EATS CONSULTING

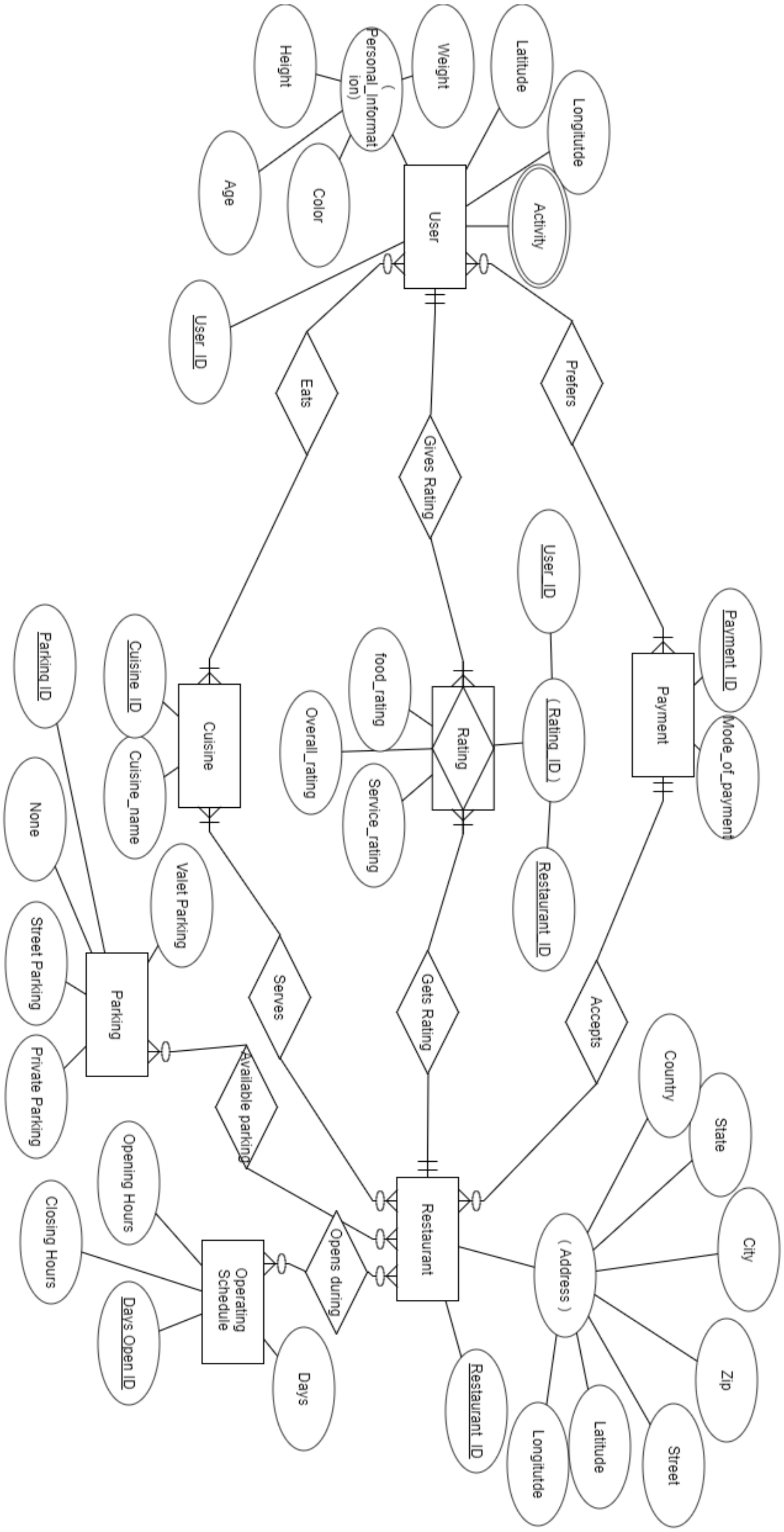
Naveen Ellappan, Anesh Krishna J N

Background

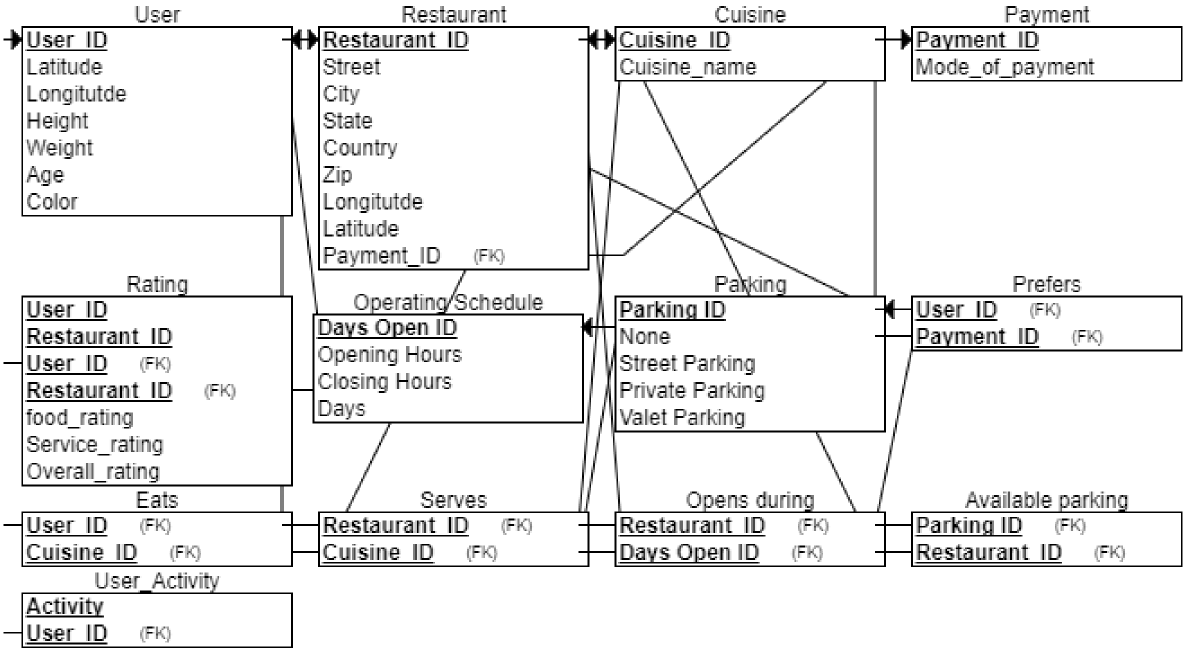
Our firm EATS Consulting works with restaurants to help them optimize the way they operate. Restaurants are one of the riskiest entrepreneurial ventures to start; so, our goal is to provide our client with the data of surrounding restaurants, to be used to make business decisions to lower costs, attract more customers, and many other important decisions vital to the business's success. Our current client is an entrepreneur in Mexico who has just opened his restaurant. We have gathered data from locations surrounding his restaurant and created a database that will allow our client to extract the data he needs and start to infer as to what to do with that data. Our client wants to know what changes he should implement in his restaurant to maximize profits. Using the database we designed, our client will be able to look at easily look through huge data sets containing information about the types of cuisine served at nearby restaurants, most popular location, preferred payment types, restaurant ratings, and customer information. Our data base is designed to help our client pick the best strategy that are aligned with their goals for the establishment. Usually in the first couple years of a restaurant opening it will take them time to try different menu options to try and set a foot in the local market. During this time, they will also be trying different marketing strategies and try to target the right segment of customers to maximize their profits. These trial and error processes will take time as stated above but more importantly to any business is the cost. For example, in two years of trial and error, if our client still cannot attract customers, there is a good possibility of the business shutting down. With our consulting services this trial and error could be cut down drastically. With the customer preference data we are providing, we can find which payment type is most preferred by counting which one is used the most. Using this type of information our client can make sure he accepts that particular form of payment to capture that segment. In our report we will focus on the most important business functions that require our database service.

Introduction: Project Objective

Our projects goal is to design, create, and implement a database that can be easily be parsed through to retrieve the information that our client wants. The first step was to create an ERD diagram connecting the most important information from our datasets so that the data tables will be able to be joined later on. The next step is to translate the ERD into a relational schema with the primary keys properly labeled. Choosing the right primary key is important to be able to get the most amount of connections between data to get even more information. From here we finish the database with normalization. Moving into MySQL, we import our datasets into the program and test the datasets. Now we focus on the business perspective of the project. We are providing our client with a large set of queries available to them at their disposal to extract the information they need. They will also be able to input their information to the database.


