|  |  |
| --- | --- |
| Database Project  05/09/2018 | **tAEHOON KIM**  **jOVANY FUNES**  **mELISSA NUBLA**  Database 302 |
|  |  |

**Executive Summary**

This project focuses on the reliability of Yelp’s user ratings or average number of stars for different establishments across the country. This looks specifically at the most important information like location, contacts, hours, menu, average stars/ratings, etc. This is then related to external factors like average income in a certain area. Instead of focusing on the internal factors that affect the ratings like the quality of food, service, and sanitary conditions, external factors such as the average income, and average ratings in a specific zip code area were considered.

Information had initially been collected through the yelp website as it provides a zip file of selected data, which contained enough information for this study. However, there had been initial issues with the database, such as difficulty extracting and running the files. A simple solution to run the files through python had initially worked, but was clearly not sustainable for the size of the database needed. Finally, another search for yelp data yielded Kaggle.com, where the same yelp data had been made available in a file that was uncorrupted. After this process, the database is then populated, cleaned, normalized, processed, and analyzed.

Some parameters that the group focused on were the relationship of the rates and the general locations, average stars in a certain area, number of restaurants near a location, average income in the area, and the like. Other things observed in the study were subjects such as the polarity of reviews in a certain area. This looks at the percentage of positive to negative ratings for a given business.

This group utilized a few different data tools and/or languages to run the study. These were: Python for general data collection and some polarity tests. SQL, Excel and Access, for the general management of the database, from cleaning up, data queries, and joining information. The group also used Insights for some simple charting and data visualization, and finally, Caspio, a website used to create a UI for the database of this project.

The group has created a page on ArcGIS using the Insights application to demonstrate the data that the group has found. The link can be found [here](https://agis.maps.arcgis.com/home/item.html?id=5203d17c0950454c9876e6d677c59c23). (Because of a glitch, some of the language on the page is in Korean. This can be fixed by using a simple translator on your browser, or reaching out to one of the researchers.) The group has also created a simple website to provide a UI to the audience. It can be accessed through this [link](http://coloreddata.com/) or the address: www.coloreddata.com

**Problem Description**

Yelp is the company that provides people information about what local businesses are located in which specific area and how people think of the business by showing the star out of 5. However, we thought that since the stars are reflected on how people think, it can be very subjective. For example, when two people give 3 stars, it may be possible that the stars mean differently. One might say that 3 stars means ‘tasty’ to him/her when other say that it indicates ‘just okay’. Therefore, we realized that it would be better if there are many factors that supplement the drawback of star systems so that users can have more objective review on the local businesses. We found that on Yelp, it does not show the average stars in that local area. Our design shows not only the average star of its business, but also the overall average stars of businesses in the location so that users can compare how high the business is rated in the area.

Also, we assume that as people have experienced fancier and more expensive foods, they tend to be stricter on giving stars; people who are financially prosperous have the experience. Therefore, we added ZIP\_INCOME table so that people can consider the same amount of stars differently by locations. For example, users compare the average income by city and if one’s mean is higher than others’, people can think that rates on restaurants in one city may be more strictly measured than in the other cities.

Furthermore, we show the review count of each business and the number of polarities of review texts on each business whether they are positive or negative. As the number of review is higher, the rate become more reliable since then the review is reflected by many people. Moreover, because the user can see the polarity, they can more accurately assume the meaning of stars without reading the whole text. For instance, if someone gave 3 stars with negative polarity text, the user can think that the 3 star is more like negative sign for the reviewer, not neutral.

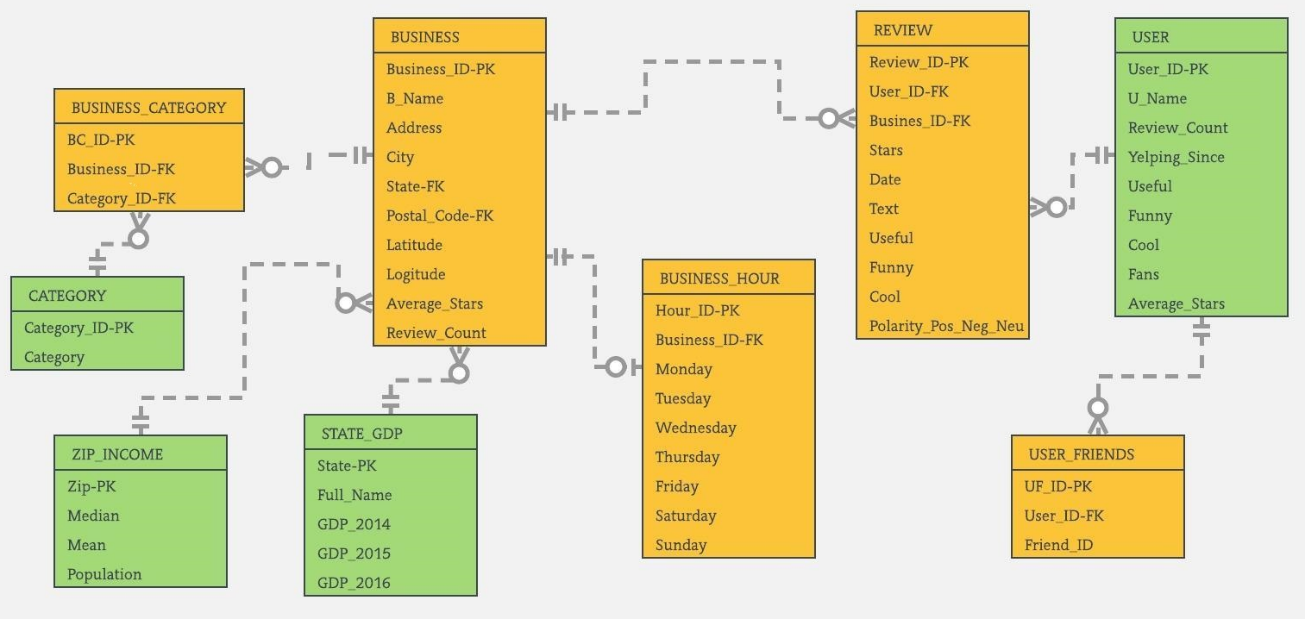
We found that McDonald’s, Subway, and Taco Bell are the three businesses that are the most widely located in our BUSINESS table. Also, we thought that they are the restaurants that provide the same quality of food in every stores. Therefore, we consider them as a good indicator to find out how strict people are on a specific area to give the stars like Big Mac index in economy. Unfortunately, those businesses do not cover every city we have in the dataset, but can be a helpful supplementary for star systems.

