**BUS243 – RESTAURANT RATING AND SEARCH MANAGEMENT SYSTEM**

Aarathi Srinivas Nadathur (013704405)

Chaithanya Reddy Bogadi (013803998)

Nandini Puppala (013752713)

San Jose State University

**CONTENTS**

|  |  |
| --- | --- |
| Abstract | 3 |
| Business Benefits and Goals: | 3 |
| Key Features | 4 |
| Requirements     Business Requirements     Functional Requirement     Technical Requirements     Infrastructure Requirements | 5  5  5    6 |
| User Interface | 7 |
| Database design | 9 |
| Test and Launch | 12 |
| Queries | 13 |
| Visualization | 17 |
| Conclusion | 21 |

**Abstract**

Hunger and craving for good food is a global phenomenon. Being foodies, we personally relied on a lot of Restaurant rating and search websites, especially after coming to the US where we had no idea of any place or restaurant. We wanted to try on different cuisines and these restaurant search and rating websites could give us a lot of information. We wanted to know how this was implemented and wanted to explore this and create a Database for the Restaurant rating and search management system.

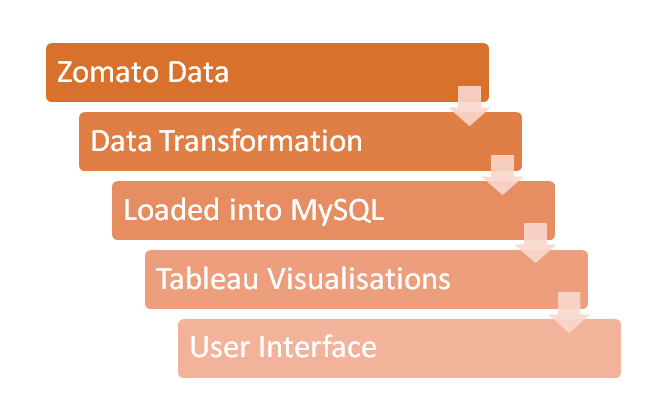
In this prototype, we created a restaurant rating and search platform that helps the user to choose restaurants according to their budget and requirements. Once the user gives the requirement, all the restaurants satisfying the requirement are fetched from the database and displayed, making it more convenient for the user to choose. This Database is created in such a way that user gets all the information about the restaurant. The Database consists of geographic location, cuisines offered and the ratings of restaurants.

**Business Benefits and Goals:**

Organize the restaurant ensuring the information is easily available to the customers. It helps the restaurants build a social media presence. It also helps to analyze the performance of restaurants at different locations. Stores the price range for each restaurant and accordingly customers can visit a place depending on our budget.

**Goals**

* Helping people to find great restaurants around them.
* Know the best rated restaurant in the location.
* Know the price range of each restaurant.
* Building amazing experiences around dining.



Brief view on the whole process

**Key Features**

Being a restaurant rating and search platform, we aim at providing all the details, a customer would want to know about a particular restaurant or restaurants in an area based on public rating. We tried in creating a database that would give information about all the features of a restaurant like food, hospitality, price range, cuisine, etc. In a developed version of the User Interface, we would like to put in lot of filters and show the following features:

Filters based on rating

* Aggregate ratings of a given restaurant
* Food rating for a given restaurant
* Text ratings given by customers

Price range of the restaurant like:

* Price range rating
* Average cost per two persons at a given restaurant

With the increase in customers preferring vegan, we have included an option to filter for vegan or non-vegan restaurants as well

* Number of Vegan restaurants in the given area

Filter for Food items in the restaurant

* Top food item offered by a given restaurant
* Cuisine offered

Filter for Restaurant timings

* Restaurant opening and closing time.
* Days on which restaurant is closed

Filters with Restaurant Address:

* Search for restaurants with locality
* Search for restaurants with pincode.
* Get the restaurants exact address.

Delivery based filters.

* If the restaurant offers home delivery.

**Requirements**

**Business Requirements:**

There were many business rules to follow for a Restaurant rating and search platform with the advancement of technology. Restaurant rating and search site must give a lot of details about the restaurants in an area. Looking at the similar systems available like Zomato, Yelp, Open Table, Zagat, Trip Advisor, Gayot, Dine etc. we tried to incorporate the features and align it with the data we had. The functionalities we could implement with the available technologies were considered. We created ERD based on the business rules mentioned below.

* From our observation, a restaurant rating and search management system, has all the details of the restaurants. The restaurant data set we obtained from Kaggle had details of each restaurant in Florida and Georgia and the restaurants are identified with a unique restaurant\_ID.
* Each restaurant is mapped to a price range rating, hospitality rating and aggregate rating given separately.
* Each restaurant is mapped to an average cost for two persons and the text rating.
* Each restaurant is mapped to a single address, which has all the details of the restaurants location. These are also updated only when a change is notified.
* Each restaurant has its own opening time, closing time and the days on which it is closed and also any extra information provided in the restaurant websites.
* We also have information on the restaurant’s cuisine, and each restaurant is mapped to a single cuisine, if a restaurant offers multiple cuisines, then we put it under international.
* Each restaurant has a top food item offered and its price. On the whole, our database has details of the restaurants in Florida and Georgia, but this prototype can be scaled up to all the restaurants in USA.

**Functional Requirement:**

The database has details regarding the restaurant address, timings, the cuisines available, price for two persons, user rating for food and the restaurant as a whole. The web User Interface is the front end where:

* User can type in the locality or pincode and it will display the restaurants in that area.
* User can fetch restaurants based on vegan or non-vegan
* Users can get address of a particular restaurant
* Users can find the restaurants in a locality that serve a given cuisine.
* Users can find best rated restaurants.
* User can search for their budget friendly restaurants.

With the user-friendly User Interface, the customers will get the data they need which is Error-free and validated.

