DATA SCIENTIST ROLE PLAY: PROFILING AND ANALYZING THE YELP DATASET

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 distinct record: 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 distinct record: 10000
vi. Checkin = Foreign Key (business_id): 493
vii. Photo = Primary Key distinct record: 10000
viii. Tip = Foreign Key (business_id): 3979; Foreign Key (user_id): 537
ix. User = Primary Key distinct record: 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

```
SELECT id, name, review_count, yelping_since, 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
FROM user
WHERE id=NULL OR name=NULL OR review_count=NULL OR yelping_since=NULL OR useful=NULL OR
  funny=NULL OR cool=NULL OR fans=NULL OR average_stars=NULL OR compliment_hot=NULL OR
  compliment_more=NULL OR compliment_profile=NULL OR compliment_cute=NULL OR
  compliment_list=NULL OR compliment_note=NULL OR compliment_plain=NULL OR
  compliment_cool=NULL OR compliment_funny=NULL OR compliment_writer=NULL OR
  compliment_photos=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:	max:	avg:
1	5	3.7082

ii. Table: Business, Column: Stars

min:	max:	avg:
1	5	3.6549

iii. Table: Tip, Column: Likes

min:	max:	avg:
0	2	0.0144

iv. Table: Checkin, Column: Count

min:	max:	avg:
1	53	1.9414

v. Table: User, Column: Review_count

min:	max:	avg:
0	2000	24.2995

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

```
SELECT city, SUM(review_count) AS Total_Review
FROM business
GROUP BY city
ORDER BY Total_Review DESC
```

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 |
+-----------------+---------------+
```

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

i. Avon

SELECT stars AS Star Rating, COUNT(DISTINCT(id)) AS Count
FROM business
WHERE city = 'Avon'
GROUP BY stars

```
+-------------+-------+
| Star Rating | Count |
+-------------+-------+
|         1.5 |     1 |
|         2.5 |     2 |
|         3.5 |     3 |
|         4.0 |     2 |
|         4.5 |     1 |
|         5.0 |     1 |
+-------------+-------+
```

ii. Beachwood

SELECT stars AS Star Rating, COUNT(DISTINCT(id)) AS Count
FROM business
WHERE city = ' Beachwood '
GROUP BY stars

```
+-------------+-------+
| Star Rating | 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: Gerald, Sara, Yuri

SELECT name, review_count
FROM user
ORDER BY review_count DESC LIMIT 3

```
+--------+--------------+
| 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:

SELECT name, SUM(fans), SUM(review_count)
FROM user
GROUP BY id
ORDER BY fans DESC

```
+-----------+-----------+-------------------+
| name      | SUM(fans) | SUM(review_count) |
+-----------+-----------+-------------------+
| Amy       |       503 |               609 |
| Mimi      |       497 |               968 |
| Harald    |       311 |              1153 |
| Gerald    |       253 |              2000 |
| Christine |       173 |               930 |
| Lisa      |       159 |               813 |
| Cat       |       133 |               377 |
| William   |       126 |              1215 |
| Fran      |       124 |               862 |
| Lissa     |       120 |               834 |
| Mark      |       115 |               861 |
| Tiffany   |       111 |               408 |
| bernice   |       105 |               255 |
| Roanna    |       104 |              1039 |
| Angela    |       101 |               694 |
| .Hon      |       101 |              1246 |
| Ben       |        96 |               307 |
| Linda     |        89 |               584 |
| Christina |        85 |               842 |
| Jessica   |        84 |               220 |
| Greg      |        81 |               408 |
| Nieves    |        80 |               178 |
| Sui       |        78 |               754 |
| Yuri      |        76 |              1339 |
| Nicole    |        73 |               161 |
+-----------+-----------+-------------------+
(Output limit exceeded, 25 of 10000 total rows shown)
```

The result shows that there is NO correlation between the number of fans and the number of posted reviews. For example, Amy has 503 fans with 609 reviews. However, in this selection the Gerald has

posted the maximum number of reviews (2000) but he has only 253 fans. I believe, there are other factors such as how long that person have been active in yelp, what kind of reviews does that person post etc.

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

Answer: love was used 1780 times and hate was used 232 times

Two separate codes were used.