**Database Design**

*Business Rules*

* If BUSINESS or USER row is deleted, a row of REVIEW that is connected to those tables is also deleted.
* Any businesses that are not located in state that is in STATE\_GDP or in postal code that is in ZIP\_INCOME cannot be inserted on BUSINESS table.
* If CATEGORY row is to be deleted and that row is connected to any BUSINESS\_CATEGORY row, the CATRGORY row deletion will not be allowed.

*ER-Diagram*

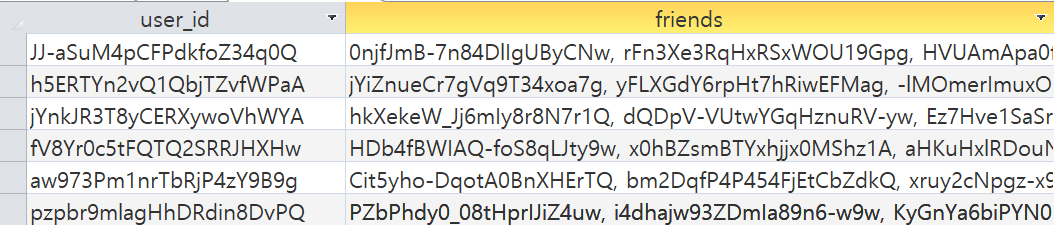


*Normalization Process*

* Started with four tables, BUSIENSS, BUSINESS\_HOUR, REVIEW, and USER
* Found the violation of 1NF in Both BUSINESS and USER table.

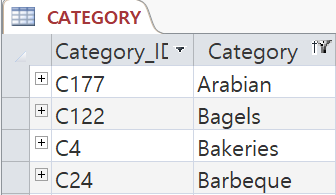
(More than one element is located in one cell)





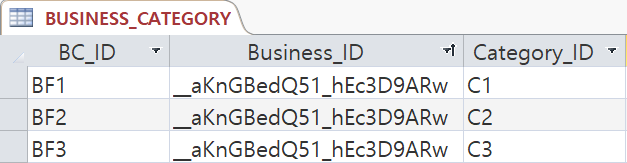
* The attributes, categories and friends, would have multivalued dependency if they were put in each cell. Therefore, our group created new tables and put each element in a cell in the newly created tables.

CATEGORY with a primary key, Category\_ID



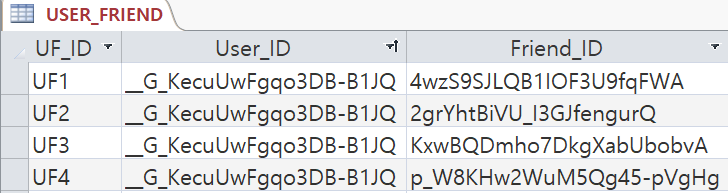
BUSIENSS\_CATEGORY (Intersection Table) with a primary key, BC\_ID

* Link the BUSINESS and CATEGORY table
* Two foreign keys: Business\_ID and Category\_ID



USER\_FRIEND with a primary key, UF\_ID

* Foreign key: User\_ID

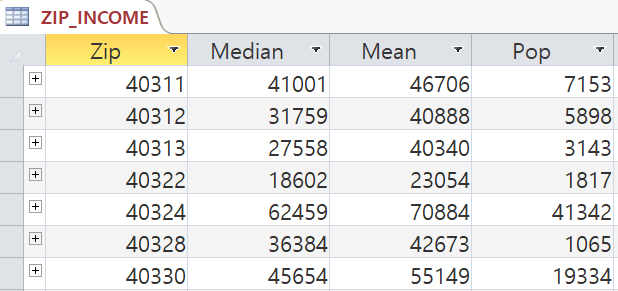


*Addition of new tables and an attribute*

* In order to supplement the star systems, we needed to add more data in our database.
* Decided to add two more tables related to income and one attribute in REVIEW table.

