Part 1: Yelp Dataset Profiling and Understanding

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

```
SELECT *
FROM (table);
```

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.

```
SELECT COUNT(DISTINCT(primary/foreign key))
FROM (table);
```

i. Business = 10000 (id) primary key
ii. Hours = 1562 (business_id) foreign key
iii. Category = 2643 (business_id) foreign key
iv. Attribute = 1115 (business_id) foreign key
v. Review = 10000 (id) primary key
= 8090 (business_id) foreign key
= 9581 (user_id) foreign key
vi. Checkin = 493 (business_id) foreign key
vii. Photo = 10000 (id) primary key
= 6493 (business_id) foreign key
viii. Tip = 3979 (business_id) foreign key
= 537 (user_id) foreign key
ix. User = 10000 (id) primary key
x. Friend = 11 (user_id) foreign key
xi. Elite_years = 2780 (user_id) foreign key

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

Answer: No

SQL code used to arrive at answer:

```
--Counts all the rows in a specific column ignoring NULL values
SELECT COUNT(id)
FROM user;
+-----------+
| COUNT(id) |
+-----------+
|     10000 |
+-----------+
```

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: 3.7082

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 SUM(review_count) AS total_reviews_by_city,
city
FROM business
GROUP BY city
ORDER BY total_reviews_by_city DESC;
```

Copy and Paste the Result Below:

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

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,
COUNT(stars) AS count
FROM business
WHERE city = 'Avon'
GROUP BY stars
ORDER BY stars ASC;
```

Copy and Paste the Resulting Table Below (2 columns – star rating and count):

```
+-------+-------+
| stars | count |
+-------+-------+
|   1.5 |     1 |
|   2.5 |     2 |
|   3.5 |     3 |
|   4.0 |     2 |
|   4.5 |     1 |
|   5.0 |     1 |
+-------+-------+
```

ii. Beachwood

SQL code used to arrive at answer:

```
SELECT stars,
COUNT(stars) AS count
FROM business
WHERE city = 'Beachwood'
GROUP BY stars
ORDER BY stars ASC;
```

Copy and Paste the Resulting Table Below (2 columns – star rating and count):

```
+-------+-------+
| stars | count |
+-------+-------+
|   2.0 |     1 |
|   2.5 |     1 |
|   3.0 |     2 |
|   3.5 |     2 |
|   4.0 |     1 |
|   4.5 |     2 |
|   5.0 |     5 |
+-------+-------+
```

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

SQL code used to arrive at answer:

```
SELECT id,
name,
SUM(review_count) AS Total_reviews
FROM user
GROUP BY id
ORDER BY Total_reviews DESC
LIMIT 3;
```

Copy and Paste the Result Below:

```
+------------------------+--------+---------------+
| id                     | name   | Total_reviews |
+------------------------+--------+---------------+
| -G7Zkl1wIWBBmD0KRy_sCw | Gerald |          2000 |
| -3s52C4zL_DHRK0ULG6qtg | Sara   |          1629 |
| -8lbUNlXVSoXqaRRiHiSNg | Yuri   |          1339 |
+------------------------+--------+---------------+
```

8. Does posting more reviews correlate with more fans?

Please explain your findings and interpretation of the results:

At first glance, it appears there is no correlation between fans and total review count. I included the SUM of fans for the top 25 users with the most reviews. However, when the results are filtered by ascending order for users with the least reviews, these users have significantly less fans. I believe there is a slight correlation between fans and total reviews, but there may be a more prominent correlation between fans and another factor.

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:

SQL code used to arrive at answer:

```
SELECT id,
name,
SUM(fans) AS Total_fans
FROM user
GROUP BY id
ORDER BY Total_fans DESC
LIMIT 10;
```

Copy and Paste the Result Below:

```
+------------------------+-----------+------------+
| id                     | name      | Total_fans |
+------------------------+-----------+------------+
| -9I98YbNQnLdAmcYfb324Q | Amy       |        503 |
| -8EnCioUmDygAbsYZmTeRQ | Mimi      |        497 |
| --2vR0DIsmQ6WfcSzKWigw | Harald    |        311 |
| -G7Zkl1wIWBBmD0KRy_sCw | Gerald    |        253 |
| -0IiMAZI2SsQ7VmyzJjokQ | Christine |        173 |
| -g3XIcCb2b-BD0QBCcq2Sw | Lisa      |        159 |
| -9bbDysuiWeo2VShFJJtcw | Cat       |        133 |
| -FZBTkAZEXoP7CYvRV2ZwQ | William   |        126 |
| -9da1xk7zgnnfO1uTVYGkA | Fran      |        124 |
| -lh59ko3dxChBSZ9U7LfUw | 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.

