Restaurant reviews in Yelp

Project\_0502\_03\_Readme

Yudong Lu, Amal Byju, Chunghao Lee, Rachel Pothen

Data source - Dataset was manually curated from the Yelp website by the team.

Reference - <https://www.yelp.com>

How to test project with screenshots:

–Objective 1

--Display restaurants in each category in decreasing order of Rating

Using this approach, we can find out the successful and lagging restaurants in each category. This might give us insights for further analysis.



GO

CREATE VIEW v\_RestaurantDisplay AS

SELECT AVG(w.reviewRating) AS 'Rating', r.restaurantName AS 'Restaurant Name', r.restaurantCategory AS 'Restaurant Category'

FROM [TIPS.Review] w INNER JOIN [TIPS.Restaurant] r

ON w.restaurantID = r.restaurantID

GROUP BY r.restaurantCategory, r.restaurantName

ORDER BY restaurantCategory, Rating DESC OFFSET 0 ROWS

GO

SELECT \* FROM v\_RestaurantDisplay

From the results, it is quite apparent that breakfast is a category with a lot of restaurants.

--Display (upto) top 5 restaurants in each category in decreasing order of Rating. In case of a tie, both restaurants are printed.

We are interested in finding the most popular restaurants in each category. Using the result, we can come up with questions that focus on why the restaurants are successful in their respective categories.



GO

CREATE VIEW v\_TopFiveRestaurants AS

SELECT rs.Rating, rs.restaurantName AS 'Restaurant Name', rs.restaurantCategory AS 'Restaurant Category'

FROM (

SELECT AVG(w.reviewRating) AS 'Rating', r.restaurantName, r.restaurantCategory, Rank()

OVER (PARTITION BY r.restaurantCategory ORDER BY AVG(w.reviewRating) DESC) AS Rank

FROM [TIPS.Review] w INNER JOIN [TIPS.Restaurant] r

ON w.restaurantID = r.restaurantID

GROUP BY r.restaurantCategory, r.restaurantName

) rs WHERE Rank <= 5

GO

SELECT \* FROM v\_TopFiveRestaurants

There is only one restaurant in the bakery category and it has got a rating of 5. The third best American restaurant is Dry 85 and it has got a rating of 2. Restaurants in the American category can be improved by the dataset.

-- What is the highest rated restaurant in the breakfast category

Instead of poring over a long list, this business transaction helps us to focus the most successful restaurant in a particular category, in this case, the ‘Breakfast’ category.



GO

CREATE VIEW v\_HighRateRestinBreakfast AS

SELECT TOP 1 rt.restaurantID AS 'Restaurant ID', rt.restaurantName AS 'Restaurant Name', rt.restaurantAddr AS 'Restaurant Address', rt.restaurantWeb AS 'Restaurant Web', rt.restaurantCategory AS 'Restaurant Category', rt.restaurantPriceRange AS 'Restaurant Price Range', (rv.reviewRating) AS 'Average Rating'

FROM [TIPS.Restaurant] rt, [TIPS.Review] rv

WHERE rv.restaurantID = rt.restaurantID AND rt.restaurantCategory = 'Breakfast'

GROUP BY rt.restaurantID, rt.restaurantName, rt.restaurantAddr, rt.restaurantCategory, rt.restaurantWeb, rt.restaurantPriceRange, rv.reviewRating

ORDER BY AVG (rv.reviewRating) DESC

GO

SELECT \* FROM v\_HighRateRestinBreakfast

It is quite apparent that the best restaurant in the breakfast category has got a high price range. It focuses on affluent customers.

--Objective 2

--Display restaurants in each price range in decreasing order of rating. Here, we are interested in the restaurants that are successful or not successful in their price ranges.

GO

CREATE VIEW v\_RestaurantinPriceRange AS

SELECT AVG(w.reviewRating) AS 'Rating', r.restaurantName AS 'Restaurant Name', r.restaurantPriceRange AS 'Restaurant Price Range'

FROM [TIPS.Review] w INNER JOIN [TIPS.Restaurant] r

ON w.restaurantID = r.restaurantID

GROUP BY r.restaurantPriceRange, r.restaurantName

ORDER BY r.restaurantPriceRange, AVG(w.reviewRating) DESC OFFSET 0 ROWS

GO

SELECT \* FROM v\_RestaurantinPriceRange

In the one-dollar price category, we can see that the Foundry Bakery performs well. It has got a rating of 5 stars and is a bakery. In the two-dollar price category, the Iron Pig does a good job and it focuses on barbeque.

