# Data Scientist: Role Play

## 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**

```
SELECT COUNT(*) FROM attribute;
```

```
+----------+
| COUNT(*) |
+----------+
|    10000 |
+----------+
```

ii. **Business table =10000**

```
SELECT COUNT(*) FROM business;
```

iii. **Category table =10000**

```
SELECT COUNT(*) FROM category;
```

iv. **Checkin table =10000**

```
SELECT COUNT(*) FROM checkin;
```

v. **elite_years table =10000**

```
SELECT COUNT(*) FROM elite_years;
```

vi. **friend table =10000**

```
SELECT COUNT(*) FROM friend;
```

vii. **hours table =10000**

```
SELECT COUNT(*) FROM hours;
```

viii. **photo table =10000**

```
SELECT COUNT(*) FROM photo;
```

ix. **review table =10000**

```
SELECT COUNT(*) FROM review;
```

x. **tip table =10000**

```
SELECT COUNT(*) FROM tip;
```

xi. **user table =10000**

```
SELECT COUNT(*) FROM user;
```

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.

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

i. **Business =10000**
id(Primary Key)

```
SELECT COUNT(DISTINCT id) FROM business ;
```

```
+--------------------+
| COUNT(DISTINCT id) |
+--------------------+
|              10000 |
+--------------------+
```

ii. **Hours =1562**
business_id(Foreign Key)

```
SELECT COUNT(DISTINCT business_id) FROM hours;
```

```
+----------------------------+
| COUNT(DISTINCT business_id) |
+----------------------------+
|                       1562 |
+----------------------------+
```

iii. **Category =2643**
business_id(Foreign Key)

```
SELECT COUNT(DISTINCT business_id) FROM category;
```

```
+----------------------------+
| COUNT(DISTINCT business_id) |
+----------------------------+
|                       2643 |
+----------------------------+
```

iv. **Attribute =1115**
business_id(Foreign Key)

```
SELECT COUNT(DISTINCT business_id) FROM attribute;
```

```
+----------------------------+
| COUNT(DISTINCT business_id) |
+----------------------------+
|                       1115 |
+----------------------------+
```

v. **Review =10000**
id(Primary Key)

```
SELECT COUNT(DISTINCT id) FROM review;
```

```
+--------------------+
| COUNT(DISTINCT id) |
+--------------------+
|              10000 |
+--------------------+
```

vi. **Checkin = 493**
business_id(Foreign Key)

```
SELECT COUNT(DISTINCT business_id) FROM checkin;
```

```
+----------------------------+
| COUNT(DISTINCT business_id) |
+----------------------------+
|                        493 |
+----------------------------+
```

vii. **Photo =10000**
id(Primary Key)

```
SELECT COUNT(DISTINCT id) FROM photo;
```

```
+--------------------+
| COUNT(DISTINCT id) |
+--------------------+
|              10000 |
+--------------------+
```

viii. **Tip =537**
**user_id(Foreign Key)**

```
SELECT COUNT(DISTINCT user_id) FROM tip;
```

```
+-------------------------+
| COUNT(DISTINCT user_id) |
+-------------------------+
|                     537 |
+-------------------------+
```

**Tip =3979**

**business_id(Foreign Key**)

```
SELECT COUNT(DISTINCT business_id) FROM tip;
```

```
+-----------------------------+
| COUNT(DISTINCT business_id) |
+-----------------------------+
|                        3979 |
+-----------------------------+
```

ix. **User = 10000**

id(Primary Key)

```
SELECT COUNT(DISTINCT id) FROM user;
```

```
+--------------------+
| COUNT(DISTINCT id) |
+--------------------+
|              10000 |
+--------------------+
```

x. **Friend = 11**

user_id(Foreign Key)

```
SELECT COUNT(DISTINCT user_id) FROM friend;
```

```
+-------------------------+
| COUNT(DISTINCT user_id) |
+-------------------------+
|                      11 |
+-------------------------+
```

xi. **Elite_years =2780**

user_id(Foreign Key)

```
SELECT COUNT(DISTINCT user_id) FROM elite_years;
```

```
+-------------------------+
| COUNT(DISTINCT user_id) |
+-------------------------+
|                    2780 |
+-------------------------+
```

