Part 1

1. Profile the data by finding the total number of records for each of the tables below:

i. Attribute table = 10000
ii. Business table = 10000
iii. Category table = 10000
iv. Checkin table = 10000
v. elite_years table = 10000
vi. friend table = 10000
vii. hours table = 10000
viii. photo table = 10000
ix. review table = 10000
x. tip table = 10000
xi. user table = 10000

2. Find the total distinct records by either the foreign key or primary key for each table. If two foreign keys are listed in the table, please specify which foreign key.

i. Business = 10000
ii. Hours = 1562
iii. Category = 2643
iv. Attribute = 1115
v. Review = 10000
vi. Checkin = 493
vii. Photo = 10000
viii. Tip = 537(user_id)
ix. User = 10000
x. Friend = 11
xi. Elite_years = 2780

Note: Primary Keys are denoted in the ER-Diagram with a yellow key icon.

3. Are there any columns with null values in the Users table? Indicate "yes," or "no."

Answer: No

SQL code used to arrive at the answer:

```
Select *
From user
Where
--using coalesce function to check if any values of the user table is null

coalesce(id,name,review_count,useful,funny,cool,fans,average_stars,compliment_hot,compliment_more,compliment_profile,compliment_cute,compliment_list,compliment_note,compliment_plain,compliment_cool,compliment_funny,compliment_writer,compliment_photos,yelping_since) is null
```

4. For each table and column listed below, display the smallest (minimum), largest (maximum), and average (mean) value for the following fields:

i. Table: Review, Column: Stars

min: 1 max: 5 avg: 3.7082

ii. Table: Business, Column: Stars

min: 1 max: 5 avg: 3.6549

iii. Table: Tip, Column: Likes

min: 0 max: 2 avg: 0.0144

iv. Table: Checkin, Column: Count

min: 1   max: 53   avg:1.9414

v. Table: User, Column: Review_count

min: 0   max: 2000 avg:24.2995

5. List the cities with the most reviews in descending order:

SQL code used to arrive at answer:

```
select city, sum(review_count) as reviews
from business
group by city
order by reviews desc
```

order by review_count desc

Copy and Paste the Result Below:

```
+-----------------+---------+
| city            | reviews |
+-----------------+---------+
| Las Vegas       |   82854 |
| Phoenix         |   34503 |
| Toronto         |   24113 |
| Scottsdale      |   20614 |
| Charlotte       |   12523 |
| Henderson       |   10871 |
| Tempe           |   10504 |
| Pittsburgh      |    9798 |
| Montréal         |     9448 |
| Chandler         |     8112 |
| Mesa             |     6875 |
| Gilbert          |     6380 |
| Cleveland        |     5593 |
| Madison          |     5265 |
| Glendale         |     4406 |
| Mississauga      |     3814 |
| Edinburgh        |     2792 |
| Peoria           |     2624 |
| North Las Vegas  |     2438 |
| Markham          |     2352 |
| Champaign        |     2029 |
| Stuttgart        |     1849 |
| Surprise         |     1520 |
| Lakewood         |     1465 |
| Goodyear         |     1155 |
+------------------+----------+
```

6. Find the distribution of star ratings to the business in the following cities:

i. Avon

SQL code used to arrive at answer:

```
select stars,sum(review_count) as Review_count
from business
where city ='Avon'
group by stars
```

Copy and Paste the Resulting Table Below (2 columns â€“ star rating and count):

```
+--------+--------------+
| stars | Review_count |
+--------+--------------+
|    1.5 |           10 |
|    2.5 |            6 |
|    3.5 |           88 |
|    4.0 |           21 |
|    4.5 |           31 |
|    5.0 |            3 |
+--------+--------------+
```

ii. Beachwood

SQL code used to arrive at answer:

```
select stars,sum(review_count) as Review_count
from business
where city ='Beachwood'
group by stars
```

Copy and Paste the Resulting Table Below (2 columns â€" star rating and count):

```
+--------+--------------+
| stars | Review_count |
+--------+--------------+
|    2.0 |            8 |
|    2.5 |            3 |
|    3.0 |           11 |
|    3.5 |            6 |
|    4.0 |           69 |
|    4.5 |           17 |
|    5.0 |           23 |
+--------+--------------+
```

7. Find the top 3 users based on their total number of reviews:

SQL code used to arrive at answer:

```
select name, review_count
from user
order by review_count desc
limit 3
```

Copy and Paste the Result Below:

```
+--------+-------------+
| name   | review_count |
+--------+-------------+
| Gerald |         2000 |
| Sara   |         1629 |
| Yuri   |         1339 |
+--------+-------------+
```

8. Does posing more reviews correlate with more fans?

Please explain your findings and interpretation of the results:

Yes, As it could be noticed from the following results that most of the people who have fans proved to be posting more reviews but there is a contradiction to this also holds as few people who are posting more reviews have less fans unlike others, but if majority was considered then we could come to the above fact.

