# Zomato Restaurants Database Analysis

EMSE 6586
Spring 2021

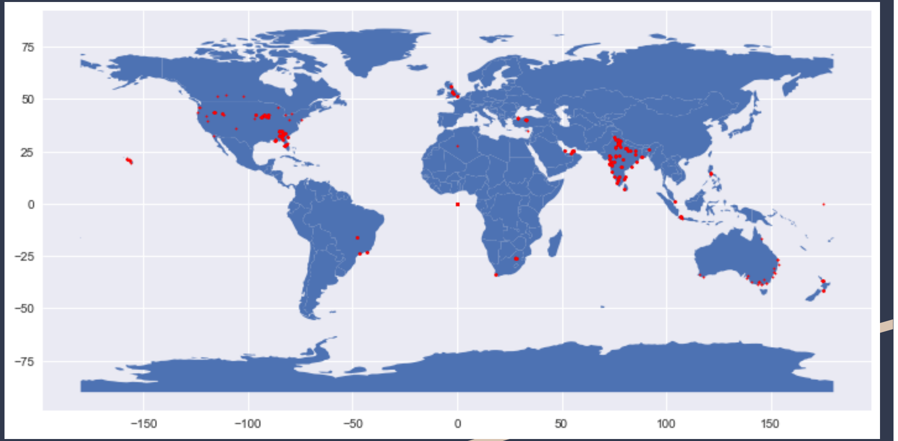
Jose Garcia
Kahang Ngau

# Objectives

OBJECTIVES

- Restructure a flattened dataset and load into a SQL database
- Demonstrate the convenience of the database
- Provide an efficient and easier way for end-users to search specific information

# Exploratory Data Analysis



Exploratory_Data_Analysis.ipynb

## SOURCE

- Zomato dataset from Kaggle
  - Dataset collected from Zomato API and stored in comma separated value file.

## RESTAURANTS

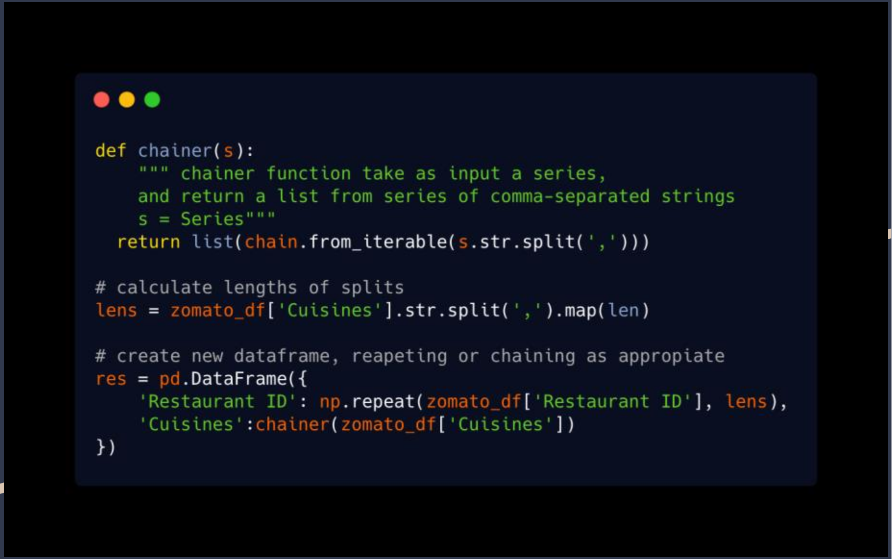
- 9542 restaurants
- Data includes: location, rating, cuisines, evaluation, price.
- Cuisines: [‘Fast food’, ‘Italian’, ‘Asian’, ‘North Indian’, ...]
- Locality: [‘Soho’, ‘Macon’, ‘Savannah’, ...]
- Yes/No: has table booking? Is delivering now?