**Technical Requirements:**

The database was developed in MySQL Workbench and we used other tools as mentioned below.

**MySQL**:  MySQL is the most popular Open Source Relational Database Management System. It can work on any operating systems varying from Windows, Linux etc.

**Python**: We used python for visualizing the data. Matplotlib and Pandas packages were used for creating charts and Data wrangling.

**Microsoft Excel**: It is a Spreadsheet developed by Microsoft. In our project, we used excel to assess the data for our initial analysis purpose.

**Tableau:** Tableau is the most powerful, secure, and flexible end-to-end analytics platform. It can be connected to various data sources, be it Databases, MS Excel, data warehouses or web data.

**XAMPP Server: (**X- platform, Apache, MySQL, PHP, Perl)

It is a free open source cross platform web application. It is used for web development on the local machine.

**Infrastructure Requirements:**

***Hardware Specifications***

i7 Core processor

16GB RAM

1GB Hard Disk Space.

***Softwares Specifications:***

Python 3.5

MySQL Workbench 8.0 CE

Microsoft Excel 2013 Version

Tableau Public

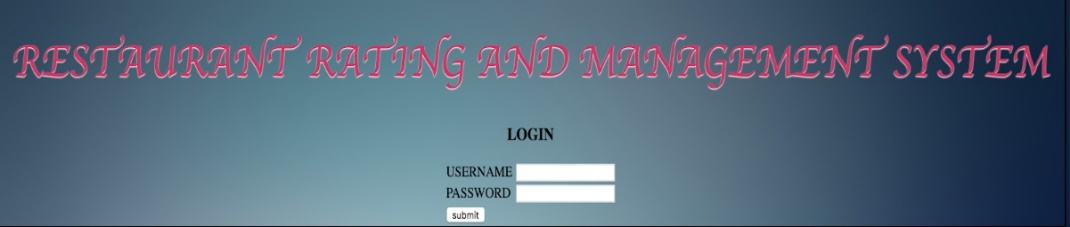
Xampp Server.

**User Interface:**

The user interface has been created using PHP and MYSQL. The Xampp server is used to connect the front end to the database. The users must first register themselves and after they complete their registration, they should login using their login credentials. The users can find the restaurants based on the locality or using vegan option availability and closing times. The data is fetched from the database and the result is shown in the user interface.

USER REGISTRATION and LOGIN:

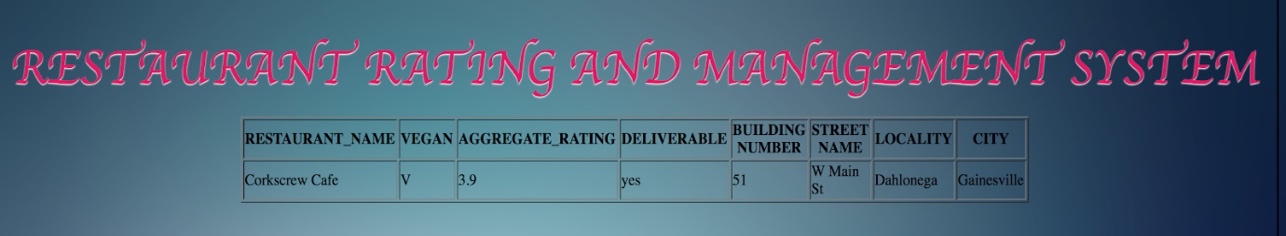
The user first registers in the website. Next the user logs in with the registered credentials.



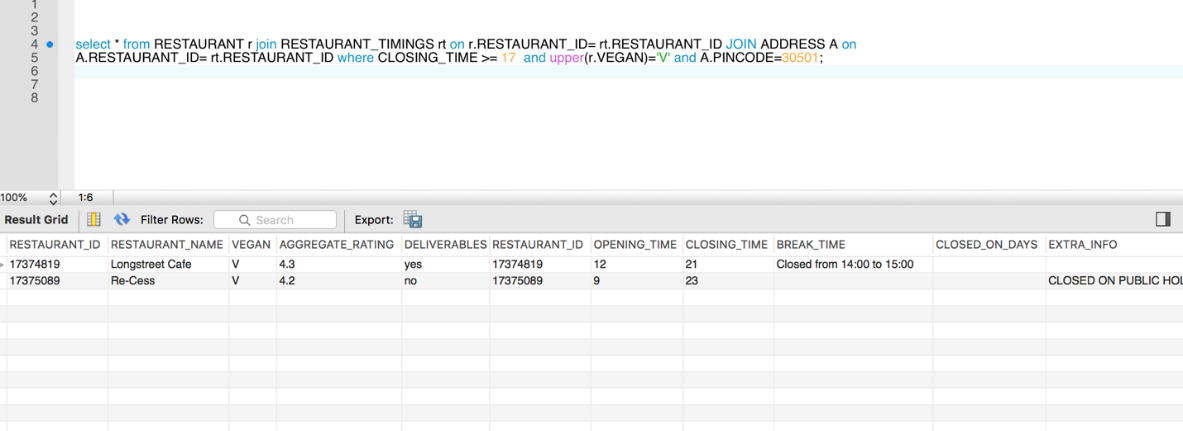
We used two filters in this prototype to fetch the records.

1. The user fills in the details like the Pincode, vegan options, closing time,etc. According to the details filled the records are retrieved from the database and displayed on the screen.