User
(Personal_Information)
Height
Weight
Age
Color
Latitude
Longitude
Activity
User_ID
Eats
Gives Rating
Prefers
Cuisine
Cuisine_ID
Cuisine_name
Rating
food_rating
Overall_rating
Service_rating
User_ID
(Rating_ID)
Restaurant_ID
Payment
Payment_ID
Mode_of_payment
Parking
Parking ID
None
Street Parking
Private Parking
Valet Parking
Serves
Gets Rating
Accepts
Available parking
Restaurant
Opens during
Operating Schedule
Opening Hours
Closing Hours
Days Open ID
Days
(Address)
Country
State
City
Zip
Street
Latitude
Longitude
Restaurant_ID

Relational Schema



ERD/Relational Schema Design Summary

When designing our data model, we first had to decide what are the basic preferences consumers want when they are going to eat out. We knew immediately that we would need a user entity to match specific users with entities like Payment. The entities we chose are Payment, User, Restaurant, Cuisine, Parking, Operating Schedule and Rating. Each of these have attributes that are very specific so the most detailed information could be extracted. The relationships design process was also very detailed.

- User-Payment, User-cuisine, Restaurant-Cuisine, Restaurant-parking, Restaurant-Operating Schedule and Restaurant-payment have many to many relationships.
- User and Restaurant are connected with the help of an associative entity, Rating. Essentially, Rating entity consists of information that could be used by both User entity and Restaurant entity.
- We can also see that the underlying Relational Schema is in sync with the above depicted ERD, where every one of the many to many relationship is broken with the help of a separate table consisting of respective foreign keys.

How will these queries help our client?

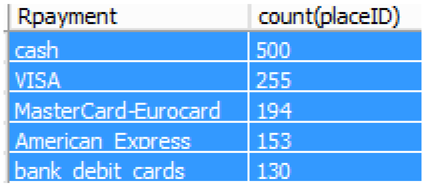
These queries are meant to optimize the decision making time of our client. With these queries the menu can be designed to meet the preferences of the customers. For example, one query helps us find the three most popular types of parking lots that are used. This information can make our client think about whether or not to make a long time investment into building a parking lot or not. This information will come in handy for cost benefit situations. Another important set of queries are our Dress Code and Budget Query. These queries can give valuable information when used separately and together. The budget query allows our client to see what the typical budget of the consumer is in that area. This is very important information to a restaurant owner. With this information our client can choose the price that most consumers are willing to pay. This information also trickles down into many other business functions. What our client decides to charge has a direct impact on how much they can pay to buy supplies. The price has to be optimal so that it is attractive to customers and doesn't lead to buying poorer quality supplies. The dress code query shows the amount of people who prefer casual, formal, or informal. These queries can help our client but not ensure it will attract customers. With the Min and Max rating queries we can see which places have the highest and lowest rating. Using this information, you can see whether the place_ID with the lowest rating is for example serving the highest rated cuisine. In this case our client can look at this information and decide whether serving the highest rated cuisine is really a factor in a higher restaurant rating. Below you can find many more queries with basic descriptions as to what information you can extract.

SQL Queries:

# Basic data extraction queries

#1 Finding the three most preferred mode of payments accepted

select Rpayment, count(placeID) from chefmozaccepts group by Rpayment order by count(placeID) Desc limit 5;

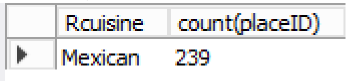


Rpayment	count(placeID)
cash	500
VISA	255
MasterCard-Eurocard	194
American_Express	153
bank_debit_cards	130

Cash and VISA credit cards are the most preferred modes of payment(MoP)

#2 Finding the most popular cuisines served

select Rcuisine, count(placeID) from chefmozcuisine group by Rcuisine order by count(placeID) Desc limit 1;

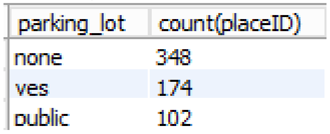


	Rcuisine	count(placeID)
▶	Mexican	239

Mexican cuisine is the most popularly served

#3 Finding three most popular parking_lots used

select parking_lot, count(placeID) from chefmozparking group by parking_lot order by count(placeID) Desc limit 3;

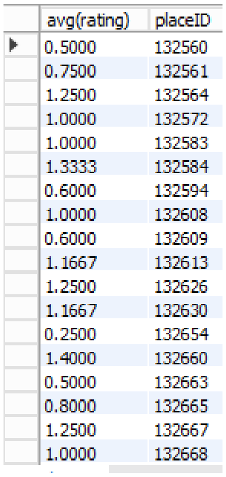


parking_lot	count(placeID)
none	348
yes	174
public	102

It's clear that there are many branches which don't have any parking slot

#4 Average ratings received by Places

select avg(rating), placeID from rating_final group by placeID;

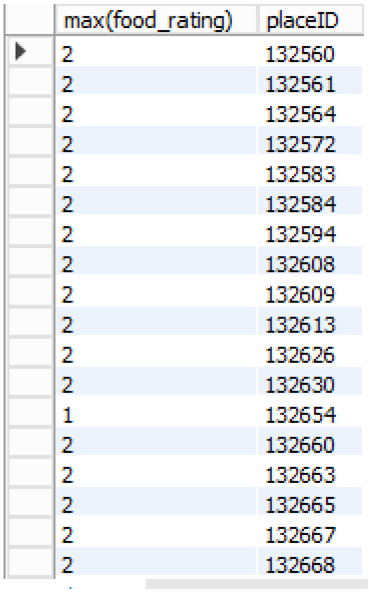


avg(rating)	placeID
0.5000	132560
0.7500	132561
1.2500	132564
1.0000	132572
1.0000	132583
1.3333	132584
0.6000	132594
1.0000	132608
0.6000	132609
1.1667	132613
1.2500	132626
1.1667	132630
0.2500	132654
1.4000	132660
0.5000	132663
0.8000	132665
1.2500	132667
1.0000	132668

Just an overview of the resulting table;

#5 Max food ratings received by Places

select max(food_rating), placeID from rating_final group by placeID;

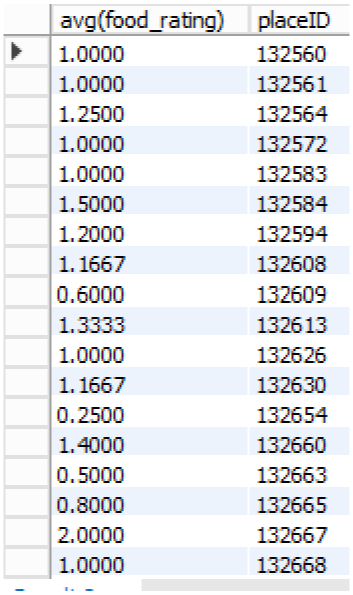


max(food_rating)	placeID
2	132560
2	132561
2	132564
2	132572
2	132583
2	132584
2	132594
2	132608
2	132609
2	132613
2	132626
2	132630
1	132654
2	132660
2	132663
2	132665
2	132667
2	132668

Just an overview of the resulting table;

#6 Avg food ratings received by Places

select avg(food_rating), placeID from rating_final group by placeID;

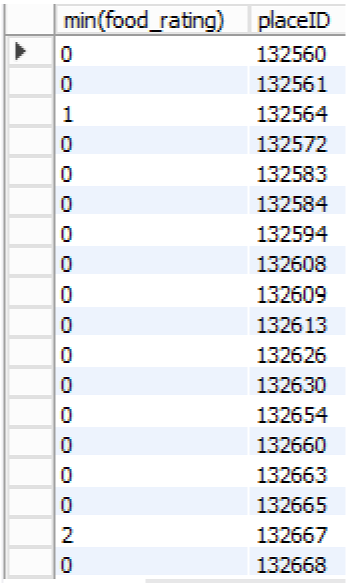


avg(food_rating)	placeID
1.0000	132560
1.0000	132561
1.2500	132564
1.0000	132572
1.0000	132583
1.5000	132584
1.2000	132594
1.1667	132608
0.6000	132609
1.3333	132613
1.0000	132626
1.1667	132630
0.2500	132654
1.4000	132660
0.5000	132663
0.8000	132665
2.0000	132667
1.0000	132668

Just an overview of the resulting table;

#7 Min food ratings received by Places

select min(food_rating), placeID from rating_final group by placeID;