## RESTRUCTURE DATA

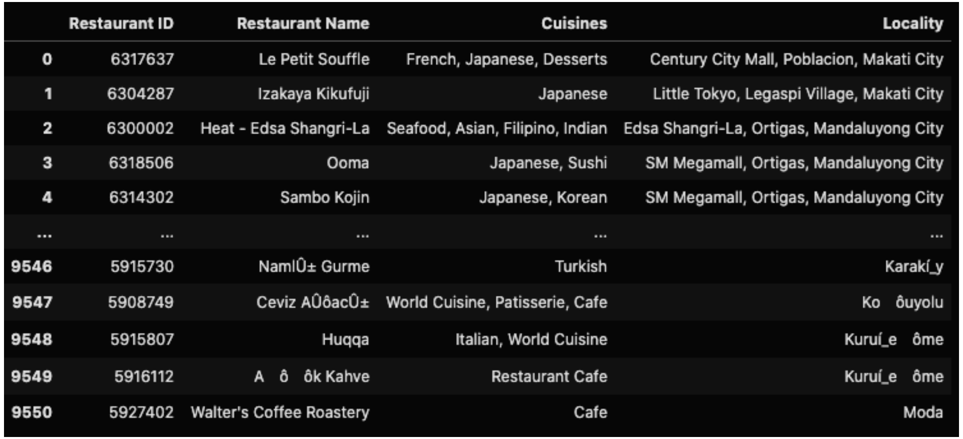
- Restaurant info
- Ratings
- Evaluation
- Cuisines
- Localities

→ Restaurant ID

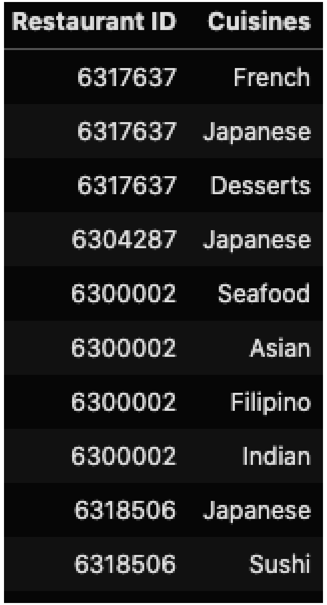
# Cuisines and Locality disaggregation



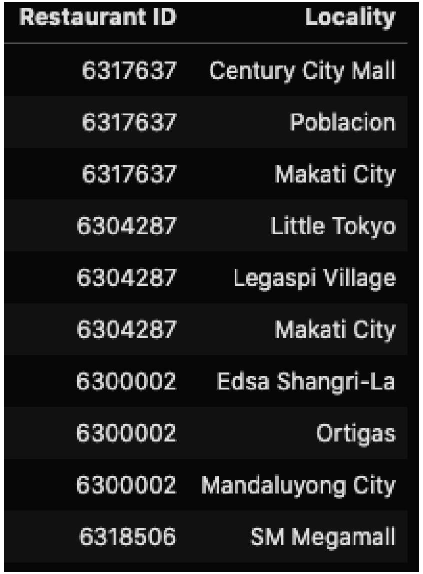
```
def chainer(s):
    """ chainer function take as input a series,
    and return a list from series of comma-separated strings
    s = Series"""
  return list(chain.from_iterable(s.str.split(',')))

# calculate lengths of splits
lens = zomato_df['Cuisines'].str.split(',').map(len)

# create new dataframe, reapeting or chaining as appropiate
res = pd.DataFrame({
    'Restaurant ID': np.repeat(zomato_df['Restaurant ID'], lens),
    'Cuisines':chainer(zomato_df['Cuisines'])
})
```



| | Restaurant ID | Restaurant Name | Cuisines | Locality |
|---|---|---|---|---|
| 0 | 6317637 | Le Petit Souffle | French, Japanese, Desserts | Century City Mall, Poblacion, Makati City |
| 1 | 6304287 | Izakaya Kikufuji | Japanese | Little Tokyo, Legaspi Village, Makati City |
| 2 | 6300002 | Heat - Edsa Shangri-La | Seafood, Asian, Filipino, Indian | Edsa Shangri-La, Ortigas, Mandaluyong City |
| 3 | 6318506 | Ooma | Japanese, Sushi | SM Megamall, Ortigas, Mandaluyong City |
| 4 | 6314302 | Sambo Kojin | Japanese, Korean | SM Megamall, Ortigas, Mandaluyong City |
| ... | ... | ... | ... | ... |
| 9546 | 5915730 | NamlÛ± Gurme | Turkish | Karakí_y |
| 9547 | 5908749 | Ceviz AÛôacÛ± | World Cuisine, Patisserie, Cafe | Ko ôuyolu |
| 9548 | 5915807 | Huqqa | Italian, World Cuisine | Kuruí_e ôme |
| 9549 | 5916112 | A ô ôk Kahve | Restaurant Cafe | Kuruí_e ôme |
| 9550 | 5927402 | Walter's Coffee Roastery | Cafe | Moda |

