Project BUDT703_0506_09

Housing Reviews in College Park
READ ME DOCUMENT

“Renter’s Plan” Team

Team Members:

Nguyet Chu, Taniya Singh, Saswati Mohanty, Yifei Li and Guanhao Sheng

Our company **Renters Plan** provides Business Analytics (BA) Consulting to the **Southern Management Corporation (SMC)** who manages the Graduate Gardens and Graduate Hills apartments in College Park.

Southern Management Company(SMC) targets students for off-campus housing and is reputed for its well-established and affordable student housing options.

- 30pts: A documentation on data sources, references, how to test your project with screenshots in Project_050#_##_Readme.docx

A. DATA SOURCE

A1. Data source:

1. https://www.apartments.com/college-park-md/
2. https://www.apartmentratings.com/md/college-park/
3. Google reviews

A2. Data set overview

1. Apartment: List of 10 popular apartments around College Park with their average rent, bed count, bathroom count, tennant count, deposit, etc.
2. Location: Apartment addresses and distances from different landmarks.
3. Review: We have studied 189 Customer reviews and added them in a Review table.
4. Data source: The reviews are extracted from three different data sources.
5. Provide: The ratings are translated into services provided accounting to whether the customer got these services by apartments.
6. Owner: The owner's name, type and phone numbers are noted for apartments.
7. Amenities: All the apartment's amenities are listed.

B. REFERENCES

Information was collected from the three data sources and compiled into an Excel spreadsheet. The spreadsheet has also been uploaded to Canvas.

C. PHYSICAL DATABASE DESIGN

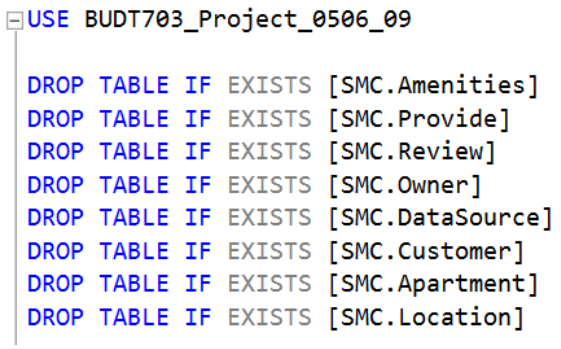
Implement the dataset in the Microsoft SQL Server Management Studio by executing the following steps.

C1. Step 1:

- Create Tables with attributes of respective data-types.
  Example: lctStreet VARCHAR(50)

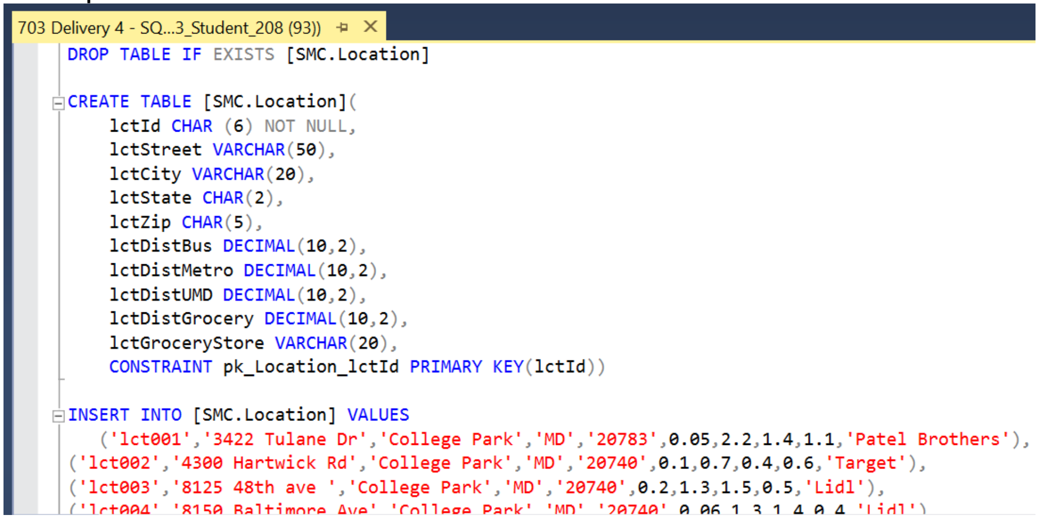
- Assign each table a primary key and make it NOT NULL
  Example: lctId CHAR (6) NOT NULL

- Access the project database using USE command and then dropping any table IF EXISTS to facilitate seamlessly running for multiple times.



```
USE BUDT703_Project_0506_09

DROP TABLE IF EXISTS [SMC.Amenities]
DROP TABLE IF EXISTS [SMC.Provide]
DROP TABLE IF EXISTS [SMC.Review]
DROP TABLE IF EXISTS [SMC.Owner]
DROP TABLE IF EXISTS [SMC.DataSource]
DROP TABLE IF EXISTS [SMC.Customer]
DROP TABLE IF EXISTS [SMC.Apartment]
DROP TABLE IF EXISTS [SMC.Location]
```

A set of 6 tables were created:
Example: The LOCATION table with values inserted.



```
DROP TABLE IF EXISTS [SMC.Location]

CREATE TABLE [SMC.Location](
    lctId CHAR (6) NOT NULL,
    lctStreet VARCHAR(50),
    lctCity VARCHAR(20),
    lctState CHAR(2),
    lctZip CHAR(5),
    lctDistBus DECIMAL(10,2),
    lctDistMetro DECIMAL(10,2),
    lctDistUMD DECIMAL(10,2),
    lctDistGrocery DECIMAL(10,2),
    lctGroceryStore VARCHAR(20),
    CONSTRAINT pk_Location_lctId PRIMARY KEY(lctId))

INSERT INTO [SMC.Location] VALUES
   ('lct001','3422 Tulane Dr','College Park','MD','20783',0.05,2.2,1.4,1.1,'Patel Brothers'),
('lct002','4300 Hartwick Rd','College Park','MD','20740',0.1,0.7,0.4,0.6,'Target'),
('lct003','8125 48th ave ','College Park','MD','20740',0.2,1.3,1.5,0.5,'Lidl'),
('lct004','8150 Baltimore Ave','College Park','MD','20740',0.06,1.3,1.4,0.4,'Lidl')
```

C2. Step 2:

According to any relations in the relation schema, ADD CONSTRAINTS and Reference it to introduce relevant foreign keys. Also add the referential integrity business rules.

Example: CONSTRAINT fk_Apartment_lctId FOREIGN KEY (lctId) REFERENCES [SMC.Location] (lctId)
ON DELETE CASCADE ON UPDATE CASCADE)

C3. Step 3:

- We have formatted the Excel data sheets and translated into INSERT INTO() VALUES() statement to fill in our tables respectively.
- Insert values into the tables using the function INSERT INTO() VALUES().

Example: INSERT INTO [SMC.Location] VALUES
('lct001','3422 Tulane Dr','College Park','MD','20783',0.05,2.2,1.4,1.1,'Patel Brothers')

This step completes our dataset replication from the excel file into the SQL Server Management Studio and creates 6 tables filled with values as below.

