Zomato Restaurant Data Analysis using SQL (MySQL)

The Zomato Restaurant Data Analysis initiative is dedicated to conducting an in-depth exploration and examination of restaurant data sourced from Zomato, a prominent online platform for food delivery and restaurant discovery. By harnessing the power of SQL for data manipulation and querying, this endeavour strives to unveil valuable insights that can guide strategic business decisions and foster a deeper comprehension of the restaurant landscape within a specific geographic area. Through meticulous analysis of this comprehensive dataset, the project aims to uncover patterns, trends, and opportunities that can potentially shape the future course of the industry and enhance the overall dining experience for consumers.

Datasets: This data analysis project utilizes two datasets.

- ✓ ***Zomato*** – Containing comprehensive details of restaurants affiliated with Zomato. Containing 9551 rows and 13 columns: RestaurantID, Res_identify, CountryCode, City, Cuisines, Has_Table_booking, Has_Online_delivery, Is_delivering_now, Switch_to_order_menu, Price_range, Votes, Average_Cost_for_two, Rating

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	RestaurantID	Res_identify	CountryCode	City	Cuisines	Has_Table_	Has_Online_d	Is_delivering	Switc	Price_ra	Votes	Average_	Rating
2	6317637	Le 6317637	162	Makati City	French, Japanese,	Yes	No	No	No	3	314	1100	4.8
3	6304287	Iza6304287	162	Makati City	Japanese	Yes	No	No	No	3	591	1200	4.5
4	6300002	Hea6300002	162	Mandaluyong City	Seafood, Asian, Fi	Yes	No	No	No	4	270	4000	4.4
5	6318506	Oom6318506	162	Mandaluyong City	Japanese, Sushi	No	No	No	No	4	365	1500	4.9
6	6314302	Sam6314302	162	Mandaluyong City	Japanese, Korean	Yes	No	No	No	4	229	1500	4.8
7	18189371	Din18189371	162	Mandaluyong City	Chinese	No	No	No	No	3	336	1000	4.4
8	6300781	Buf6300781	162	Pasay City	Asian, European	Yes	No	No	No	4	520	2000	4
9	6301290	Vik6301290	162	Pasay City	Seafood, Filipino,	Yes	No	No	No	4	677	2000	4.2
10	6300010	Spi6300010	162	Pasay City	European, Asian, I	Yes	No	No	No	4	621	6000	4.9
11	6314987	Loc6314987	162	Pasig City	Filipino	Yes	No	No	No	3	532	1100	4.8
12	6309903	Sil6309903	162	Pasig City	Filipino, Mexican	No	No	No	No	3	1070	800	4.9
13	6309455	Mad6309455	162	Pasig City	American, Ice Crea	Yes	No	No	No	3	488	900	4.2
14	6318433	Sil6318433	162	Quezon City	Filipino, Mexican	No	No	No	No	3	294	800	4.8
15	6310470	Gue6310470	162	San Juan City	Filipino	Yes	No	No	No	3	458	1000	4.2
16	6314605	Sod6314605	162	San Juan City	Korean	No	No	No	No	3	223	700	4.3
17	18185059	Caf18185059	162	Santa Rosa	Cafe, American, It	No	No	No	No	3	29	800	3.6
18	18182702	Non18182702	162	Santa Rosa	Italian, Pizza	No	No	No	No	3	72	850	4
19	6318213	Bal6318213	162	Tagaytay City	Filipino	Yes	No	No	No	3	211	1200	4.5
20	18255654	Hob18255654	162	Taguig City	Cafe, Korean, Des	No	No	No	No	2	118	600	4.5
21	6308205	Wil6308205	162	Taguig City	Cafe, Bakery, Ame	Yes	No	No	No	4	392	1500	4.4
22	6315438	NIU6315438	162	Taguig City	Seafood, America	Yes	No	No	No	4	535	3000	4.7

- ✓ ***CountryTable*** – containing 15 rows and 2 columns: CountryCode, Country

	A	B
1	CountryCode	Country
2	1	India
3	214	UAE
4	215	UK
5	216	USA
6	14	South Africa
7	30	Malayasia
8	37	Indonesia
9	94	Singapore
10	148	Hongkong
11	162	Nigeria
12	166	France
13	184	Switzerland
14	189	Australia
15	191	New Zealand
16	208	Canada

Through this comprehensive analysis, the project aims to offer a deeper understanding of the Zomato restaurant ecosystem, enhance user experiences, and optimize business strategies.