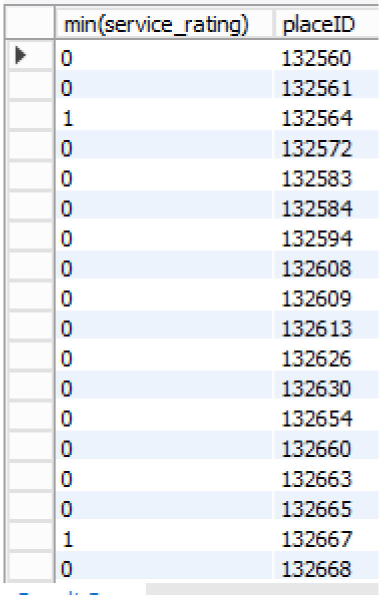


min(food_rating)	placeID
0	132560
0	132561
1	132564
0	132572
0	132583
0	132584
0	132594
0	132608
0	132609
0	132613
0	132626
0	132630
0	132654
0	132660
0	132663
0	132665
2	132667
0	132668

Just an overview of the resulting table;

#8 Min service ratings received by Places

select min(service_rating), placeID from rating_final group by placeID;

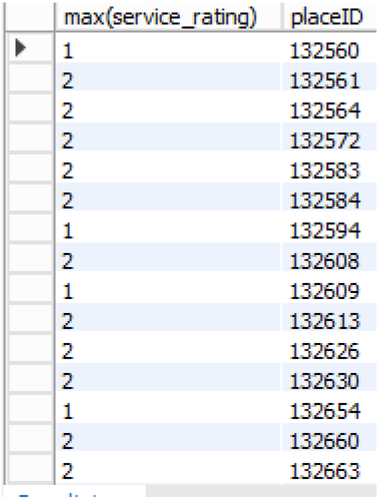


min(service_rating)	placeID
0	132560
0	132561
1	132564
0	132572
0	132583
0	132584
0	132594
0	132608
0	132609
0	132613
0	132626
0	132630
0	132654
0	132660
0	132663
0	132665
1	132667
0	132668

Just an overview of the resulting table;

#9 Max service ratings received by Places

select max(service_rating), placeID from rating_final group by placeID;

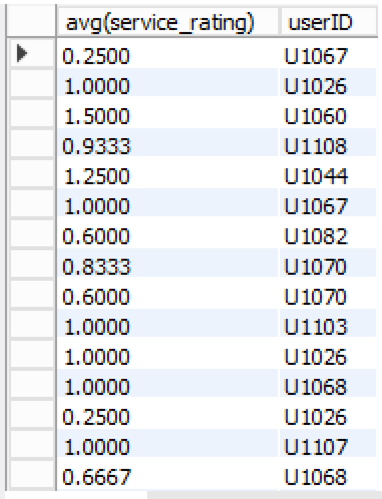


max(service_rating)	placeID
1	132560
2	132561
2	132564
2	132572
2	132583
2	132584
1	132594
2	132608
1	132609
2	132613
2	132626
2	132630
1	132654
2	132660
2	132663

Just an overview of the resulting table;

#10 Avg service ratings received by places

select avg(service_rating), userID from rating_final group by placeID;

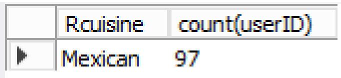


avg(service_rating)	userID
0.2500	U1067
1.0000	U1026
1.5000	U1060
0.9333	U1108
1.2500	U1044
1.0000	U1067
0.6000	U1082
0.8333	U1070
0.6000	U1070
1.0000	U1103
1.0000	U1026
1.0000	U1068
0.2500	U1026
1.0000	U1107
0.6667	U1068

Just an overview of the resulting table;

#11 Finding the most popular cuisines among users

select Rcuisine, count(userID) from usercuisine group by Rcuisine order by count(userID) Desc limit 1;

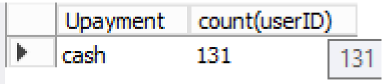


Rcuisine	count(userID)
Mexican	97

It's clear that Mexican is the most popular cuisine among users.

#12 Finding the most popular payments used

select Upayment, count(userID) from userpayment group by Upayment order by count(userID) Desc limit 1;



Upayment	count(userID)
cash	131

It's clear that Cash is the most preferred mode of payments among users. So, the restaurant must accept cash at all of its branches, irrespective of any other mode of payments they accept.

# INFERENCES AFTER COMPARING

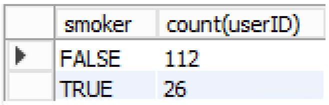
#13 Smoking behavior

select smoker, count(userID) from userprofile group by smoker;

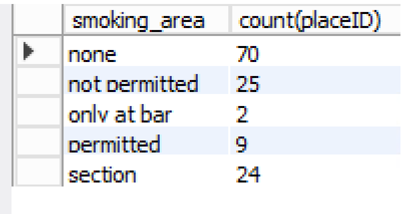
AND

#14 Smoking area

select smoking_area, count(placeID) from geoplaces group by smoking_area;



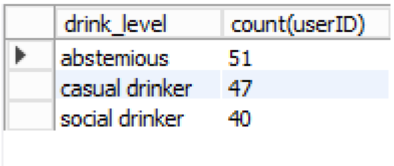
smoker	count(userID)
FALSE	112
TRUE	26



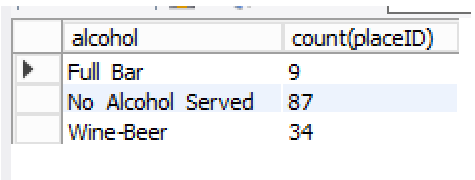
smoking_area	count(placeID)
none	70
not permitted	25
only at bar	2
permitted	9
section	24

It's clear that many people don't smoke in this region. Hence it's obvious that it's not a good idea for the restaurants to spend more on providing smoking facilities. Currently only 70 of their 130 places do not have smoking facilities. They should plan on increasing this and utilising the existing smoking facility for someother purpose.

#15 Drinking Behavior
select drink_level, count(userID) from userprofile group by drink_level;
AND
#16 Alcohol
select alcohol, count(placeID) from geoplaces group by alcohol;



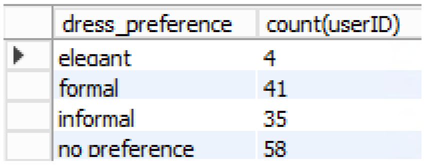
drink_level	count(userID)
abstemious	51
casual drinker	47
social drinker	40



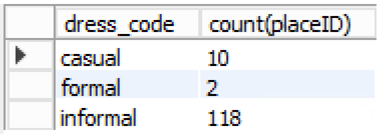
alcohol	count(placeID)
Full Bar	9
No Alcohol Served	87
Wine-Beer	34

It's clear that many people drink in this region. Hence it's obvious that more than restaurants, they should plan on coming up with more resto-bars. Currently 87 of their 130 places do not have serve alcohol at all. They should plan on increasing to increase their service ratings and also their profits.

#17 Dressing behavior
select dress_preference, count(userID) from userprofile group by dress_preference;
AND
#18 Dress code
select dress_code, count(placeID) from geoplaces group by dress_code;



dress_preference	count(userID)
elegant	4
formal	41
informal	35
no preference	58

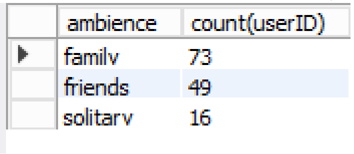


dress_code	count(placeID)
casual	10
formal	2
informal	118

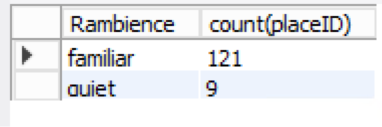
It's clear that more people do not have any preference or like informal. So the dress code should be informal in most of the places, which is the case now. So they are doing good on this aspect here.

#19 accompanying people kind
select ambience, count(userID) from userprofile group by ambience;
AND
#20 Ambience

select Rambience, count(placeID) from geoplaces group by Rambience;



ambience	count(userID)
family	73
friends	49
solitary	16



Rambience	count(placeID)
familiar	121
quiet	9

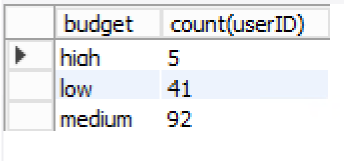
#21 budget

select budget, count(userID) from userprofile group by budget;

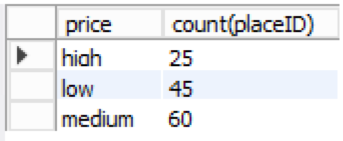
AND

#22 price

select price, count(placeID) from geoplaces group by price;



budget	count(userID)
high	5
low	41
medium	92



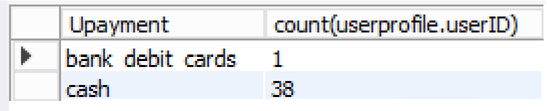
price	count(placeID)
high	25
low	45
medium	60

High Budget people are very less compared to other two category people. So, it's not a good idea sto have high priced restaurants. So, they should plan accordingly and reduce the count of High priced restaurants from 25 to a much smaller count.