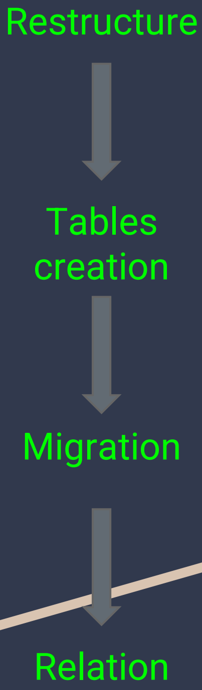


| Restaurant ID | Cuisines |
|---|---|
| 6317637 | French |
| 6317637 | Japanese |
| 6317637 | Desserts |
| 6304287 | Japanese |
| 6300002 | Seafood |
| 6300002 | Asian |
| 6300002 | Filipino |
| 6300002 | Indian |
| 6318506 | Japanese |
| 6318506 | Sushi |



| Restaurant ID | Locality |
|---|---|
| 6317637 | Century City Mall |
| 6317637 | Poblacion |
| 6317637 | Makati City |
| 6304287 | Little Tokyo |
| 6304287 | Legaspi Village |
| 6304287 | Makati City |
| 6300002 | Edsa Shangri-La |
| 6300002 | Ortigas |
| 6300002 | Mandaluyong City |
| 6318506 | SM Megamall |

# Approach

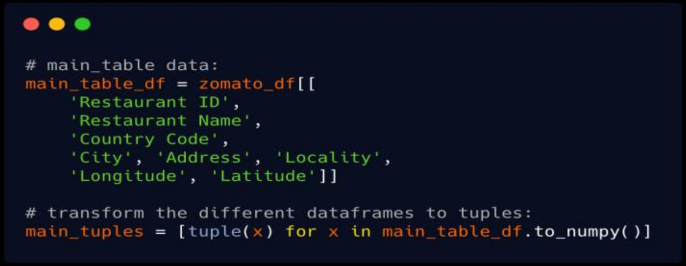


Restructure

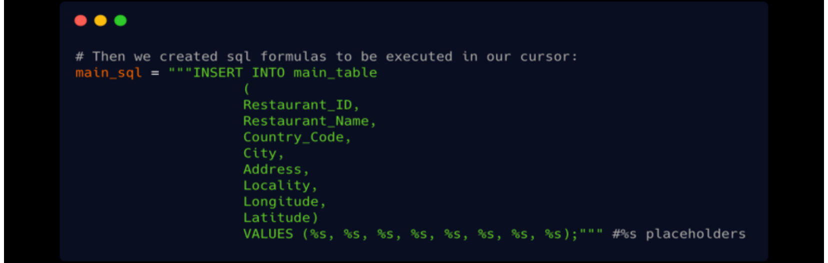
Tables creation

Migration

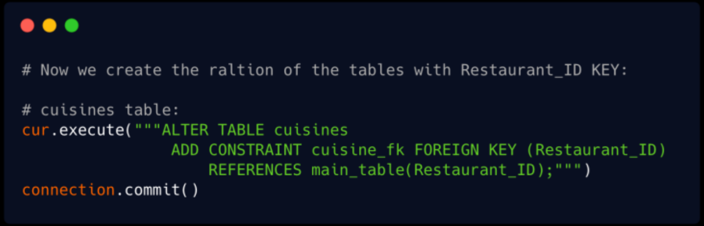
Relation



```
# main_table data:
main_table_df = zomato_df[[
     'Restaurant ID',
     'Restaurant Name',
     'Country Code',
     'City', 'Address', 'Locality',
     'Longitude', 'Latitude']]

# transform the different dataframes to tuples:
main_tuples = [tuple(x) for x in main_table_df.to_numpy()]
```



```
# Then we created sql formulas to be executed in our cursor:
main_sql = """INSERT INTO main_table
                    (
                    Restaurant_ID,
                    Restaurant_Name,
                    Country_Code,
                    City,
                    Address,
                    Locality,
                    Longitude,
                    Latitude)
                    VALUES (%s, %s, %s, %s, %s, %s, %s, %s);""" #%s placeholders
```



```
# Now we create the raltion of the tables with Restaurant_ID KEY:

# cuisines table:
cur.execute("""ALTER TABLE cuisines
                ADD CONSTRAINT cuisine_fk FOREIGN KEY (Restaurant_ID)
                    REFERENCES main_table(Restaurant_ID);""")
connection.commit()
```

