ASSIGNMENT PART 2

By

Syed Ali Murtaza

1 INTRODUCTION

In this report I am going to implement the second part of the assignment. In this part we are given with the dataset on blackboard which include CSV file on restaurant data, Ratings on that restaurant, Consumers and the restaurant which offers cuisine. The data set includes 4 tables Restaurants, Ratings, Consumers, and Restaurants cuisine. Now in this part I am going to discuss about the creation of tables and their primary and

foreign keys and implementation of T-SQL queries and stored procedure to accomplish the assigned tasks.

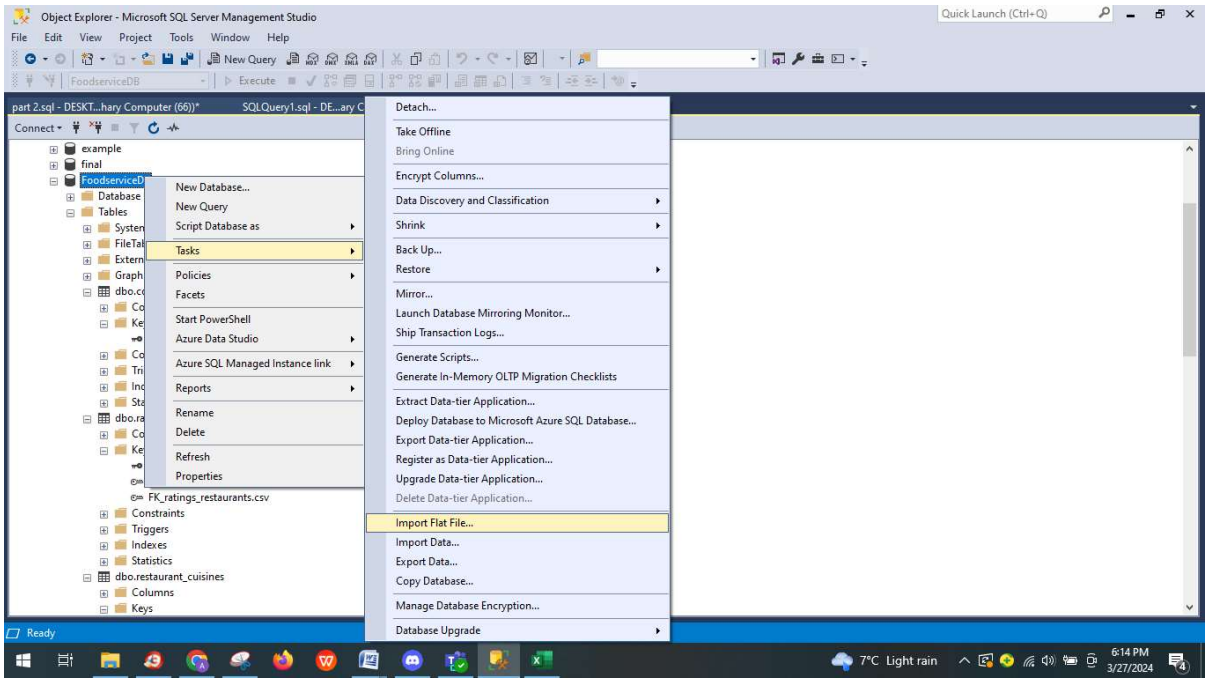
2 Part 1

2.1 DATABASE CREATION AND DATA IMPORT

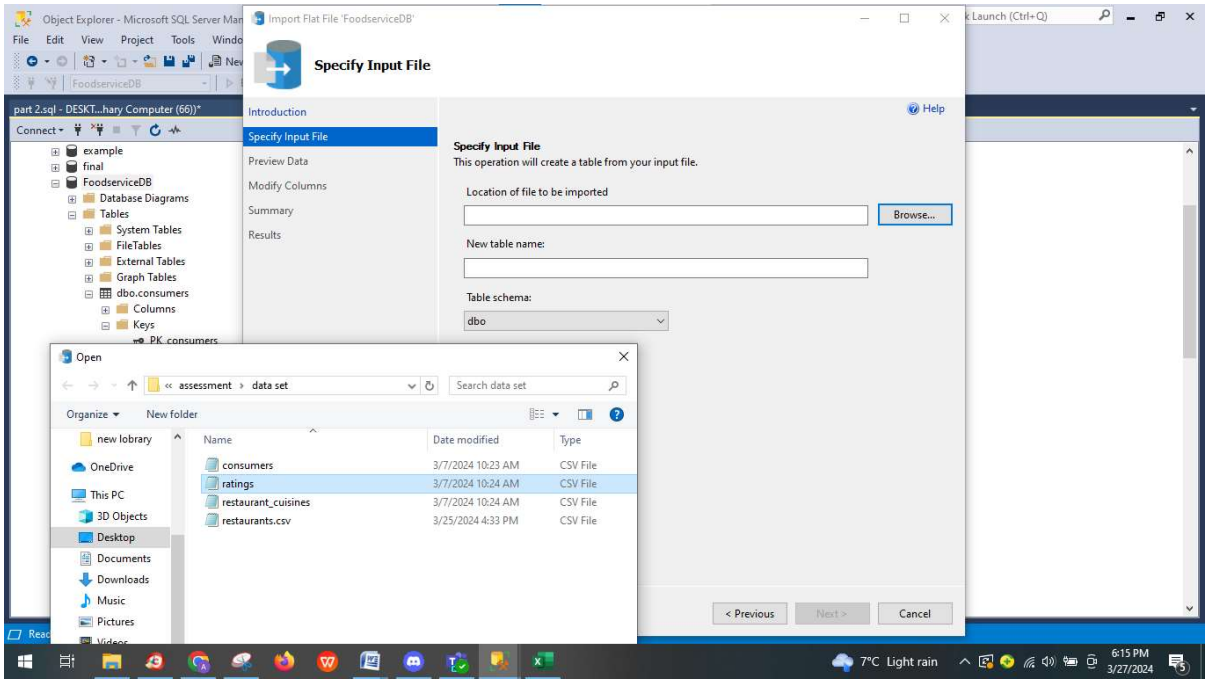
First of all I created a database with the name of FoodserviceDB due to demand of instructor.

```
1  create database FoodserviceDB
2  go
```

And then I download the dataset given in the blackboard into my computer. And Than using this database environment I imported data into data database by right clicking the FoodserviceDB (database name) in the object explorer and than select task and than select Import flat files.