MySQL Query for this page:



1. The user can find the restaurants based on the locality as well.



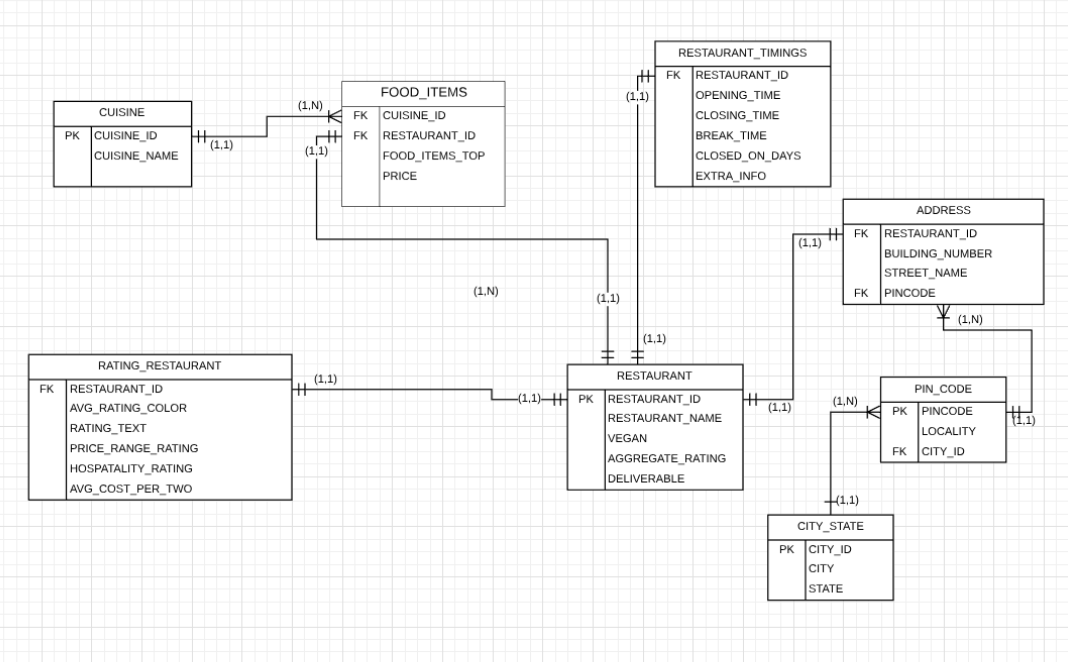
~~~~

MySQL query 2 for this page:



**Database design:**

We got the Zomato dataset from Kaggle. Data gathering was relatively easy as we got the file, but we spent a lot of time on data wrangling as it had many missing records, null values, characters that does not make any sense and many duplicate records. After cleansing we were left with a table with 240 rows and 45 columns in a de-normalized form.The database we designed has 8 normalized tables in total.



Entity Relationship Diagram of Restaurant Rating and Search Management system

* **Restaurant table**: This is the primary table and has a unique key for each restaurant and has some details of the restaurant including the Aggregate rating given to a restaurant.
* **Address**: This table has the location details of the restaurant, where it is located, and other details are linked with the pincode table.
* **Pin\_code**: This has the locality and city\_id, the pincode is associated with City\_state.
* **City\_state:**This has the city\_id associated to a city and its state details
* **Rating\_restaurant**: Restaurant\_rating has all the details of various ratings given by the customers.
* **Cuisine**: This table has all the details of different cuisines being offered by all the restaurants.
* **Food\_items**: This has details of the cuisine offered by a particular restaurant, its top food item and the price.
* **Restaurant\_timings**: This has all the details on the opening and closing time, break time of the restaurant, when the restaurant is closed and any extra info.

|  |  |  |  |
| --- | --- | --- | --- |
| **Entity** | **Relationship** | **Cardinality** | **Entity** |