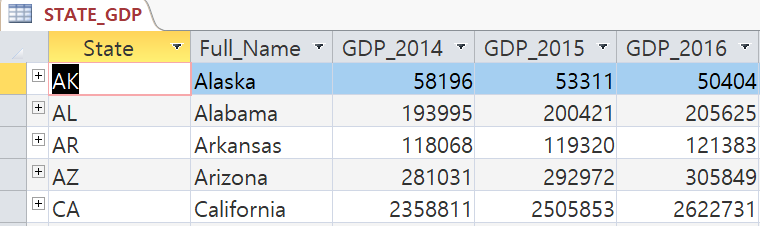
ZIP\_INCOME with a primary key, Zip

* The primary key is placed in BUSINESS table as a foreign key. It is called Postal\_Code in BUSINESS.



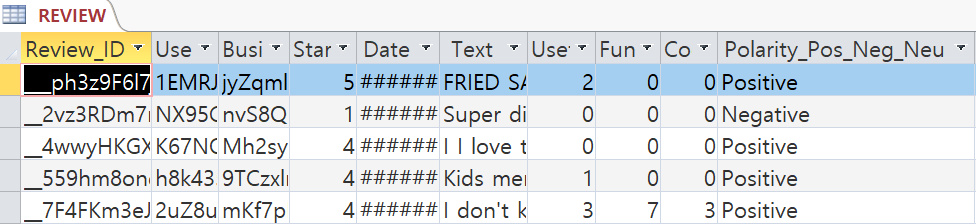
STATE\_GDP with a primary key, State

* The primary key is placed in BUSINESS table as a foreign key.



New Attribute in REVIEW

* A new attribute, Polarity\_Pos\_Neg\_Neu, is inserted.



**Data Model**

CATEGORY ----- BUSINESS\_CATEGORY (1:M)

BUSINESS ----- BUSINESS\_CATEGORY (1:M)

* BUSINESS and CATEGORY tables have a Many to Many (N:M) relationship. One BUSINESS can have many categories and one CATEGORY can be involved in many businesses. Therefore, we made an intersection table, BUSINESS\_CATEGORY. The table has one row for each line between BUSINESS and CATEGORY and each of two tables gets 1:M relationship with the intersection table. Originally, the intersection table has the combination of parents keys (Primary keys of BUSINESS and CATEGORY) as primary key. However, we made its own primary key, BC\_ID, since we did not want the composite key, so the parent keys, Business\_ID and Category\_ID, became foreign keys in the REVIEW table.

BUSINESS ----- REVIEW (1:M)

USER ----- REVIEW (1:M)

* BUSINESS and USER each has 1:M relationship with REVIEW: one BUSINESS or one USER can have many reviews, but one REVIEW can only have one BUSINESS and USER. Therefore, BUSINESS and USER are the parent sides while REVIEW is child side, so primary keys of BUSINESS and USER, Business\_ID and User\_ID, become foreign key when they are linked to REVIEW.

ZIP\_INCOME ----- BUSINESS (1:M)

STATE\_GDP ----- BUSINESS(1:M)

* In our model, every BUSINESS is enforced to be in one of postal code in ZIP\_INCOME and states in STATE\_GDP. Therefore, BUSINESS have 1:M relationship with those two tables: one ZIP or STATE can have many businesses, but one BUSIENSS can have only one zip and state. Since they have 1:M relationship, primary keys of ZIP\_INCOME and STATE\_GDP, Zip and State, are placed in BUSINESS when the tables are linked.

BUSINESS ----- BUSINESS\_HOUR (1:1)

* One BUSINESS have their own open hour, so BUSINESS and BUSINESS\_HOUR have a 1:1 relationship. Primary key of BUSINESS, Business\_ID, is placed in BUSINESS\_HOUR as a foreign key when they are linked together. Since sometimes BUSINESS does not provide its open hour, minimum cardinality is zero to BUSINESS\_HOUR.

USER ----- USER\_FRIENDS (1:M)

* USER\_FRIEND is an intersection table between USER and FRIEND, which have M to N relationship, but we did not include FRIEND table since we would not use it. Therefore, USER and USER\_FRIEND have 1:M relationship. Also, friend can be logically existed only when user exist, so minimum cardinality of USER is one while it is zero to USER\_FRIEND. Primary key of USER, User\_ID, is placed in USER\_FRIEND as a foreign key when they are linked together.

**System Documentation**

1. **SELECT** BU.City, BU.Business\_ID, [B\_Name], Review\_Count, Avg\_Star\_Business, Avg\_Star\_City