SQL Code

```
select name,review_count, fans
```

```
from user
order by fans desc
```

Resuls:

```
+-----------+--------------+------+
| name      | review_count | fans |
+-----------+--------------+------+
| Amy       |          609 |  503 |
| Mimi      |          968 |  497 |
| Harald    |         1153 |  311 |
| Gerald    |         2000 |  253 |
| Christine |          930 |  173 |
| Lisa      |          813 |  159 |
| Cat       |          377 |  133 |
| William   |         1215 |  126 |
| Fran      |          862 |  124 |
| Lissa     |          834 |  120 |
| Mark      |          861 |  115 |
| Tiffany   |          408 |  111 |
| bernice   |          255 |  105 |
| Roanna    |         1039 |  104 |
| Angela    |          694 |  101 |
| .Hon      |         1246 |  101 |
| Ben       |          307 |   96 |
| Linda     |          584 |   89 |
| Christina |          842 |   85 |
| Jessica   |          220 |   84 |
| Greg      |          408 |   81 |
| Nieves    |          178 |   80 |
| Sui       |          754 |   78 |
| Yuri      |         1339 |   76 |
| Nicole    |          161 |   73 |
+-----------+--------------+------+
```

9. Are there more reviews with the word "love" or with the word "hate" in them?

Answer:

Love

SQL code used to arrive at answer:

```
select count(*)
from review
where text like '%love%'

select count(*)
from review
where text like '%hate%'
```

10. Find the top 10 users with the most fans:

```
+-----------+------+
| name      | fans |
+-----------+------+
| Amy       |  503 |
| Mimi      |  497 |
| Harald    |  311 |
| Gerald    |  253 |
| Christine |  173 |
| Lisa      |  159 |
| Cat       |  133 |
| William   |  126 |
| Fran      |  124 |
| Lissa     |  120 |
+-----------+------+
```

SQL code used to arrive at answer:

```
select name,fans
```

```
from user
order by fans desc
limit 10
```

PART -2 :

1.) Do the two groups you chose to analyze have a different distribution of hours?

Ans:) Yes, both groups have a different distribution of hours and 2 to 3 groups seem to have a longer working hours on average, unlike the others.

2.) Do the two groups you chose to analyze have a different number of reviews?

Ans:) Yes, the 4 to 5 groups have more reviews as compared with the 2 to 3 reviews group.

3.). Can you infer anything from the location data provided between these two groups? Explain.

Ans:) Most of the 2 to 3-star ratings are from two locations whereas the 4 to 5 group has been diversified between various locations across the city.

SQL code used for analysis:

```
Select    b.name,h.hours,b.stars,b.postal_code,
count(b.postal_code) as con

from business b join

category c on b.id = c.business_id

join hours h

on h.business_id = b.id
```

```
where (b.city =  'Phoenix' and
c.category = 'Restaurants')

and

(b.stars between 2 and 3 )

or

(b.stars between 4 and 5)


group by b.stars,h.hours, b.postal_code

order by b.stars asc , b.postal_code desc
```

2. Group business based on the ones that are open and the ones that are closed. What differences can you find between the ones that are still open and the ones that are closed? List at least two differences and the SQL code you used to arrive at your answer.

i. Difference 1:

The average rating for the restaurants that are open now is slightly higher than that of the restaurants that are closed now.

ii. Difference 2:

The total review count could also tend to be more for the restaurants that are open now unlike the ones that are closed now.

Please find the result below:

```
+---------+---------------+--------------+
| is_open | average_rating | Total_reviews |
+---------+---------------+--------------+
|       0 |  3.52039473684 |        35261 |
|       1 |  3.67900943396 |       269300 |
+---------+---------------+--------------+
```

SQL code used for analysis:

```
select
b.is_open,
avg(b.stars) as average_rating,
SUM(review_count) as Total_reviews
from business b
group by b.is_open
```

3. For this last part of your analysis, you are going to choose the type of analysis you want to conduct on the Yelp dataset and are going to prepare the data for analysis.

i. Indicate the type of analysis you chose to do:

I tried to analyze customer reviews from a certain city to have a comprehensive understanding of the industry and the level of consumer satisfaction there.

During which I categorized the customer reviews into three groups (Positive, Negative, and Neutral) based on an analysis of the wording of their feedback.

ii. Write 1-2 brief paragraphs on the type of data you will need for your analysis and why you chose that data:

In this assessment, I selected the business table and reviews table to carry out my analysis because the first contains crucial information like city and location, while the latter has text that is supplied by the consumer.

Additionally, in order to filter the reviews based on those words, I downloaded some positive, negative, and neutral words from the internet.