Code for love

```
SELECT COUNT(*)
FROM review
WHERE text like '%love%'
```

```
+----------+
| COUNT(*) |
+----------+
|     1780 |
+----------+
```

Code for hate

```
SELECT COUNT(*)
FROM review
WHERE text like '%hate%'
```

```
+----------+
| COUNT(*) |
+----------+
|      232 |
+----------+
```

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

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

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

11. Is there a strong relationship (or correlation) between having a high number of fans and being listed as "useful" or "funny?" Out of the top 10 users with the highest number of fans, what percent are also listed as “useful” or “funny”?

Code 1: To determine people with most fans and their respective total funny and useful marking

```
SELECT name, SUM(fans) AS 'Total_Fans', SUM(useful), SUM(funny)
FROM user
GROUP BY id
ORDER BY Total_Fans DESC LIMIT 10
```

name	Total_Fans	SUM(useful)	SUM(funny)
Amy	503	3226	2554
Mimi	497	257	138
Harald	311	122921	122419
Gerald	253	17524	2324
Christine	173	4834	6646
Lisa	159	48	13
Cat	133	1062	672
William	126	9363	9361
Fran	124	9851	7606
Lissa	120	455	150

Code 2: To determine the people with most funny marking

```
SELECT name, SUM(funny) AS 'Total_Funny'
FROM user
GROUP BY id
ORDER BY Total_Funny DESC LIMIT 10
```

name	Total_Funny
Harald	122419
William	9361
Fran	7606
Christine	6646
W	6033
.Hon	5851
Alan	4567
Susie	3823
Jen	3164
Jim	2913

Code 3: To determine the people with most useful marking

```
SELECT name, SUM(useful) AS 'Total_Useful'
FROM user
GROUP BY id
ORDER BY Total_Useful DESC LIMIT 10
```

name	Total_Useful
Harald	122921
Gerald	17524
Susie	14703
Fran	9851
William	9363
.Hon	7850
W	6974

```
| Alan      |         5640 |
| Christine |         4834 |
| Mike      |         4656 |
+-----------+--------------+
```

The result shows that all of the top 10 users with the highest number of fans are listed as useful and funny. However, Amy, who has maximum fans (503) is not marked as the most funny or most useful. Harald, however, has only 311 fans but has been listed as funny and useful four times more than Amy. In the lists of top 10 users with highest number of listing as useful and funny, only Harald, Gerald, Fran, William, and Christine appear who also appear in the list of top 10 users with highest number of fans. Therefore, it seems there is a medium level correlation between having a high number of fans and being listed as "useful" or "funny".

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 picked Food and Las Vegas as the business category and city respectively.

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

Restaurants with 2-3 stars have longer hours compared to the highest rated ones

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

The highest rated ones have more reviews

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

All the high rated food businesses are located in the southeast neighborhood of the city and others are located in the Eastside neighborhood

```
SELECT b.stars, b.review_count, h.hours, b.neighborhood
FROM (business b INNER JOIN category c ON b.id = c.business_id) INNER JOIN hours h ON b.id =
h.business_id
WHERE c.category = 'Food' AND b.city = 'Las Vegas'
+-------+--------------+----------------------+--------------+
| stars | review_count | hours                | neighborhood |
+-------+--------------+----------------------+--------------+
|   4.0 |           30 | Monday|10:00-19:00    | Southeast    |
|   4.0 |           30 | Tuesday|10:00-19:00   | Southeast    |
|   4.0 |           30 | Friday|10:00-19:00    | Southeast    |
|   4.0 |           30 | Wednesday|10:00-19:00 | Southeast    |
|   4.0 |           30 | Thursday|10:00-19:00  | Southeast    |
|   4.0 |           30 | Saturday|10:00-19:00  | Southeast    |
|   2.5 |            6 | Monday|8:00-22:00     | Eastside     |
|   2.5 |            6 | Tuesday|8:00-22:00    | Eastside     |
|   2.5 |            6 | Friday|8:00-22:00     | Eastside     |
|   2.5 |            6 | Wednesday|8:00-22:00  | Eastside     |
|   2.5 |            6 | Thursday|8:00-22:00   | Eastside     |
|   2.5 |            6 | Sunday|8:00-22:00     | Eastside     |
|   2.5 |            6 | Saturday|8:00-22:00   | Eastside     |
+-------+--------------+----------------------+--------------+
```

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:

Closed businesses have received less reviews and lower stars compared to the open businesses

ii. Difference 2:

Closed businesses have received less number of check-ins compared to the open ones.

```
SELECT b.is_open,
MAX(b.stars) AS 'Max-Star',
AVG(b.stars) AS 'Avg-Stars',
MAX(b.review_count) AS 'Max-review',
COUNT(ch.count) AS 'CheckIn#'
FROM business b INNER JOIN checkin ch ON b.id = ch.business_id
GROUP BY b.is_open
```

```
+---------+----------+---------------+------------+----------+
| is_open | Max-Star |     Avg-Stars | Max-review | CheckIn# |
+---------+----------+---------------+------------+----------+
|       0 |      2.0 |           2.0 |         16 |       12 |
|       1 |      4.5 | 3.32128514056 |         27 |      498 |
+---------+----------+---------------+------------+----------+
```

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 wanted to see if the stars and check-in count have any correlation with negative words used in the actual review.

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

I will be using review and checkin data for this analysis as they contain the star rating, checkin count for businesses.

For this analysis, I have used positive words, such as, best, good, love, and like and for negative reviews, hate, terrible, and bad were used.

CODES FOR NEGATIVE SENTIMENT:

```
SELECT DISTINCT(r.business_id), r.stars, ch.count
FROM  checkin ch INNER JOIN review r ON r.business_id = ch.business_id
WHERE  text like '%terrible%'
ORDER BY r.stars DESC
```

```
+------------------------+-------+-------+
| business_id            | stars | count |
+------------------------+-------+-------+
| 3C5cDapsxHKckHn5-cmCzg |     2 |     3 |
| 3C5cDapsxHKckHn5-cmCzg |     2 |     2 |
| 3C5cDapsxHKckHn5-cmCzg |     2 |     1 |
| 3C5cDapsxHKckHn5-cmCzg |     2 |     4 |
+------------------------+-------+-------+
```

```
SELECT DISTINCT(r.business_id), r.stars, ch.count
FROM  checkin ch INNER JOIN review r ON r.business_id = ch.business_id
WHERE  text like '%hate%' OR text like '%bad%' OR text like '%terrible%'
ORDER BY r.stars DESC
```

```
+------------------------+-------+-------+
| business_id            | stars | count |
+------------------------+-------+-------+
| 3C5cDapsxHKckHn5-cmCzg |     2 |     3 |
| 3C5cDapsxHKckHn5-cmCzg |     2 |     2 |
| 3C5cDapsxHKckHn5-cmCzg |     2 |     1 |
| 3C5cDapsxHKckHn5-cmCzg |     2 |     4 |
+------------------------+-------+-------+
```

AS shown above, there are only four matching records were found which have the word "terrible" in their reviews. However, none of the matching records has the other three negative words. As expected, these businesses have low rating (2/5) and low checkin count