--Display top 5 restaurants in each price range in decreasing order of rating. In case of a tie, both restaurants are printed.

Instead of poring over the successful and lagging restaurant lists in each price range, we can instead look at the top 5 restaurants in each price range. This query helps us to focus on the categories of each restaurant. It is also possible to extract more data about certain restaurants through more SQL queries in order to do a deeper analysis.



GO

CREATE VIEW v\_TopFiveRestinPriceRange AS

SELECT rs.Rating, rs.restaurantName AS 'Restaurant Name', rs.restaurantPriceRange AS 'Restaurant Price Range'

FROM (

SELECT AVG(w.reviewRating) AS 'Rating', r.restaurantName , r.restaurantPriceRange , Rank()

OVER (PARTITION BY r.restaurantPriceRange ORDER BY AVG(w.reviewRating) DESC) AS Rank

FROM [TIPS.Review] w INNER JOIN [TIPS.Restaurant] r

ON w.restaurantID = r.restaurantID

GROUP BY r.restaurantPriceRange, r.restaurantName

) rs WHERE Rank <= 5

GO

SELECT \* FROM v\_TopFiveRestinPriceRange

--Objective 3

--We find the number of 1-star, 2-star, 3-star, 4-star and 5-star reviews for a chosen restaurant.

--In this case, we have chosen 'Preserve' restaurant. This is an example of a focused analysis on the customer reviews and popularity of the restaurant.



GO

CREATE VIEW v\_StarsForChosenRestaurant AS

SELECT reviewRating AS 'Review Rating', COUNT(\*) AS 'Frequency' FROM [TIPS.Review]

WHERE restaurantID = (SELECT restaurantID FROM [TIPS.Restaurant] WHERE restaurantName = 'Preserve')

GROUP BY reviewRating

ORDER BY reviewRating DESC OFFSET 0 ROWS

GO

SELECT \* FROM v\_StarsForChosenRestaurant

We found that 2 customers had a great experience with this restaurant. The lowest review rating is 3, which means OK experience. It seems that no one was disappointed with this restaurant.

--Objective 4

We find the number of 1-star, 2-star, 3-star, 4-star and 5-star reviews for a chosen user. In this case, we have chosen 'Kevin W.' as the user. We display the frequency of each rating for a chosen user. Using this result, we can find out the rating behaviour and the distribution of the rating for that user, and evaluate whether the rating for that user is relatively high or low by comparing to the average.

GO

CREATE VIEW v\_UserRatings AS

SELECT reviewRating AS 'Review Rating', COUNT(\*) AS 'Frequency' FROM [TIPS.Review]

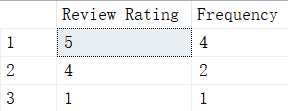
WHERE userID = (SELECT userID FROM [TIPS.User] WHERE userFName = 'Kevin' AND userLName = 'W.')

GROUP BY reviewRating

ORDER BY reviewRating DESC OFFSET 0 ROWS

GO

SELECT \* FROM v\_UserRatings



--Objective 5

We display the top 5 most popular collections and their details. By using this result, we can know which collections are the most popular to all users, so prospective entrepreneurs, reviewers, and food enthusiasts can study the trend of restaurants using it.

GO

CREATE VIEW v\_MostPopularCollection AS

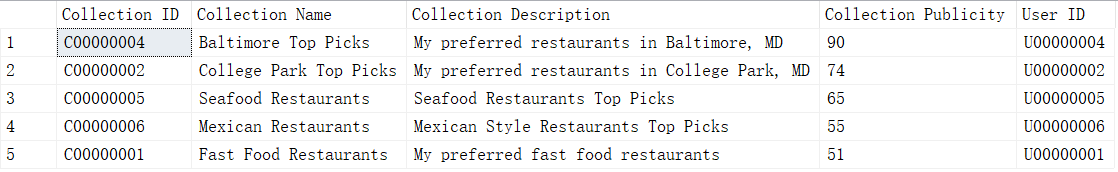
SELECT TOP 5 collectionID AS 'Collection ID', collectionName AS 'Collection Name', collectionDescription AS 'Collection Description',

collectionPublicity AS 'Collection Publicity', userID AS 'User ID' FROM [TIPS.Collection]

ORDER BY collectionPublicity DESC

GO

SELECT \* FROM v\_MostPopularCollection



--Objective 6

We display the top 5 most useful reviews for a chosen restaurant. In this case, we have chosen Kemoll's Chophouse as the restaurant. By using the result, we can know which reviews are the most useful to users. So these reviews could be displayed at the beginning of the review section for best user experiences.