1. The Apartment table
2. The Location table
3. The Review table
4. The Data source table
5. The Provide table
6. The Owner table
7. The Amenities table

D. BUSINESS TRANSACTIONS AND QUERIES

D1. Queries in the SQL Server Management Studio

Based on our **mission statement,** we query the dataset in the SQL Server Management Studio to get insights on the data. This is done to fulfil our outcome of advising the Southern Management company to:

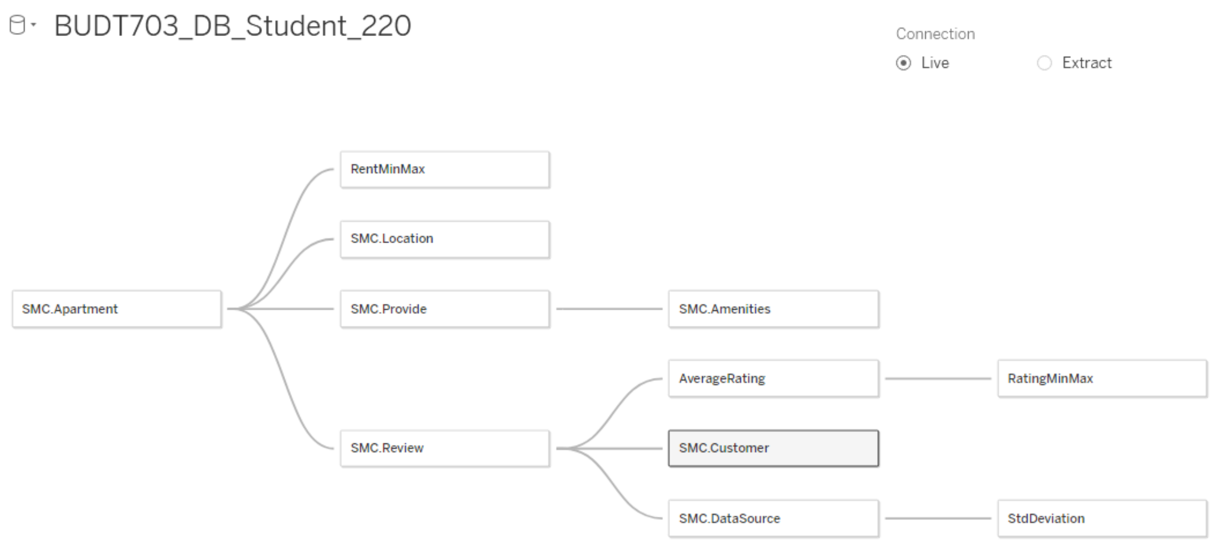
1. Know the average Rent min and Rent max of apartments around College Park to know which apartments stand out as affordable housing.
2. Identify apartments with an average distance of no more than 1 mile to different landmarks in College Park and a minimum average rating of 3.
3. Find the highest-rated and the lowest-rated apartments in College Park that have a rent of less than $1000, have gym centers, and are no more than 1 mile from Lidl..
4. Identify the data source that performs the best for data collection based on the closest information to the average reviews for each apartment.

The above objectives are translated into Queries in the SQL Server Management Studio commands using a wide variety of functions like CREATE VIEW, COUNT( ), AVG( ), JOIN( ), etc.

D2. Visualization in Tableau

The SQL results are visualized using Tableau with a series of charts and graphs for a better understanding of the outcomes in the SQL Server Management Studio

The data sources are connected in Tableau as follows.



BUDT703_DB_Student_220

Connection

● Live ○ Extract

D3. Business transactions

D3.1. Business Transaction 1:

What is the difference in the minimum and maximum rents of each apartment to the average minimum and maximum of all the listed apartments with the number of amenities they offer and their average review?

SQL command

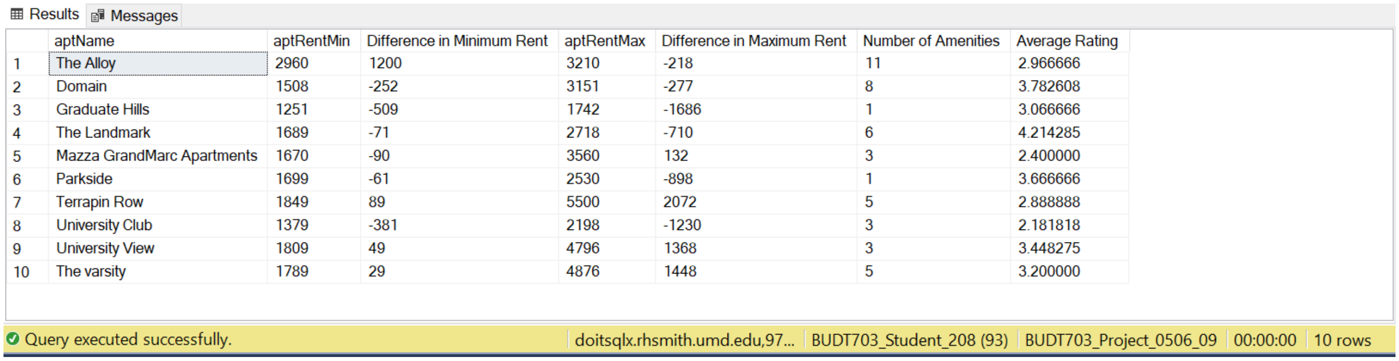
```
CREATE VIEW [RentMinMax] AS
        SELECT AVG(a.aptRentMin) AS 'AvgRentMin', AVG(a.aptRentMax) AS
'AvgRentMax'
        FROM [SMC.Apartment] a

SELECT a.aptName,
        a.aptRentMin,
        (a.aptRentMin - (
        SELECT b.AvgRentMin
        FROM [RentMinMax] b ) ) AS 'Difference in Minimum Rent',
        a.aptRentMax,
        (a.aptRentMax - (
        SELECT b.AvgRentMax
        FROM [RentMinMax] b ) ) AS 'Difference in Maximum Rent',
        COUNT(DISTINCT m.srvAmenity) AS 'Number of Amenities',
        AVG(r.rating) as 'Average Rating'
FROM [SMC.Apartment] a JOIN [SMC.Amenities] m
        ON a.aptId = m.aptId
        JOIN [SMC.Review] r
        ON a.aptId = r.aptId
```

GROUP BY a.aptId, a.aptName, a.aptRentMin, a.aptRentMax

Method

- CREATE VIEW is used to store the Average Rent Min and Average Rent Max values of all of the apartments.
- The main statement:
    - FROM: JOIN statements are used to link the Apartment, Reviews and Amenities tables using apartment ID.
    - SELECT: Using SELECT statements to select the Apartment name, Apt Rent Min, Difference of Apt Rent Min from the Average Rent Min (calculated), Apt Rent Max, Difference of Apt Rent Max from the Average Rent Max (calculated), display the number of unique service amenities (using COUNT(DISTINCT) and average rating of each apartment using AVG function
    - Results are then grouped by a.aptId, a.aptName, a.aptRentMin, a.aptRentMax