| CUISINE  PK CUISINE\_ID | Has | 1:N | FOOD\_ITEMS  FK CUISINE\_ID |
| FOOD\_ITEMS  FK CUISINE\_ID  FK RESTAURANT\_ID | Belongs to    Belongs | 1:1    1:1 | CUISINE  PK CUISINE\_ID    RESTAURANT  PK RESTAURANT\_ID |
| RESTAURANT  PK RESTAURANT\_ID | Has | 1:1    1:1      1:1      1:1 | RATING\_RESTAURANT  FK RESTAURANT\_ID    RESTAURANT\_TIMINGS  FK RESTAURANT\_ID    FOOD\_ITEMS  FK RESTAURANT\_ID    ADDRESS  FK RESTAURANT |
| RATING\_RESTAURANT  FK RESTAURANT\_ID | Has | 1:1 | RESTAURANT  PK RESTAURANT\_ID |
| RESTAURANT\_TIMINGS  FK RESTAURANT\_ID | Has | 1:1 | RESTAURANT  PK RESTAURANT\_ID |
| ADDRESS  FK RESTAURANT    FK PINCODE | Belongs to    Has | 1:1    1:1 | RESTAURANT  PK RESTAURANT\_ID    PIN\_CODE  PK PINCODE |
| PIN\_CODE  PK PINCODE    FK CITY\_ID | Belongs to  Belongs to | 1:N  1:1 | ADDRESS  FK PINCODE    CITY\_STATE  PK CITY\_ID |
| CITY\_STATE  PK CITY\_ID | Has | 1:N | PINCODE  FK CITY\_CODE |

**Test and Launch:**

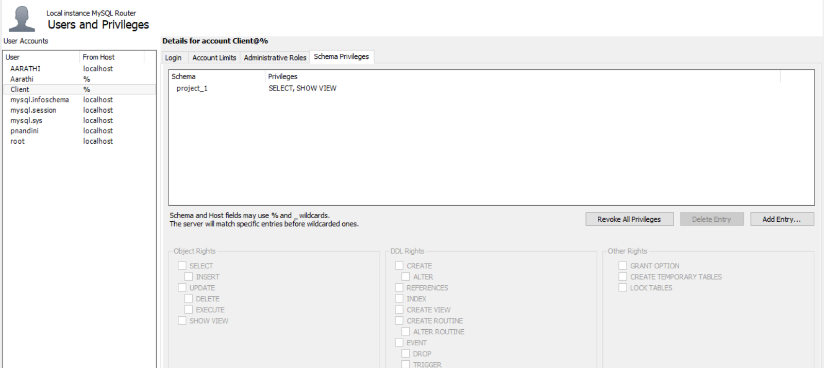
As mentioned earlier we got the data in .csv format and after lot of cleansing we normalized the data and created the DB. We normalized it as required for the project as shown in the ERD. We created eight .csv files for all the eight tables. We created the Database in MySQL Workbench by creating a create table script for all the tables with the constraints placed. Using the import table wizard, we could populate the tables with the eight data files we had.

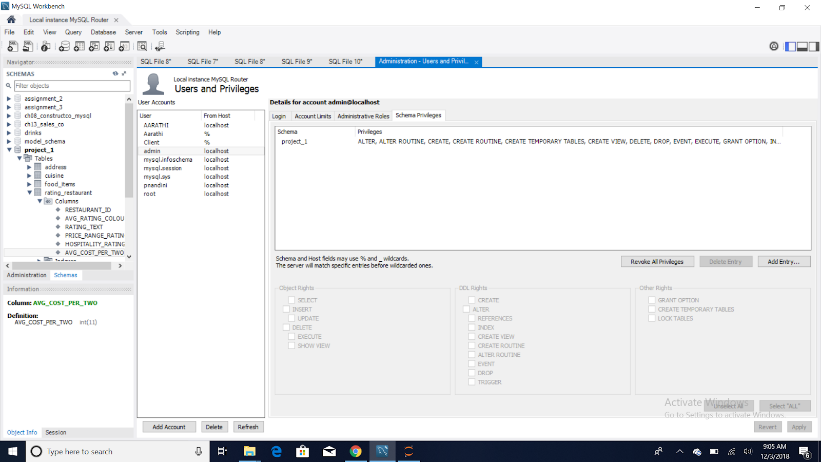
We created three users in the MySQL Database and changed the access for each user.

Aarathi - Database developer who has access to creating and updating the table structures and access the data and create triggers.

Client - User has access to only select queries and views.

Admin - user with full access to the database, including updating, deleting or creation.