#Simple Join,

#23 24 25 USERPROFILE AND USER MoP

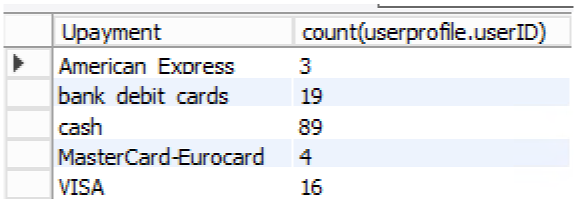
```
select Upayment, count(userprofile.userID)
from userpayment
join userprofile
on userpayment.userID = userprofile.userID
where budget = "low"
group by Upayment;
```



Upayment	count(userprofile.userID)
bank_debit_cards	1
cash	38

Low Budget people prefer only cash and Debit cards. So, it becomes mandatory that the restaurant accept cash at all of its branches, irrespective of any other mode of payments they accept.

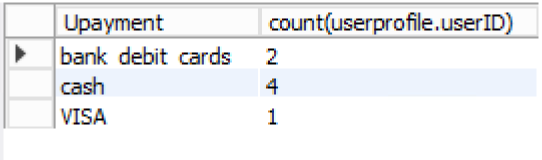
```
select Upayment, count(userprofile.userID)
from userpayment
join userprofile
on userpayment.userID = userprofile.userID
where budget = "medium"
group by Upayment;
```



Upayment	count(userprofile.userID)
American Express	3
bank debit cards	19
cash	89
MasterCard-Eurocard	4
VISA	16

Most of the Medium budget people prefer cash over other modes, even though Bank debit cards and VISA cards are prominent amongst few of them.

```
select Upayment, count(userprofile.userID)
from userpayment
join userprofile
on userpayment.userID = userprofile.userID
where budget = "high"
group by Upayment;
```



Upayment	count(userprofile.userID)
bank debit cards	2
cash	4
VISA	1

All high budget people prefer one of cash, Bank debit cards and VISA cards.

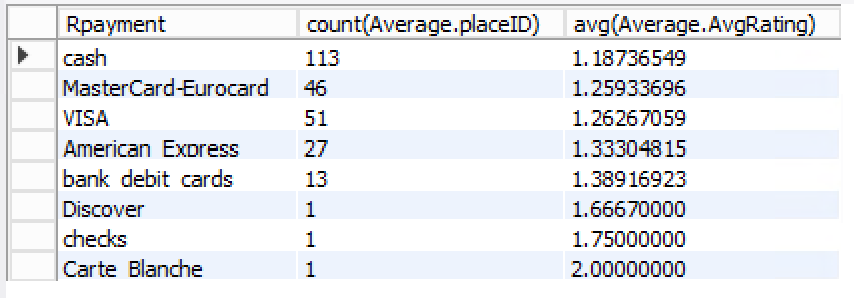
Major inference is that, they will be good enough with just three payment options: cash, Bank debit cards and VISA cards.

#Sub Query

# MoP, cuisines and parking lots with Ratings, food and service ratings from rating final

#26 Avg Ratings recieved as per mode of payment

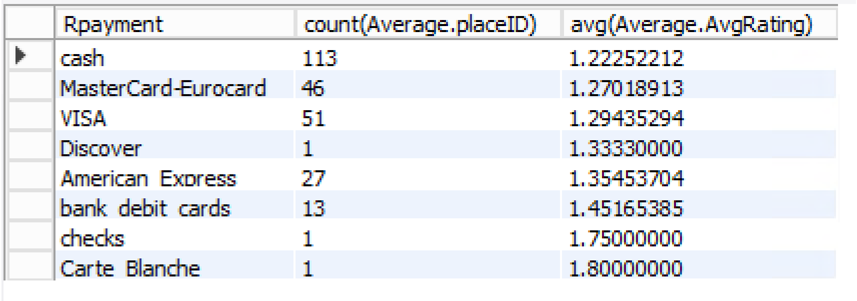
```
Select c.Rpayment, count(Average.placeID), avg(Average.AvgRating)
from chefmozaccepts as c,
        (select avg(rating) as AvgRating, placeID from rating_final group by placeID
  ) as Average
where c.placeID = Average.placeID
group by Rpayment
order by avg(Average.AvgRating);
```



Rpayment	count(Average.placeID)	avg(Average.AvgRating)
cash	113	1.18736549
MasterCard-Eurocard	46	1.25933696
VISA	51	1.26267059
American Express	27	1.33304815
bank debit cards	13	1.38916923
Discover	1	1.66670000
checks	1	1.75000000
Carte Blanche	1	2.00000000

#27 Avg food Ratings recieved as per mode of payment

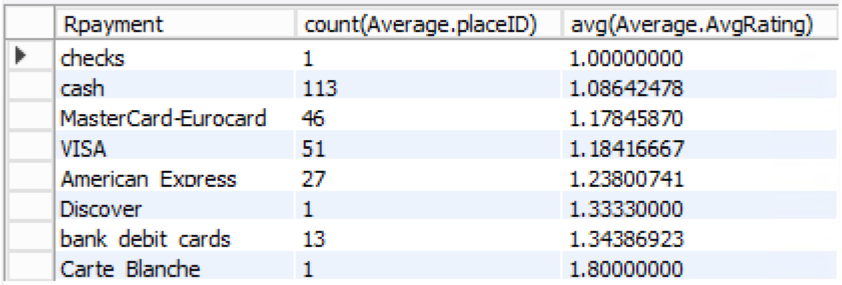
```
Select c.Rpayment, count(Average.placeID), avg(Average.AvgRating)
from chefmozaccepts as c,
        (select avg(food_rating) as AvgRating, placeID from rating_final group by placeID
  ) as Average
where c.placeID = Average.placeID
group by Rpayment
order by avg(Average.AvgRating);
```



Rpayment	count(Average.placeID)	avg(Average.AvgRating)
cash	113	1.22252212
MasterCard-Eurocard	46	1.27018913
VISA	51	1.29435294
Discover	1	1.33330000
American_Express	27	1.35453704
bank_debit_cards	13	1.45165385
checks	1	1.75000000
Carte_Blanche	1	1.80000000

#28 Avg service Ratings recieved as per mode of payment

```
Select c.Rpayment, count(Average.placeID), avg(Average.AvgRating)
from chefmozaccepts as c,
        (select avg(service_rating) as AvgRating, placeID from rating_final group by placeID
  ) as Average
where c.placeID = Average.placeID
group by Rpayment
order by avg(Average.AvgRating);
```

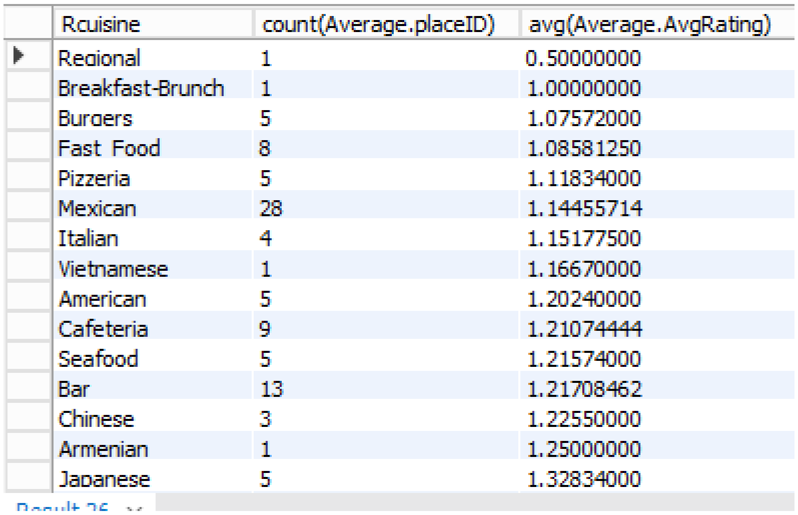


Rpayment	count(Average.placeID)	avg(Average.AvgRating)
checks	1	1.00000000
cash	113	1.08642478
MasterCard-Eurocard	46	1.17845870
VISA	51	1.18416667
American_Express	27	1.23800741
Discover	1	1.33330000
bank_debit_cards	13	1.34386923
Carte_Blanche	1	1.80000000

MAJOR INFERENCE: Ignoring Discover, Carte Blanche and Checks, it's clear that, the average rating in all 3 aspects, received by places that use Bank debit cards is considerably high compared to other mode of payments.