While analyzing Zomato restaurant data, the focus was on the following aspects:

- ✓ ***City Ratings Analysis:*** Identified cities with poor restaurant ratings to help Zomato target areas for improvement.
- ✓ **Affordable and Highly Rated Restaurants:** Enlisted the most affordable and highly rated restaurants city-wise to assist users in finding value-for-money dining options.
- ✓ **Offline Services Evaluation:** Identified restaurants with poor offline services to enhance customer satisfaction.
- ✓ **Restaurant Categorization:** Grouped restaurants based on average cost for two into categories such as Luxurious Expensive, Very Expensive, Expensive, High, Medium High, and Average, then quantified the number of restaurants in each category.
- ✓ **Top Rated Restaurants:** Listed the top 5 restaurants with the highest rating and maximum votes to highlight premier dining options.
- ✓ **High-Rated Cities Identification:** Identified cities with at least 3 restaurants having ratings >= 4.9, sorted alphabetically in case of ties.
- ✓ **Top Countries by Restaurant Count:** Identified the top 5 countries with the most restaurants linked with Zomato.

Analysis:

```
5
6     -- Business Questions: --
7     /* 1) Help Zomato in identifying the cities with poor Restaurant ratings */
8     select city, round(avg(Rating),1) as AvgRating
9     from zomato
10    group by city
11    having AvgRating < 2.5; -- we can set the value of AvgRating as per business needs
12
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

city	AvgRating
Mc Millan	2.4
Montville	2.4
Faridabad	2.3
Noida	2.4

** Here, cities with poor restaurant ratings were identified. Initially, the data was grouped by cities and the average restaurant rating for each city was calculated. Subsequently, a condition was applied on the average rating. The 'HAVING' clause was utilized instead of the 'WHERE' clause because the condition was applied to aggregate values derived from the 'GROUP BY' clause.

```sql
/* 2)Enlist most affordable and highly rated restaurants city wise. */
select city, RestaurantID, Res_identify, Rating, Average_Cost_for_two
from (select city, RestaurantID, Res_identify, Rating, Average_Cost_for_two,
rank() over (partition by city order by Rating desc, Average_Cost_for_two) as rnk
from zomato) as temp
where rnk = 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

city	RestaurantID	Res_identify	Rating	Average_Cost_for_two
Abu Dhabi	5703500	Pun5703500	4.9	330
Agra	3400346	She3400346	4.9	0
Ahmedabad	18385201	Cry18385201	4.6	350
Albany	17284409	Gua17284409	3.9	10
Albany	17284158	Jim17284158	3.9	10
Allahabad	2400052	Eat2400052	3.7	200
Amritsar	2200175	Gur2200175	4.1	100
Ankara	6004011	Gag6004011	4.9	80
Armidale	16611114	Whi16611114	3.5	20
Athens	17293409	Sr.17293409	4.6	10