CODES FOR POSITIVE SENTIMENT:

```
SELECT DISTINCT(r.business_id), r.stars, ch.count
FROM  checkin ch INNER JOIN review r ON r.business_id = ch.business_id
WHERE  text like '%best%'
ORDER BY r.stars DESC
```

```
+------------------------+-------+-------+
| business_id            | stars | count |
+------------------------+-------+-------+
| qchCmKqTY46vKS1Y8zXdQQ |     5 |     1 |
| u_1EIg-c3d8XYfrmcdHX3w |     4 |     8 |
| u_1EIg-c3d8XYfrmcdHX3w |     4 |     2 |
| u_1EIg-c3d8XYfrmcdHX3w |     4 |     1 |
| u_1EIg-c3d8XYfrmcdHX3w |     4 |    15 |
| u_1EIg-c3d8XYfrmcdHX3w |     4 |     4 |
| u_1EIg-c3d8XYfrmcdHX3w |     4 |     9 |
| u_1EIg-c3d8XYfrmcdHX3w |     4 |     3 |
| u_1EIg-c3d8XYfrmcdHX3w |     4 |     5 |
| u_1EIg-c3d8XYfrmcdHX3w |     4 |     6 |
| u_1EIg-c3d8XYfrmcdHX3w |     4 |     7 |
| u_1EIg-c3d8XYfrmcdHX3w |     4 |    16 |
+------------------------+-------+-------+
```

```
SELECT DISTINCT(r.business_id), r.stars, ch.count
FROM  checkin ch INNER JOIN review r ON r.business_id = ch.business_id
WHERE  text like '%love%'
ORDER BY r.stars DESC
```

```
+------------------------+-------+-------+
| business_id            | stars | count |
+------------------------+-------+-------+
| 6RxtO_MxnLwZ7D7iYDE8Jg |     5 |     1 |
| 6RxtO_MxnLwZ7D7iYDE8Jg |     5 |     3 |
| 6RxtO_MxnLwZ7D7iYDE8Jg |     5 |     2 |
| 6RxtO_MxnLwZ7D7iYDE8Jg |     5 |     4 |
| 6RxtO_MxnLwZ7D7iYDE8Jg |     5 |     5 |
| 6RxtO_MxnLwZ7D7iYDE8Jg |     5 |     8 |
| lxsXOZNKoKPwXgkigdWPng |     4 |     3 |
| lxsXOZNKoKPwXgkigdWPng |     4 |     1 |
| lxsXOZNKoKPwXgkigdWPng |     4 |     2 |
+------------------------+-------+-------+
```

```
SELECT DISTINCT(r.business_id), r.stars, ch.count
FROM  checkin ch INNER JOIN review r ON r.business_id = ch.business_id
WHERE  text like '%like%'
ORDER BY r.stars DESC
```

```
+------------------------+-------+-------+
| business_id            | stars | count |
+------------------------+-------+-------+
| qchCmKqTY46vKS1Y8zXdQQ |     5 |     1 |
+------------------------+-------+-------+
```

```
SELECT DISTINCT(r.business_id), r.stars, ch.count
FROM  checkin ch INNER JOIN review r ON r.business_id = ch.business_id
WHERE  text like '%good%'
ORDER BY r.stars DESC
```

```
+------------------------+-------+-------+
| business_id            | stars | count |
+------------------------+-------+-------+
| ebgu1HjoE5S45U7oKyq4sw |     5 |     2 |
| ebgu1HjoE5S45U7oKyq4sw |     5 |     1 |
| ebgu1HjoE5S45U7oKyq4sw |     5 |     5 |
| ebgu1HjoE5S45U7oKyq4sw |     5 |     3 |
| ebgu1HjoE5S45U7oKyq4sw |     5 |     4 |
| 8C65DwwnazwYB2z9naHyaw |     5 |     2 |
| 8C65DwwnazwYB2z9naHyaw |     5 |     1 |
| 8C65DwwnazwYB2z9naHyaw |     5 |     3 |
| 8C65DwwnazwYB2z9naHyaw |     5 |     4 |
| u_1EIg-c3d8XYfrmcdHX3w |     4 |     8 |
| u_1EIg-c3d8XYfrmcdHX3w |     4 |     2 |
| u_1EIg-c3d8XYfrmcdHX3w |     4 |     1 |
| u_1EIg-c3d8XYfrmcdHX3w |     4 |    15 |
| u_1EIg-c3d8XYfrmcdHX3w |     4 |     4 |
| u_1EIg-c3d8XYfrmcdHX3w |     4 |     9 |
| u_1EIg-c3d8XYfrmcdHX3w |     4 |     3 |
| u_1EIg-c3d8XYfrmcdHX3w |     4 |     5 |
| u_1EIg-c3d8XYfrmcdHX3w |     4 |     6 |
| u_1EIg-c3d8XYfrmcdHX3w |     4 |     7 |
| u_1EIg-c3d8XYfrmcdHX3w |     4 |    16 |
| 5MXId9GCUlYgokbvqayzVg |     4 |     1 |
| 3C5cDapsxHKckHn5-cmCzg |     2 |     3 |
| 3C5cDapsxHKckHn5-cmCzg |     2 |     2 |
| 3C5cDapsxHKckHn5-cmCzg |     2 |     1 |
| 3C5cDapsxHKckHn5-cmCzg |     2 |     4 |
+------------------------+-------+-------+
```

As shown above, there were 35 matching records with positive sentiments. All of them have high star rating (4-5) except the last four with "good" in their reviews. However, the check in count does not show any direct correlation with reviews or the ratings. Although there are few businesses in the positive sentiment records with high check in counts most of them small number of check ins similar to the "terrible" ones. Although there are many other factors that need to be considered which are beyond the scope of this analysis, such as, the context of those words, this analysis shows that most often there is a correlation between positive review and star rating but no correlation between rating or reviews and check in count. Therefore based on this data it is difficult to conclude whether more people visit businesses with higher ratings and positive reviews only.