**FROM** (SELECT B.City, RS.Business\_ID, [B\_Name], Review\_Count, Avg\_Star\_Business

FROM BUSINESS AS B,

(SELECT Business\_ID, count(Business\_ID) AS Review\_Count,

Round(Sum(Stars)/count(Business\_ID), 1) AS Avg\_Star\_Business

FROM REVIEW GROUP BY Business\_ID) AS RS

WHERE RS.Business\_ID = B.Business\_ID) AS BU

**RIGHT OUTER JOIN**

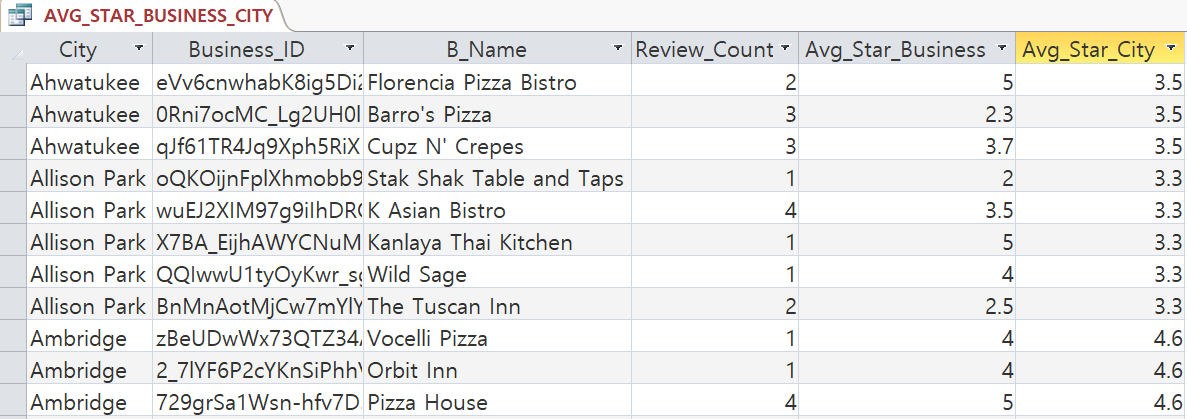
(SELECT City, ROUND(AVG(Stars),1) AS AVG\_Star\_City

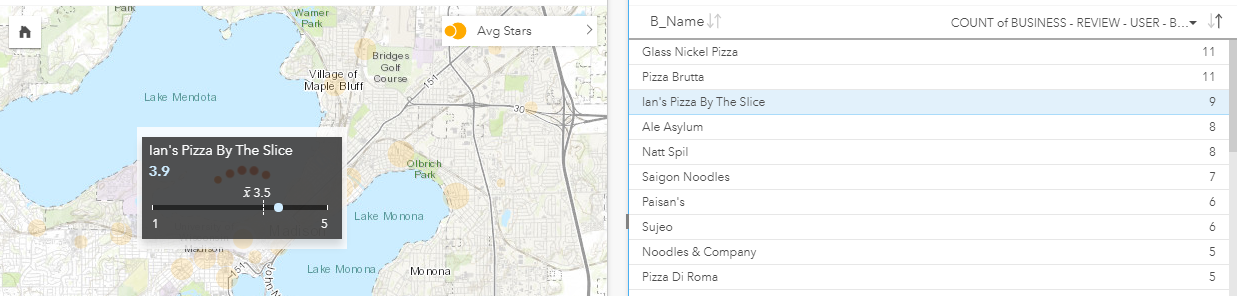
FROM REVIEW AS R, BUSINESS AS B

WHERE R.Business\_ID = B.Business\_ID GROUP BY City) AS CI

**ON** BU.City = CI.City;

* It enables people to understand how high the business is rated in its city by comparing its own star and average star of whole businesses in the city. Also, by looking at the review count, users are able to know that how many people’s opinions are reflected on the stars. Therefore, it shows the level of objectivity.





1. **SELECT** DISTINCT State, City, Postal\_Code, B\_Name, Category

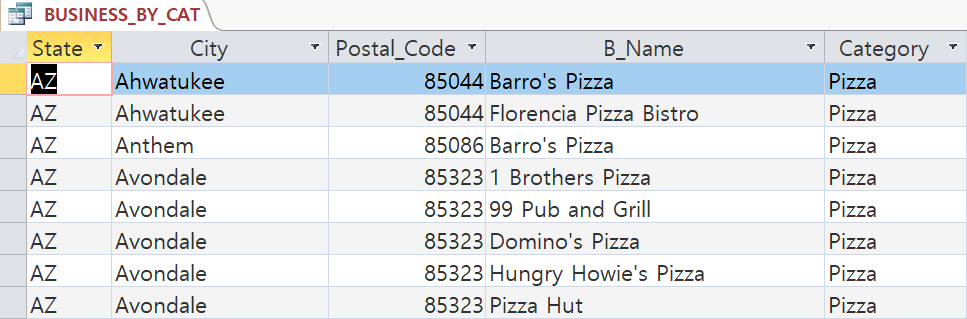
**FROM** BUSINESS AS B, BUSINESS\_CATEGORY AS BC, CATEGORY AS C

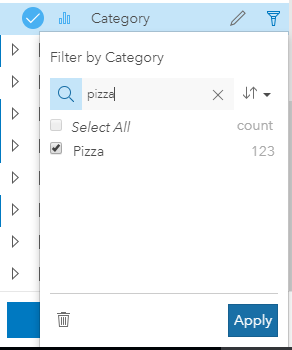
**WHERE** B.Business\_ID = BC.Business\_ID and BC.Category\_ID = C.Category\_ID and

C.Category LIKE '\*Pizza\*'

**ORDER BY** State, City;

* It shows every pizza stores in our database. By adding the state or city and changing the category in WHERE condition, people are able to search specific kinds of restaurant in a specific location.