GO

CREATE VIEW v\_TopFiveReviewsRestaurant AS

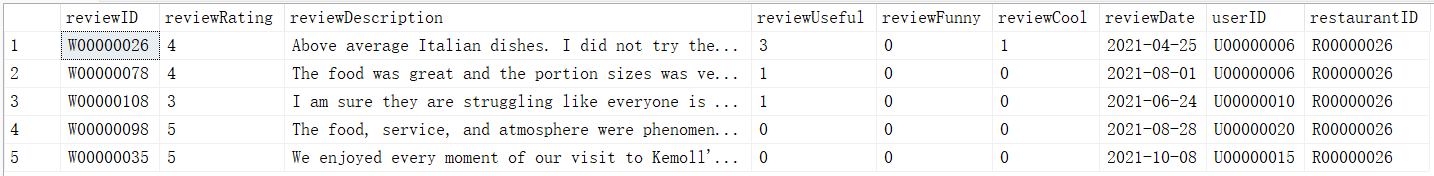
SELECT TOP 5 \* FROM [TIPS.Review]

WHERE restaurantID = (SELECT restaurantID FROM [TIPS.Restaurant] WHERE restaurantName = 'Kemoll''s Chophouse')

ORDER BY reviewUseful DESC

GO

SELECT \* FROM v\_TopFiveReviewsRestaurant



--Objective 7

We display the top 5 categories with the least number of restaurants. Using this result, we can learn which category is the most rare in the area, and the entrepreneurs could avoid competition by using this data to start their restaurant.

CREATE VIEW v\_TopFiveCategories AS

SELECT TOP 5 COUNT(\*) AS 'No. of Restaurants', restaurantCategory AS 'Restaurant Category' FROM [TIPS.Restaurant]

GROUP BY restaurantCategory

ORDER BY COUNT(\*) ASC

GO

SELECT \* FROM v\_TopFiveCategories



--Objective 8

We display the top 5 categories with the most number of restaurants. Using these results, we

can find out the most famous restaurant categories. These results can be useful

for entrepreneurs when trying to understand the prospects of opening a

restaurant for different categories.

Query:

CREATE VIEW v\_TopFiveCatRestaurants AS

SELECT TOP 5 COUNT(\*) AS 'No. of Restaurants', restaurantCategory AS 'Restaurant Category' FROM [TIPS.Restaurant]

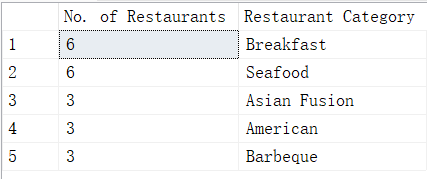
GROUP BY restaurantCategory

ORDER BY COUNT(\*) DESC

GO

SELECT \* FROM v\_TopFiveCatRestaurants

Output:



From these results, we can infer that the restaurant categories ‘Breakfast’ and ‘Seafood’ have the most restaurants which indicate that the restaurants belonging to these categories could have the highest demand.

--Objective 9

We display the top 10 users who have posted the maximum number of reviews. Using these results, we can identify the most active users and their information and demographics. This information can be useful for entrepreneurs when trying to analyze their customers.

Query:

CREATE VIEW v\_TopTenUserMaxReviews AS

SELECT TOP 10 u.userID AS 'User ID', u.userFName AS 'First Name', u.userLName AS 'Last Name', u.userDOJ AS 'Date of Join', u.userLoc AS 'Location', u.userElite AS 'Elite User', COUNT(\*) AS 'Review Count'

FROM [TIPS.User] u INNER JOIN [TIPS.Review] r

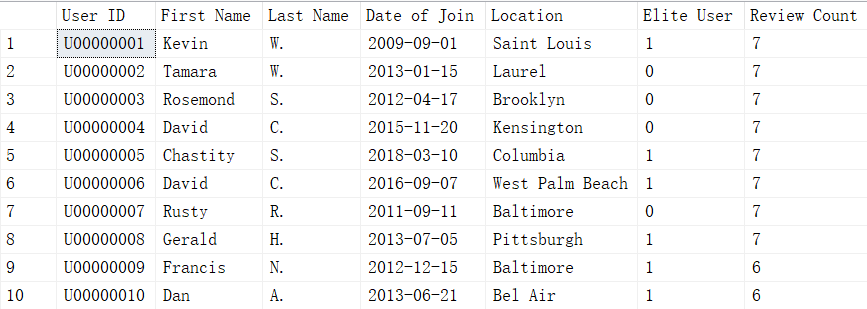
ON u.userID = r.userID

GROUP BY u.userID, u.userFName, u.userLName, u.userLoc, u.userDOJ, u.userElite

GO

SELECT \* FROM v\_TopTenUserMaxReviews

Output:



Top 10 users who have posted the maximum number of posts.