Result 2

** Here, highly rated and most affordable restaurants for each city were identified. Windows functions were employed for the calculation. The RANK window function grouped the restaurants for each city and ranked them based on high ratings and low costs. This query was utilized as a Common Table Expression (CTE), and based on that another query was written where the condition 'rank=1' was applied to obtain the highly rated and most affordable restaurant for each city.

```sql
/* 3)Help Zomato in identifying the restaurants with poor offline services ??????*/
select RestaurantID, Res_identify, Has_Table_booking, Has_Online_delivery, Votes, Rating
from zomato
where Has_Table_booking = "yes" and Has_Online_delivery = "no"
        and Rating < 2.5 -- we can set the value of AvgRating as per business needs
order by Rating;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

RestaurantID	Res_identify	Has_Table_booking	Has_Online_delivery	Votes	Rating
17977757	Cof17977757	Yes	No	0	1
308961	Ang308961	Yes	No	0	1
308962	The308962	Yes	No	0	1
309368	Sil309368	Yes	No	3	1
18466937	The18466937	Yes	No	0	1
301468	Viv301468	Yes	No	0	1
311629	Ill311629	Yes	No	2	1
18419884	Tes18419884	Yes	No	1	1

zomato 11

** For identifying restaurants with poor offline services, three conditions were applied. The restaurant should offer offline table booking services, while online delivery services should be unavailable. Additionally, restaurants with ratings below a specified threshold value were considered to provide poor service. The threshold value of the rating can be adjusted based on business needs.

```sql
/* 4)Help zomato in identifying those cities which have atleast 3 restaurants with ratings >= 4.9
  In case there are two cities with the same result, sort them in alphabetical order.*/
SELECT city,
       count(restaurantid) as noOfRestaurants
FROM zomato
WHERE rating >= 4.9
GROUP BY city
having count(restaurantid) >=3
ORDER BY NoOfRestaurants DESC, city ASC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

city	noOfRestaurants
Dubai	3
Jakarta	3
London	3
New Delhi	3

** To identify cities with at least 3 restaurants having ratings of 4.9 or higher, the cities and restaurants with ratings greater than or equal to 4.9 were selected. The WHERE clause was utilized for this purpose. Subsequently, the data was grouped by cities using the GROUP BY clause, and the count of restaurants for each city was determined. The HAVING clause was then applied to the aggregate value count to filter cities with at least 3 restaurants with ratings of 4.9 or higher.

```sql
/* 5) What are the top 5 countries with most restaurants linked with Zomato? */
select Country, count(*) TotalRestaurants
from zomato INNER JOIN CountryTable ON zomato.countrycode = CountryTable.CountryCode
group by CountryTable.Country
order by 2 desc
limit 5;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Country	TotalRestaurants
India	8652
USA	434
UK	80
Malayasia	60
UAE	60

** To determine the top 5 countries with the most restaurants linked with Zomato, two tables, Zomato (containing restaurant details) and CountryCode (containing country names), were joined using an INNER JOIN to obtain restaurant data along with country names. Next, the data was grouped by country using the GROUP BY clause, and the count of restaurants associated with Zomato in each country was calculated. The ORDER BY clause was then utilized to reorder the countries in descending order based on the number of restaurants. Finally, the LIMIT clause was applied to retrieve only the top 5 countries with the highest number of restaurants linked with Zomato.

```sql
/* 6) Group the restaurants basis the average cost for two into:
Luxurious Expensive, Very Expensive, Expensive, High, Medium High, Average.
Then, find the number of restaurants in each category.
*/
with tempdf as(
                select RestaurantID,
                case
                    when Average_Cost_for_two between 0 and 1000 then "Average"
                    when Average_Cost_for_two between 1001 and 5000 then  "Medium High"
                    when Average_Cost_for_two between 5001 and 10000 then "High"
                    when Average_Cost_for_two between 10001 and 50000 then "Expensive"
                    when Average_Cost_for_two between 50001 and 100000 then "Very Expensive"
                    else "Luxurious Expensive"
                end as Status
                from zomato)

select Status, count(*) as `Number of Resturant`
from tempdf
group by Status
order by 2 desc;
```

Result Grid | Filter Rows: | Export: | Wrap Cell

Status	Number of Resturant
Average	8376
Medium High	1143
Luxurious Expensive	18
High	11
Very Expensive	3

** Here, the CASE WHEN clause was initially employed to group restaurants based on the average cost for two persons. The cost range can be adjusted according to business needs to effectively group the restaurants.

Subsequently, this query was utilized as a common table expression to construct another query aimed at determining the total number of restaurants belonging to each category. The GROUP BY clause was utilized to group restaurants by each category, and the count of restaurants within each category was calculated.