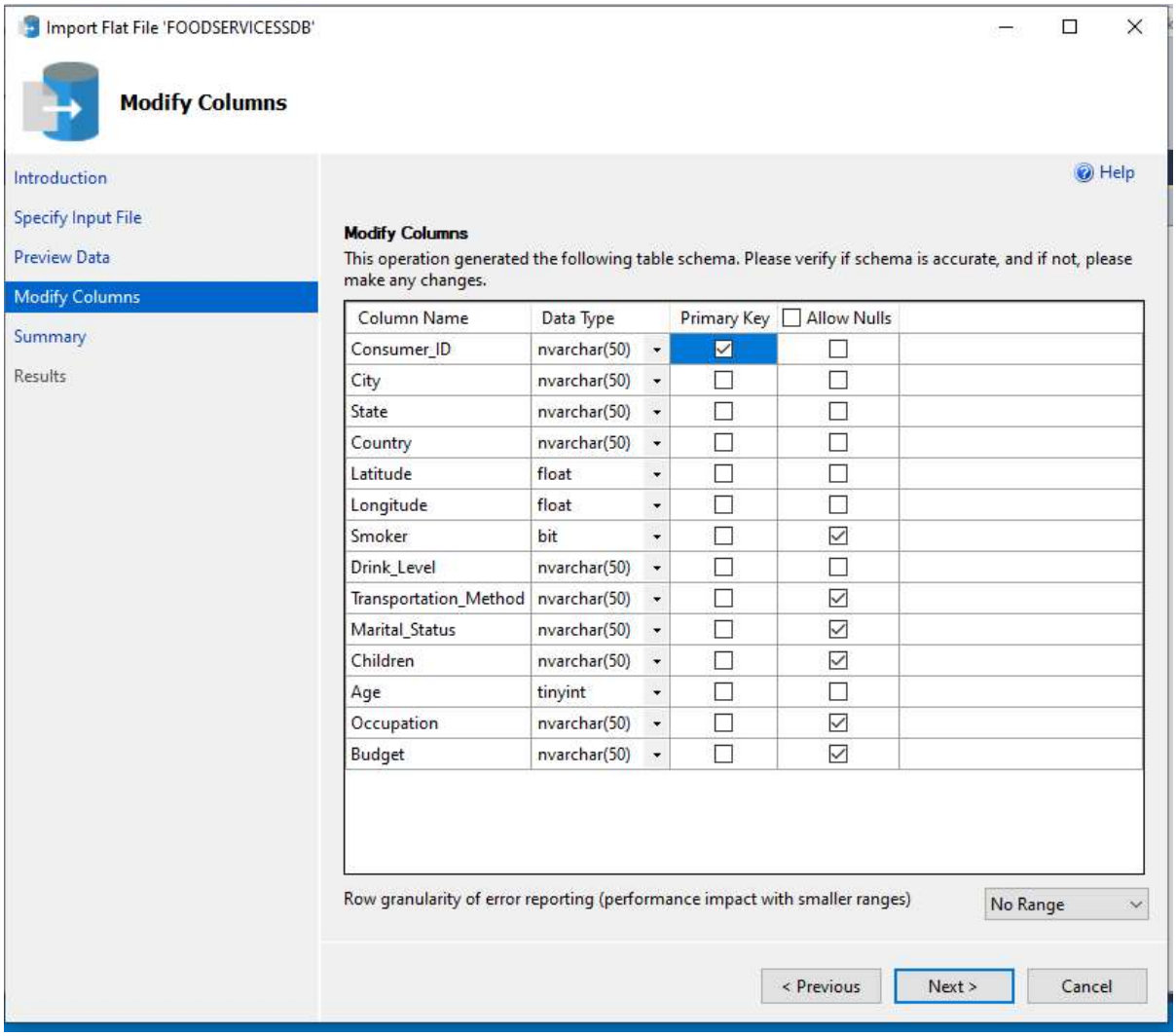


and than I select the given data one by one because it can't load all the files at once. So as you can see from given SS.



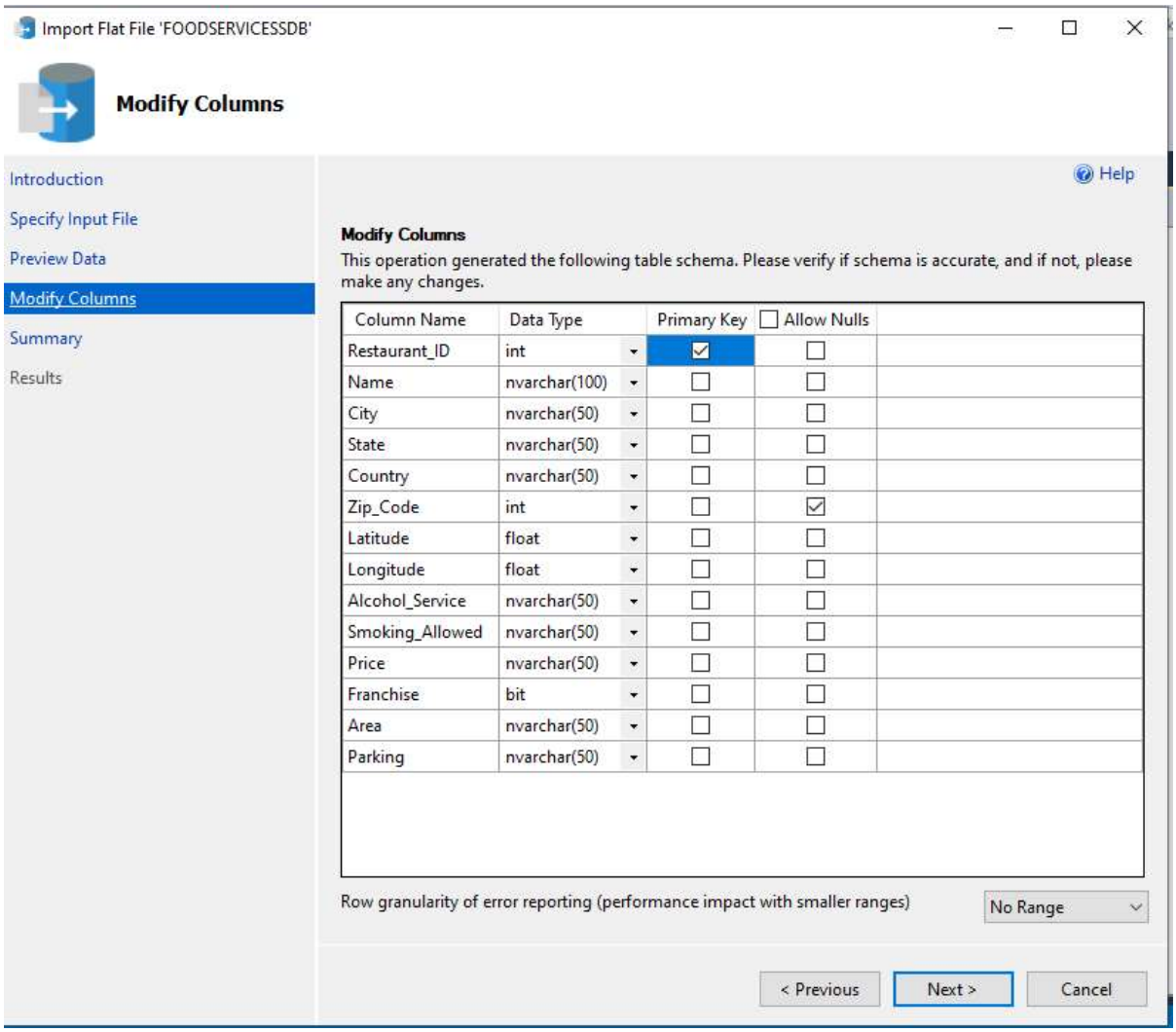
Now the logic behind doing this all is that first I defined primary key for each table and define 2 primary keys(composite key) for ratings table and restaurant cuisine table.