```
SELECT b.name,
b.city,
b.stars,
b.review_count,
h.hours
FROM business AS b LEFT JOIN hours AS h
ON b.id = h.business_id
WHERE b.city = 'Charlotte'
AND h.hours IS NOT NULL
GROUP BY b.name, h.hours
--HAVING stars >=2 AND stars <=3
HAVING stars >=4 AND stars <=5;
```

```
SELECT SUM(b.review_count) total_reviews,
b.city,
AVG(b.stars)
FROM business AS b
WHERE b.city = 'Charlotte'
AND stars >=2 AND stars <=3
--AND stars >=4 AND stars <=5;
```

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

There is not a large enough sample size of businesses with hours provided to conduct a proper analysis.
Charlotte 2-3 Star Business - No business hour data.
Charlotte 4-5 Star Business - 5 total businesses with hours data.

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

Charlotte 2-3 Star Business - 2865 total reviews
Charlotte 4-5 Star Business - 7023 total reviews

iii. Are you able to infer anything from the location data provided between these two groups? Explain.

SQL code used for analysis:

```
SELECT COUNT(id) AS num_of_bus,
neighborhood,
SUM(review_count) AS total_reviews,
AVG(stars) AS avg_stars
FROM business
WHERE city = 'Charlotte'
--AND stars >=2 AND stars <=3
AND stars >=4 AND stars <=5
GROUP BY neighborhood
ORDER BY num_of_bus DESC;
```

OUTCOME

```
+------------+-----------------+---------------+---------------+
| num_of_bus | neighborhood    | total_reviews |     avg_stars |
+------------+-----------------+---------------+---------------+
|         59 |                 |          2267 | 4.38983050847 |
|         13 | Elizabeth       |           262 | 4.30769230769 |
|         12 | Ballantyne      |           447 | 4.45833333333 |
|         12 | South Park      |           424 |         4.375 |
|         11 | Highland Creek  |           120 | 4.45454545455 |
|         11 | University City |           217 |           4.5 |
|         10 | Eastland        |           226 |          4.45 |
|          9 | NoDa            |           358 | 4.38888888889 |
|          9 | Starmount       |           237 | 4.27777777778 |
|          8 | Arboretum       |            63 |           4.5 |
|          8 | South End       |           290 |        4.5625 |
|          6 | Dilworth        |           547 | 4.33333333333 |
|          6 | First Ward      |           294 | 4.16666666667 |
|          6 | Plaza Midwood   |           495 | 4.33333333333 |
|          6 | Steele Creek    |            44 | 4.66666666667 |
|          6 | Uptown          |            75 |          4.25 |
|          4 | Cotswold        |            42 |         4.625 |
|          4 | Fourth Ward     |           112 |         4.375 |
|          4 | Sedgefield      |           109 |          4.25 |
|          3 | Biddleville     |            10 | 4.66666666667 |
|          3 | Derita          |            11 |           5.0 |
|          3 | Myers Park      |           157 | 4.16666666667 |
|          3 | Third Ward      |           182 | 4.66666666667 |
|          2 | Paw Creek       |            14 |          4.75 |
|          2 | Sherwood Forest |            16 |           4.0 |
```

```sql
SELECT COUNT(id) AS num_of_bus,
neighborhood,
SUM(review_count) AS total_reviews,
AVG(stars) AS avg_stars
FROM business
WHERE city = 'Charlotte'
AND stars >=2 AND stars <=3
--AND stars >=4 AND stars <=5
GROUP BY neighborhood
ORDER BY num_of_bus DESC;
```

OUTCOME

```
+------------+-----------------+---------------+---------------+
| num_of_bus | neighborhood    | total_reviews |     avg_stars |
+------------+-----------------+---------------+---------------+
|         45 |                 |           748 | 2.68888888889 |
|         12 | Ballantyne      |           269 | 2.83333333333 |
|         12 | Eastland        |           102 |         2.375 |
|         10 | South Park      |           453 |           2.8 |
|          7 | Derita          |           127 | 2.42857142857 |
|          7 | Elizabeth       |           145 | 2.85714285714 |
|          6 | First Ward      |           189 |          2.75 |
|          6 | Steele Creek    |            66 | 2.41666666667 |
|          6 | University City |            78 | 2.58333333333 |
|          5 | Cotswold        |            18 |           2.2 |
|          5 | Uptown          |           101 |           2.5 |
|          4 | Highland Creek  |            56 |          2.75 |
|          4 | Starmount       |           101 |          2.75 |
|          3 | Arboretum       |           128 | 2.66666666667 |
|          3 | North Charlotte |            23 |           2.0 |
|          3 | South End       |            15 | 2.83333333333 |
|          2 | Biddleville     |             6 |          2.25 |
|          1 | Fourth Ward     |             6 |           2.5 |
|          1 | Myers Park      |             4 |           3.0 |
|          1 | NoDa            |            12 |           3.0 |
|          1 | Plaza Midwood   |           194 |           3.0 |
|          1 | Quail Hollow    |             4 |           2.5 |
|          1 | Sherwood Forest |             3 |           2.5 |
|          1 | Third Ward      |            17 |           3.0 |
+------------+-----------------+---------------+---------------+
```

**I chose to sort the two groups of Star ratings (2-3 and 4-5) from Charlotte by neighborhood to analyze them by location information. The query results show the count of businesses in Charlotte by neighborhood in each group, total reviews, and average star rating.

Difference between 4/5 star and 2/3 star businesses by neighborhood (+/-)

- NoDa (+8)
- Highland Creek (+7)
- Plaza Midwood (+6)
- University City (+5)
- Starmount (+5)
- Arboretum (+5)
- South End (+5)
- Dilworth (+5)
- Elizabeth (+4)
- Fourth Ward (+3)
- Sedgefield (+3)
- South Park (+2)
- Uptown (+1)
- Biddleville (+1)
- Ballantyne (0)
- Steel Creek (0)
- Eastland (-2)
- Derita (-4)
- Cotswold (-1)
- North Charlotte (-3)

I also queried the total number of businesses in Charlotte grouped by neighborhood and sorted by average star rating descending. The results are filtered by only neighborhoods containing more than 10 businesses, and I removed the row containing no neighborhood data as seen below.

