Part 1: Yelp Dataset Profiling and Understanding

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

The command used is

```
SELECT COUNT(*)
FROM tablename
```

i. Attribute table = 10,000
ii. Business table = 10,000
iii. Category table = 10,000
iv. Checkin table = 10,000
v. elite_years table = 10,000
vi. friend table = 10,000
vii. hours table = 10,000
viii. photo table = 10,000
ix. review table = 10,000
x. tip table = 10,000
xi. user table = 10,000

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 = 10,000

```
SELECT Count(distinct id)
FROM business
```

ii. Hours = 1562

```
SELECT Count(distinct business_id)
FROM hours
```

iii. Category = 2643

```
SELECT Count(distinct business_id)
FROM category
```

iv. Attribute = 1115

```
SELECT Count(distinct business_id)
FROM attribute
```

v. Review = 10,000

SELECT Count(distinct id)
FROM review

vi. Checkin = 493

SELECT Count(distinct business_id)
FROM checkin

vii. Photo = 10,000

SELECT Count(distinct id)
FROM photo

viii. Tip = 537

SELECT Count(distinct user_id)
FROM tip

ix. User = 10,000

SELECT Count(distinct id)
FROM user

x. Friend = 11

SELECT Count(distinct user_id)
FROM friend

xi. Elite_years = 2780

SELECT Count(distinct user_id)
FROM elite_years

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 id IS NULL OR
name IS NULL OR
review_count IS NULL OR
yelping_since IS NULL OR
useful IS NULL OR
funny IS NULL OR
cool IS NULL OR
fans IS NULL OR
average_stars IS NULL OR
compliment_hot IS NULL OR
compliment_more IS NULL OR
```

```
compliment_profile IS NULL OR
compliment_cute IS NULL OR
compliment_list IS NULL OR
compliment_note IS NULL OR
compliment_plain IS NULL OR
compliment_cool IS NULL OR
compliment_funny IS NULL OR
compliment_writer IS NULL OR
compliment_photos 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: 1.0 **max:** 5.0 **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 total_review
FROM business
```

```
GROUP BY city
ORDER BY total_review DESC
```

Copy and Paste the Result Below:

```
+-----------------+--------------+
| city            | total_review |
+-----------------+--------------+
| 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 |
+-----------------+--------------+
(Output limit exceeded, 25 of 362 total rows shown)
```

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 star_rating_count
FROM business
WHERE city = "Avon"
GROUP BY stars
```

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

```
+-------+-------------------+
| stars | star_rating_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 star_rating_count
FROM business
WHERE city = "Beachwood"
GROUP BY stars
```

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

```
+-------+-------------------+
| stars | star_rating_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,
   sum(review_count) AS review
FROM user
GROUP BY id
ORDER BY review DESC
LIMIT 3
```

Copy and Paste the Result Below:

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

8. Does posing more reviews correlate with more fans?

No,I can rhis with confident due to the result I got below as my query result. As per the table Gerald has highest review as 2000 but merely 253 fans which makes the average of 7 fans per review. Whereas Sara has reviews way less than Gerald i.e.1629 and only 50 fans and gives average of 32 fan per review to her. With the above observation we can conclude that posing more reviews doesn't correlate with more fans.