iii. Output of your finished dataset:

```
+-----------------------------------------------+---------+-------+-------
+
| name                                          | city    | stars | score
|
+-----------------------------------------------+---------+-------+-------
+
| Bootleggers Modern American Smokehouse        | Phoenix |   4.0 |      1
|
| Anyplace Auto Repair                          | Phoenix |   4.5 |      1
|
| Mayflower Cab Company                         | Phoenix |   1.5 |    0.5
|
| KFC                                           | Phoenix |   2.0 |    0.5
|
| Lunch Box                                     | Phoenix |   2.5 |    0.5
|
| Mandarin Super Buffet                         | Phoenix |   2.5 |    0.5
|
| Scott Roofing Company                         | Phoenix |   2.5 |    0.5
|
| Showcase Honda                                | Phoenix |   2.5 |   0.5
|
| US Post Office                                | Phoenix |   2.5 |   0.5
|
| Burrito Bandito                               | Phoenix |   3.0 |   0.5
|
| Chipotle Mexican Grill                        | Phoenix |   3.0 |   0.5
|
| Fill A Seat Phoenix                           | Phoenix |   3.0 |   0.5
|
| Hotel San Carlos                              | Phoenix |   3.0 |   0.5
|
| Mellow Mushroom                               | Phoenix |   3.0 |   0.5
|
| Nichole Schaffer - State Farm Insurance Agent | Phoenix |   3.0 |   0.5
|
| Arizona Frybread                              | Phoenix |   3.5 |   0.5
|
| Autowits Auto Dealership                      | Phoenix |   3.5 |   0.5
|
| Corleone's                                    | Phoenix |   3.5 |   0.5
|
| Dubliner                                      | Phoenix |   3.5 |   0.5
|
| Harley's Italian Bistro                       | Phoenix |   3.5 |   0.5
|
| Herbal Nails & Spa -  - Happy Valley          | Phoenix |   3.5 |   0.5
|
| Julio G's Tatum                               | Phoenix |   3.5 |   0.5
|
| Lenny's Burger Shop                           | Phoenix |   3.5 |   0.5
|
| Lucky Strike                                  | Phoenix |   3.5 |   0.5
|
| Luke's of Chicago's                           | Phoenix |   3.5 |   0.5
|
+-------------------------------------------------+---------+-------+-------
+
(Output limit exceeded, 25 of 54 total rows shown)
```

iv. Provide the SQL code you used to create your final dataset:

```
select b.name,b.city,b.stars,

case

-- Assigning score to the customer feedback from 0 to 1 (0 -
Negitive review, 0.5 - Nuetral review and 1 - Postive review)
when r.text like '%it!!%' or '%turns!%' or '%blown%' or
'%good!%' or '%wow!%' or '%down!%' or '%amazingly%' or '%book!%'
or '%book!%' or '%next!%' or '%read!%' or '%movie!%' or
'%brilliantly%' or '%masterful%' or '%awesome%' or '%superb%' or
'%fabulous%' or '%it!%' or '%wait%' or '%wonderfully%' or
'%highly%' or '%turner!%' or '%incredible%' or '%toes%' or
'%fantastic%' or '%bed%' or '%masterfully%' or '%thank%' or
'%prime%' or '%loved%' or '%favour!%' or '%blew%' or
'%excellent%' or '%master%' or '%time!%' or '%chilling%' or
'%amazing%' or '%crafted%' or '%end!%' or '%roller%' or
'%story!%' or '%seat%' or '%loves%' or '%edge%' or '%gift%' or
'%twice%' or '%beautiful%' or '%insightful%' or '%layers%' or
'%constantly%' or '%wow%' or '%keeps%' or '%night%' or
'%coaster%' or '%pieces%' or '%terrific%' or '%sleep%' or
'%genius%' or '%predict%' or '%unpredictable%' or '%morning%' or
'%thrilling%' or '%reading!%' or '%intricate%' or '%complex%' or
'%fascinating%' or '%funny%' or '%immediately%' or '%enjoys%' or
'%woven%' or '%late%' or '%unfolds%' or '%minute%' or '%love%'
or '%beautifully%' or '%brilliant%' or '%surface%' or
```

```
'%perfect%' or '%witty%' or '%till%' or '%fast-paced%' or
'%intense%' THEN 1
when r.text like '%waste%' or '%poorly%' or '%wasted%' or
'%worst%' or '%ridiculous%' or '%dumb%' or '%badly%' or
'%awful%' or '%skipped%' or '%horrible%' or '%worse%' or
'%depressing%' or '%pathetic%' or '%terrible%' or '%stupid%' or
'%silly%' or '%boring%' or '%annoying%' or '%unrealistic%' or
'%bother%' or '%poor%' or '%contrived%' or '%unbelievable%' or
'%stuck%' or '%miserable%' or '%profanity%' or '%implausible%'
or '%selfish%' or '%sorry%' or '%mistake%' or '%unlikeable%' or
'%unlikable%' or '%struggled%' or '%bothered%' or '%mess%' or
'%quit%' or '%hated%' or '%death%' or '%book?%' or '%shallow%'
or '%negative%' or '%disliked%' or '%cliche%' or '%dull%' or
'%care%' or '%really?%' or '%annoyed%' or '%suspend%' or
'%pass%' or '%sadly%' or '%cared%' or '%skip%' or '%holes%' or
'%stopped%' or '%plain%' THEN 0
else 0.5
end as score

--Joining busniess and review tables
from review r join business b on r.business_id=b.id

--Analysis on a any city about the customer feedback(here we
considered Phoenix)
where city ='Phoenix'
group by b.stars,b.name, score

--Retrieing the data to analyse here we considered mid and high
review data for analysis to check customer happiness
having avg(score) >= 0.5
order by b.city asc, score desc
```