Now we are ready to test and launch the product.

**Queries:**

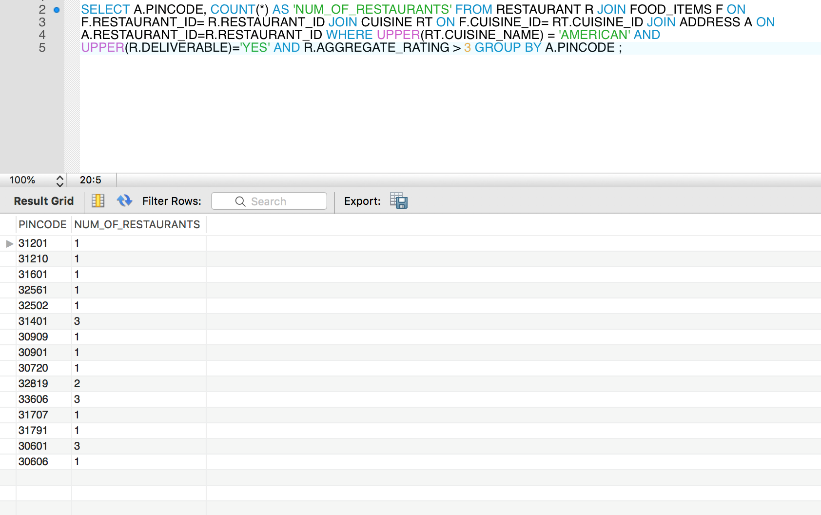
Here are some queries which can be used in the advanced version of the User Interface. Another feature which we thought would be helpful with this Restaurant rating and search platform is for business people who would like to set up new businesses, can analyze the data to find useful insights on current market trends. We included those queries as well.

**Query 1**: To get the count of restaurants that have an aggregate rating of greater than 3 for each place based on pin codes.

‘SELECT A.PINCODE, COUNT(\*) AS 'NUM\_OF\_RESTAURANTS' FROM RESTAURANT R JOIN FOOD\_ITEMS F ON

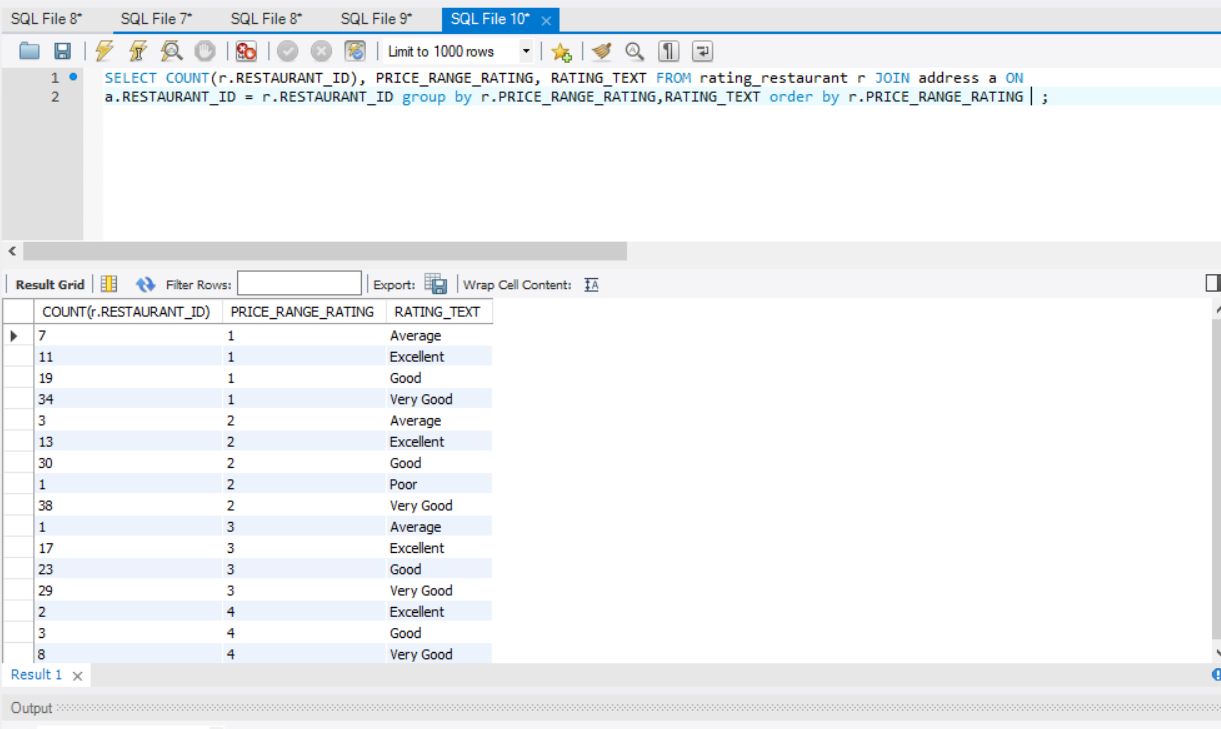
F.RESTAURANT\_ID= R.RESTAURANT\_ID JOIN CUISINE RT ON F.CUISINE\_ID= RT.CUISINE\_ID JOIN ADDRESS A ON A.RESTAURANT\_ID=R.RESTAURANT\_ID WHERE UPPER(RT.CUISINE\_NAME) = 'AMERICAN' AND UPPER(R.DELIVERABLE)='YES' AND R.AGGREGATE\_RATING > 3 GROUP BY A.PINCODE ;’

Output:



**Query 2**: To get the count of restaurants grouped by price\_range\_rating and rating text.

‘SELECT COUNT(R.RESTAURANT\_ID), PRICE\_RANGE\_RATING, RATING\_TEXT FROM RATING\_RESTAURANT R JOIN ADDRESS A ON A.RESTAURANT\_ID = R.RESTAURANT\_ID GROUP BY R.PRICE\_RANGE\_RATING,RATING\_TEXT ORDER BY R.PRICE\_RANGE\_RATING  ;’



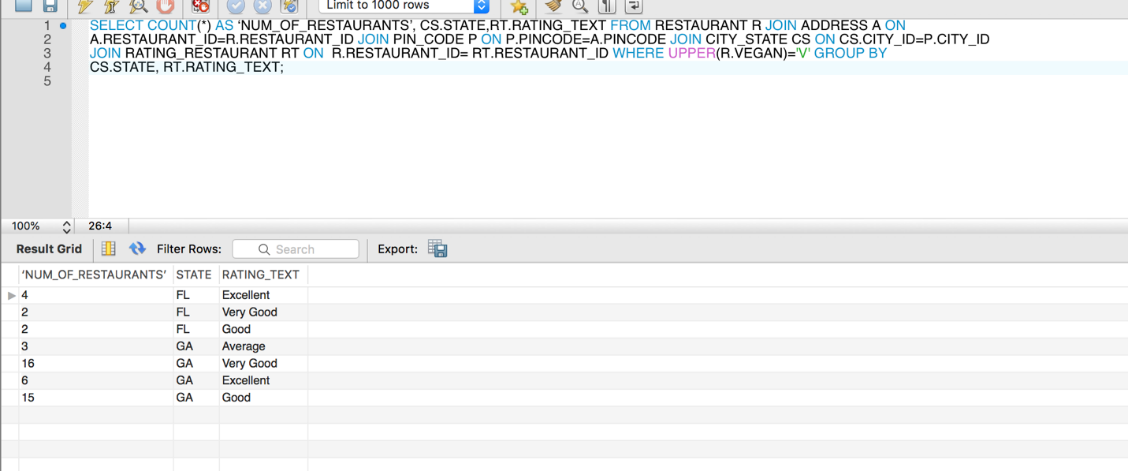
**Query 3**: To get the number of Vegan restaurants having good ratings, grouped by rating text in each state.

‘SELECT COUNT(\*) AS ‘NUM\_OF\_RESTAURANTS’, CS.STATE,RT.RATING\_TEXT FROM RESTAURANT R JOIN ADDRESS A ON

A.RESTAURANT\_ID=R.RESTAURANT\_ID JOIN PIN\_CODE P ON P.PINCODE=A.PINCODE JOIN CITY\_STATE CS ON CS.CITY\_ID=P.CITY\_ID

JOIN RATING\_RESTAURANT RT ON R.RESTAURANT\_ID= RT.RESTAURANT\_ID WHERE UPPER(R.VEGAN)='V' GROUP BY