```
SELECT COUNT(id) AS num_of_bus,
neighborhood,
SUM(review_count) AS total_reviews,
AVG(stars) AS avg_stars
FROM business
WHERE city = 'Charlotte'
GROUP BY neighborhood
HAVING num_of_bus > 10
AND num_of_bus <> 130 --remove row containing no neighborhood data
ORDER BY avg_stars DESC;
```

OUTCOME

```
+------------+-----------------+----------------+----------------+
| num_of_bus | neighborhood    | total_reviews |      avg_stars |
+------------+-----------------+----------------+----------------+
|         11 | NoDa            |            373 | 4.18181818182 |
|         13 | South End       |            320 |            4.0 |
|         16 | Highland Creek  |            186 |        3.96875 |
|         21 | Elizabeth       |            560 | 3.78571428571 |
|         15 | Arboretum       |            214 |            3.7 |
|         26 | University City |            450 | 3.63461538462 |
|         32 | Ballantyne      |            896 |       3.609375 |
|         18 | Starmount       |            402 | 3.58333333333 |
|         18 | Steele Creek    |            290 | 3.52777777778 |
|         28 | South Park      |           1021 | 3.46428571429 |
|         14 | First Ward      |            792 | 3.32142857143 |
|         15 | Uptown          |            253 |            3.3 |
|         29 | Eastland        |            399 | 3.27586206897 |
|         11 | Derita          |            141 | 3.22727272727 |
|         14 | Cotswold        |            333 | 3.03571428571 |
+------------+-----------------+----------------+----------------+
```

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:

Average Star Rating

- Open Businesses: 3.68
- Closed Businesses: 3.52

ii. Difference 2:

Total Photos

- Open Businesses: 585
- Closed Businesses: 66

SQL code used for analysis:

```
SELECT AVG(stars)
FROM business
WHERE is_open = 0
--WHERE is_open = 1;
```

```
SELECT COUNT(p.id)
FROM business AS b
LEFT JOIN photo AS p
ON b.id = p.business_id
WHERE b.is_open = 0
--WHERE b.is_open = 1
AND p.id IS NOT NULL;
```

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:

How does a user's total number of reviews affect the way they rate businesses?

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

I first want to find the average number of reviews for all users. I want to know how many (of the 10,000 users) are above and below the average review amount. Then I will calculate average star rating for users above and below the average number of reviews to determine if the amount of reviews per user relates to the average stars given to businesses. I will also calculate how many users have only posted 1 review and their average star rating.

iii. Output of your finished dataset:

I combined the SQL code (along with comments) and the output of my codes together in the next section.

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

```
--Finding the average amount of reviews for all users.
SELECT AVG(review_count)
FROM user;
+-------------------+
| AVG(review_count) |
+-------------------+
|           24.2995 |
+-------------------+

--Finding how many indivual users with more than 25 reviews.
SELECT COUNT(DISTINCT id) AS users_plus25,
AVG(average_stars)
FROM user
WHERE review_count>25;
+--------------+--------------------+
| users_plus25 | AVG(average_stars) |
+--------------+--------------------+
|         1727 |      3.76341053851 |
+--------------+--------------------+

--Finding how many indivual users with less than 25 reviews
--and their average star rating.
SELECT COUNT(DISTINCT id) AS users_less25,
AVG(average_stars)
FROM user
WHERE review_count<25;
+--------------+--------------------+
| users_less25 | AVG(average_stars) |
+--------------+--------------------+
|         8209 |       3.6850785723 |
+--------------+--------------------+
--Finding how many indivual users with only 1 review
--and their average star rating.
SELECT COUNT(DISTINCT id),
AVG(average_stars)
FROM user
WHERE review_count = 1;
+--------------------+--------------------+
| COUNT(DISTINCT id) | AVG(average_stars) |
+--------------------+--------------------+
|               1815 |      3.54095316804 |
+--------------------+--------------------+

--Finding how many indivual users with only 1 review
--gave a 1 star rating
SELECT COUNT(DISTINCT id),
AVG(average_stars)
FROM user
WHERE review_count = 1
AND average_stars = 1;
+--------------------+--------------------+
| COUNT(DISTINCT id) | AVG(average_stars) |
+--------------------+--------------------+
|                473 |                1.0 |
+--------------------+--------------------+
```