Query:

CREATE VIEW v\_MostPostUser AS

SELECT TOP 10 u.userID AS 'User ID', u.userFName AS 'First Name', u.userLName AS 'Last Name', u.userDOJ AS 'Date of Join', u.userLoc AS 'Location', u.userElite AS 'Elite User', COUNT(\*) AS 'Post Count'

FROM [TIPS.User] u INNER JOIN [TIPS.Post] p

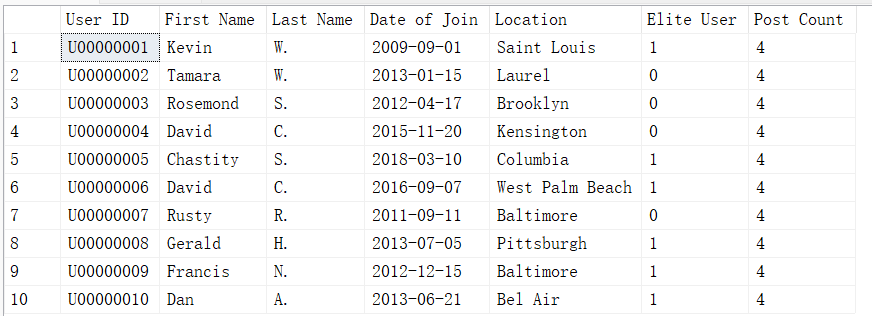
ON u.userID = p.userID

GROUP BY u.userID, u.userFName, u.userLName, u.userLoc, u.userDOJ, u.userElite

GO

SELECT \* FROM v\_MostPostUser

Output:



--Objective 10

We display the top 3

most recent posts for a chosen restaurant. Using these results, a

restaurant can find out the most recent questions and answers posted related to

their restaurant. By analyzing the post name, restaurants can understand the

recent topics on which questions were asked.

Query:

CREATE VIEW v\_TopThreeMostRecentPosts AS

SELECT TOP 3 postID AS 'Post ID', postName AS 'Post Name', postContent AS 'Post Content', postType AS 'Post Type',

postDate AS 'Post Date', userID AS 'User ID', restaurantID AS 'Restaurant ID'

FROM [TIPS.Post]

WHERE restaurantID =

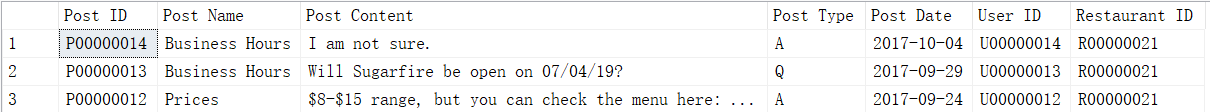
(SELECT restaurantID FROM [TIPS.Restaurant] WHERE restaurantName = 'Sugarfire Smoke House')

ORDER BY postDate DESC

GO

SELECT \* FROM v\_TopThreeMostRecentPosts

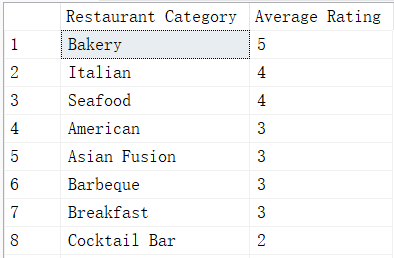
Output:



From these results, we can see that two out of the three recent posts were related to Business Hours. Using this information, the restaurant can add more information regarding their business hours so that the users can be well aware about it.

--Objective 11

--What are the favorite restaurant categories for elite user in descending order. From this analysis, we wish to find the popular restaurant categories among the foodies and critics and see whether they differ from the popular categories of the general public.



