Part 1: Yelp Dataset Profiling and Understanding

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 = PRIMARY KEY **id** :10000
ii. Hours = FOREIGN KEY **business_id:** 1562
iii. Category = FOREIGN KEY **business_id:** 2643
iv. Attribute = FOREIGN KEY **business_id:** 1115
v. Review = PRIMARY KEY **id:** 10000
vi. Checkin = FOREIGN KEY **business_id:** 493
vii. Photo = PRIMARY KEY **id:** 10000
viii. Tip = FOREIGN KEY **user_id:** 537
ix. User = PRIMARY KEY **id:** 10000
x. Friend = FOREIGN KEY **user_id:** 11
xi. Elite_years = FOREIGN KEY **user_id:** 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 answer:

```
SELECT count(name)

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_cool IS NULL OR
compliment_plain 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 **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 sum_of_reviews
FROM business
GROUP BY city
ORDER BY sum_of_reviews DESC;
```

Copy and Paste the Result Below:

```
+------------------+-----------------+
| city             | sum_of_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 |
+------------------+-----------------+
(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 DISTINCT stars, COUNT(stars) AS count
      FROM business
      WHERE city = 'Avon'
      GROUP BY stars
      ORDER BY stars;
```

Copy and Paste the Resulting Table Below (2 columns â€“ 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 DISTINCT stars, COUNT(stars) AS count
	FROM business
	WHERE city = 'Beachwood'
	GROUP BY stars
	ORDER BY stars;
```

Copy and Paste the Resulting Table Below (2 columns â€“ 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 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:

No, it doesn't. When we compare the average of reviews of the TOP 10 most popular users (more fans), we have 796 reviews and 240 fans on average. Then, when we do the same with the TOP 10 reviewers we have 1260 reviews per user with only 170 fans on average.
Inducing me to believe that there is no strong connection between number of reviews and number of fans.

```
	-- TOP 10 REVIEWERS

SELECT name, review_count, fans
FROM user
ORDER BY review_count DESC
LIMIT 10
+-----------+--------------+------+
| 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 |
+-----------+--------------+------+
```

```
-- TOP 10 MOST POPULAR
+-----------+--------------+------+
| 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 |
+-----------+--------------+------+
```

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(id)
FROM review
WHERE text LIKE "%hate%";
      (232)
```

```
SELECT COUNT(id)
FROM review
WHERE text LIKE "%love%";
      (1780)
```

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

SQL code used to arrive at answer:

```
SELECT name, fans
FROM user
ORDER BY fans DESC
LIMIT 10;
```

Copy and Paste the Result Below:

```
+-----------+------+
| name      | fans |
+-----------+------+
| 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.

Las Vegas and Shopping

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

YES. The 2-3 stars rating group is concentrated on the night shift.

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

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

YES. The best reviewed businesses are concentrated on Southeast and Spring Valley neighborhood, while the least rated business are concentrated on the Eastside.

SQL code used for analysis:

```sql
SELECT b.name,
      c.category,
      b.stars,
      COUNT(b.stars),
      h.hours,
      b.review_count,
      b.neighborhood
FROM business b
JOIN category c ON b.id = c.business_id
JOIN hours h ON h.business_id = b.id
WHERE city = 'Las Vegas'

            AND b.stars BETWEEN 4 AND 5

            OR b.stars BETWEEN 2 AND 3

GROUP BY b.name
ORDER BY b.stars 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:

There is over 5x more businesses opened than closed

ii. Difference 2:

Among opened businesses, rewiews with the word LOVE were more common (114) than those businesses who already closed the doors (12).

SQL code used for analysis:

```sql
SELECT count(id),
      CASE
            WHEN is_open = 1 THEN 'Open'
            ELSE 'Closed'
      END status
FROM business
GROUP BY is_open;
```

```sql
SELECT c.category, count(c.category) AS opened
FROM business b
JOIN category c ON b.id = c.business_id
WHERE is_open = 1
GROUP BY c.category
ORDER BY opened DESC;

SELECT c.category, count(c.category) AS closed
FROM business b
JOIN category c ON b.id = c.business_id
WHERE is_open = 0
GROUP BY c.category
ORDER BY opened DESC;
```

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:

Descriptive Analysis

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 task I put myself as a first-time entrepreneur who was still not sure about what kind of category he should put his time and money on.

In order to get a better grasp, he decided to combine in a single table a list of the most popular business categories in America. Analyzing how many businesses are open and the average of stars ratings and reviews.

iii. Output of your finished dataset:

```
+-------------------+------+-------------------+------------------------+
| TOP 10 Categories | Open | Average of Reviews | Average of Stars Rating |
+-------------------+------+-------------------+------------------------+
| Restaurants       |   53 |              71.0 |                    3.5 |
| Shopping          |   25 |              38.0 |                    4.0 |
| Food              |   20 |              79.0 |                    3.7 |
| Health & Medical  |   16 |              12.0 |                    4.2 |
| Home Services     |   15 |               6.0 |                    3.9 |
| Beauty & Spas     |   12 |              10.0 |                    3.8 |
| Nightlife         |   12 |              79.0 |                    3.6 |
| Bars              |   11 |              86.0 |                    3.6 |
| Active Life       |   10 |              13.0 |                    4.2 |
| Local Services    |   10 |               9.0 |                    4.3 |
+-------------------+------+-------------------+------------------------+
```

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

```sql
SELECT  c.category AS 'TOP 10 Categories',
        count(c.category) AS Open,
        ROUND(AVG(b.review_count)) AS 'Average of Reviews',
        ROUND(AVG(b.stars),1) AS 'Average of Stars Rating'
FROM business b
JOIN category c ON b.id = c.business_id
WHERE b.is_open=1
GROUP BY c.category
ORDER BY Open DESC
LIMIT 10
```