[MySQLconnection.ipynb](MySQLconnection.ipynb)

# Database schema:

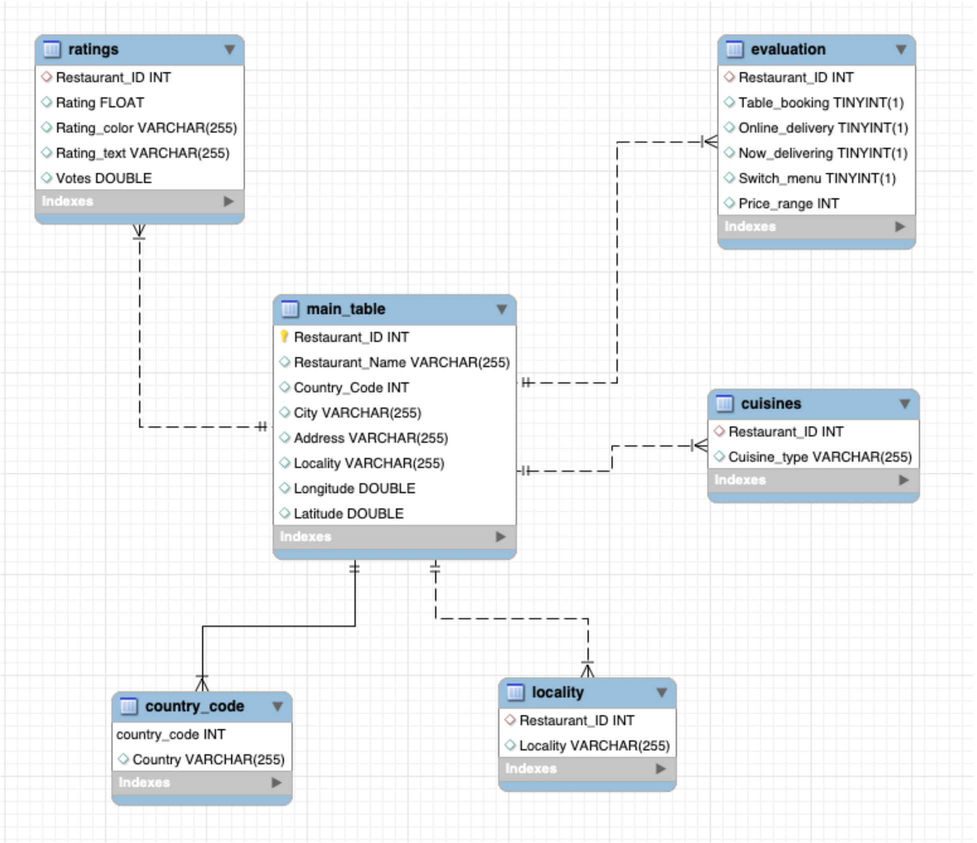
Primary keys: Restaurant_ID

‘main_table’

Foreign keys: Restaurant_ID

‘Ratings’, ‘evaluation’, ‘cuisines’, ‘locality’

Keys: Country_code



**ratings**
- Restaurant_ID INT
- Rating FLOAT
- Rating_color VARCHAR(255)
- Rating_text VARCHAR(255)
- Votes DOUBLE

Indexes

**evaluation**
- Restaurant_ID INT
- Table_booking TINYINT(1)
- Online_delivery TINYINT(1)
- Now_delivering TINYINT(1)
- Switch_menu TINYINT(1)
- Price_range INT