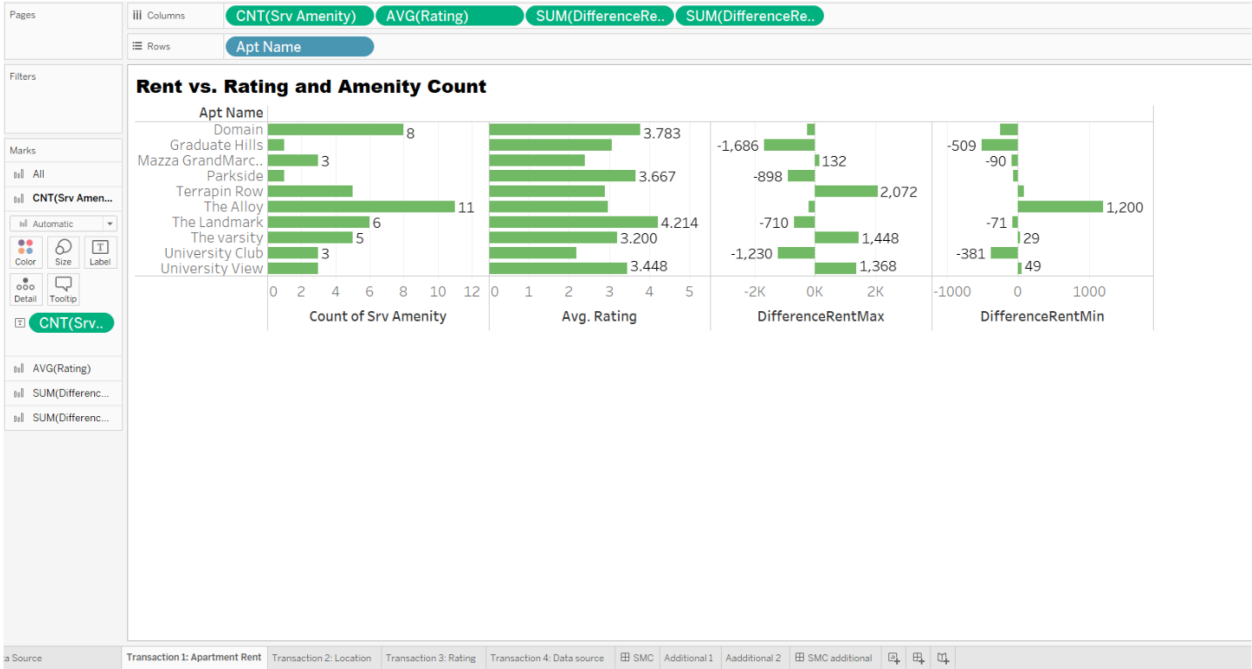
Results



	aptName	aptRentMin	Difference in Minimum Rent	aptRentMax	Difference in Maximum Rent	Number of Amenities	Average Rating
1	The Alloy	2960	1200	3210	-218	11	2.966666
2	Domain	1508	-252	3151	-277	8	3.782608
3	Graduate Hills	1251	-509	1742	-1686	1	3.066666
4	The Landmark	1689	-71	2718	-710	6	4.214285
5	Mazza GrandMarc Apartments	1670	-90	3560	132	3	2.400000
6	Parkside	1699	-61	2530	-898	1	3.666666
7	Terrapin Row	1849	89	5500	2072	5	2.888888
8	University Club	1379	-381	2198	-1230	3	2.181818
9	University View	1809	49	4796	1368	3	3.448275
10	The varsity	1789	29	4876	1448	5	3.200000

Query executed successfully. | doitsqlx.rhsmith.umd.edu,97... | BUDT703_Student_208 (93) | BUDT703_Project_0506_09 | 00:00:00 | 10 rows

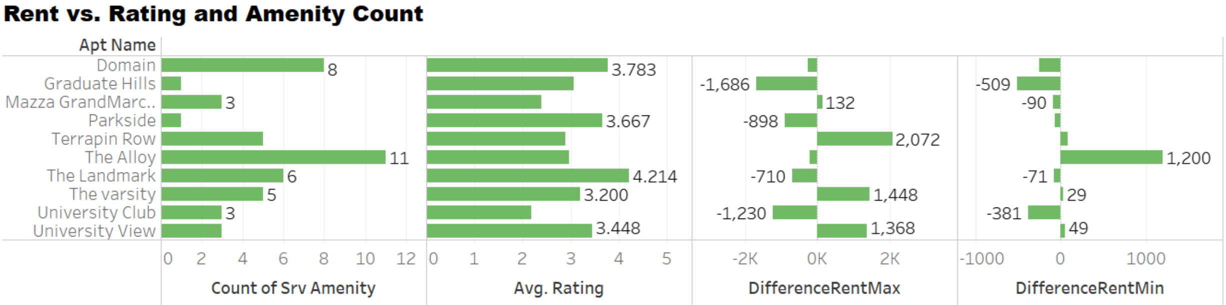
Tableau Visualization:



Columns: CNT(Srv Amenity) AVG(Rating) SUM(DifferenceRe.. SUM(DifferenceRe..

Rows: Apt Name

Tableau Result



D3.2. Business Transaction 2:

Which apartment has an average distance of no more than 1 mile to different landmarks in College Park and has a minimum average rating of 3? List the average distance and the average rating of each apartment.

SQL command

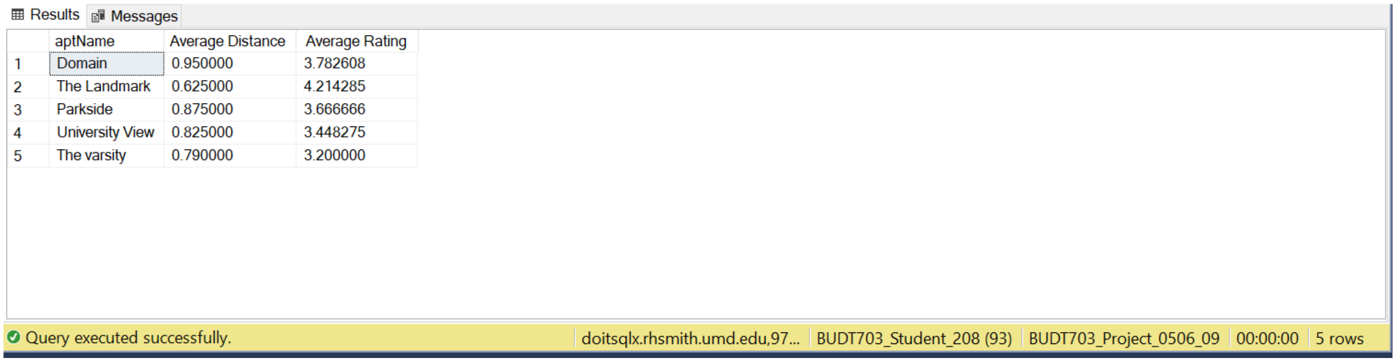
```
SELECT a.aptName, c1.avgrd AS 'Average Distance',c2.avgrr AS' Average Rating'
FROM [SMC.Apartment] a,
        (SELECT l.lctId,(l.lctDistBus+l.lctDistGrocery+l.lctDistMetro+l.lctDistUMD)/4 AS
avgrd
        FROM [SMC.Location] l
        WHERE (l.lctDistBus+l.lctDistGrocery+l.lctDistMetro+l.lctDistUMD)/4 <= 1) c1,
        (SELECT r.aptId, AVG(r.rating) as avgrr
        FROM [SMC.Review] r
```

```
          GROUP BY r.aptId
          HAVING AVG(r.rating)>=3) c2
WHERE a.lctId = c1.lctId
          AND a.aptId = c2.aptId
```