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

Answer:**NO,None of the column have null values.**

SQL code used to arrive at 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

```
SELECT MIN(stars), MAX(stars), AVG(stars)
FROM review;
```

```
+------------+------------+------------+
| MIN(stars) | MAX(stars) | AVG(stars) |
+------------+------------+------------+
|          1 |          5 |     3.7082 |
+------------+------------+------------+
```

ii. Table: Business, Column: Stars

min:1.0 max:5.0 avg:3.6549

```
SELECT MIN(stars), MAX(stars), AVG(stars)
FROM business;
```

```
+------------+------------+------------+
| MIN(stars) | MAX(stars) | AVG(stars) |
+------------+------------+------------+
|        1.0 |        5.0 |     3.6549 |
+------------+------------+------------+
```

iii. Table: Tip, Column: Likes

min:0 max:2 avg:0.0144

```
SELECT MIN(likes), MAX(likes), AVG(likes)
FROM tip;
```

```
+------------+------------+------------+
| MIN(likes) | MAX(likes) | AVG(likes) |
+------------+------------+------------+
|          0 |          2 |     0.0144 |
+------------+------------+------------+
```

iv. Table: Checkin, Column: Count

min:1 max:53 avg:1.9414

```
SELECT MIN(count), MAX(count), AVG(count)
FROM checkin;
```

```
+------------+------------+------------+
| MIN(count) | MAX(count) | AVG(count) |
+------------+------------+------------+
|          1 |         53 |     1.9414 |
+------------+------------+------------+
```

v. Table: User, Column: Review_count

min:0 max:2000 avg:24.2995

```
SELECT MIN(review_count), MAX(review_count), AVG(review_count)
FROM user;
```

| MIN(review_count) | MAX(review_count) | AVG(review_count) |
|---|---|---|
| 0 | 2000 | 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_Reviews
FROM business
GROUP BY city
ORDER BY Total_Reviews DESC;
```

Copy and Paste the Result Below:

| city | Total_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 stars AS star_rating, SUM(review_count) AS count
FROM business
WHERE city = 'Avon'
GROUP BY stars;
```

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

| 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 AS star_rating, SUM(review_count) AS count
FROM business
WHERE city = 'Beachwood'
GROUP BY stars;
```

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

| 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 id,name, review_count
FROM user
ORDER BY review_count DESC
LIMIT 3;
```

Copy and Paste the Result Below:

| id | name | review_count |
|---|---|---|
| -G7Zkl1wIWBBmD0KRy_sCw | Gerald | 2000 |
| -3s52C4zL_DHRK0ULG6qtg | Sara | 1629 |
| -8lbUNlXVSoXqaRRiHiSNg | Yuri | 1339 |

8. Does posing more reviews correlate with more fans?

Please explain your findings and interpretation of the results:

SQL code used to arrive at answer:

```
SELECT name, review_count, fans
FROM user
ORDER BY review_count DESC;
```

Copy and Paste the Result Below:

| name | review_count | fans |
|---|---|---|
| Gerald | 2000 | 253 |
| Sara | 1629 | 50 |
| Yuri | 1339 | 76 |
| .Hon | 1246 | 101 |
| William | 1215 | 126 |
| Harald | 1153 | 311 |
| eric | 1116 | 16 |
| Roanna | 1039 | 104 |
| Mimi | 968 | 497 |
| Christine | 930 | 173 |
| Ed | 904 | 38 |
| Nicole | 864 | 43 |
| Fran | 862 | 124 |
| Mark | 861 | 115 |
| Christina | 842 | 85 |
| Dominic | 836 | 37 |
| Lissa | 834 | 120 |
| Lisa | 813 | 159 |
| Alison | 775 | 61 |
| Sui | 754 | 78 |
| Tim | 702 | 35 |
| L | 696 | 10 |
| Angela | 694 | 101 |
| Crissy | 676 | 25 |
| Lyn | 675 | 45 |

(Output limit exceeded, 25 of 10000 total rows shown)

**Interpretation:**
In the table , review count is arranged in descending order and there is relation between review count and number of fans.Hence , from above table we can conclude that review count and number of are not correlated .

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

Answer:**love-1780**
SQL code used to arrive at answer:

```
SELECT
(SELECT COUNT(id)
FROM review
WHERE text LIKE '%love%') AS love,
(SELECT COUNT(id)
```

```
FROM review
WHERE text LIKE '%hate%') AS hate;
```

```
+------+------+
| love | hate |
+------+------+
| 1780 |  232 |
+------+------+
```

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.

**The City and category I choose is Toronto and Restaurants.**

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

**Answer:No, when we look into the average days it is almost same.**

SQL code:

```
SELECT CASE WHEN stars >= 4.0 THEN '4-5 stars'
            WHEN stars >= 2.0 THEN '2-3 stars'
            ELSE 'below 2' END AS 'STARS',
       COUNT(DISTINCT business.id) AS id_count,
       COUNT(hours) AS open_days_total,
       COUNT(hours)*1.0 / COUNT(DISTINCT business.id)  AS open_days_avg
FROM ((business INNER JOIN hours ON business.id = hours.business_id)
     INNER JOIN category ON business.id = category.business_id)
WHERE city = 'Toronto' AND category.category ='Restaurants'
GROUP BY STARS
```

Table:

| STARS | id_count | open_days_total | open_days_avg |
|---|---|---|---|
| 2-3 stars | 3 | 21 | 7.0 |
| 4-5 stars | 3 | 19 | 6.33333333333 |