CS.STATE, RT.RATING\_TEXT;’



**Query 4**: Query to find the count of restaurants grouped by their rating colors in each state.

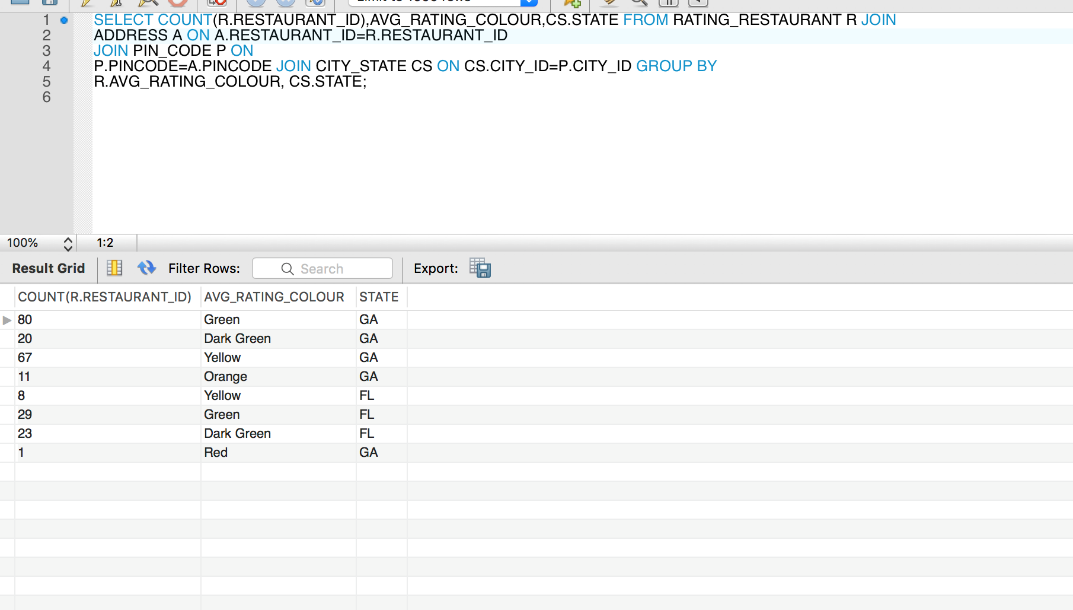
‘SELECT COUNT(R. RESTAURANT\_ID),AVG\_RATING\_COLOUR,CS.STATE FROM RATING\_RESTAURANT R JOIN

ADDRESS A ON A.RESTAURANT\_ID=R.RESTAURANT\_ID

JOIN PIN\_CODE P ON

P.PINCODE=A.PINCODE JOIN CITY\_STATE CS ON CS.CITY\_ID=P.CITY\_ID GROUP BY

R.AVG\_RATING\_COLOUR, CS.STATE;’



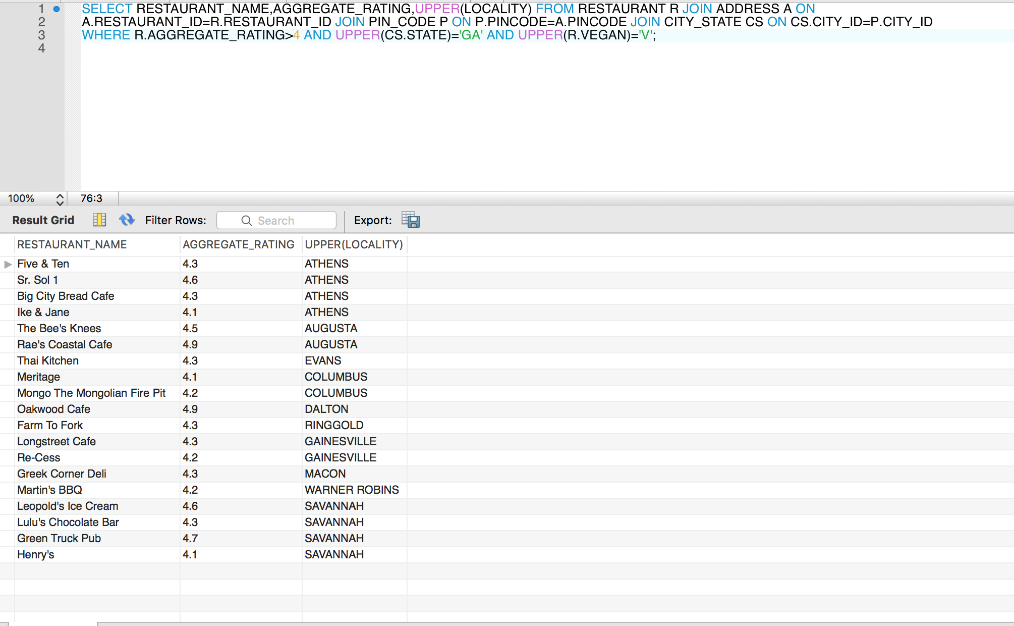
**Query 5**: To find a vegan restaurant with rating 5 in Georgia.

‘SELECT RESTAURANT\_NAME, AGGREGATE\_RATING,UPPER(LOCALITY) FROM RESTAURANT R JOIN ADDRESS A ON

A.RESTAURANT\_ID=R.RESTAURANT\_ID JOIN PIN\_CODE P ON P.PINCODE=A.PINCODE JOIN CITY\_STATE CS ON CS.CITY\_ID=P.CITY\_ID

WHERE R.AGGREGATE\_RATING>4 AND UPPER(CS.STATE)='GA' AND UPPER(R.VEGAN)='V';’

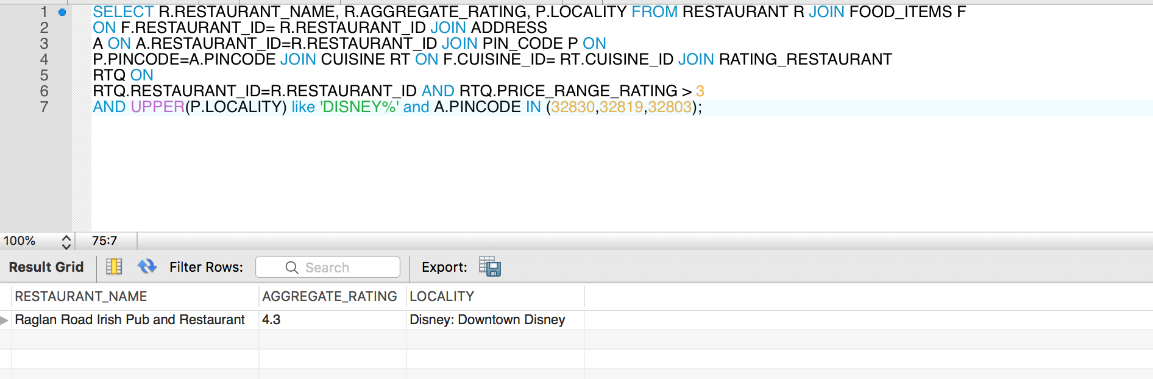
Output:



**Query 6:** To find a restaurant with aggregate rating greater than 3 in the Disney locality.

‘SELECT R.RESTAURANT\_NAME, R.AGGREGATE\_RATING, P.LOCALITY FROM RESTAURANT R JOIN FOOD\_ITEMS FON F.RESTAURANT\_ID= R.RESTAURANT\_ID JOIN ADDRESS A ON A.RESTAURANT\_ID=R.RESTAURANT\_ID JOIN PIN\_CODE P ON P.PINCODE=A.PINCODE JOIN CUISINE RT ON F.CUISINE\_ID= RT.CUISINE\_ID JOIN RATING\_RESTAURANT RTQ ON RTQ.RESTAURANT\_ID=R.RESTAURANT\_ID AND RTQ.PRICE\_RANGE\_RATING > 3 AND UPPER(P.LOCALITY) like 'DISNEY%' and A.PINCODE IN (32830,32819,32803);’

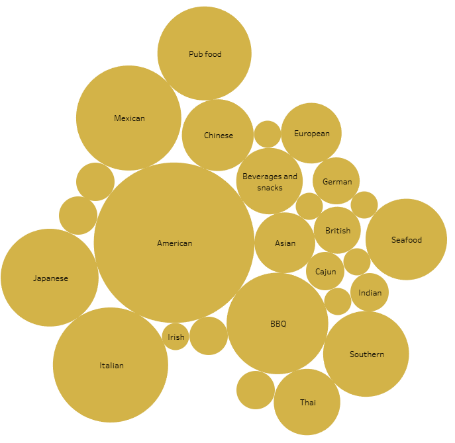
Output:



**Visualization:**

With the analytical information we could gather from the Data, we tried visualizing them using Tableau and python (Matplotlib and Pandas Packages).

* Below is the bubble chart showing the distribution of cuisines of Restaurants in Georgia and Florida, made using Tableau .