#29 Avg Ratings recieved as per cuisines served.

```
Select c.Rcuisine, count(Average.placeID), avg(Average.AvgRating)
from chefmozcuisine as c,
        (select avg(rating) as AvgRating, placeID from rating_final group by placeID
    ) as Average
where c.placeID = Average.placeID
group by Rcuisine
order by avg(Average.AvgRating);
```

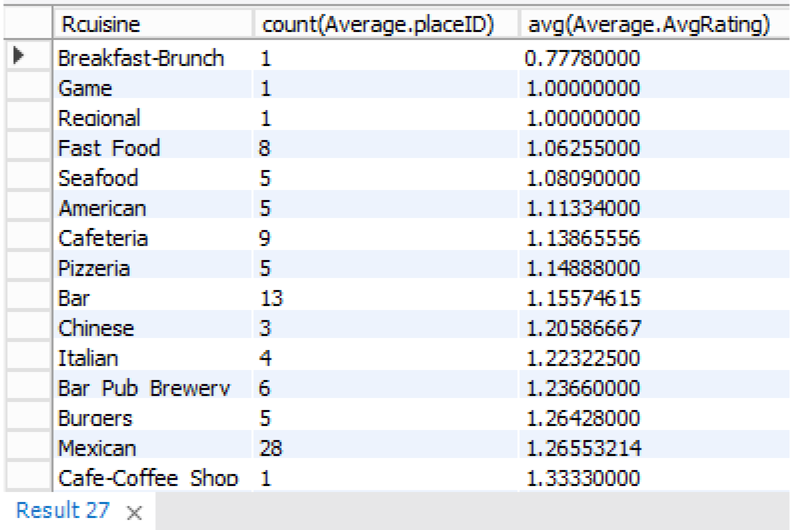


Rcuisine	count(Average.placeID)	avg(Average.AvgRating)
Regional	1	0.50000000
Breakfast-Brunch	1	1.00000000
Burgers	5	1.07572000
Fast_Food	8	1.08581250
Pizzeria	5	1.11834000
Mexican	28	1.14455714
Italian	4	1.15177500
Vietnamese	1	1.16670000
American	5	1.20240000
Cafeteria	9	1.21074444
Seafood	5	1.21574000
Bar	13	1.21708462
Chinese	3	1.22550000
Armenian	1	1.25000000
Japanese	5	1.32834000

It's just an overview of the table.

#30 Avg food Ratings recieved as per cuisines served.

```
Select c.Rcuisine, count(Average.placeID), avg(Average.AvgRating)
from chefmozcuisine as c,
        (select avg(food_rating) as AvgRating, placeID from rating_final group by placeID
    ) as Average
where c.placeID = Average.placeID
group by Rcuisine
order by avg(Average.AvgRating);
```

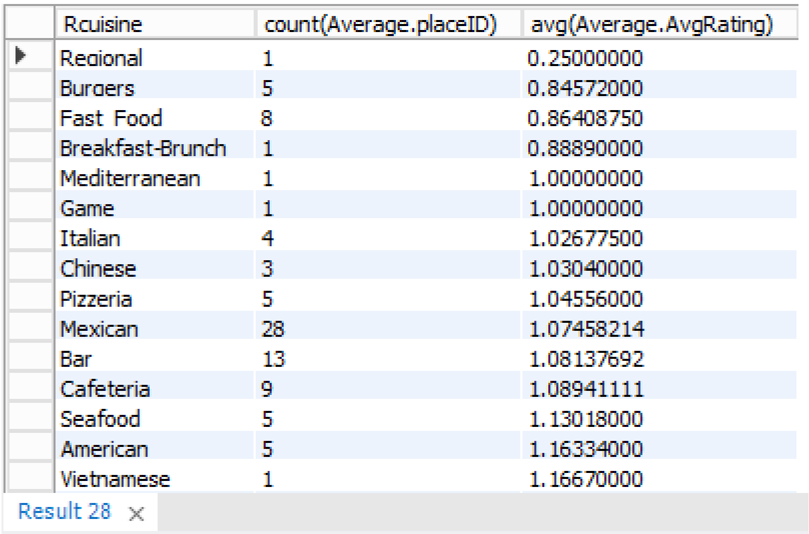


Rcuisine	count(Average.placeID)	avg(Average.AvgRating)
Breakfast-Brunch	1	0.77780000
Game	1	1.00000000
Regional	1	1.00000000
Fast_Food	8	1.06255000
Seafood	5	1.08090000
American	5	1.11334000
Cafeteria	9	1.13865556
Pizzeria	5	1.14888000
Bar	13	1.15574615
Chinese	3	1.20586667
Italian	4	1.22322500
Bar_Pub_Brewery	6	1.23660000
Burgers	5	1.26428000
Mexican	28	1.26553214
Cafe-Coffee_Shop	1	1.33330000

It's just an overview of the table.

#31 Avg service Ratings recieved as per cuisines served.

```
Select c.Rcuisine, count(Average.placeID), avg(Average.AvgRating)
from chefmozcuisine as c,
          (select avg(service_rating) as AvgRating, placeID from rating_final group by
placeID
  ) as Average
where c.placeID = Average.placeID
group by Rcuisine
order by avg(Average.AvgRating);
```

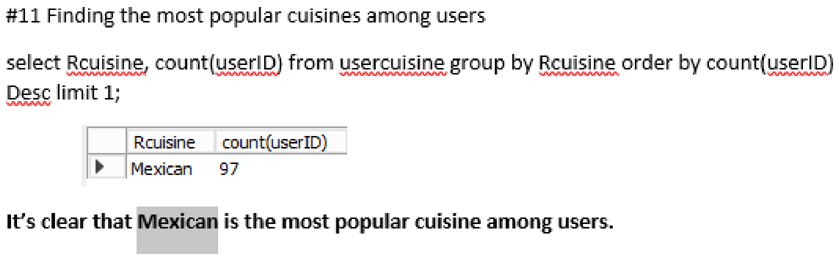


Rcuisine	count(Average.placeID)	avg(Average.AvgRating)
Regional	1	0.25000000
Burgers	5	0.84572000
Fast Food	8	0.86408750
Breakfast-Brunch	1	0.88890000
Mediterranean	1	1.00000000
Game	1	1.00000000
Italian	4	1.02677500
Chinese	3	1.03040000
Pizzeria	5	1.04556000
Mexican	28	1.07458214
Bar	13	1.08137692
Cafeteria	9	1.08941111
Seafood	5	1.13018000
American	5	1.16334000
Vietnamese	1	1.16670000

It's just an overview of the table.

MAJOR INFERENCE: International cuisines have received the maximum average ratings in all 3 aspects.

But the point of concern is Mexican Cuisines. From the above inferences, it's clear that, Mexican cuisines are the most favourite among most of the users. But, they have received very low ratings in all 3 aspects. It's something the management has got to worry about improving.



#11 Finding the most popular cuisines among users

```
select Rcuisine, count(userID) from usercuisine group by Rcuisine order by count(userID)
Desc limit 1;
```

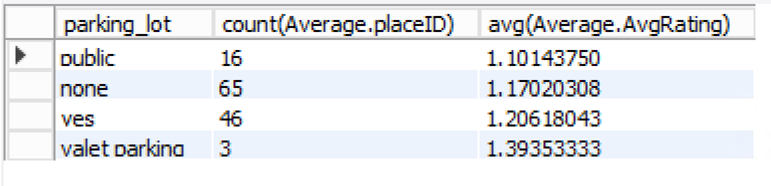
Rcuisine	count(userID)
Mexican	97

It's clear that Mexican is the most popular cuisine among users.