Indexes

**main_table**
- Restaurant_ID INT
- Restaurant_Name VARCHAR(255)
- Country_Code INT
- City VARCHAR(255)
- Address VARCHAR(255)
- Locality VARCHAR(255)
- Longitude DOUBLE
- Latitude DOUBLE

Indexes

**cuisines**
- Restaurant_ID INT
- Cuisine_type VARCHAR(255)

Indexes

**country_code**
- country_code INT
- Country VARCHAR(255)

Indexes

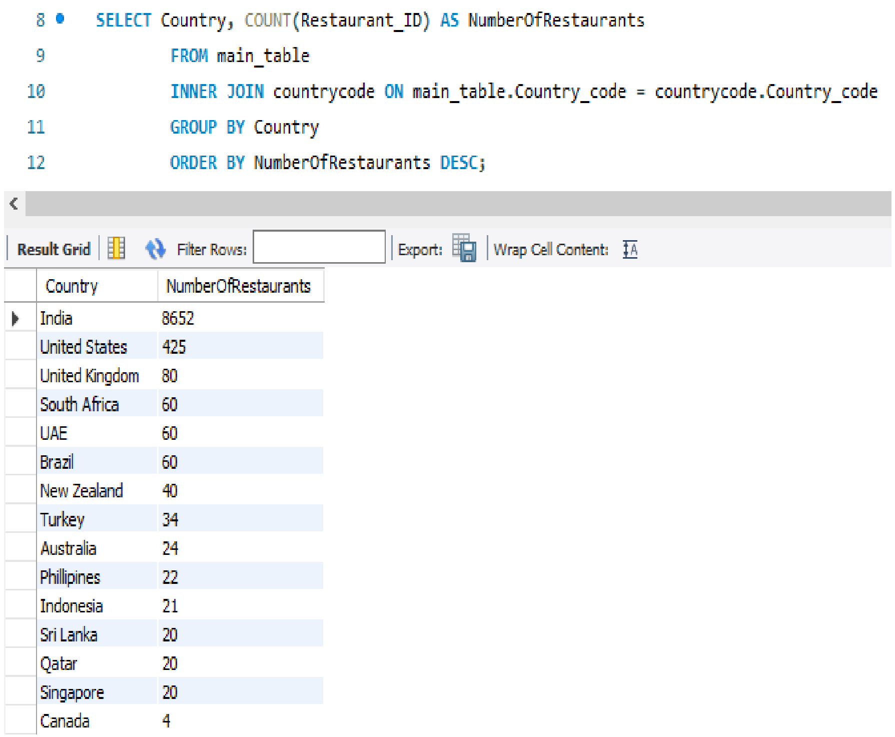
**locality**
- Restaurant_ID INT
- Locality VARCHAR(255)

Indexes

# Benefits & Advantages

- Minimize the size of the main table.
- Relatively easy to update / modify the database.
- Convenient to search for information based on the characteristics of different tables.
- Additional tables can be added to the database to have a more complex structure.

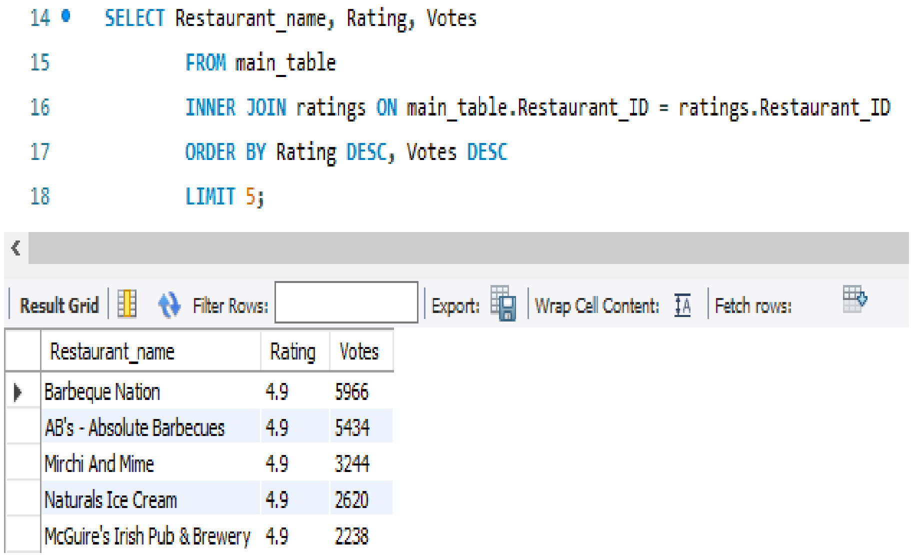
# Which country has the majority of restaurants in the dataset?



```sql
SELECT Country, COUNT(Restaurant_ID) AS NumberOfRestaurants
        FROM main_table
        INNER JOIN countrycode ON main_table.Country_code = countrycode.Country_code
        GROUP BY Country
        ORDER BY NumberOfRestaurants DESC;
```