1. **SELECT** \*

**FROM** (SELECT State, City, Count(City) AS NumOfRes\_In\_City, ROUND(AVG(Mean),2)

AS Avg\_Mean\_City\_Income

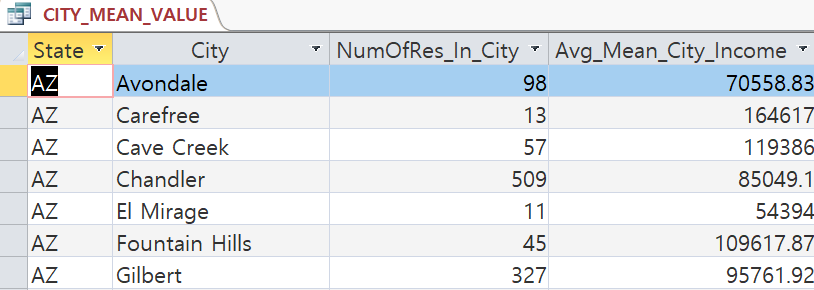
FROM BUSINESS AS B, ZIP\_INCOME AS Z

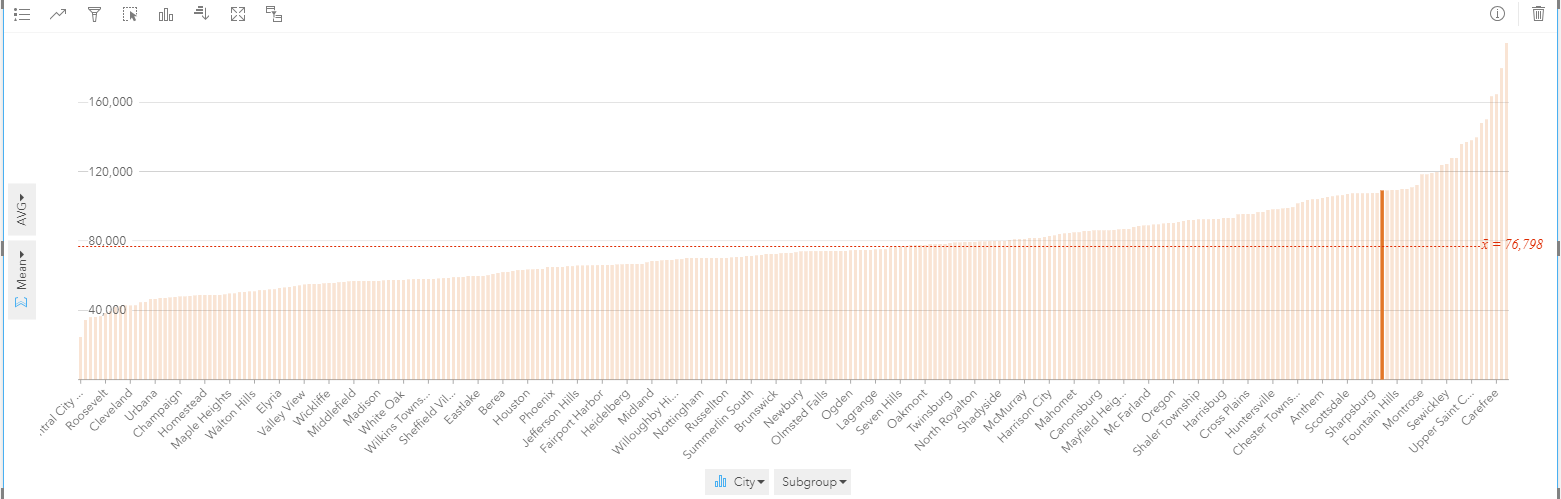
WHERE B.Postal\_Code = Z.Zip

GROUP BY City, State

ORDER BY State, City) AS A;

* It shows the average income of each city. Our group assumes that people with higher income would be stricter on giving stars. Therefore, it would be one of good indicators to understand about the rate objectively. Higher the income of a city, stricter people in the city are to give star.





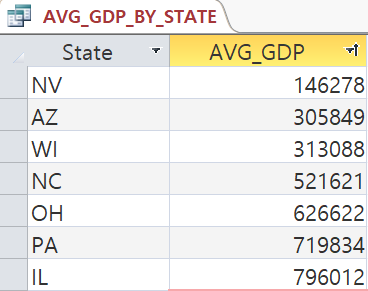
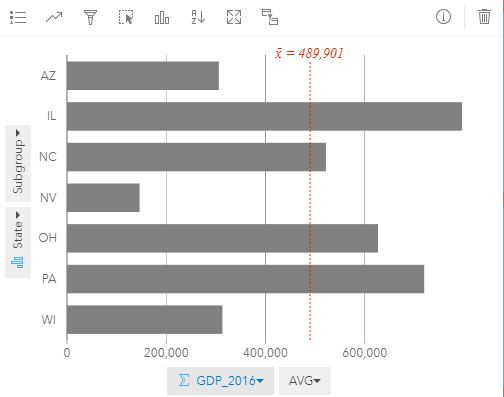
1. **SELECT** State, ROUND(GDP\_2016, 2) AS AVG\_GDP

**FROM** STATE\_GDP

**WHERE** State IN (SELECT DISTINCT State

FROM BUSINESS);

* Like #3 above, our group thought that average GDP can be another good indicator to let people understand the true meaning of the stars. Higher the GDP of state, stricter people in the state are to give star.