Now in consumer table, I defined consumer_id as a primary key and keep the other columns as default



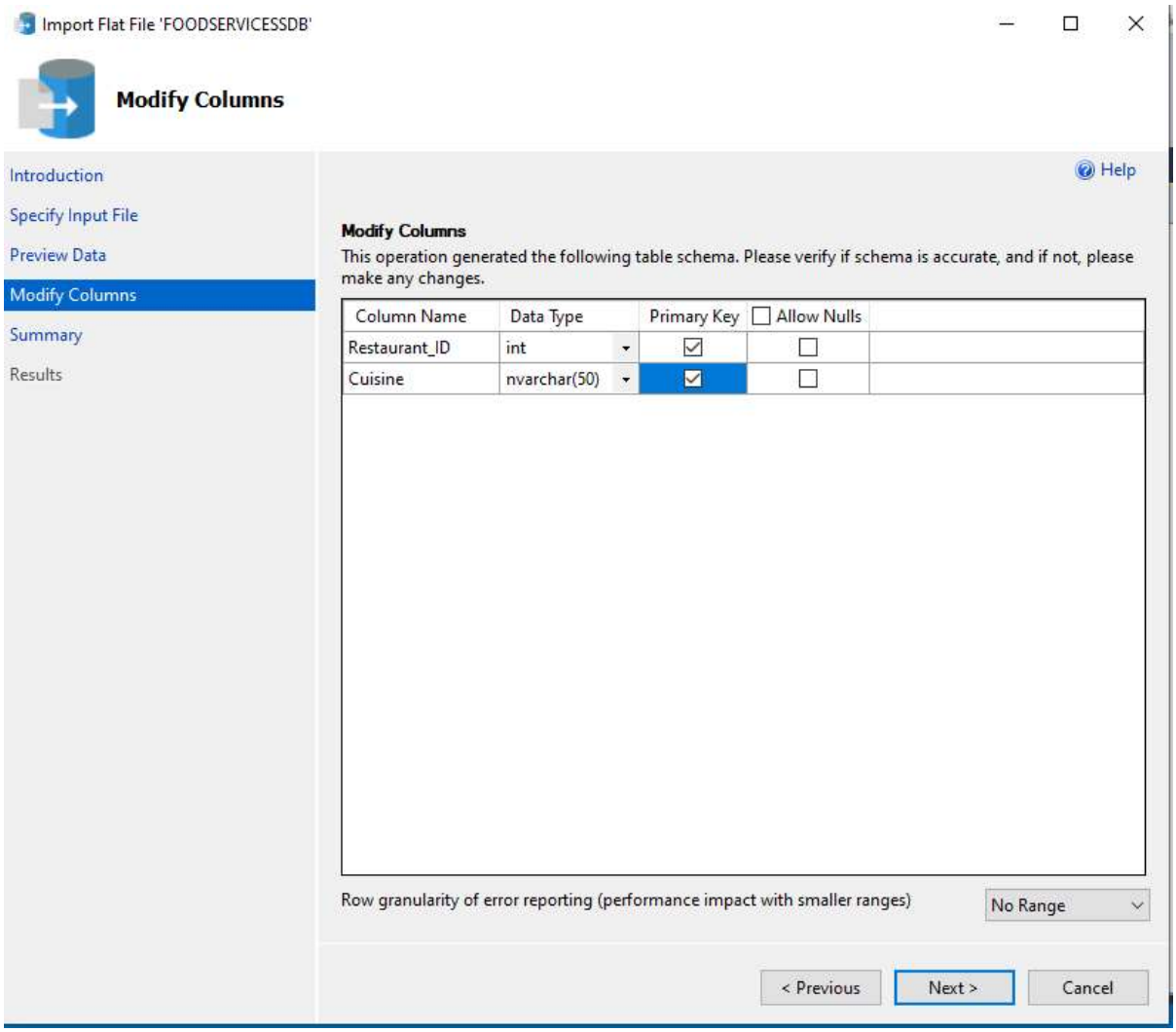
In restaurant table I marked restaurant_id as a primary key and keep other columns as default.

1



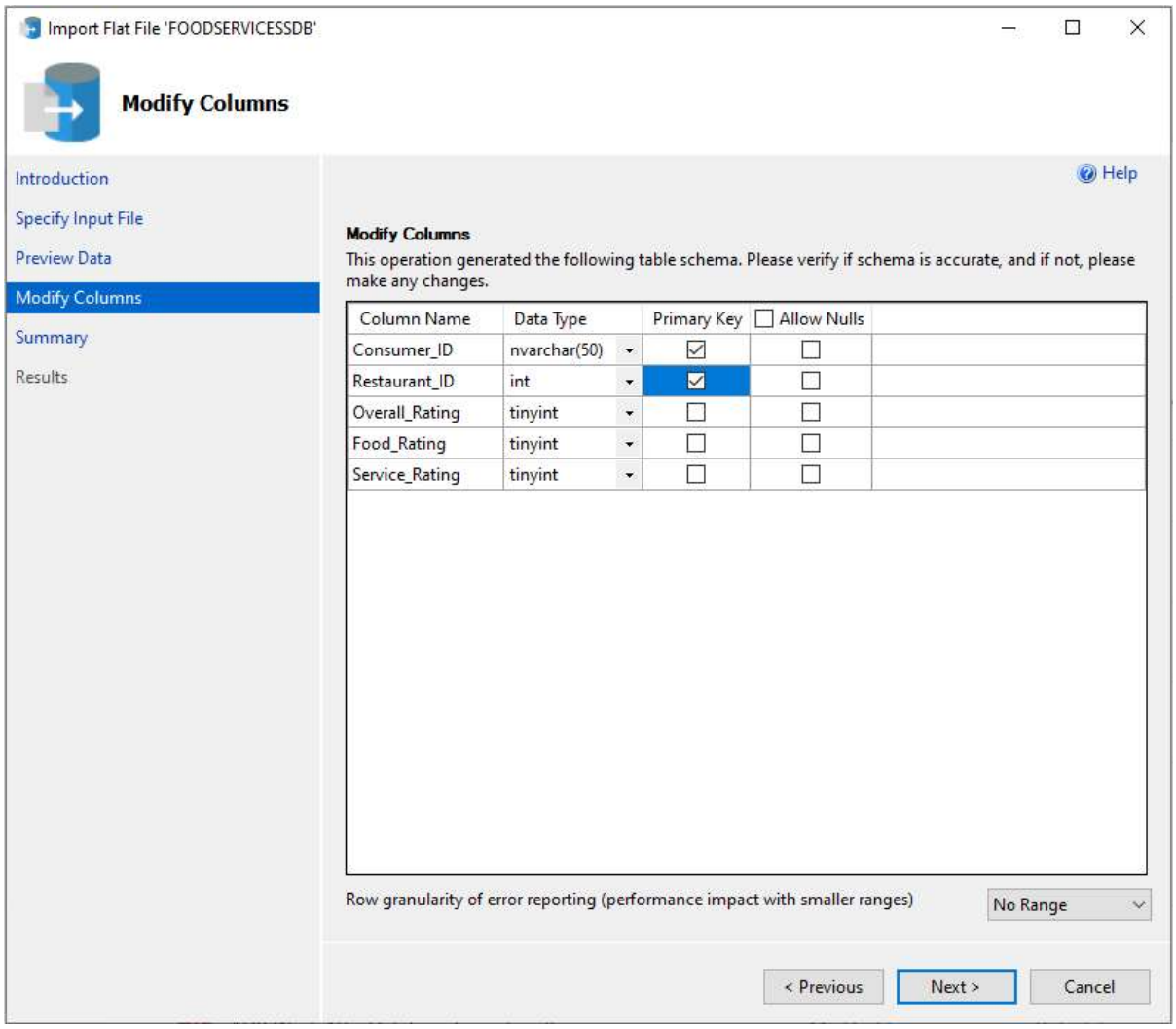
In restaurant cuisine table I marked restaurants_ID and Cuisine both as a primary key