| Country | NumberOfRestaurants |
|---|---|
| India | 8652 |
| United States | 425 |
| United Kingdom | 80 |
| South Africa | 60 |
| UAE | 60 |
| Brazil | 60 |
| New Zealand | 40 |
| Turkey | 34 |
| Australia | 24 |
| Phillipines | 22 |
| Indonesia | 21 |
| Sri Lanka | 20 |
| Qatar | 20 |
| Singapore | 20 |
| Canada | 4 |

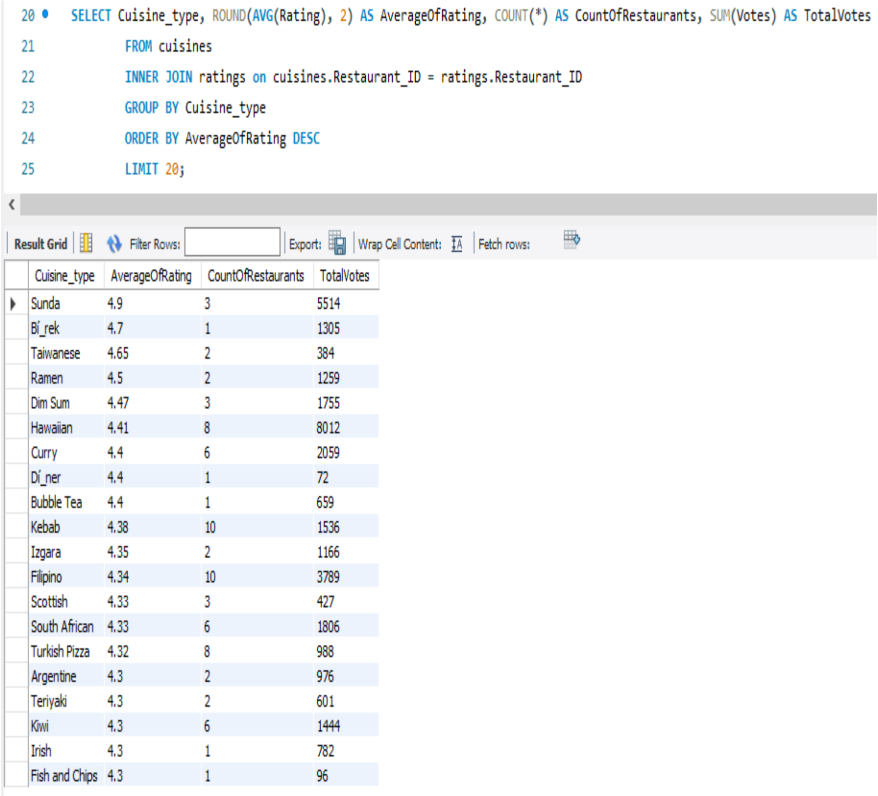
# What are the top 5 rating restaurants and how many votes does each of them collected?



```sql
SELECT Restaurant_name, Rating, Votes
        FROM main_table
        INNER JOIN ratings ON main_table.Restaurant_ID = ratings.Restaurant_ID
        ORDER BY Rating DESC, Votes DESC
        LIMIT 5;
```

| Restaurant_name | Rating | Votes |
|---|---|---|
| Barbeque Nation | 4.9 | 5966 |
| AB's - Absolute Barbecues | 4.9 | 5434 |
| Mirchi And Mime | 4.9 | 3244 |
| Naturals Ice Cream | 4.9 | 2620 |
| McGuire's Irish Pub & Brewery | 4.9 | 2238 |