Method

- FROM:
    - Subquery 1 is used to create a list of apartments that have an average distance of no more than 1 mile to different landmarks in College Park and display the average distance of each apartment
    - Subquery 2 is used to create a list of apartments that has an average rating of at least 3 and display the average rating of each apartment
    - The results of subquery 1 and subquery 2 are joined with the Apartment table using the location ID and apartment ID in the WHERE statement
- SELECT: SELECT is used to select apartment name, average distance and average rating from the table

Results



	aptName	Average Distance	Average Rating
1	Domain	0.950000	3.782608
2	The Landmark	0.625000	4.214285
3	Parkside	0.875000	3.666666
4	University View	0.825000	3.448275
5	The varsity	0.790000	3.200000

```
Query executed successfully.    doitsqlx.rhsmith.umd.edu,97...    BUDT703_Student_208 (93)    BUDT703_Project_0506_09    00:00:00    5 rows
```

Tableau Visualization

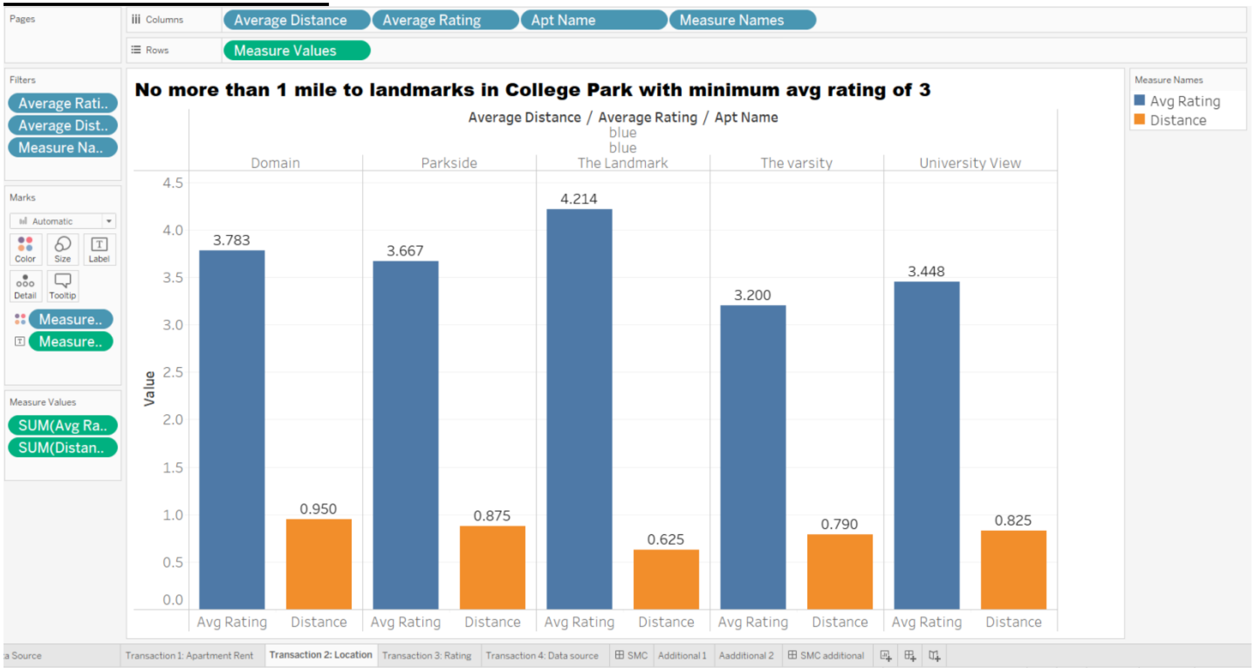
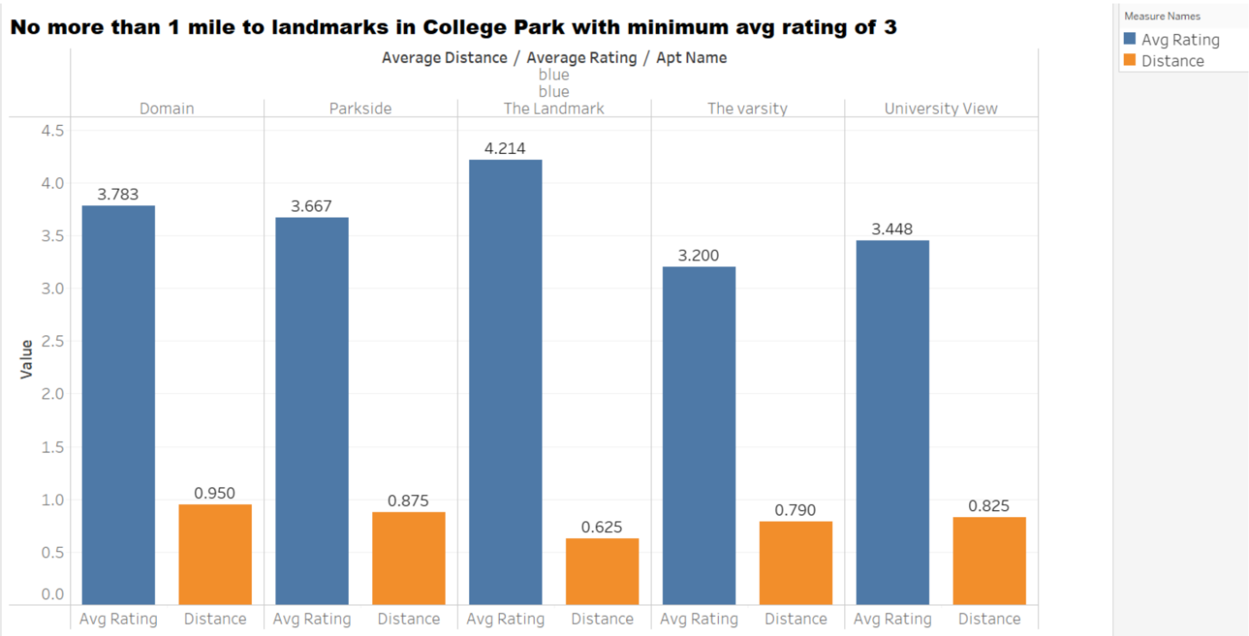


Tableau Result



D3.3. Business Transaction 3

Which is the highest-rated and lowest-rated apartment in College Park that has a rent of less than $1000 and has a gym center?