1

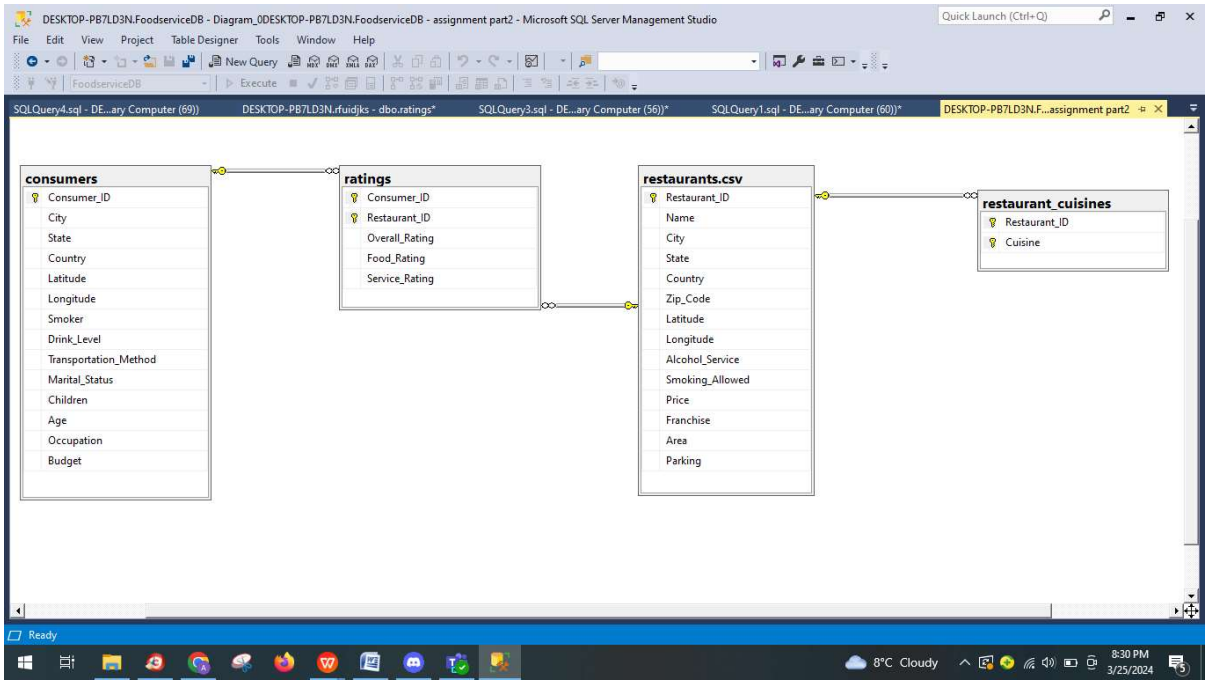


In ratings table I marked Customer_id and restaurant_id both as a primary key to maintain the link between tables that I show you latter.

1



Now after doing all above steps I link the tables manually you can see here



2.2 Database Diagram:-

Now from this database diagram, you can see that I defined the primary keys for each table and then connect the ratings table to consumers table and restaurants table. So in this way consumer_ID and Resturant_id in the ratings table become the foreign key referencing consumerID in the consumer table and restaurant-id in the restaurant table. On the other hand I linked the restaurant cuisine table with the restaurant table so in this way the restaurant id in the restaurant cuisine table becomes common for restaurant_id in the restaurant table and resturant_id in resturant cuisine table.

2.3 Key Relationships:

- **Restaurant to Restaurant_Cuisines:** One-to-many, as a single restaurant can serve multiple cuisines.

- **Restaurant to Ratings:** One-to-many, indicating that a restaurant can have multiple ratings from different consumers.

- **Consumers to Ratings:** One-to-many, showing that a consumer can rate multiple restaurants.

3 Part 2

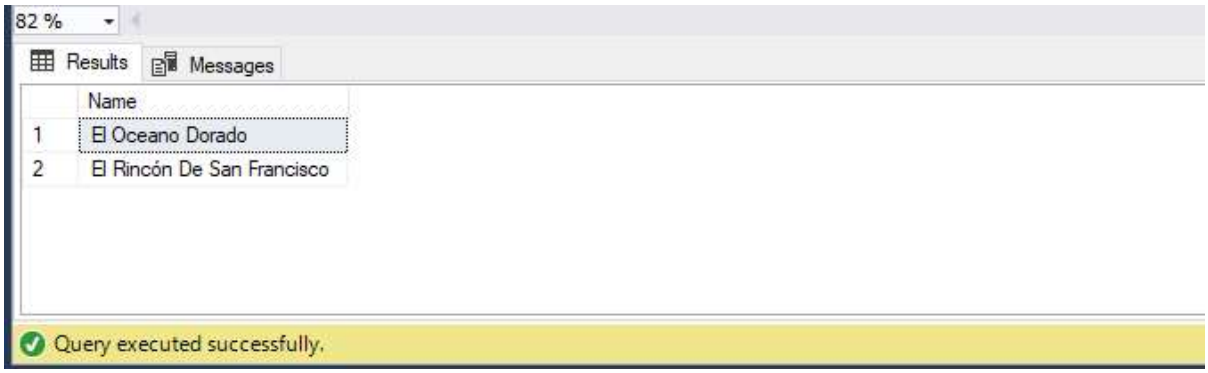
Task Execution Using SQL Queries

3.1 Write a querry that list all the restaurants with a medium range price with open area, serving Mexican food.

For writing above querry as he asked for all the restaurants so I merge the restaurant table and restaurant cuisine table using Join, because type of food is present in restaurant cuisine table and other attributes are present in restaurant table and also instructor asked for all the restaurant and than I use WHERE clause to only select the columns which indicate restaurant serving Maxican food and having open area with a medium range price.

```
SELECT r.Name
FROM [restaurants.csv] r
JOIN Restaurant_Cuisines rc ON r.Restaurant_id = rc.Restaurant_id
WHERE r.Price = 'Medium'
AND r.Area = 'open'
AND rc.Cuisine = 'Mexican'
```

Result:-



From the result we can see that there are only 2 resturants with given attributes.

3.2 Write a query that returns the total number of restaurants who have the overall rating as 1 and are serving Mexican food.

Compare the results with the total number of restaurants who have the overall rating as 1 serving Italian food (please give explanations on their comparison)