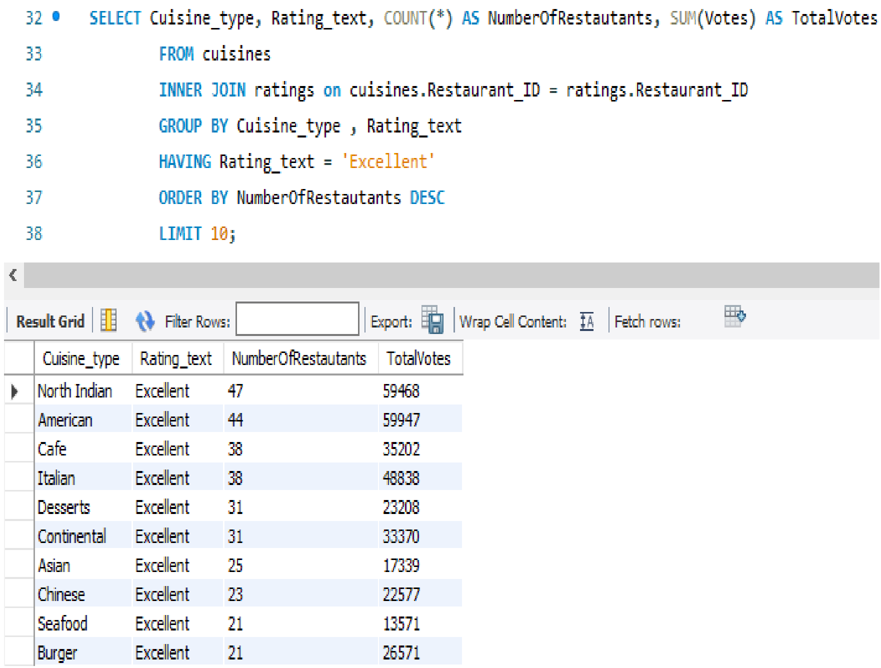
# What are the top 20 cuisine types that have the highest average rating?



```sql
SELECT Cuisine_type, ROUND(AVG(Rating), 2) AS AverageOfRating, COUNT(*) AS CountOfRestaurants, SUM(Votes) AS TotalVotes
          FROM cuisines
          INNER JOIN ratings on cuisines.Restaurant_ID = ratings.Restaurant_ID
          GROUP BY Cuisine_type
          ORDER BY AverageOfRating DESC
          LIMIT 20;
```

| Cuisine_type | AverageOfRating | CountOfRestaurants | TotalVotes |
|---|---|---|---|
| Sunda | 4.9 | 3 | 5514 |
| Bí_rek | 4.7 | 1 | 1305 |
| Taiwanese | 4.65 | 2 | 384 |
| Ramen | 4.5 | 2 | 1259 |
| Dim Sum | 4.47 | 3 | 1755 |
| Hawaiian | 4.41 | 8 | 8012 |
| Curry | 4.4 | 6 | 2059 |
| Dí_ner | 4.4 | 1 | 72 |
| Bubble Tea | 4.4 | 1 | 659 |
| Kebab | 4.38 | 10 | 1536 |
| Izgara | 4.35 | 2 | 1166 |
| Filipino | 4.34 | 10 | 3789 |
| Scottish | 4.33 | 3 | 427 |
| South African | 4.33 | 6 | 1806 |
| Turkish Pizza | 4.32 | 8 | 988 |
| Argentine | 4.3 | 2 | 976 |
| Teriyaki | 4.3 | 2 | 601 |
| Kiwi | 4.3 | 6 | 1444 |
| Irish | 4.3 | 1 | 782 |
| Fish and Chips | 4.3 | 1 | 96 |