SQL command

```
CREATE VIEW [RatingMinMax] AS
        SELECT a.aptName, AVG(r.rating) AS 'AvgRating'
        FROM [SMC.Apartment] a JOIN [SMC.Review] r
                ON a.aptId = r.aptId
        GROUP BY a.aptId, a.aptName, a.aptRentMax, a.aptTennantMax,
a.aptRentMin, a.aptTennantMin, a.lctId
        HAVING (a.aptRentMax/a.aptTennantMax)<1000 OR
(a.aptRentMin/a.aptTennantMin)<1000
                AND a.aptId IN (
                        SELECT m.aptId
                        FROM [SMC.Amenities] m
                        WHERE m.srvAmenity='Gym')

SELECT m.aptName, m.[AvgRating]
FROM [RatingMinMax] m
WHERE m.[AvgRating] = (
        SELECT MAX(m.[AvgRating])
        FROM [RatingMinMax] m )
        OR m.[AvgRating] = (
        SELECT MIN(m.[AvgRating])
        FROM [RatingMinMax] m )
```

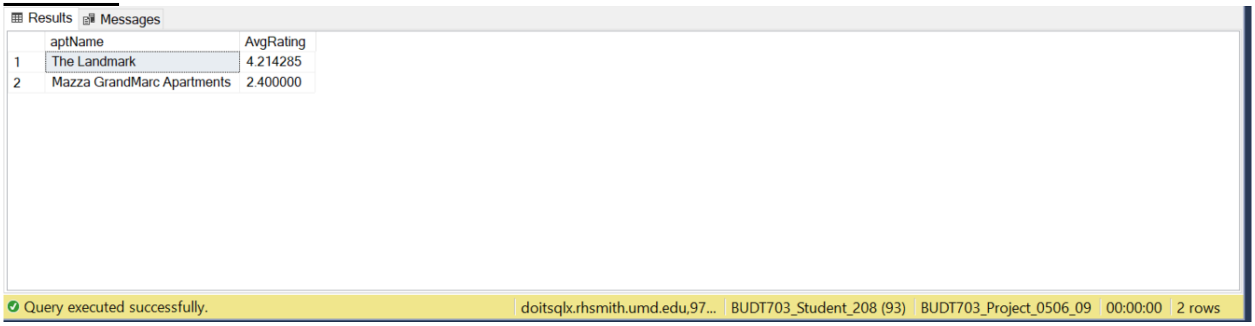
```
GROUP BY m.aptName, m.[AvgRating]
ORDER BY m.[AvgRating] DESC
```

Method

- CREATE VIEW is used to create a virtual view named “RatingMinMax” that stores the list of apartments that have a rent of less than $1000 and have a gym center, and show the calculated average ratings of those apartments.
  - Firstly, the Apartment and Review tables are joined using the apartment ID and the rows are grouped by apartment ID, apartment Name, maximum rent, maximum number of tenants, maximum rent, maximum number of tenants and location ID.
  - Secondly, HAVING is used to define the two criteria
    a. Criteria 1 - Rent per person is less than $1000
       We need to find the rent per person by dividing the maximum rent by the maximum number of tenants and dividing the minimum rent by the minimum number of tenants. We set the condition that either of the results is smaller than $1000.
    b. Criteria 2 - Apartment that has a gym center
       We use a subquery to look for apartments that have their apartment IDs in the Amenities tables and has the “Gym” values
  - Lastly, SELECT is used to create a list of apartments from the table that satisfy the two criteria and show the average rating of each apartment using the AVG function.

- The main statement:
  - WHERE: MIN and MAX are used to find the rows in the “RatingMinMax” view that have the highest and the lowest ratings.
  - SELECT is used to select the apartments from the “RatingMinMax” view that have highest and lowest rating values and show the ratings of those apartments. The apartments are arranged in the descending order of average rating.

Results



	aptName	AvgRating
1	The Landmark	4.214285
2	Mazza GrandMarc Apartments	2.400000

Query executed successfully. doitsqlx.rhsmith.umd.edu,97... BUDT703_Student_208 (93) BUDT703_Project_0506_09 00:00:00 2 rows

Tableau Visualization

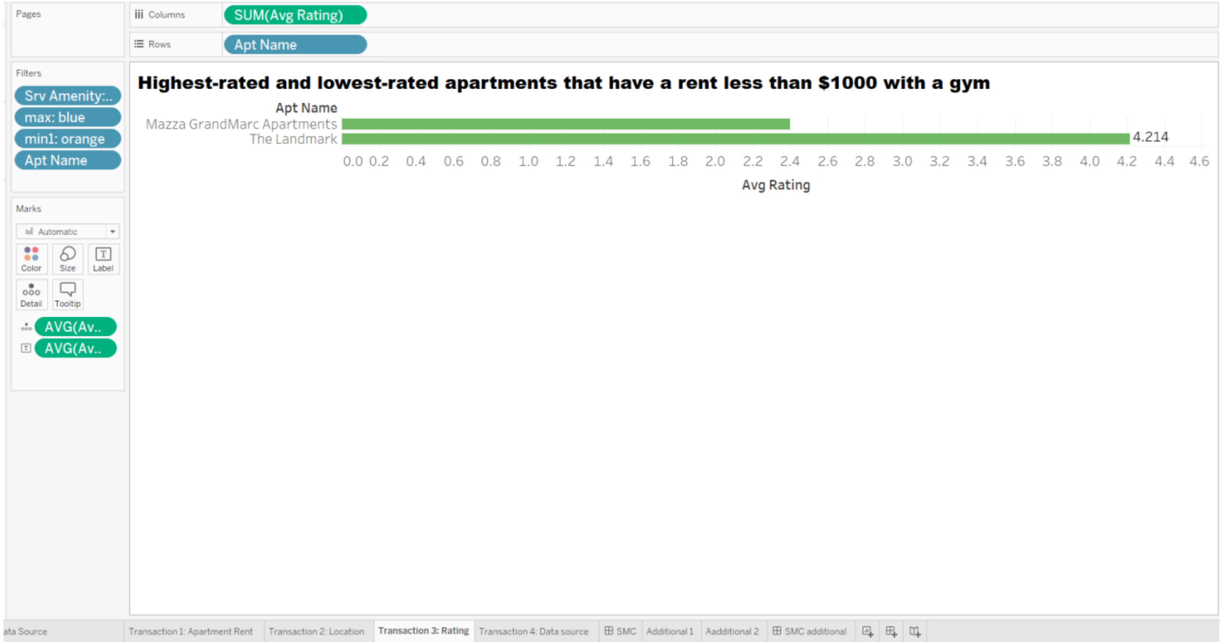
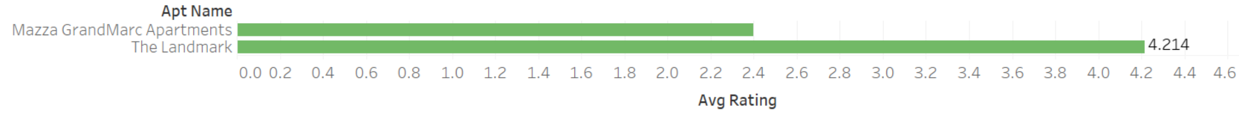


__Tableau Result__

Highest-rated and lowest-rated apartments that have a rent less than $1000 with a gym



D3.4. Business Transaction 4

Which data source performs the best for data collection based on the closest information to the average reviews for each apartment?