#32 Avg Ratings recieved as per parking lots used.

```
Select c.parking_lot, count(Average.placeID), avg(Average.AvgRating)
from chefmozparking as c,
          (select avg(rating) as AvgRating, placeID from rating_final group by placeID
  ) as Average
```

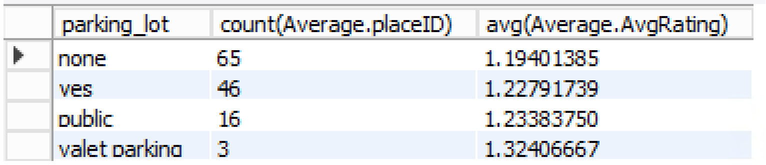
```
where c.placeID = Average.placeID
group by parking_lot
order by avg(Average.AvgRating);
```



	parking_lot	count(Average.placeID)	avg(Average.AvgRating)
▶	public	16	1.10143750
	none	65	1.17020308
	yes	46	1.20618043
	valet parking	3	1.39353333

#33 Avg food Ratings recieved as per parking lots used.

```
Select c.parking_lot, count(Average.placeID), avg(Average.AvgRating)
from chefmozparking as c,
            (select avg(food_rating) as AvgRating, placeID from rating_final group by placeID
   ) as Average
where c.placeID = Average.placeID
group by parking_lot
order by avg(Average.AvgRating);
```

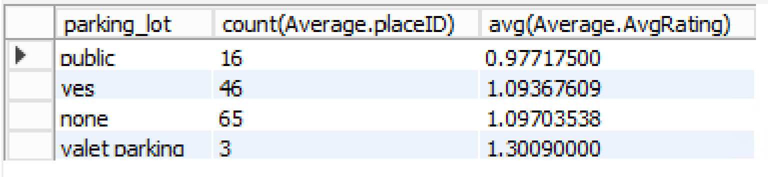


	parking_lot	count(Average.placeID)	avg(Average.AvgRating)
▶	none	65	1.19401385
	yes	46	1.22791739
	public	16	1.23383750
	valet parking	3	1.32406667

#34 Avg service Ratings recieved as per parking lots used.

```
Select c.parking_lot, count(Average.placeID), avg(Average.AvgRating)
from chefmozparking as c,

            (select avg(service_rating) as AvgRating, placeID from rating_final group by placeID
   ) as Average
where c.placeID = Average.placeID
group by parking_lot
order by avg(Average.AvgRating);
```



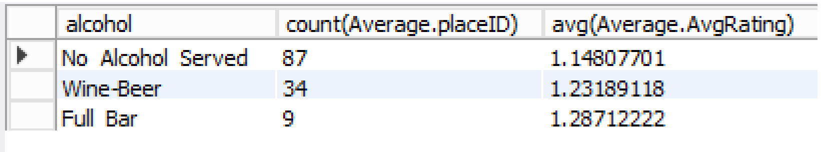
	parking_lot	count(Average.placeID)	avg(Average.AvgRating)
▶	public	16	0.97717500
	yes	46	1.09367609
	none	65	1.09703538
	valet parking	3	1.30090000

MAJOR INFERENCE: Places with Valet Parking has received the maximum average ratings in all 3 aspects.

# GEOPLACES AND RATING FINAL

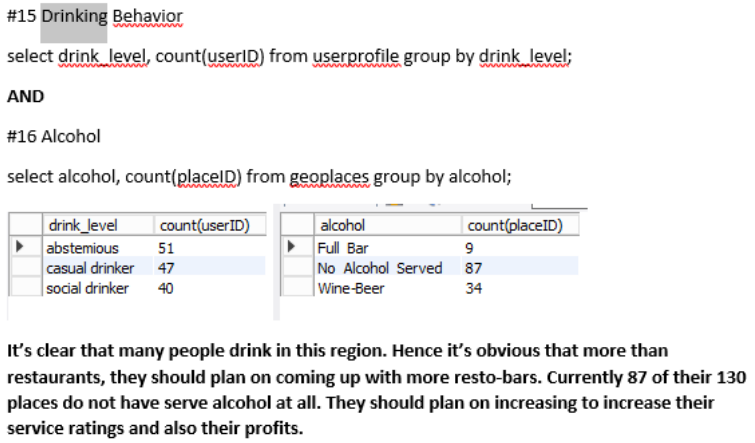
#35 Avg Ratings recieved as per alcohol

```
Select c.alcohol, count(Average.placeID), avg(Average.AvgRating)
from geoplaces as c,
        (select avg(rating) as AvgRating, placeID from rating_final group by placeID
  ) as Average
where c.placeID = Average.placeID
group by alcohol
order by avg(Average.AvgRating);
```



alcohol	count(Average.placeID)	avg(Average.AvgRating)
No Alcohol Served	87	1.14807701
Wine-Beer	34	1.23189118
Full Bar	9	1.28712222

Places that don't serve alcohol has got very less ratings which could be attributed to the fact that there are many people who drink in this region.



#15 Drinking Behavior

select drink_level, count(userID) from userprofile group by drink_level;

AND

#16 Alcohol

select alcohol, count(placeID) from geoplaces group by alcohol;

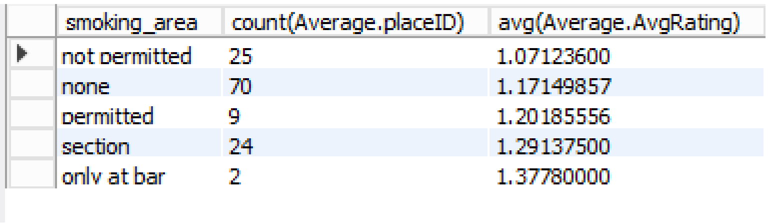
drink_level	count(userID)
abstemious	51
casual drinker	47
social drinker	40

alcohol	count(placeID)
Full Bar	9
No Alcohol Served	87
Wine-Beer	34

It's clear that many people drink in this region. Hence it's obvious that more than restaurants, they should plan on coming up with more resto-bars. Currently 87 of their 130 places do not have serve alcohol at all. They should plan on increasing to increase their service ratings and also their profits.

#36 Avg Ratings recieved as per smoking area

```
Select c.smoking_area, count(Average.placeID), avg(Average.AvgRating)
from geoplaces as c,
        (select avg(rating) as AvgRating, placeID from rating_final group by placeID
  ) as Average
where c.placeID = Average.placeID
group by smoking_area
order by avg(Average.AvgRating);
```

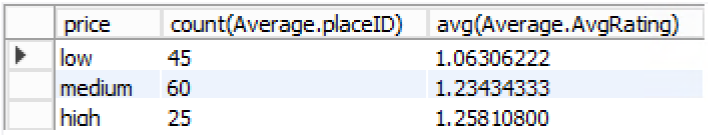


smoking_area	count(Average.placeID)	avg(Average.AvgRating)
not permitted	25	1.07123600
none	70	1.17149857
permitted	9	1.20185556
section	24	1.29137500
only at bar	2	1.37780000

Places that don’t allow smoking have got very less ratings

#37 Avg Ratings recieved as per price

```
Select c.price, count(Average.placeID), avg(Average.AvgRating)
from geoplaces as c,
          (select avg(rating) as AvgRating, placeID from rating_final group by placeID
   ) as Average
where c.placeID = Average.placeID
group by price
order by avg(Average.AvgRating);
```

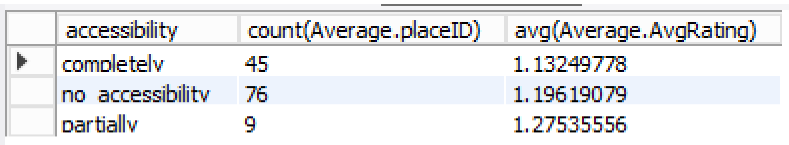


price	count(Average.placeID)	avg(Average.AvgRating)
low	45	1.06306222
medium	60	1.23434333
high	25	1.25810800

Places which are priced low have got very less ratings, which goes a long way in saying the management is actually compromising on quality by reducing the price; this should definitely be addressed.

#38 Avg Ratings recieved as per accessibility

```
Select c.accessibility, count(Average.placeID), avg(Average.AvgRating)
from geoplaces as c,
          (select avg(rating) as AvgRating, placeID from rating_final group by placeID
   ) as Average
where c.placeID = Average.placeID
group by accessibility
order by avg(Average.AvgRating);
```

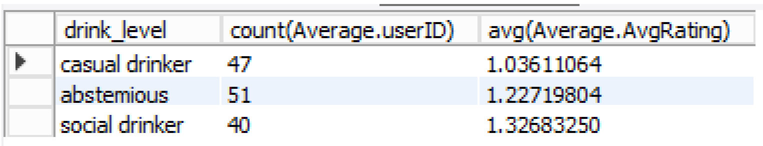


accessibility	count(Average.placeID)	avg(Average.AvgRating)
completely	45	1.13249778
no_accessibility	76	1.19619079
partially	9	1.27535556

#USERPROFILE AND RATING FINAL

#39 Avg Ratings given grouped by drinking behavior

```
Select c.drink_level, count(Average.userID), avg(Average.AvgRating)
from userprofile as c,
          (select avg(rating) as AvgRating, userID from rating_final group by userID
   ) as Average
where c.userID = Average.userID
group by drink_level
order by avg(Average.AvgRating);
```

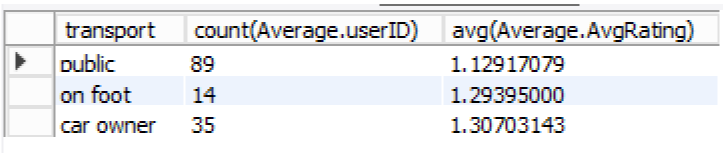


drink_level	count(Average.userID)	avg(Average.AvgRating)
casual drinker	47	1.03611064
abstemious	51	1.22719804
social drinker	40	1.32683250

Users who are casual drinkers have given less ratings to places.