Please explain your findings and interpretation of the results:

```
SELECT
   name,
   sum(review_count) AS total_review,
   fans,
   sum(review_count)/fans AS review_per_fan
FROM user
GROUP BY id
ORDER BY total_review DESC
```

```
+-----------+--------------+------+----------------+
| name      | total_review | fans | review_per_fan |
+-----------+--------------+------+----------------+
| Gerald    |         2000 |  253 |              7 |
| Sara      |         1629 |   50 |             32 |
| Yuri      |         1339 |   76 |             17 |
| .Hon      |         1246 |  101 |             12 |
| William   |         1215 |  126 |              9 |
| Harald    |         1153 |  311 |              3 |
| eric      |         1116 |   16 |             69 |
| Roanna    |         1039 |  104 |              9 |
| Mimi      |          968 |  497 |              1 |
| Christine |          930 |  173 |              5 |
| Ed        |          904 |   38 |             23 |
| Nicole    |          864 |   43 |             20 |
| Fran      |          862 |  124 |              6 |
| Mark      |          861 |  115 |              7 |
| Christina |          842 |   85 |              9 |
| Dominic   |          836 |   37 |             22 |
```

```
| Lissa      |          834 |  120 |               6 |
| Lisa       |          813 |  159 |               5 |
| Alison     |          775 |   61 |              12 |
| Sui        |          754 |   78 |               9 |
| Tim        |          702 |   35 |              20 |
| L          |          696 |   10 |              69 |
| Angela     |          694 |  101 |               6 |
| Crissy     |          676 |   25 |              27 |
| Lyn        |          675 |   45 |              15 |
+------------+--------------+------+-----------------+
(Output limit exceeded, 25 of 10000 total rows shown)
```

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

Answer: Reviews with the word “love” is more than the reviews with the word “hate”.

SQL code used to arrive at answer:

```
SELECT
   feelings,
   count(*) AS total_count
FROM (SELECT
     CASE  WHEN text LIKE "%love%" THEN "love"
           WHEN text LIKE "%hate%" THEN "hate"
           ELSE "Others"
           END feelings
     FROM review)
GROUP BY feelings
ORDER BY total_count DESC
```

Output:

```
+----------+-------------+
| feelings | total_count |
+----------+-------------+
| Others   |        8042 |
| love     |        1780 |
| hate     |         178 |
+----------+-------------+
```

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

SQL code used to arrive at answer:

```
SELECT
   name,
   sum(fans) AS total_fan
FROM user
GROUP BY id
ORDER BY total_fan DESC
LIMIT 10
```

Copy and Paste the Result Below:

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

Part 2: Inferences and Analysis

1. Pick one city and category of your choice and group the businesses in that city or category by their overall star rating. Compare the businesses with 2-3 stars to the businesses with 4-5 stars and answer the following questions. Include your code.

I choose city as “Las Vegas” and category as “Food”

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

Yes, but not a huge difference. 2-3 stars has a total of 7 working hours and 4-5 stars has 6.

SQL code used for analysis:

```
SELECT  CASE WHEN stars >= 4 THEN "4-5 stars"
             WHEN stars >= 2 THEN "2-3 stars"
```

```
                    ELSE "below 2"
                    END star_rating,
               city,
               c.category,
               count(distinct business.id) AS company_count,
               count(h.hours) AS working_hours
          FROM business
          JOIN hours h ON business.id = h.business_id
          JOIN category c ON business.id = c.business_id
          WHERE city = "Las Vegas" AND c.category = "Food"
          GROUP BY star_rating
```

```
+-------------+-----------+----------+---------------+---------------+
| star_rating | city      | category | company_count | working_hours |
+-------------+-----------+----------+---------------+---------------+
| 2-3 stars   | Las Vegas | Food     |             1 |             7 |
| 4-5 stars   | Las Vegas | Food     |             1 |             6 |
+-------------+-----------+----------+---------------+---------------+
```

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

Yes, one with 4-5 star rating has more number of reviews as compare to one having 2-3 star ratings.

SQL code used for analysis:

```
          SELECT  CASE WHEN stars >= 4 THEN "4-5 stars"
                    WHEN stars >= 2 THEN "2-3 stars"
                    ELSE "below 2"
                    END star_rating,
               city,
               c.category,
               count(distinct business.id) AS company_count,
               sum(review_count) AS total_review
          FROM business
          JOIN category c ON business.id = c.business_id
          WHERE city = "Las Vegas" AND c.category = "Food"
          GROUP BY star_rating
```

```
+-------------+-----------+----------+---------------+--------------+
| star_rating | city      | category | company_count | total_review |
+-------------+-----------+----------+---------------+--------------+
| 2-3 stars   | Las Vegas | Food     |             1 |            6 |
| 4-5 stars   | Las Vegas | Food     |             1 |           30 |
+-------------+-----------+----------+---------------+--------------+
```

iii. Are you able to infer anything from the location data provided between these two groups? Explain.
No, every business is in a two different zip-code.