1. **SELECT** S5.state, STAR\_1, STAR\_2, STAR\_3, STAR\_4, STAR\_5

**FROM** (SELECT S123.state, STAR\_1, STAR\_2, STAR\_3, STAR\_4

FROM (SELECT S12.state, STAR\_1, STAR\_2, STAR\_3

FROM (SELECT S1.state, STAR\_1, STAR\_2

FROM (SELECT state, count(state) AS STAR\_1

FROM Review AS R, Business AS B

WHERE R.business\_id = B.business\_id and R.stars = 1

GROUP BY state) AS S1,

(SELECT state, count(state) AS STAR\_2

FROM Review AS R, Business AS B

WHERE R.business\_id = B.business\_id and R.stars = 2

GROUP BY state) AS S2

WHERE S1.state = S2.state) AS S12,

(SELECT state, count(state) AS STAR\_3

FROM Review AS R, Business AS B

WHERE R.business\_id = B.business\_id and R.stars = 3

GROUP BY state) AS S3

WHERE S12.state = S3.state) AS S123,

(SELECT state, count(state) AS STAR\_4

FROM Review AS R, Business AS B

WHERE R.business\_id = B.business\_id and R.stars = 4

GROUP BY state) AS S4

WHERE S123.state = S4.state) AS S1234,

(SELECT state, count(state) AS STAR\_5

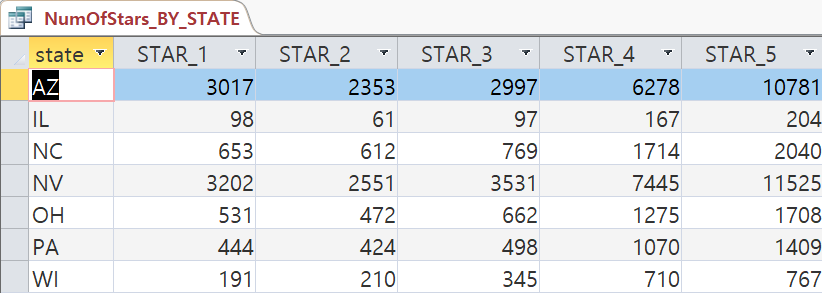
FROM Review AS R, Business AS B

WHERE R.business\_id = B.business\_id and R.stars = 5

GROUP BY state) AS S5

**WHERE** S1234.state = S5.state;

* It indicates how people in one’s state tend to give stars. For example, we can tell people in WI tend to give a higher star than people in IL.



1. **SELECT** N.State, N.City, N.B\_Name, Count\_Positive, Count\_Negative

**FROM** (SELECT State, City, B\_Name, R.Business\_ID, COUNT(R.Business\_ID) AS Count\_Positive

FROM BUSINESS AS B, REVIEW AS R

WHERE B.Business\_ID = R.Business\_ID and Polarity\_Pos\_Neg\_Neu = 'Positive'

GROUP BY R.Business\_ID,B\_Name, City, State) AS P

**RIGHT OUTER JOIN**

(SELECT State, City, B\_Name, R.Business\_ID, COUNT(R.Business\_ID) AS Count\_Negative

FROM BUSINESS AS B, REVIEW AS R

WHERE B.Business\_ID = R.Business\_ID and Polarity\_Pos\_Neg\_Neu = 'Negative'

GROUP BY R.Business\_ID,B\_Name, City, State) AS N

**ON** P.Business\_ID = N.Business\_ID

**UNION**

**SELECT** P.State, P.City, P.B\_Name, Count\_Positive, Count\_Negative

**FROM** (SELECT State, City, B\_Name, R.Business\_ID, COUNT(R.Business\_ID) AS Count\_Positive

FROM BUSINESS AS B, REVIEW AS R

WHERE B.Business\_ID = R.Business\_ID and Polarity\_Pos\_Neg\_Neu = 'Positive'

GROUP BY R.Business\_ID,B\_Name, City, State) AS P

**LEFT OUTER JOIN**

(SELECT State, City, B\_Name, R.Business\_ID, COUNT(R.Business\_ID) AS Count\_Negative