__SQL command__

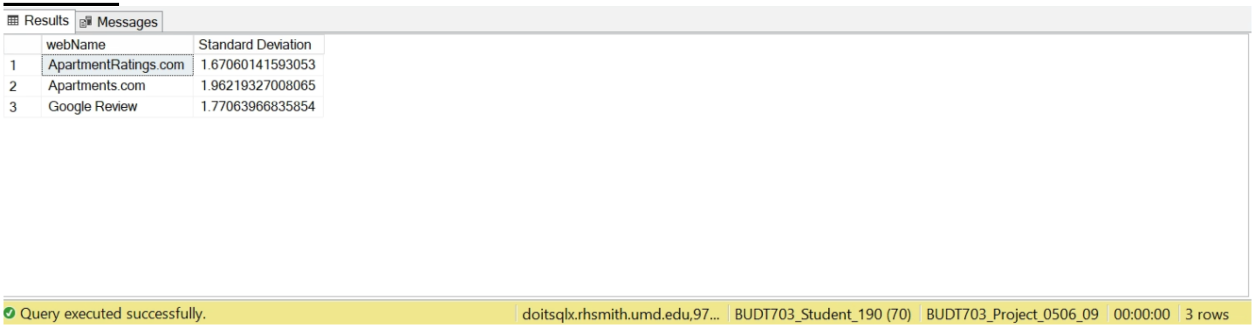
```
SELECT m.webName, STDEV(m.[Rating]) as 'Standard Deviation'
FROM (
        SELECT s.webName, a.aptName,  r.rating
        FROM [SMC.Review] r JOIN [SMC.DataSource] s
                ON r.webId = s.webId
                JOIN  [SMC.Apartment] a
                ON r.aptId = a.aptId ) m
GROUP BY m.webName
```

__Method__

- FROM: Subquery is to create a table which shows the full list of web names, the apartments on each website and the rating of each apartment
  This table is created by joining the Review and DataSource tables using web ID, then joining the apartment table using apartment ID. Then the SELECT statement is used to select the web name, apartment name and rating from this table to create a new table named “m”.
- GROUP BY is used to group the rows in the “m” table by the webname.

- SELECT: From the ‘m” table, SELECT statement is used to select the web name and show the calculated standard deviation of its rating values (using STDEV).

Results



	webName	Standard Deviation
1	ApartmentRatings.com	1.67060141593053
2	Apartments.com	1.96219327008065
3	Google Review	1.77063966835854

Query executed successfully. | doitsqlx.rhsmith.umd.edu,97... | BUDT703_Student_190 (70) | BUDT703_Project_0506_09 | 00:00:00 | 3 rows