SQL code used for analysis:

```
SELECT  CASE WHEN stars >= 4 THEN "4-5 stars"
             WHEN stars >= 2 THEN "2-3 stars"
             ELSE "below 2"
             END star_rating,
       address,
       neighborhood,
       city,
       postal_code
FROM business
JOIN category c ON business.id = c.business_id
WHERE city = "Las Vegas" AND c.category = "Food"
ORDER BY star_rating
```

```
+-------------+-----------------------------+--------------+-----------+-----
---------+
| star_rating | address                     | neighborhood | city      |
postal_code |
+-------------+-----------------------------+--------------+-----------+-----
--------+
| 2-3 stars   | 3808 E Tropicana Ave        | Eastside     | Las Vegas |
89121       |
| 4-5 stars   | 8975 S Eastern Ave, Ste 3-B | Southeast    | Las Vegas |
89123       |
+-------------+-----------------------------+--------------+-----------+----
---------+
```

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:

Total number of reviews are significantly higher between still open and closed businesses.

ii. Difference 2:

The average star rating given are very closed to each other as 3.68 and 3.52. We can inspect from this record that businesses which got closed aren't solely due to the poor customer service or poor quality.

SQL code used for analysis:

```
SELECT  CASE    WHEN is_open = 1 THEN "STILL OPEN"
                WHEN is_open = 0 THEN "CLOSED"
                END status,
        count(distinct id) AS number_company,
        sum(review_count) AS total_review,
        round(avg(review_count),2) AS avg_review,
        round(avg(stars),2) AS avg_star_rating
FROM business
GROUP BY is_open
ORDER BY status DESC
```

status	number_company	total_review	avg_review	avg_star_rating
STILL OPEN	8480	269300	31.76	3.68
CLOSED	1520	35261	23.2	3.52

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.

Ideas for analysis include: Parsing out keywords and business attributes for sentiment analysis, clustering businesses to find commonalities or anomalies between them, predicting the overall star rating for a business, predicting the number of fans a user will have, and so on. These are just a few examples to get you started, so feel free to be creative and come up with your own problem you want to solve. Provide answers, in-line, to all of the following:

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

The analysis is to find out whether or not the business will stay open. Here in this analysis we didn't explicitly perform the analysis on the etx of the reviews each business recieved but would be going to be an exciting analysis to perform.

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

In order to perform this analysis well, we gonna need data such as id, number of reviews, star rating of business, hours open, and of course location from the business table and category from category table. We will need to count the numbers of companies within each category, the average stars given by the consumers to see how they perform, and the total reviews given to see if the data is relevant and ensure it's not biased.
Lastly, we're only going to look at categories with at least 10 companies and an average of 3.5+ stars to reduce any irrelevant data.

iii. Output of your finished dataset:

category	num_companies	avg_star_rating	total_reviews
Local Services	12	4.21	100
Active Life	10	4.15	131
Health & Medical	17	4.09	203
Home Services	16	4.0	94
Shopping	30	3.98	977
Beauty & Spas	13	3.88	119
American (Traditional)	11	3.82	1128
Food	23	3.78	1781
Bars	17	3.5	1322

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

```
SELECT  category,
       count(distinct id) AS num_companies,
       round(avg(stars),2) AS avg_star_rating,
       sum(review_count) total_reviews
FROM business
JOIN category ON business.id = category.business_id
GROUP BY category
HAVING avg_star_rating >= 3.5 AND num_companies >= 10
ORDER BY avg_star_rating DESC
```