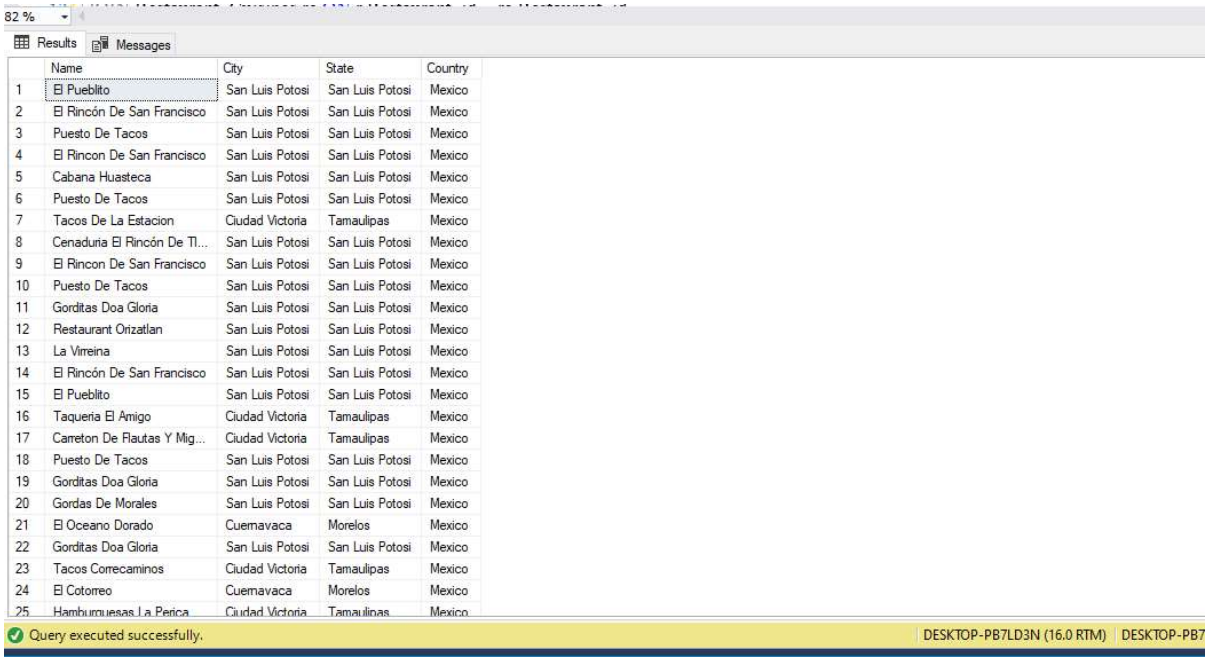
For this task first of all I list all the restaurant with the rating as 1 and serving Mexican food.

For this purpose I make a join on 3 tables i.e restaurants, ratings and restaurant cuisine.Using WHERE command I checked out of our desire attributes i.e rating=1 and cuisine=Maxican.

```
20  SELECT r.Name, r.City, r.State, r.Country
21  FROM [restaurants.csv] r
22  JOIN Ratings ra ON r.Restaurant_id = ra.Restaurant_id
23  JOIN Restaurant_Cuisines rc ON r.Restaurant_id = rc.Restaurant_id
24  WHERE ra.Overall_Rating = 1
25  AND rc.Cuisine = 'Mexican';
26
```

Result:-

There are 87 restaurants as follow



	Name	City	State	Country
1	El Pueblito	San Luis Potosi	San Luis Potosi	Mexico
2	El Rincón De San Francisco	San Luis Potosi	San Luis Potosi	Mexico
3	Puesto De Tacos	San Luis Potosi	San Luis Potosi	Mexico
4	El Rincon De San Francisco	San Luis Potosi	San Luis Potosi	Mexico
5	Cabana Huasteca	San Luis Potosi	San Luis Potosi	Mexico
6	Puesto De Tacos	San Luis Potosi	San Luis Potosi	Mexico
7	Tacos De La Estacion	Ciudad Victoria	Tamaulipas	Mexico
8	Cenaduria El Rincón De Tl...	San Luis Potosi	San Luis Potosi	Mexico
9	El Rincon De San Francisco	San Luis Potosi	San Luis Potosi	Mexico
10	Puesto De Tacos	San Luis Potosi	San Luis Potosi	Mexico
11	Gorditas Doa Gloria	San Luis Potosi	San Luis Potosi	Mexico
12	Restaurant Orizatlan	San Luis Potosi	San Luis Potosi	Mexico
13	La Vireina	San Luis Potosi	San Luis Potosi	Mexico
14	El Rincón De San Francisco	San Luis Potosi	San Luis Potosi	Mexico
15	El Pueblito	San Luis Potosi	San Luis Potosi	Mexico
16	Taqueria El Amigo	Ciudad Victoria	Tamaulipas	Mexico
17	Carreton De Flautas Y Mig...	Ciudad Victoria	Tamaulipas	Mexico
18	Puesto De Tacos	San Luis Potosi	San Luis Potosi	Mexico
19	Gorditas Doa Gloria	San Luis Potosi	San Luis Potosi	Mexico
20	Gordas De Morales	San Luis Potosi	San Luis Potosi	Mexico
21	El Oceano Dorado	Cuernavaca	Morelos	Mexico
22	Gorditas Doa Gloria	San Luis Potosi	San Luis Potosi	Mexico
23	Tacos Correcaminos	Ciudad Victoria	Tamaulipas	Mexico
24	El Cotorreo	Cuernavaca	Morelos	Mexico
25	Hamburguesas La Perica	Ciudad Victoria	Tamaulipas	Mexico

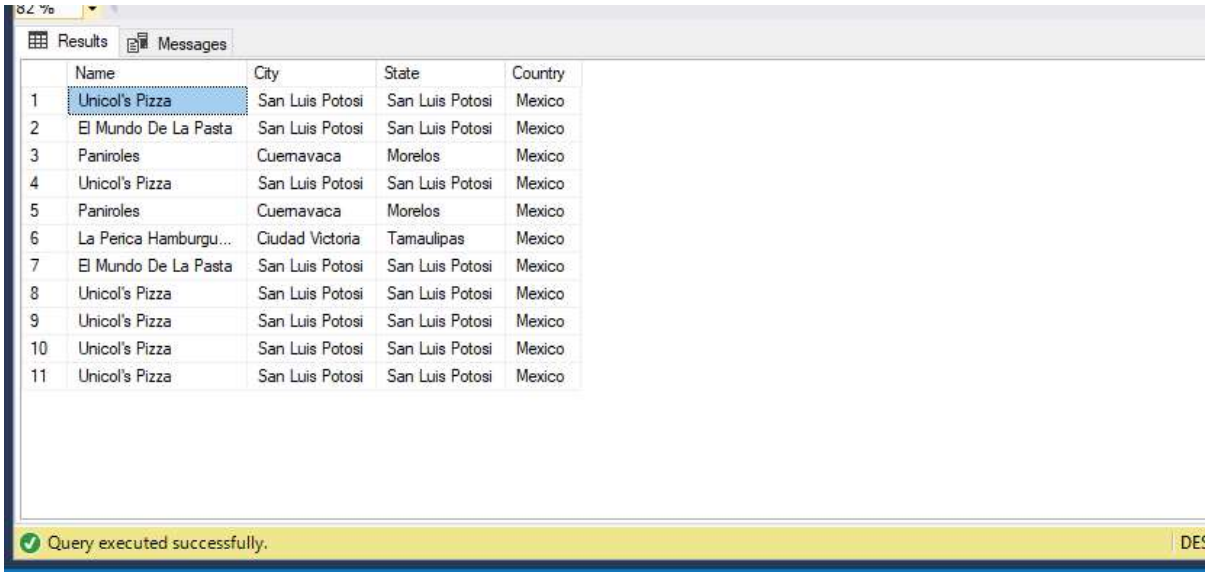
Query executed successfully. DESKTOP-PB7LD3N (16.0 RTM) DESKTOP-PB7

Now I write a querry for all the restaurant serving Italian cuisine and rating=1 using the same logic that I did for Mexican cuisine.

```
SELECT r.Name, r.City, r.State, r.Country
FROM [restaurants.csv] r
JOIN Ratings ra ON r.Restaurant_id = ra.Restaurant_id
JOIN Restaurant_Cuisines rc ON r.Restaurant_id = rc.Restaurant_id
WHERE ra.Overall_Rating = 1
AND rc.Cuisine = 'Italian';
```

Result:-

There are only 11 restaurants as follow



Results Messages

	Name	City	State	Country
1	Unicol's Pizza	San Luis Potosi	San Luis Potosi	Mexico
2	El Mundo De La Pasta	San Luis Potosi	San Luis Potosi	Mexico
3	Paniroles	Cuernavaca	Morelos	Mexico
4	Unicol's Pizza	San Luis Potosi	San Luis Potosi	Mexico
5	Paniroles	Cuernavaca	Morelos	Mexico
6	La Perica Hamburgu...	Ciudad Victoria	Tamaulipas	Mexico
7	El Mundo De La Pasta	San Luis Potosi	San Luis Potosi	Mexico
8	Unicol's Pizza	San Luis Potosi	San Luis Potosi	Mexico
9	Unicol's Pizza	San Luis Potosi	San Luis Potosi	Mexico
10	Unicol's Pizza	San Luis Potosi	San Luis Potosi	Mexico
11	Unicol's Pizza	San Luis Potosi	San Luis Potosi	Mexico

Query executed successfully.

