
        AS Row_Number_street
    FROM Student_FS2020.[UM-AD\ptd9pk].SacRealEstate
) AS Count_Dup_street
INNER JOIN Student_FS2020.[UM-AD\ptd9pk].SacRealEstate AS p2
ON Count_Dup_street.street = p2.street
WHERE Row_Number_street = 2
```

	street	city	zip	state	beds	baths	sq_ft	type	sale_date	price	latitude	longitude	Row_Ni
1	1223 LAMBERTON CIR	SACRAMENTO	95838	CA	3	2	1370	Residential	Mon May 19 00:00:00 EDT 2008	155435	38.646677	-121.437573	2
2	1223 LAMBERTON CIR	SACRAMENTO	95838	CA	3	2	1370	Residential	Mon May 19 00:00:00 EDT 2008	155500	38.646677	-121.437573	2

The only difference is in the prices (155435 and 155500). We may guess that the latter might be rounded from the former. Therefore, for better precision, I decide to keep the former and delete the latter. Since each house address should have unique latitude-longitude pairs, we will identify the row to delete based on this geometry pair

```
--Delete one row with duplicate value on street field

DELETE FROM Student_FS2020.[UM-AD\ptd9pk].SacRealEstate
WHERE price = 155500 AND latitude = 38.646677 AND longitude = -121.437573
```

```
--Now that all values in "street" field are unique, let's assign primary key to it
ALTER TABLE Student_FS2020.[UM-AD\ptd9pk].SacRealEstate
ADD PRIMARY KEY street
```

Appendix 4. A guide on question topic – report section – SQL file matching

Data preparation – analysis part	Report section	SQL/ Tableau file
1A. Average home Price per Zip code 1B. 10 Zip codes with highest ratio of avg. home price to median HH income 1C. The home sales within 20 miles from ABCD Real Estate Office	2.3 Business Analysis	[SQL] data-analysis.sql
2. Visualization on 1A, 1B and 1C	2.3 Business Analysis	[Tableau]
3. Data Model	2.1. Data Preparation • Designing Data Schema	[SQL] design-data-schema.sql
4A. Loading data	2.1. Data Preparation • Loading datasets	
4A. Handling bad data	2.1. Data Preparation Converting data type, handling data errors and creating new calculated columns	[SQL] 1. datatype-baddata.sql 2. primarykey-nullvalues.sql 3. geographydata.sql
4B. Analytics tool selection	2.2. Analytics tools	
4C. Conclusions & recommendations	III. Managerial Implications	

REFERENCES

DB-Engines (2020, May 11), DB-Engines Ranking - Trend of Amazon Aurora vs. Microsoft Azure SQL Database vs. Microsoft SQL Server Popularity. Retrieved from <https://db-engines.com/en/system/Amazon+Aurora%3BMicrosoft+Azure+SQL+Database%3BMicrosoft+SQL+Server>

DB-Engines (2020, May 11), Method of calculating the scores of the DB-Engines Ranking. Retrieved from https://db-engines.com/en/ranking_definition

Longtermrends (2020, May 11), Shiller Case Homes Price Index / US Median Annual Income. Retrieved from <https://www.longtermrends.net/home-price-median-annual-income-ratio/>

Michigan Population Studies Center (2020, May 12), Zip Code Characteristics: Mean and Median Household Income. Retrieved from <https://www.psc.isr.umich.edu/dis/census/Features/tract2zip/>

Redfin (2020, May 11), Sacramento Housing Market. Retrieved from <https://www.redfin.com/city/16409/CA/Sacramento/housing-market>

United States Census Bureau (2020, May 11), "FIND GEOGRAPHIES USING..." OPTION. Retrieved from <https://geocoding.geo.census.gov/geocoder/geographies/address?street=1315+10th+St.&city=Sacramento&state=CA&zip=95814&benchmark=4&vintage=4>