#40 Avg Ratings given grouped by transportation

```
Select c.transport, count(Average.userID), avg(Average.AvgRating)
from userprofile as c,
        (select avg(rating) as AvgRating, userID from rating_final group by userID
  ) as Average
where c.userID = Average.userID
group by transport
order by avg(Average.AvgRating);
```

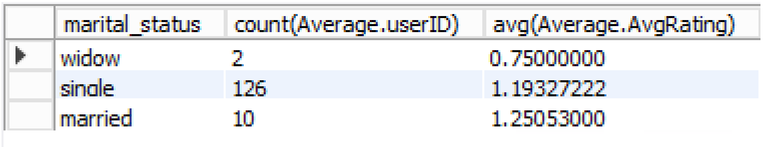


transport	count(Average.userID)	avg(Average.AvgRating)
public	89	1.12917079
on foot	14	1.29395000
car owner	35	1.30703143

Users who take public transport have given less ratings to places.

#41 Avg Ratings given grouped by marital status

```
Select c.marital_status, count(Average.userID), avg(Average.AvgRating)
from userprofile as c,
        (select avg(rating) as AvgRating, userID from rating_final group by userID
  ) as Average
where c.userID = Average.userID
group by marital_status
order by avg(Average.AvgRating);
```

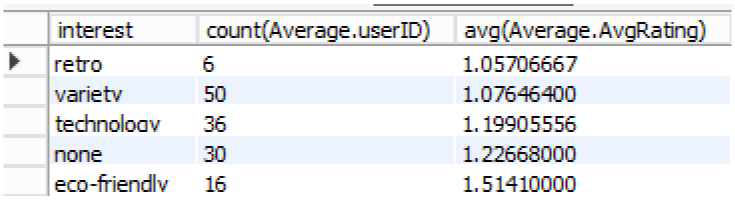


marital_status	count(Average.userID)	avg(Average.AvgRating)
widow	2	0.75000000
single	126	1.19327222
married	10	1.25053000

Married couples and widows have given less ratings compared to bachelors.

#42 Avg Ratings given grouped by interest

```
Select c.interest, count(Average.userID), avg(Average.AvgRating)
from userprofile as c,
        (select avg(rating) as AvgRating, userID from rating_final group by userID
  ) as Average
where c.userID = Average.userID
group by interest
order by avg(Average.AvgRating);
```

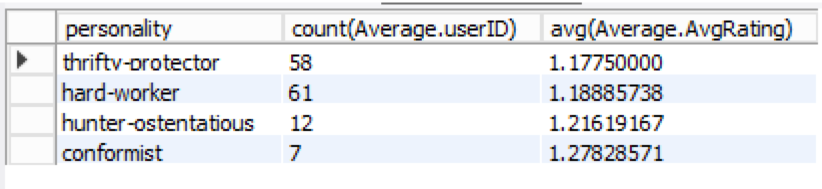


interest	count(Average.userID)	avg(Average.AvgRating)
retro	6	1.05706667
variety	50	1.07646400
technology	36	1.19905556
none	30	1.22668000
eco-friendly	16	1.51410000

Old people and people with varying interests have given less ratings

#43 Avg Ratings given grouped by personality

```
Select c.personality, count(Average.userID), avg(Average.AvgRating)
from userprofile as c,
        (select avg(rating) as AvgRating, userID from rating_final group by userID
  ) as Average
where c.userID = Average.userID
group by personality
order by avg(Average.AvgRating);
```

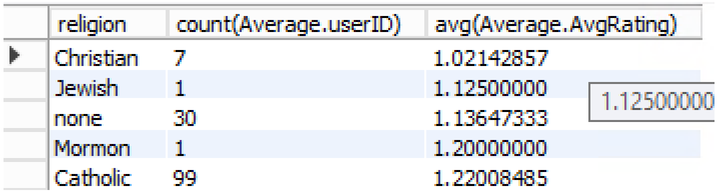


personality	count(Average.userID)	avg(Average.AvgRating)
thrifty-protector	58	1.17750000
hard-worker	61	1.18885738
hunter-ostentatious	12	1.21619167
conformist	7	1.27828571

Hard working people and thrifty protectors have given less ratings in general.

#44 Avg Ratings givven grouped by religion

```
Select c.religion, count(Average.userID), avg(Average.AvgRating)
from userprofile as c,
        (select avg(rating) as AvgRating, userID from rating_final group by userID
  ) as Average
where c.userID = Average.userID
group by religion
order by avg(Average.AvgRating);
```

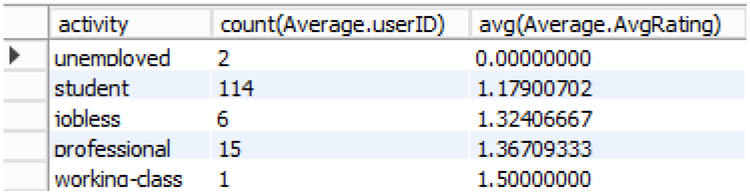


religion	count(Average.userID)	avg(Average.AvgRating)
Christian	7	1.02142857
Jewish	1	1.12500000
none	30	1.13647333
Mormon	1	1.20000000
Catholic	99	1.22008485

Christians have given less ratings in general.

#45 Avg Ratings given grouped by profession/activity

```
Select c.activity, count(Average.userID), avg(Average.AvgRating)
from userprofile as c,
        (select avg(rating) as AvgRating, userID from rating_final group by userID
   ) as Average
where c.userID = Average.userID
group by activity
order by avg(Average.AvgRating);
```

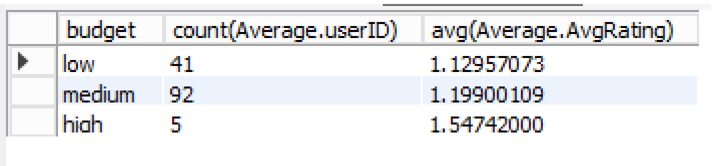


activity	count(Average.userID)	avg(Average.AvgRating)
unemployed	2	0.00000000
student	114	1.17900702
jobless	6	1.32406667
professional	15	1.36709333
working-class	1	1.50000000

Unemployed people and students have given less ratings in general.

#46 Avg Ratings given grouped by budget

```
Select c.budget, count(Average.userID), avg(Average.AvgRating)
from userprofile as c,
        (select avg(rating) as AvgRating, userID from rating_final group by userID
   ) as Average
where c.userID = Average.userID
group by budget
order by avg(Average.AvgRating);
```



budget	count(Average.userID)	avg(Average.AvgRating)
low	41	1.12957073
medium	92	1.19900109
high	5	1.54742000

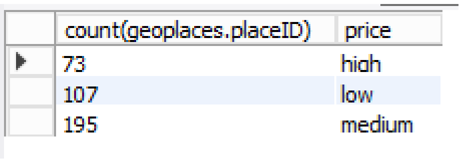
Low budget people have given less ratings in general.

# join using Three tables

#GEO AND USER

#47 48 49 Checking where the respective budget category people are going the most(high/med/low price rest.)