GO

CREATE VIEW v\_FavCatEliteUser AS

SELECT rt.restaurantCategory AS 'Restaurant Category', AVG (rv.reviewRating) as 'Average Rating'

FROM [TIPS.Restaurant] rt, [TIPS.Review] rv, [TIPS.User] u

WHERE rv.restaurantID = rt.restaurantID AND rv.userID =u.userID AND u.userElite=1

GROUP BY rt.restaurantCategory

ORDER BY AVG (rv.reviewRating) DESC OFFSET 0 ROWS

GO

SELECT \* FROM v\_FavCatEliteUser

It seems like the favorite restaurant categories of the elite users definitely differ from that of the general public.

--Objective 12

--What are the restaurants in the collection of a chosen user.

--In this case, we have chosen Kevin W. as the user. This is an example of focused analysis of the preferred restaurants of a chosen user.

GO

CREATE VIEW v\_CollUserRestaurants AS

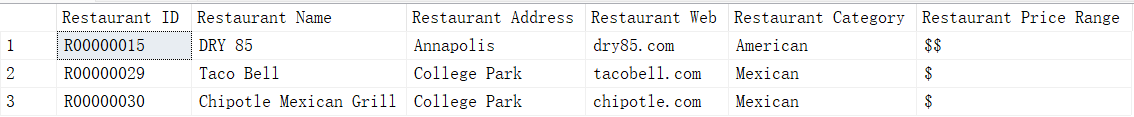
SELECT r.restaurantID AS 'Restaurant ID', r.restaurantName AS 'Restaurant Name', r.restaurantAddr AS 'Restaurant Address', r.restaurantWeb AS 'Restaurant Web', r.restaurantCategory AS 'Restaurant Category', r.restaurantPriceRange AS 'Restaurant Price Range'

FROM [TIPS.User] u, [TIPS.Collection] cl, [TIPS.Restaurant] r, [TIPS.Contains] ct

WHERE cl.userID=u.userID AND ct.collectionID=cl.collectionID AND ct.restaurantID= r.restaurantID AND u.userFName='Kevin' AND u.userLName='W.'

GO

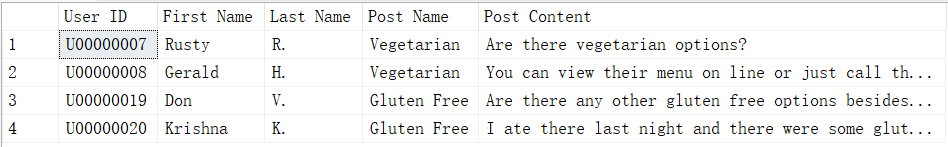
SELECT \* FROM v\_CollUserRestaurants



We can see that this user may prefer American and Mexican restaurants, which might be the reason why they have included these restaurants in a collection.

--Objective 13

-- WHO are asking or answering questions about vegetarien(vegan) or gluten. Here, we can look into the concerns and questions posed by vegetarians and vegans. This also includes people concerned about gluten.



GO

CREATE VIEW v\_QnVeganGluten AS

SELECT u.userID AS 'User ID', u.userFName AS 'First Name', u.userLName AS 'Last Name', p.postName AS 'Post Name', p.postContent AS 'Post Content'

FROM [TIPS.User] u, [TIPS.Post] p

WHERE p.userID=u.userID AND ((p.postContent LIKE '%gluten%' OR p.postName LIKE '%gluten%') OR (p.postContent LIKE '%veg%' OR p.postName LIKE '%veg%'))

GO

SELECT \* FROM v\_QnVeganGluten

It is quite clear that restaurants should provide menu items that are vegetarian or gluten free in order to cater to some customers.

--Objective 14

-- Find all crab dishes and rank all crab dishes by price from low to high. Here, we attract the attention of the users or clients that are interested in crab dishes and want to find out the cheapest and most expensive crab dishes served across restaurants.



DROP VIEW v\_CrabDishes

GO

CREATE VIEW v\_CrabDishes AS

SELECT r.restaurantID AS 'Restaurant ID',r.restaurantName AS 'Restaurant Name', m.itemName AS 'Item Name', m.itemDesc AS 'Item Description', m.itemPrice AS 'Item Price'

FROM [TIPS.Restaurant] r, [TIPS.MenuItem] m

WHERE m.restaurantID = r.restaurantID AND m.itemDesc LIKE '%crab%'

ORDER BY m.itemPrice OFFSET 0 ROWS

GO

SELECT \* FROM v\_CrabDishes

From the analysis, it is found that crab dishes can be as cheap as 5.5 dollars and can be as expensive as 37 dollars.