****

**Georgia Florida**

**MySQL Queries for the above graph:**

‘SELECT COUNT(\*) AS A, CS.CUISINE\_NAME FROM RESTAURANT R JOIN FOOD\_ITEMS C ON C.RESTAURANT\_ID = R.RESTAURANT\_ID

JOIN CUISINE CS ON CS.CUISINE\_ID = C.CUISINE\_ID JOIN ADDRESS AD ON AD.RESTAURANT\_ID=R.RESTAURANT\_ID JOIN PIN\_CODE P

ON P.PINCODE=AD.PINCODE JOIN CITY\_STATE CSA ON CSA.CITY\_ID=P.CITY\_ID WHERE

CSA.STATE='FL' GROUP BY C.CUISINE\_ID ORDER BY A DESC ;’

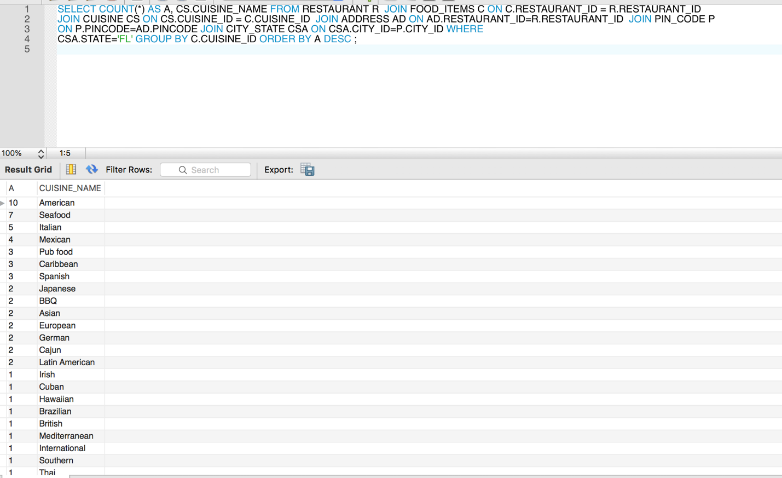
‘SELECT COUNT(\*) AS A, CS.CUISINE\_NAME FROM RESTAURANT R JOIN FOOD\_ITEMS C ON C.RESTAURANT\_ID = R.RESTAURANT\_ID

JOIN CUISINE CS ON CS.CUISINE\_ID = C.CUISINE\_ID JOIN ADDRESS AD ON AD.RESTAURANT\_ID=R.RESTAURANT\_ID JOIN PIN\_CODE P

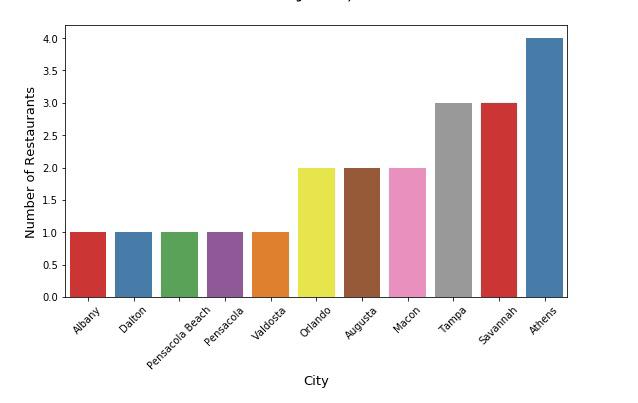
ON P.PINCODE=AD.PINCODE JOIN CITY\_STATE CSA ON CSA.CITY\_ID=P.CITY\_ID WHERE

CSA.STATE='GA' GROUP BY C.CUISINE\_ID ORDER BY A DESC ;’

OUTPUT:

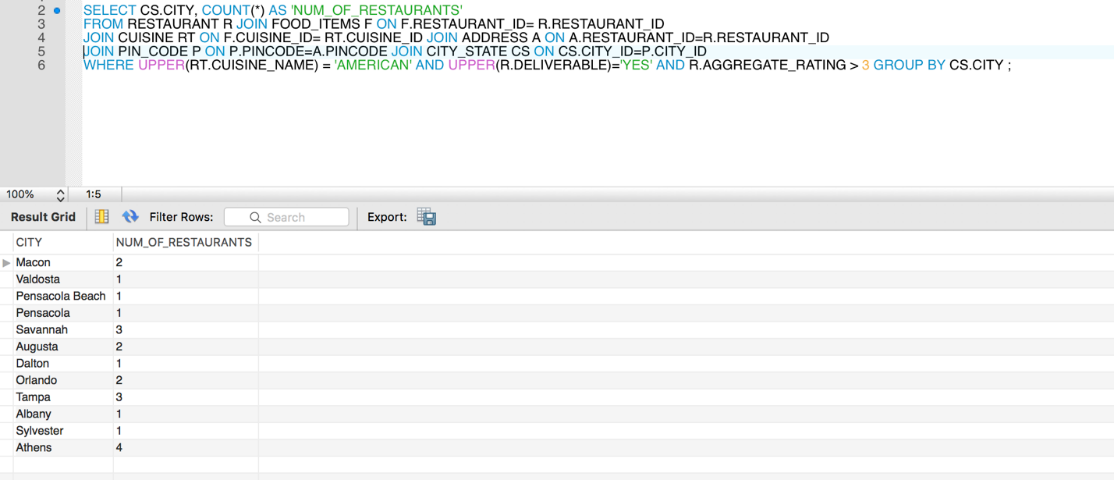


* Below is the Bar chart projecting the restaurants in different cities that deliver American food with an aggregate rating greater than three. This is done in Jupyter Notebook using Python.



MySQL Query for the graph:  
‘SELECT CS.CITY, COUNT(\*) AS 'NUM\_OF\_RESTAURANTS' FROM RESTAURANT R JOIN FOOD\_ITEMS F ON F.RESTAURANT\_ID= R.RESTAURANT\_ID JOIN CUISINE RT ON F.CUISINE\_ID= RT.CUISINE\_ID JOIN ADDRESS A ON A.RESTAURANT\_ID=R.RESTAURANT\_ID JOIN PIN\_CODE P ON P.PINCODE=A.PINCODE JOIN CITY\_STATE CS ON CS.CITY\_ID=P.CITY\_ID WHERE UPPER(RT.CUISINE\_NAME) = 'AMERICAN' AND UPPER(R.DELIVERABLE)='YES' AND R.AGGREGATE\_RATING > 3 GROUP BY CS.CITY ;’

Output:



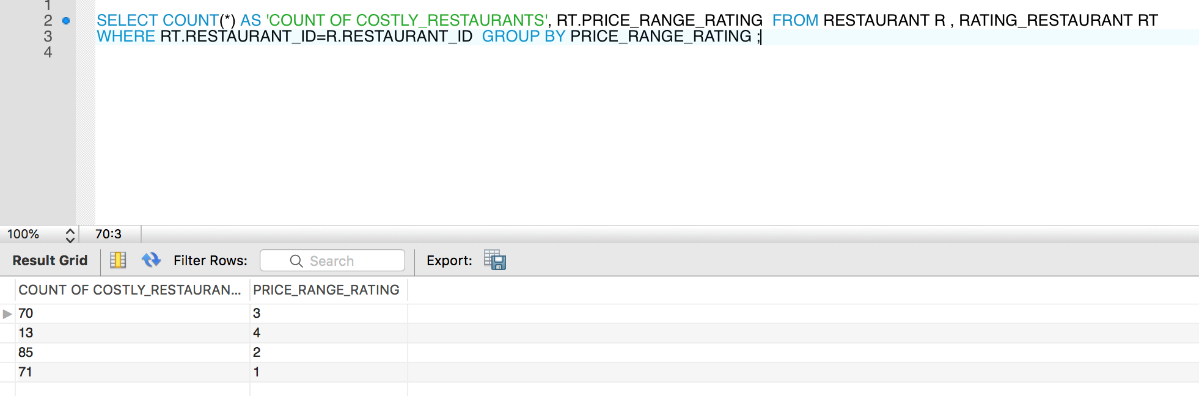
* Below is the Bar plot made using Python Matplotlib to find the number of restaurants in both the states categorized by the price rating.



MySQL Query for the above graph:

‘SELECT COUNT(\*) AS 'COUNT OF COSTLY\_RESTAURANTS', RT.PRICE\_RANGE\_RATING  FROM RESTAURANT R , RATING\_RESTAURANT RT  
WHERE RT.RESTAURANT\_ID=R.RESTAURANT\_ID  GROUP BY PRICE\_RANGE\_RATING ;

Output:



**Conclusion.**

The main goal of our project was to enable the users to find information and reviews on the restaurants according to their requirement. We got the Zomato dataset from Kaggle and we cleansed and normalized the data. The database was created with 8 tables following the Business rules established. The project, as of now has been implemented only for two states, Florida and Georgia which can later be expanded to other states as well. The analysis part was done using tableau and python to make the analysts infer results easily and it will also help others who are interested to setup a business to analyze existing market conditions. From analyzing the data we got the following results:

* In Florida the top 3 popular cuisines are American, Seafood and Italian.
* There are more above average restaurants in Athens compared to other places.
* There are more vegan restaurants in Georgia as compared to Florida, but the rating of vegan restaurants in Florida is higher than Georgia.

With the developed version of the User Interface we want to implement all the queries mentioned above, where the customers could fetch the information needed about the restaurants based on rating filters, Price range filters, Location, cuisine, delivery options etc. Our Database can give all the above-mentioned information and help customers find the address or location of the restaurants they are looking for. This project gave us the opportunity to try our new skills in practice. While doing this project we also gained deeper understanding on database design and how it can be implemented in real time projects.