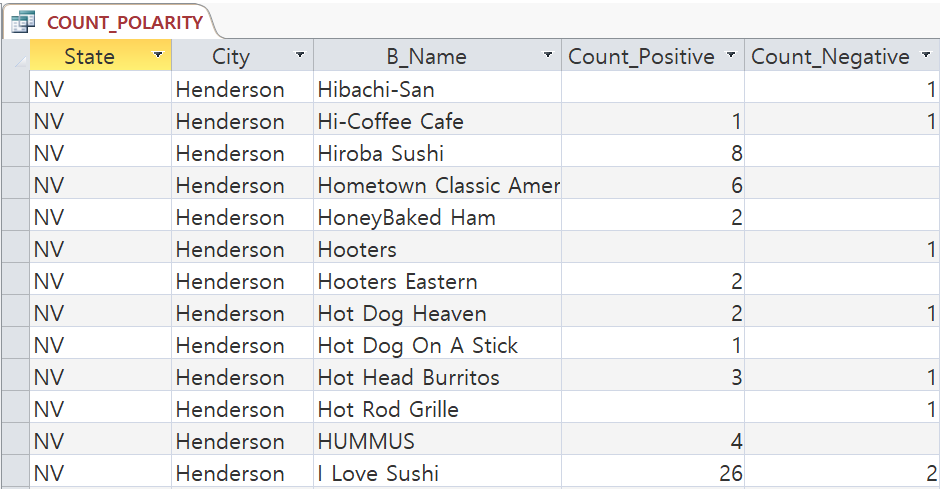
FROM BUSINESS AS B, REVIEW AS R

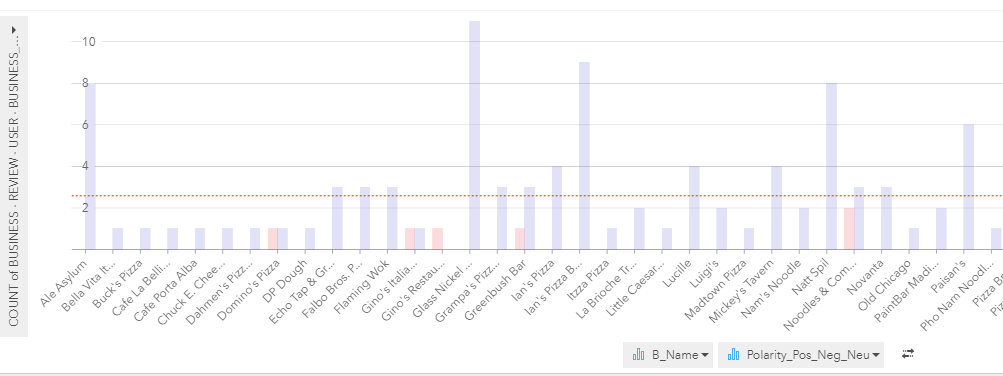
WHERE B.Business\_ID = R.Business\_ID and Polarity\_Pos\_Neg\_Neu = 'Negative'

GROUP BY R.Business\_ID,B\_Name, City, State) AS N

**ON** P.Business\_ID = N.Business\_ID;

* It shows the polarity of review text of each business in REVIEW table. By looking at the chart below, people can know what the rate means more accurately. For example, let’s say that there are two restaurants with average 3 stars: one with only positive polarity reviews and the other one with some positive and some negative reviews. Now user can understand those 3stars are not equivalent.





1. **SELECT** State, City, B.business\_id, B\_name, R.stars

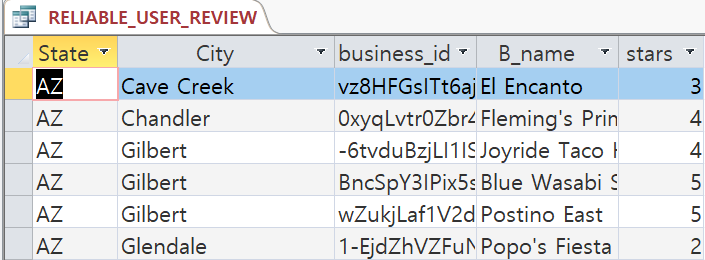
**FROM** [User] AS U, Review AS R, Business AS B

**WHERE** U.user\_id = R.user\_id and R.business\_id = B.business\_id

and U.review\_count >1000 and U.useful > 100

**ORDER BY** State, City;

* Many review count and useful mark can be a good supplementary in order to understand the true meaning of the stars. The table below shows only reviews that have more than 1000 review count and more than 100 useful mark.

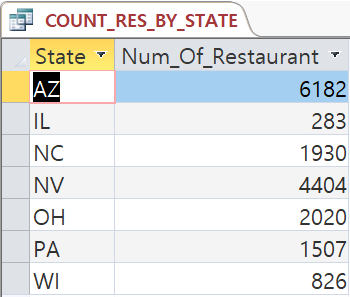


1. **SELECT** State, count(state) AS Num\_Of\_Restaurant

**FROM** Business

**GROUP** BY State;

* It shows how many restaurants in each state are in our database.



1. **SELECT** B\_Name, AVG\_Star

**FROM** BUSINESS AS B,

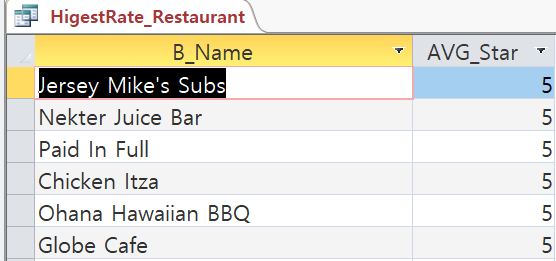
(SELECT Business\_ID, AVG(Stars) AS AVG\_Star

FROM REVIEW GROUP BY Business\_ID) AS R

**WHERE** B.Business\_ID = R.Business\_ID and

AVG\_Star = 5 and State = 'NV' and City = 'Las Vegas';

* This one shows all the 5 star restaurants in a specific city.