Comparison:-

Now I am going to compare both the result set by looking into the result set and using my intuitions

1. There is a huge difference in the number of restaurants who are serving Mexican food and rating as 1 and the restaurants who are serving Italian food and ratings=1. This can be use to several reasons like geographic location, the different standards for these cuisines etc.
2. The low rating of Mexican restaurants might or might not reflect the lower quality of Mexican cuisine; however it could also reflect some other factors like customer expectations, competition in the market etc.
3. The dataset represent a region with higher number of Mexican restaurant having lower number of rating and it has large dataset than that of Italian cuisine which better in terms of rating than Mexican cuisine.

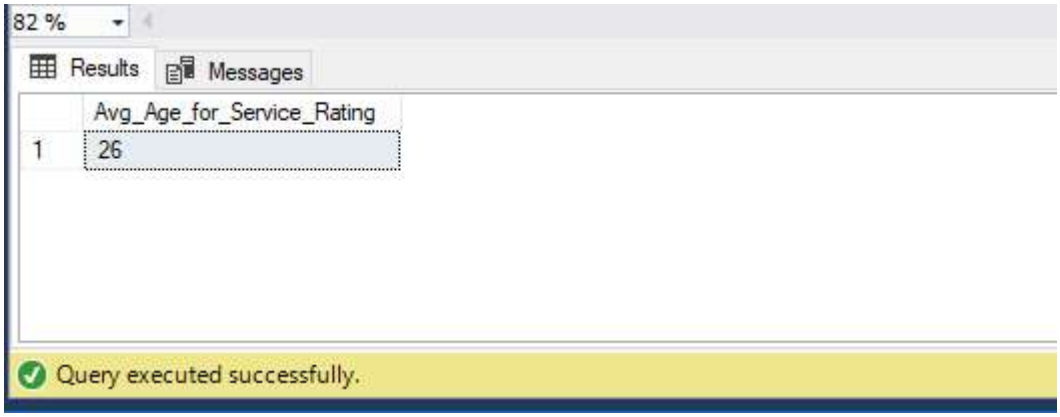
3.3. Calculate the average age of consumers who have given a 0 rating to the 'Service_rating' column. (NB: round off the value if it is a decimal).

Now for this querry I make a inner join on Consumer table and rating table on consumer_ID column so that I can select consumers who have rated, and then calculate the average age for the consumers using AVG(age) command and then use Where command to check for consumer who have given 0 rating respectively.

```
37  SELECT ROUND(AVG(Age), 0) AS Avg_Age_for_Service_Rating
38  FROM ratings
39  INNER JOIN Consumers ON ratings.Consumer_ID = consumers.Consumer_id
40  WHERE Service_Rating = 0;
41
42
```

Result:-

The average age for consumers is 26 who has given 0 rating



3.4 Write a query that returns the restaurants ranked by the youngest consumer. You should include the restaurant name and food rating that is given by that customer to the restaurant

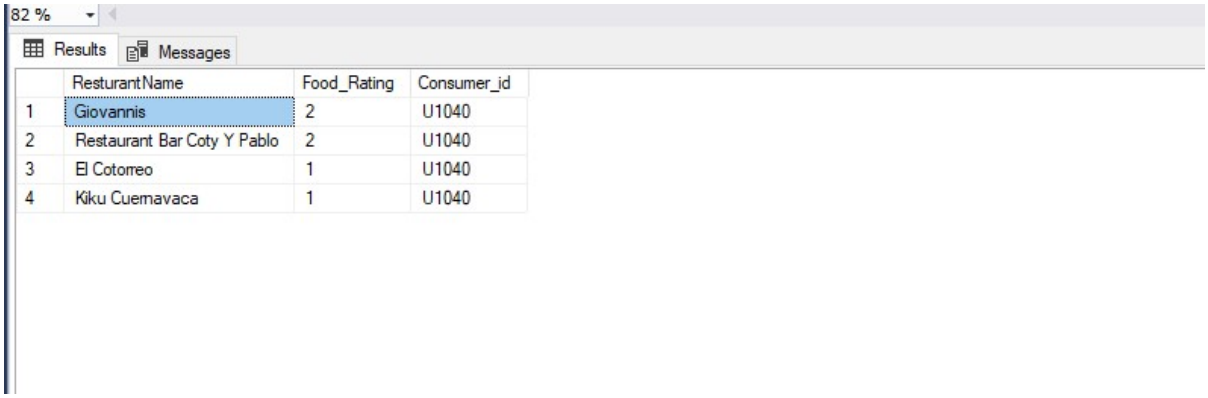
in your result. Sort the results based on food rating from high to low.

To answer the above querry, first of all I select the restaurant name from the restaurant table, Food rating and consumer_id column from the ratings table. Then I join the restaurant table and rating table using join command on column name restaurant_id. Than I write a subquerry in which I select consumer_id from consumer table and calculate their minimum age and than groupby it by their Comsumer_id and use a WHERE command to select only youngest consumer. And then sort the whole result by the Food rating of the restaurants.

Code:-

```
SELECT
    r.Name AS ResturantName,
    ra.Food_Rating,
    ra.Consumer_id
FROM
    [restaurants.csv] r
JOIN
    Ratings ra ON r.Restaurant_id = ra.Restaurant_id
JOIN
    (SELECT
        Consumer_id,
        MIN(Age) AS MinAge
     FROM
        Consumers
     GROUP BY
        Consumer_id) c ON ra.Consumer_id = c.Consumer_id
    WHERE
    c.MinAge = (SELECT MIN(Age) FROM Consumers)
ORDER BY
    ra.Food_Rating DESC;
```

Result:-



82 %

Results | Messages

	ResturantName	Food_Rating	Consumer_id
1	Giovannis	2	U1040
2	Restaurant Bar Coty Y Pablo	2	U1040
3	El Cotorreo	1	U1040
4	Kiku Cuernavaca	1	U1040

Now from the above result we can see that the youngest consumer with the age of 18 has rated the restaurants as follow and the result set is sorted with the Food rating from maximum to minimum

3.5 Write a stored procedure for the query given as: Update the Service_rating of all restaurants to '2' if they have parking available, either 'yes' or 'public'.

For this querry I simply create a stored procedure in which I set the service rating=2 and use a WHERE command to select the entire restaurant which have parking or public parking.