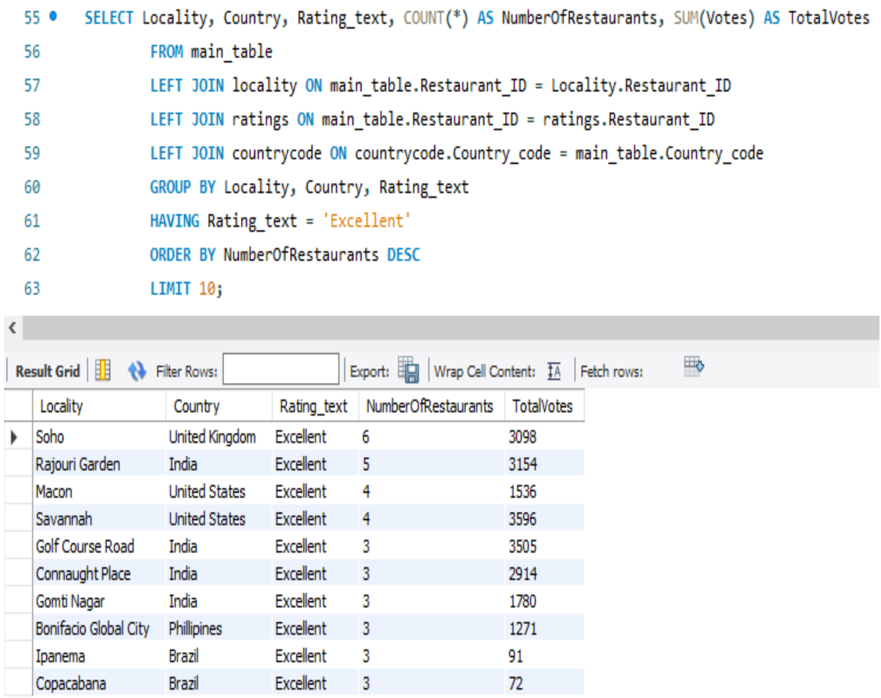
# What are the top 10 cuisine types that have the most rating as “Excellent” ?



```
SELECT Cuisine_type, Rating_text, COUNT(*) AS NumberOfRestautants, SUM(Votes) AS TotalVotes
        FROM cuisines
        INNER JOIN ratings on cuisines.Restaurant_ID = ratings.Restaurant_ID
        GROUP BY Cuisine_type , Rating_text
        HAVING Rating_text = 'Excellent'
        ORDER BY NumberOfRestautants DESC
        LIMIT 10;
```

| Cuisine_type | Rating_text | NumberOfRestautants | TotalVotes |
|---|---|---|---|
| North Indian | Excellent | 47 | 59468 |
| American | Excellent | 44 | 59947 |
| Cafe | Excellent | 38 | 35202 |
| Italian | Excellent | 38 | 48838 |
| Desserts | Excellent | 31 | 23208 |
| Continental | Excellent | 31 | 33370 |
| Asian | Excellent | 25 | 17339 |
| Chinese | Excellent | 23 | 22577 |
| Seafood | Excellent | 21 | 13571 |
| Burger | Excellent | 21 | 26571 |

# What are the top 10 localities have the most restaurants that are rated as “Excellent” ?



```sql
SELECT Locality, Country, Rating_text, COUNT(*) AS NumberOfRestaurants, SUM(Votes) AS TotalVotes
        FROM main_table
        LEFT JOIN locality ON main_table.Restaurant_ID = Locality.Restaurant_ID
        LEFT JOIN ratings ON main_table.Restaurant_ID = ratings.Restaurant_ID
        LEFT JOIN countrycode ON countrycode.Country_code = main_table.Country_code
        GROUP BY Locality, Country, Rating_text
        HAVING Rating_text = 'Excellent'
        ORDER BY NumberOfRestaurants DESC
        LIMIT 10;
```

| Locality | Country | Rating_text | NumberOfRestaurants | TotalVotes |
|---|---|---|---|---|
| Soho | United Kingdom | Excellent | 6 | 3098 |
| Rajouri Garden | India | Excellent | 5 | 3154 |
| Macon | United States | Excellent | 4 | 1536 |
| Savannah | United States | Excellent | 4 | 3596 |
| Golf Course Road | India | Excellent | 3 | 3505 |
| Connaught Place | India | Excellent | 3 | 2914 |
| Gomti Nagar | India | Excellent | 3 | 1780 |
| Bonifacio Global City | Phillipines | Excellent | 3 | 1271 |
| Ipanema | Brazil | Excellent | 3 | 91 |
| Copacabana | Brazil | Excellent | 3 | 72 |

# Thank you!