1. **SELECT** State, ROUND(AVG(Stars),1) AS AVG\_Star

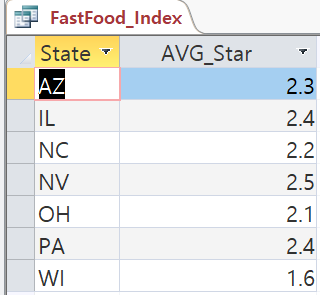
**FROM** BUSINESS AS B, REVIEW AS R

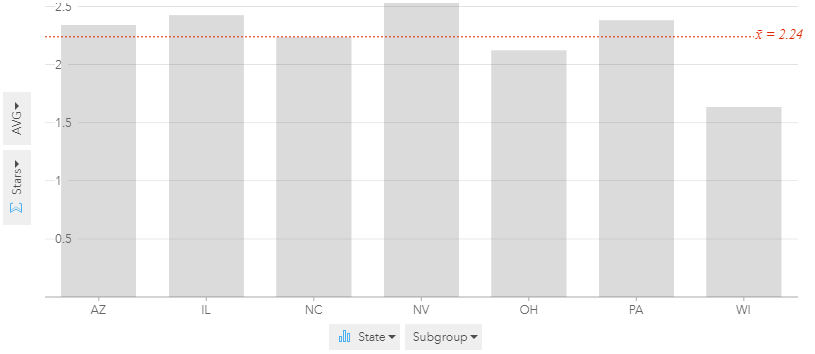
**WHERE** B.Business\_ID = R.Business\_ID and

B\_Name IN ("McDonald's", "Subway", "Taco Bell")

**GROUP BY** State;

* We figured out that "McDonald's", "Subway”, and “Taco Bell" are the three restaurants that are the most widely distributed ones in our dataset. Also, our group assumed that they are the cheapest and the easiest ones, which people have a low preference. Therefore, we thought that the chart we made below can be like Big mac index in economy. By looking at the average stars of three restaurants in each state, people can have an idea how people in one location tend to give stars. For example, we could say that people in WI are the pickiest ones since the average star in WI in the chart below is the lowest.





**Database Administration**

The easiest way to do this would be to have the YELP company have their own data server and save the database, as it might be easier to be able to connect the external data with what they already have. This might also allow them to have more security and protection in their process as they have a little more control over the database.

Additionally, it should probably be password protected. Otherwise, business owner might be tempted to change the rates and comments on its business and users would receive wrong information. Therefore, the log-in page(authentication) should be made and DBA grant the rights, following the user’s roles.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| DATABASE RIGHTS GRANTED | | | | |
| Table | User(Yelper) | Business Owner | Management | System Administrator |
| USER | Read, Insert, Change, Delete | Read | Read, Insert, Change | Grant Rights, Modify Structure |
| USER\_FRIENDS | Read, Insert, Change, Delete | Read | Read, Insert, Change | Grant Rights, Modify Structure |
| REVIEW | Read, Insert, Change, Delete | Read | Read, Insert, Change, Delete | Grant Rights, Modify Structure |
| BUSINESS | Read | Read, Insert, Change, Delete | Read, Insert, Change | Grant Rights, Modify Structure |
| BUSINESS\_CATEGORY | Read | Read, Insert, Change, Delete | Read, Insert, Change, Delete | Grant Rights, Modify Structure |
| CATEGORY | Read | Read, Insert | Read, Insert, Change, Delete | Grant Rights, Modify Structure |
| BUSINESS\_HOUR | Read | Read, Insert, Change, Delete | Read, Insert, Change, Delete | Grant Rights, Modify Structure |
| ZIP\_INCOME | Read | Read | Read, Insert, Change, Delete | Grant Rights, Modify Structure |
| STATE\_GDP | Read | Read | Read, Insert, Change, Delete | Grant Rights, Modify Structure |

Yelp will mainly be collecting data from users and business owners. Business owners will leave the information that is in BUSINESS and BUSINESS\_HOUR table and Users will provide the attributes of USER table. Also, User will write the comments on business and rate them with stars. When user gives the stars in the REVIEW table, the attributes of BUSINESS, Average\_Stars and Review\_Count, will automatically be recalculated by the amount.

Therefore, those two attributes of the business should be locked with shared lock. Other users can view the average stars and review count of the business, but cannot change at that moment by inserting row of REVIEW. Otherwise, they might be able to have concurrent update problem. The lock would be optimistic since there will not be many cases when two or more people provide review to the same business at the same time.

This might be helpful to the yelp company itself as we live in an age where knowledge equals power. This means that the more information yelp is able to give out about the businesses, it may attract more users and additionally, traffic as it would be much more useable to a larger range of people. For example, it adds a certain element of customization for the user as it gives them more choices or factors to add into their search for their perfect establishment. Next, it could be helpful for businesses marketing aspect as it will know more about which audience to focus on, what sells, what markets are available. This goes similarly for establishment owners as it gives them much more room to find more and different avenues to focus on improving their businesses.

Some future suggestions could be to create a more sustainable system that can be automatically updated, and involve much more data. This database uses only a sample set provided by yelp but having this run on a larger scale, not only for yelp, but also different companies that use an interface similar to that of yelp’s can possibly be a good business move. Aside from the different stakeholders mentioned earlier, it might also be a great venture for yelp to look into the research business as it generates such a great set of data, especially considering the number of users and reviews in gets per day.