Tableau Visualization

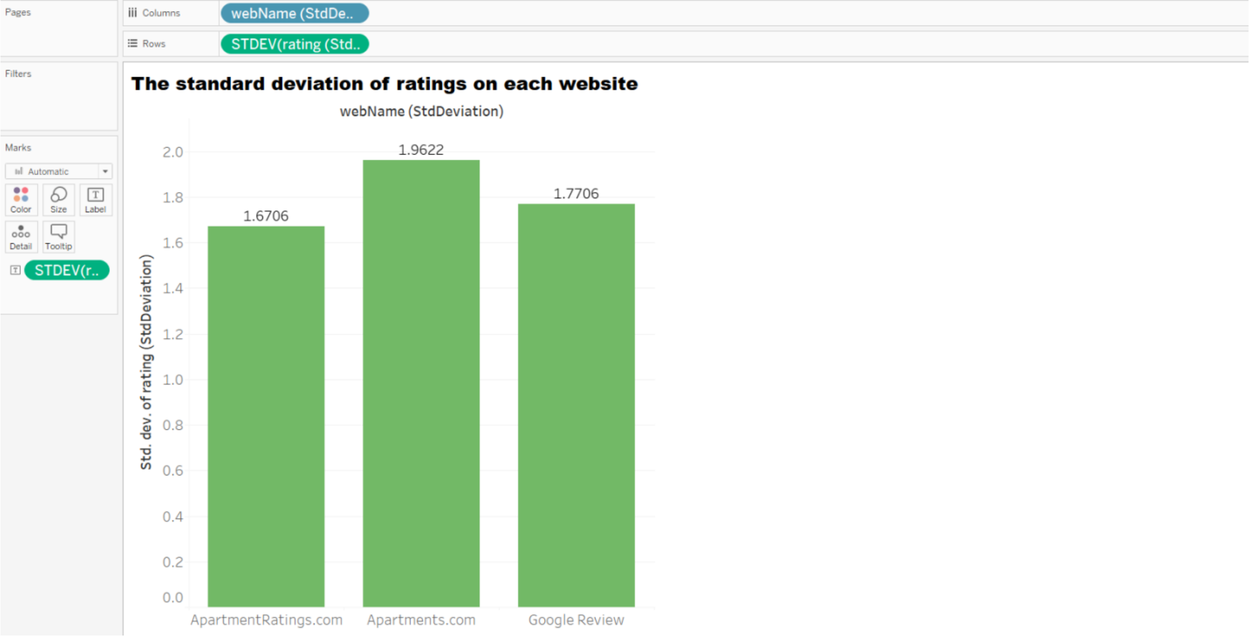
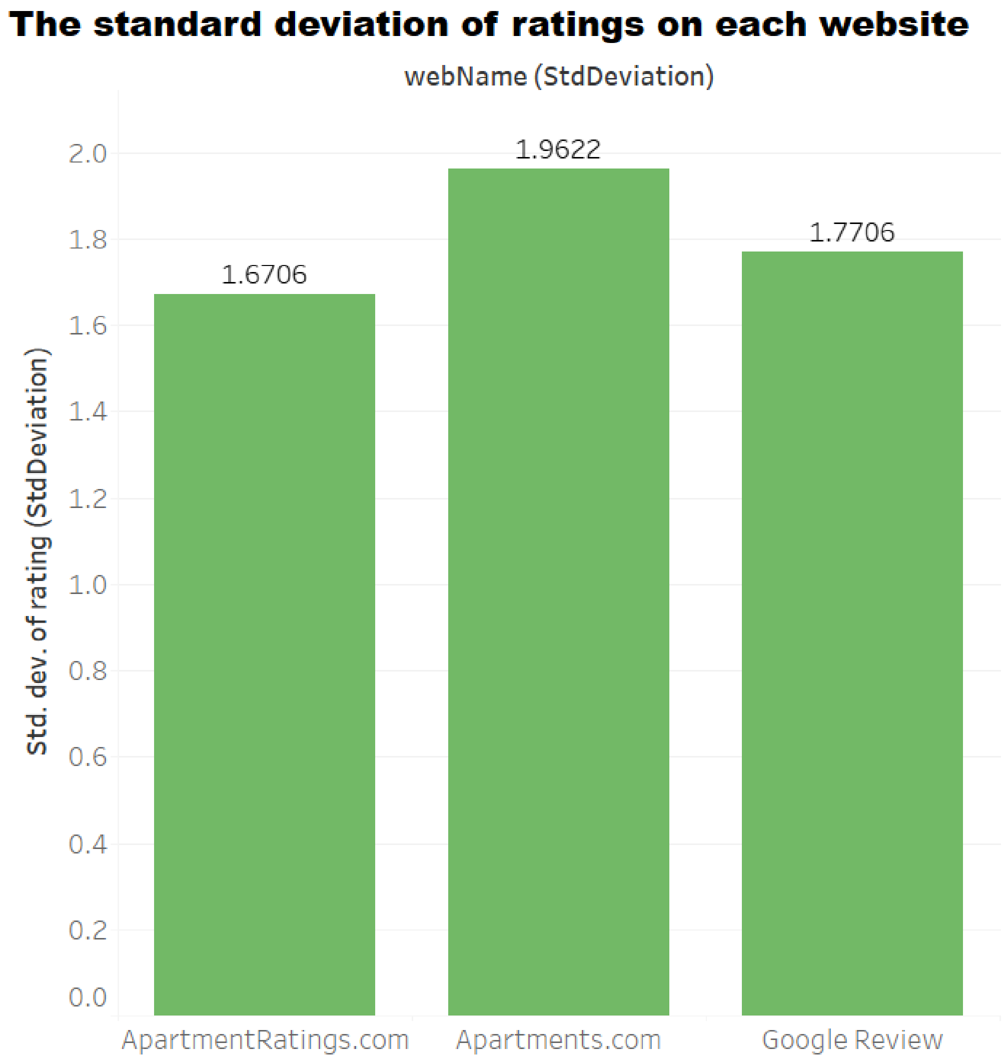
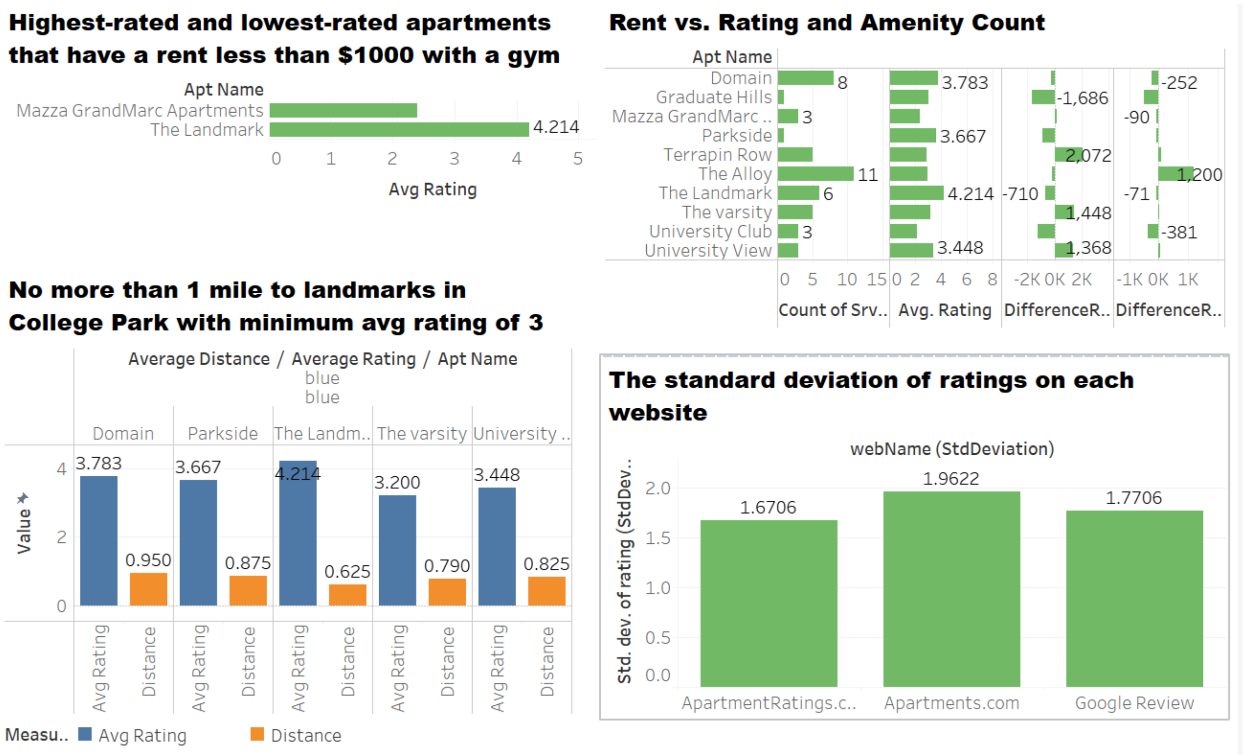


Tableau Results



Visualization Dashboard

We have created a Tableau dashboard to explain our final findings with the SMC group for a better clarity of results.



Highest-rated and lowest-rated apartments that have a rent less than $1000 with a gym

Rent vs. Rating and Amenity Count

No more than 1 mile to landmarks in College Park with minimum avg rating of 3

The standard deviation of ratings on each website

From the dashboard attached above, we can see the difference of maximum rent fee and minimum rent fee for each apartment with their average ratings, the highest-rated and lowest-rated apartment which is less than $1000 in average per month with a gym center, the apartments that are no more than one mile to different landmarks in College Park with the minimum average rating of 3, and how each data source we used performs.

Recommendations

- SMC should provide more amenities as it has the least amenities as compared to its competitors.
- SMC should improve its overall average rating as five other apartments have higher ratings than Graduate Hills.
- SMC should partner with local transportation services to assist students especially in grocery shopping as it is far from different landmarks in College Park compared to other apartments.
- ApartmentRatings.com is the most accurate source on customer review and satisfaction.