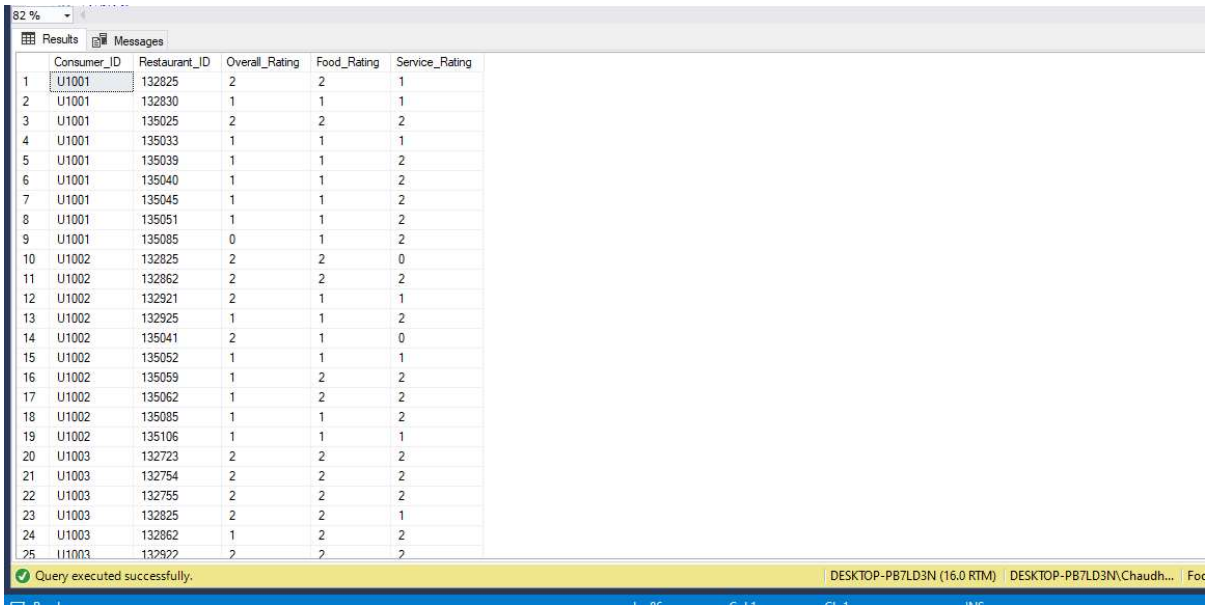
```
68  CREATE PROCEDURE UpdateServiceRating
69  AS
70  BEGIN
71      -- Update Service_rating of restaurants with parking available to '2'
72      UPDATE Ratings
73      SET Service_Rating = '2'
74      WHERE Restaurant_id IN (
75          SELECT Restaurant_id
76          FROM [restaurants.csv]
77          WHERE Parking IN ('yes', 'public')
78      );
79  END;
80
```

And then execute the above stored procedure

```
80
81  EXEC UpdateServiceRating;
82
```

Result:-

Now when we see the ratings table than the service ratings would be change to 2 for the restaurants which have parking or public parking.



	Consumer_ID	Restaurant_ID	Overall_Rating	Food_Rating	Service_Rating
1	U1001	132825	2	2	1
2	U1001	132830	1	1	1
3	U1001	135025	2	2	2
4	U1001	135033	1	1	1
5	U1001	135039	1	1	2
6	U1001	135040	1	1	2
7	U1001	135045	1	1	2
8	U1001	135051	1	1	2
9	U1001	135085	0	1	2
10	U1002	132825	2	2	0
11	U1002	132862	2	2	2
12	U1002	132921	2	1	1
13	U1002	132925	1	1	2
14	U1002	135041	2	1	0
15	U1002	135052	1	1	1
16	U1002	135059	1	2	2
17	U1002	135062	1	2	2
18	U1002	135085	1	1	2
19	U1002	135106	1	1	1
20	U1003	132723	2	2	2
21	U1003	132754	2	2	2
22	U1003	132755	2	2	2
23	U1003	132825	2	2	1
24	U1003	132862	1	2	2
25	U1003	132922	2	2	2

Query executed successfully.

3.6 You should also write four queries of your own and provide a brief explanation of the results which each query returns. You should make use of all of the following at least once: Nested queries-EXISTS

Nested queries-IN

System functions

Use of GROUP BY, HAVING and ORDER BY clauses

Now to answer above, here I am going to make 4 querries according to the demand of instructor.

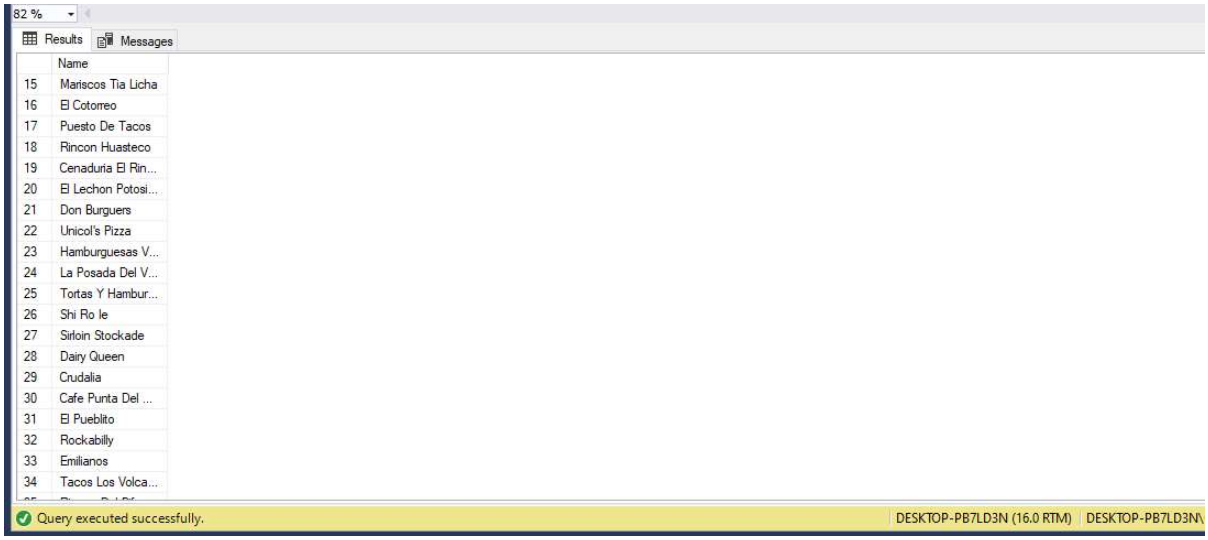
1- Restaurants with Average Rating Above 1

```
92
93  SELECT Name
94  FROM [restaurants.csv] r
95  WHERE EXISTS (
96    SELECT 1
97    FROM ratings rt
98    WHERE rt.Restaurant_id = r.Restaurant_id
99    AND rt.Service_Rating = 1
100 );
101
```

Now in this querry I select names of restaurants and then use a exist querry to check if the condition exist for each restaurant where service rating is 1. This querry aims for filtering the restaurants whose service rating is 1.