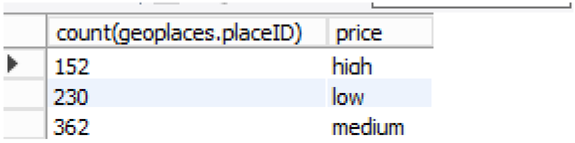
```
select count(geoplaces.placeID), price
from geoplaces, (select userID, budget from userprofile where budget = "low") a,
rating_final
where geoplaces.placeID = rating_final.placeID
and a.userID = rating_final.userID
group by price;
```



count(geoplaces.placeID)	price
73	high
107	low
195	medium

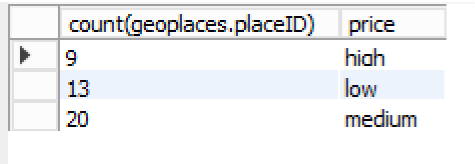
```
select count(geoplaces.placeID), price
```

```
from geoplaces, (select userID, budget from userprofile where budget = "medium") a,
rating_final
where geoplaces.placeID = rating_final.placeID
and a.userID = rating_final.userID
group by price;
```



count(geoplaces.placeID)	price
152	high
230	low
362	medium

```
select count(geoplaces.placeID), price
from geoplaces, (select userID, budget from userprofile where budget = "high") a,
rating_final
where geoplaces.placeID = rating_final.placeID
and a.userID = rating_final.userID
group by price;
```

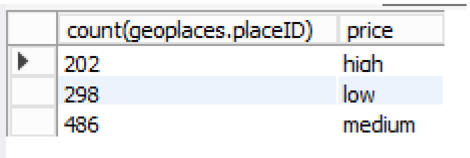


count(geoplaces.placeID)	price
9	high
13	low
20	medium

Major Inference: Medium priced restaurants are the most popular among all users across all three budget categories.

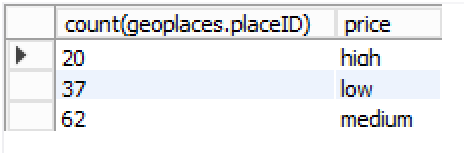
#50 51 52 53 Checking where the respective activity category are going the most(high/med/low price rest.)

```
select count(geoplaces.placeID), price
from geoplaces, (select userID, activity from userprofile where activity = "student") a,
rating_final
where geoplaces.placeID = rating_final.placeID
and a.userID = rating_final.userID
group by price;
```



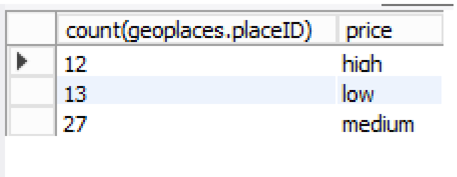
count(geoplaces.placeID)	price
202	high
298	low
486	medium

```
select count(geoplaces.placeID), price
from geoplaces, (select userID, activity from userprofile where activity = "professional")
a, rating_final
where geoplaces.placeID = rating_final.placeID
and a.userID = rating_final.userID
group by price;
```



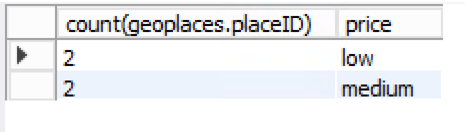
	count(geoplaces.placeID)	price
▶	20	high
	37	low
	62	medium

```
select count(geoplaces.placeID), price
from geoplaces, (select userID, activity from userprofile where activity = "unemployed"
or activity = "jobless") a, rating_final
where geoplaces.placeID = rating_final.placeID
and a.userID = rating_final.userID
group by price;
```



	count(geoplaces.placeID)	price
▶	12	high
	13	low
	27	medium

```
select count(geoplaces.placeID), price
from geoplaces, (select userID, activity from userprofile where activity = "working-
class") a, rating_final
where geoplaces.placeID = rating_final.placeID
and a.userID = rating_final.userID
group by price;
```

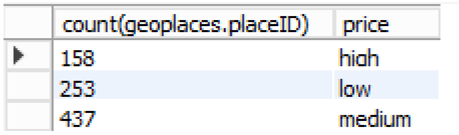


	count(geoplaces.placeID)	price
▶	2	low
	2	medium

Major Inference: Medium priced restaurants are the most popular across all four categories of activity.

#54 Checking where the respective transport category people are going the most(high/med/low price rest.)

```
select count(geoplaces.placeID), price
from geoplaces, (select userID, activity from userprofile where transport = "public" or
transport = "on foot") a, rating_final
where geoplaces.placeID = rating_final.placeID
and a.userID = rating_final.userID
group by price;
```



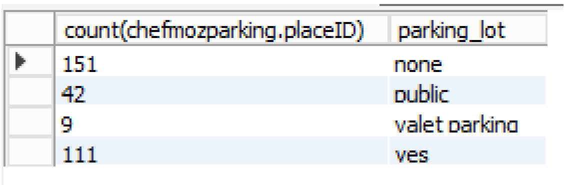
	count(geoplaces.placeID)	price
▶	158	high
	253	low
	437	medium

Major Inference: Medium priced restaurants are the most popular among people who use public transport

#USER AND PARKING

#55 Checking where the car owners are going to rest. with different parking lots

select count(chefmozparking.placeID), parking_lot
from chefmozparking, (select userID, budget from userprofile where transport = "car owner") a, rating_final
where chefmozparking.placeID = rating_final.placeID
and a.userID = rating_final.userID
group by parking_lot;



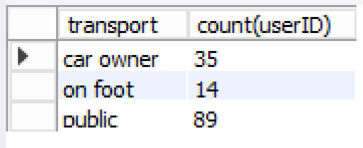
count(chefmozparking.placeID)	parking_lot
151	none
42	public
9	valet parking
111	yes

Major Inference: Among people who own car, they don't necessarily look for parking options at the restaurant which is evident from the table.

# INFERENCES without comparison

#56 transportation nature

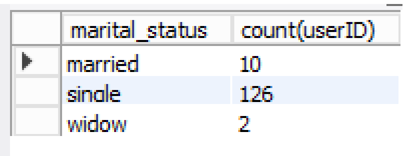
select transport, count(userID) from userprofile group by transport;



transport	count(userID)
car owner	35
on foot	14
public	89

#57 marital Status

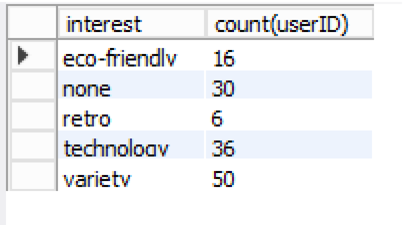
select marital_status, count(userID) from userprofile group by marital_status;



marital_status	count(userID)
married	10
single	126
widow	2

#58 interest variance

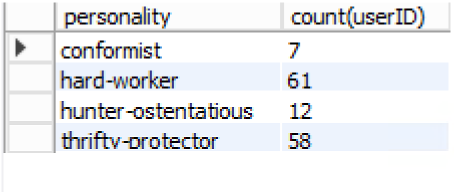
select interest, count(userID) from userprofile group by interest;



interest	count(userID)
eco-friendly	16
none	30
retro	6
technology	36
variety	50

#59 personality behavior

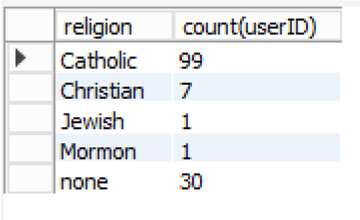
select personality, count(userID) from userprofile group by personality;



personality	count(userID)
conformist	7
hard-worker	61
hunter-ostentatious	12
thrifty-protector	58

#60 religion charectiristic

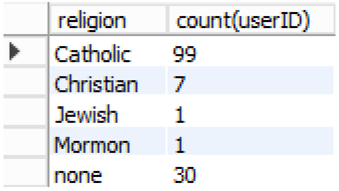
select religion, count(userID) from userprofile group by religion;



religion	count(userID)
Catholic	99
Christian	7
Jewish	1
Mormon	1
none	30

#61 Profession

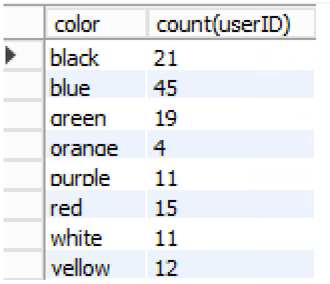
select activity, count(userID) from userprofile group by activity;



religion	count(userID)
Catholic	99
Christian	7
Jewish	1
Mormon	1
none	30

#62 color

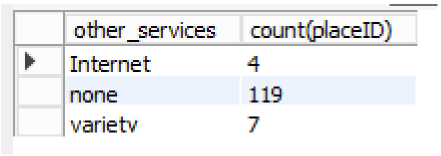
select color, count(userID) from userprofile group by color;



color	count(userID)
black	21
blue	45
green	19
orange	4
purple	11
red	15
white	11
yellow	12

#63 Other Services

select other_services, count(placeID) from geoplaces group by other_services;



other_services	count(placeID)
Internet	4
none	119
variety	7