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

**Answer:Yes,the average review count is greater for 4-5 stars.**

SQL code:

```
SELECT CASE WHEN stars >= 4.0 THEN '4-5 stars'
            WHEN stars >= 2.0 THEN '2-3 stars'
            ELSE 'below 2' END AS 'STARS',
       COUNT(DISTINCT business.id) AS id_count,
       SUM(review_count) AS review_count_total,
       SUM(review_count)*1.0/COUNT(DISTINCT business.id)  AS
review_count_avg
FROM business INNER JOIN category ON business.id = category.business_id
WHERE city = 'Toronto' AND category.category ='Restaurants'
GROUP BY STARS
```

Table:

| STARS | id_count | review_count_total | review_count_avg |
|---|---|---|---|
| 2-3 stars | 4 | 89 | 22.25 |
| 4-5 stars | 5 | 206 | 41.2 |
| below 2 | 1 | 4 | 4.0 |

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

**Answer:We can see that from the table the 2-3 stars restaurant are more in the neighbourhood Downtown Core and 4-5 stars restaurant are more in the neighbourhood Willowdale.**

SQL code used for analysis:

```
SELECT CASE WHEN stars >= 4.0 THEN '4-5 stars'
            WHEN stars >= 2.0 THEN '2-3 stars'
            ELSE 'below 2' END AS 'stars',
            neighborhood, address, postal_code
FROM business b JOIN Category c
ON c.business_id = b.id
WHERE city = 'Toronto' AND category = 'Restaurants'
ORDER BY stars;
```

Table:

| stars | neighborhood | address | postal_code |
|---|---|---|---|
| 2-3 stars | Downtown Core | 260 Yonge Street | M4B 2L9 |
| 2-3 stars | Downtown Core | 389 Church Street | M5B 2E5 |
| 2-3 stars | Entertainment District | 270 Adelaide Street W | M5H 1X6 |
| 2-3 stars | | 3003 Bathurst Street | M6B |
| 4-5 stars | | 816 Saint Clair Avenue W | M6C 1B6 |
| 4-5 stars | High Park | 1669 Bloor Street W | M6P 1A6 |
| 4-5 stars | Etobicoke | 5084 Dundas Street W | M9A 1C2 |
| 4-5 stars | Willowdale | 7 Byng Avenue | M2N 5R6 |
| 4-5 stars | Niagara | 169 Niagara Street | M5V |
| below 2 | Willowdale | 5 Northtown Way, Unit 7 | M2N 7A1 |

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 business is more for open than closed.**

**open:8480**

**closed:1520**

**ii. Difference 2:The number of reviews and average stars is also greater for open.**

SQL code used for analysis:

```
SELECT is_open, COUNT(DISTINCT id) AS total_businesses,
    AVG(stars) AS avg_stars, SUM(review_count) AS total_reviews
FROM business
GROUP BY is_open;
```

Table:

| is_open | total_businesses | avg_stars | total_reviews |
|---|---|---|---|
| 0 | 1520 | 3.52039473684 | 35261 |
| 1 | 8480 | 3.67900943396 | 269300 |

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:

Which business has highest star count and the city ,the average star the business got?

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

I have calculated the average star and the star count of a business  and their corresponding city where it is most.From the below table we can see it.The Buffet has the highest star count which is in Las Vegas City. We can also see that Las Vegas City has the more number of business with more star count.

This analysis is appropriate for examining the distribution of star ratings and calculating the number of reviews for each business in various cities. This study will help you identify top-rated businesses and locations with strong review activity by providing insights into the popularity and consumer sentiment connected with different businesses in different cities.

iii. Output of your finished dataset:

| city | average_stars | star_count | name |
|---|---|---|---|
| Las Vegas | 3.8 | 10 | The Buffet |
| Las Vegas | 3.28571428571 | 7 | Diablo's Cantina |
| Gilbert | 5.0 | 5 | Joe's Farm Grill |
| Las Vegas | 2.4 | 5 | Rainforest Café |
| Las Vegas | 4.25 | 4 | Delmonico Steakhouse |
| Las Vegas | 4.25 | 4 | The Cheesecake Factory |
| Phoenix | 4.0 | 4 | Matt's Big Breakfast |
| Scottsdale | 5.0 | 4 | Portillo's Hot Dogs |
| Chandler | 4.33333333333 | 3 | Boba Tea House |
| Chandler | 3.33333333333 | 3 | El Zocalo Mexican Grill |

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

```
SELECT b.city, AVG(r.stars) AS average_stars, COUNT(r.stars) AS
star_count, b.name
FROM business b
JOIN review r ON b.id = r.business_id
GROUP BY b.city, b.name
HAVING COUNT(r.stars) > 0
ORDER BY star_count DESC
LIMIT 10;
```