Results:-

1



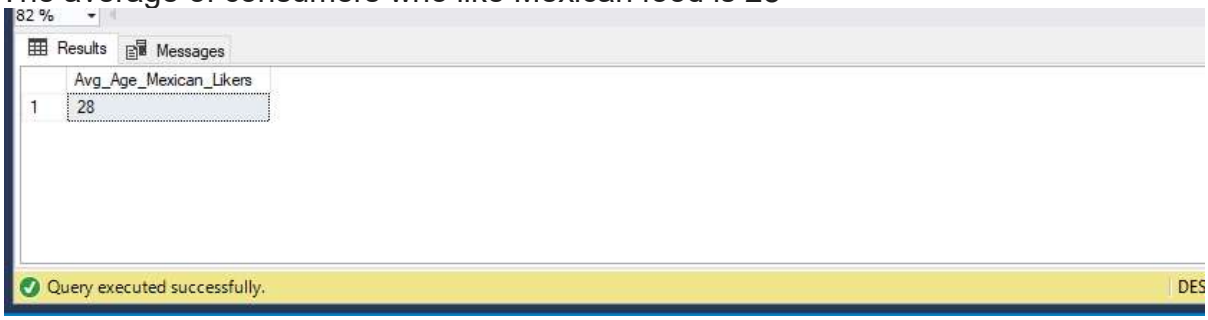
2- Average Age of Consumers Who Like Mexican Food

```
103 SELECT ROUND(AVG(Age), 0) AS Avg_Age_Mexican_Likers
104 FROM consumers c
105 WHERE Consumer_id IN (
106   SELECT Consumer_id
107   FROM ratings rt
108   WHERE rt.Food_Rating > 1
109   AND rt.Restaurant_id IN (
110     SELECT Restaurant_id
111     FROM restaurant_cuisines
112     WHERE Cuisine = 'Mexican'
113   )
114 );
115
```

In this querry first of all I calculate the average age of the consumers and than querry, used nested IN operator to filter consumers based on their food rating which is greater than 1 and than it use IN operator to select restaurants associated with those consumers and having Mexican cuisine.

Result:-

The average of consumers who like Mexican food is 28



1

3. Restaurants with Average Food Rating Above 1

```
117
118 SELECT
119   r.Name AS RestaurantName,
120   AVG(ra.Food_Rating) AS AvgFoodRating,
121   AVG(ra.Service_Rating) AS AvgServiceRating
122 FROM
123   [restaurants.csv] r
124 LEFT JOIN
125   Ratings ra ON r.Restaurant_id = ra.Restaurant_id
126 GROUP BY
127   r.Name
128 HAVING
129   AVG(ra.Food_Rating) > 1  -- Filter for restaurants with average food rating above 1
130 ORDER BY
131   AvgFoodRating DESC;
```

In this querry I have used Orderby, having and groupby commands. This querry join the restaurant table with ratings tables and than groupby by their names and than calculate the average food ratings of restaurants and than orderby by their average food ratings.

Result:-

From the results you can see that there are 4 resturants.

4. Restuarants that have received more than 1 over all rating

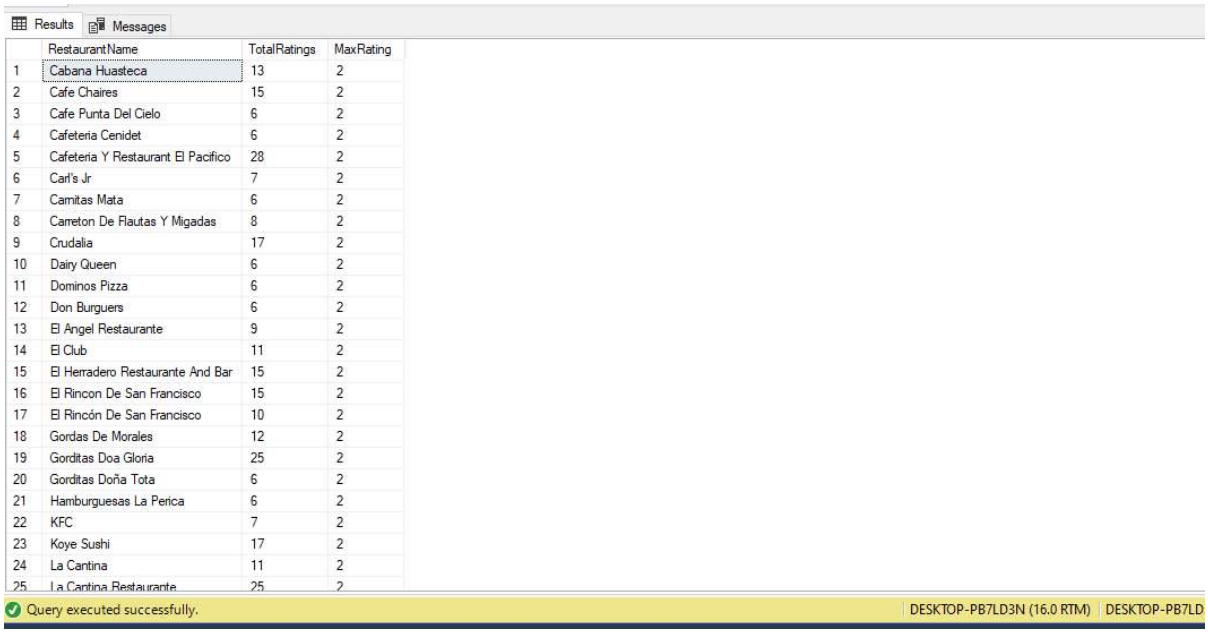
```
135
136 SELECT
137   r.Name AS RestaurantName,
138   COUNT(*) AS TotalRatings,
139   MAX(ra.Overall_Rating) AS MaxRating
140 FROM
141   [restaurants.csv] r
142 LEFT JOIN
143   Ratings ra ON r.Restaurant_id = ra.Restaurant_id
144 GROUP BY
145   r.Name
146 HAVING
147   COUNT(*) > 5
148 ORDER BY
149   MaxRating DESC;
150
```

This querry returns all the restaurants with their total number of ratings using system function Count(*) and the maximum ratings and than groupby them by their name than

1

using having function to only check out those restaurant which have total number of ratings greater than 5 and then at the last used orderby to order the output by their maximum rating pattern in descending order.

Result:-



Results | Messages

	RestaurantName	TotalRatings	MaxRating
1	Cabana Huasteca	13	2
2	Cafe Chaires	15	2
3	Cafe Punta Del Cielo	6	2
4	Cafeteria Cenidet	6	2
5	Cafeteria Y Restaurant El Pacifico	28	2
6	Carl's Jr	7	2
7	Carnitas Mata	6	2
8	Carreton De Flautas Y Migadas	8	2
9	Crudalia	17	2
10	Dairy Queen	6	2
11	Dominos Pizza	6	2
12	Don Burguers	6	2
13	El Angel Restaurante	9	2
14	El Club	11	2
15	El Herradero Restaurante And Bar	15	2
16	El Rincon De San Francisco	15	2
17	El Rincón De San Francisco	10	2
18	Gordas De Morales	12	2
19	Gorditas Doa Gloria	25	2
20	Gorditas Doña Tota	6	2
21	Hamburguesas La Perica	6	2
22	KFC	7	2
23	Koye Sushi	17	2
24	La Cantina	11	2
25	La Cantina Restaurante	25	2

Query executed successfully. DESKTOP-PB7LD3N (16.0 RTM) DESKTOP-PB7LD

4 Conclusion:-

The database design contain 4 tables restaurant, ratings, restaurant cuisine, consumers. There are following key functionalities of the database

I. **Data integrity:-**
Primary keys are associated with each table to ensure data integrity and foreign key are also created to maintain the connection between the tables. This will ensure the data integrity of the database.

II. **Relationships:-**
Relationships are build between the tables that will enables the querries to fetch comprehensive data like fetching data of restaurants only with the consumers detail.

III. **Scalability:-**
The database design ensure scalability like accommodating additional restaurants, consumers etc in case if we need to increase the dataset.