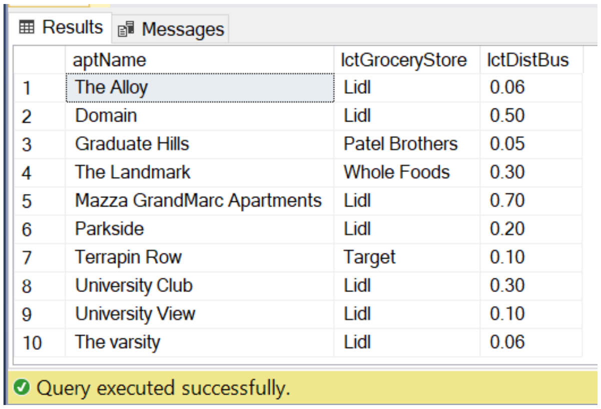
Additional Queries:

1. **What is the distance to the nearest bus-stop to the nearest grocery shop from the apartment for shopping.**

```
SELECT a.aptName,l.lctGroceryStore,l.lctDistBus
```

```
FROM [SMC.Location] l , [SMC.Apartment] a

WHERE a.lctId=l.lctId
```

Results



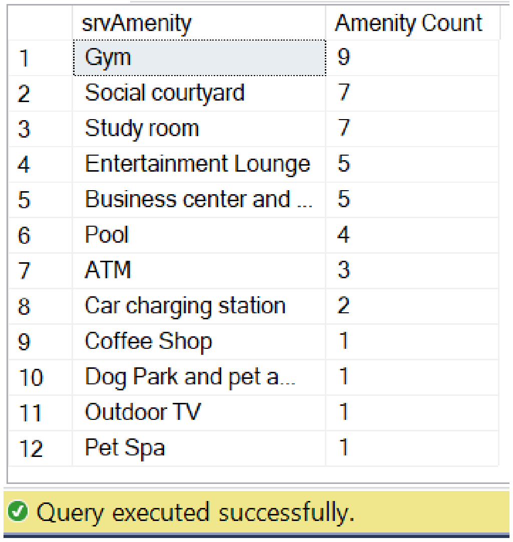
	aptName	lctGroceryStore	lctDistBus
1	The Alloy	Lidl	0.06
2	Domain	Lidl	0.50
3	Graduate Hills	Patel Brothers	0.05
4	The Landmark	Whole Foods	0.30
5	Mazza GrandMarc Apartments	Lidl	0.70
6	Parkside	Lidl	0.20
7	Terrapin Row	Target	0.10
8	University Club	Lidl	0.30
9	University View	Lidl	0.10
10	The varsity	Lidl	0.06

Query executed successfully.

2. List of most popular amenities amongst all apartments

```
SELECT am.srvAmenity,COUNT(am.srvAmenity) as 'Amenity Count'

FROM [SMC.Amenities] am

GROUP BY am.srvAmenity

ORDER BY COUNT(am.srvAmenity) DESC
```

Results



	srvAmenity	Amenity Count
1	Gym	9
2	Social courtyard	7
3	Study room	7
4	Entertainment Lounge	5
5	Business center and ...	5
6	Pool	4
7	ATM	3
8	Car charging station	2
9	Coffee Shop	1
10	Dog Park and pet a...	1
11	Outdoor TV	1
12	Pet Spa	1

Query executed successfully.

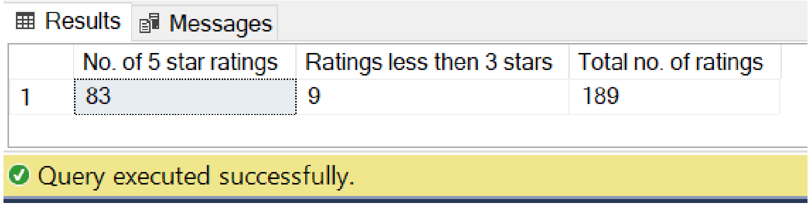
<u>Tableau Visualization</u>

<u>Tableau Results</u>

3. How many 5 star ratings, ratings less than 3 stars compared to total ratings.

```
SELECT COUNT(r.rating) as 'No. of 5 star ratings',
        (SELECT COUNT(r.rating) FROM [SMC.Review] r WHERE r.rating=3) as
        'Ratings less then 3 stars',
        (SELECT COUNT(r.rating) FROM [SMC.Review] r) as 'Total no. of ratings'
FROM [SMC.Review] r
WHERE r.rating=5
```

Results:



	No. of 5 star ratings	Ratings less then 3 stars	Total no. of ratings
1	83	9	189

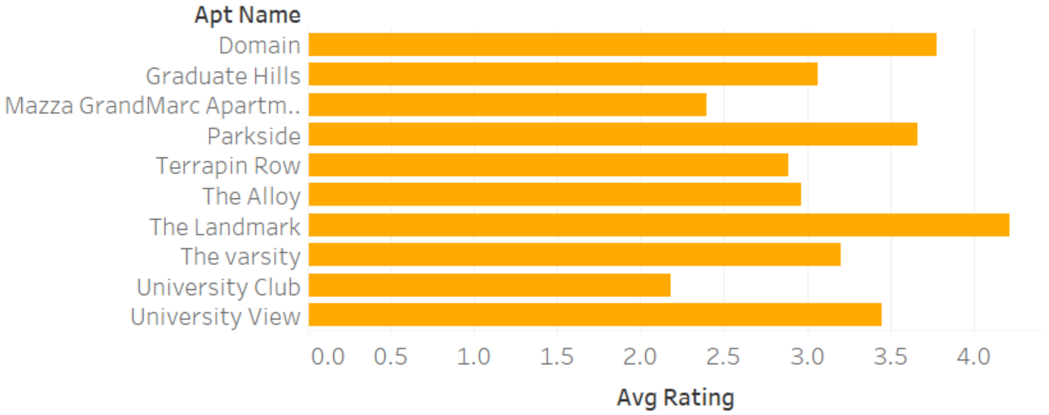
Query executed successfully.

Visualization Dashboard 2

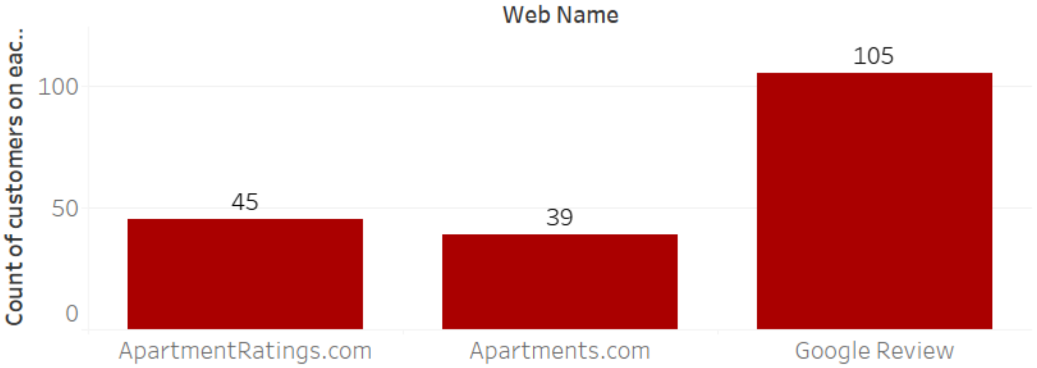
Additionally, we have visualized the average ratings throughout all the apartments to provide a clear picture of ratings. As SMC has to work on increasing it.

Also, to judge our data collection process, the count of the number of customer’s reviews collected for each datasource is displayed. This is done to check which datasource was more accessible and easy to collect data from.

Average ratings of each apartment



This graph shows how many customer’s reviews were recorded from each datasource



We observe that the highest average rating is of the Landmark apartments and the maximum customer reviews are collected from Google Reviews.