```
64
65    /* 7) List the two top 5 restaurants with highest rating with maximum votes. */
66    select RestaurantID, Res_identify, Rating, Votes
67    from zomato
68    order by Rating desc, Votes desc
69    limit 5;
70
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

RestaurantID	Res_identify	Rating	Votes
20842	Bar20842	4.9	5966
94286	AB'94286	4.9	5434
17806994	Mir17806994	4.9	3244
310143	Nat310143	4.9	2620
17580142	McG17580142	4.9	2238

** To identify the top 5 restaurants with the highest ratings and maximum votes, the dataset was reordered using the ORDER BY clause, sorting by rating and votes in descending order. Subsequently, the top 5 rows were extracted using the LIMIT clause.

```
71    /* 8) If someone is looking for a restaurant in kolkata which provides online
72    delivery. Help him choose the best restaurant */
73    select RestaurantID, Res_identify, City, Has_Online_delivery, Rating, Average_Cost_for_two
74    from zomato
75    where City = "Kolkata" and Has_Online_delivery = "yes"
76    order by Rating desc, Average_Cost_for_two asc
77    limit 1;  -- For more suggestions we can set the limit value more than 1
78
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

RestaurantID	Res_identify	City	Has_Online_delivery	Rating	Average_Cost_for_two
20747	Ind20747	Kolkata	Yes	4.6	800

** Here, an attempt was made to find the best restaurant in Kolkata that provides online delivery. To achieve this, the WHERE clause was utilized to apply conditions such as the city being Kolkata and the restaurant offering online delivery. For the best restaurant, one should consider establishments with high ratings and low costs. Consequently, the ORDER BY clause was employed to reorder the dataset, with ratings sorted in descending order and costs in ascending order. Finally, the LIMIT clause with a value of 1 was applied to identify the top restaurant in Kolkata providing online delivery. The limit value can be adjusted to explore more restaurant.

```sql
/* 9) Help someone in finding the best rated Restaurant for Pizza in New Delhi. */
select  RestaurantID, Res_identify, City, Cuisines, Rating
from zomato
where Cuisines like "%pizza%" And city = "New delhi"
order by Rating desc
limit 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

RestaurantID	Res_identify	City	Cuisines	Rating
18400736	Owl18400736	New Delhi	Burger, American, Fast Food, Italian, Pizza	4.5

** In this scenario, an attempt was made to identify the highest-rated restaurant for pizza in New Delhi. The WHERE clause was utilized to apply conditions such as the city being New Delhi and pizza being available in their cuisines. To determine the best-rated restaurant, the ORDER BY clause was used to reorder the dataset, with ratings sorted in descending order. Finally, the LIMIT clause with a value of 1 was applied to pinpoint the top-rated restaurant for pizza in New Delhi. The limit value can be adjusted to explore additional restaurant options.

Results:

- ✓ In the above analysis, cities with poor restaurant ratings and restaurants with poor offline services were identified. This analysis can assist Zomato in taking necessary actions to enhance their services in those areas.

- ✓ Additionally, the most affordable and highly rated restaurants were identified city-wise, and restaurants were categorized into various cost brackets, such as Luxurious Expensive, Very Expensive, Expensive, High, Medium High, and Average. This categorization aids customers in selecting restaurants according to their preferences.

- ✓ Furthermore, the top 5 countries with the highest number of restaurants linked with Zomato were identified. This information enables Zomato to concentrate on countries with a low number of restaurants linked with their platform to expand their business.

- ✓ Moreover, Zomato can assist its customers by providing information about the best restaurants offering online delivery in any city or allowing customers to order specific cuisines from highly rated restaurants in any city.