```
/*
Data Scientist Role Play: Profiling and Analyzing the Yelp Dataset Coursera Worksheet

This is a 2-part assignment. In the first part, you are asked a series of questions that will help you profile
and understand the data just like a data scientist would.
For this first part of the assignment, you will be assessed both on the correctness of your findings,
as well as the code you used to arrive at your answer.
You will be graded on how easy your code is to read, so remember to use proper formatting
and comments where necessary.

In the second part of the assignment, you are asked to come up with your own inferences and analysis of the data
for a particular research question you want to answer. You will be required to prepare the dataset for the analys
is you choose to do. As with the first part, you will be graded, in part, on how easy your code is to read,
so use proper formatting and comments to illustrate and communicate your intent as required.

For both parts of this assignment, use this "worksheet." It provides all the questions you are being asked,
and your job will be to transfer your answers and SQL coding where indicated into this worksheet so that your
peers can review your work. You should be able to use any Text Editor (Windows Notepad, Apple TextEdit, Notepad ++,
Sublime Text, etc.) to copy and paste your answers. If you are going to use Word or some other page layout application,
just be careful to make sure your answers and code are lined appropriately.
In this case, you may want to save as a PDF to ensure your formatting remains intact for you reviewer.
*/

/*
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 = 10000
ii. Hours = 1562 business_id (FK)
iii. Category = 2643 business_id (FK)
iv. Attribute = 1115 business_id (FK)
v. Review = 10000
vi. Checkin = 493 business_id (FK)
vii. Photo = 10000
viii. Tip = 537 user_id (FK)
ix. User = 10000
x. Friend = 11 user?id (FK)
xi. Elite_years = 2780 user?id (FK)

/*
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 id, name, review_count FROM user
--COMPARING ALL ATTRIBUTES USING OR STATEMENT
WHERE
    id IS NOT NULL AND
    name IS NOT NULL AND
    review_count IS NOT NULL AND
    yelping_since IS NOT NULL AND
    useful IS NOT NULL AND
    funny IS NOT NULL AND
    cool IS NOT NULL AND
    fans IS NOT NULL AND
    average_stars IS NOT NULL AND
    compliment_hot IS NOT NULL AND
    compliment_more IS NOT NULL AND
    compliment_profile IS NOT NULL AND
    compliment_cute IS NOT NULL AND
    compliment_list IS NOT NULL AND
    compliment_note IS NOT NULL AND
    compliment_plain IS NOT NULL AND
    compliment_cool IS NOT NULL AND
    compliment_funny IS NOT NULL AND
    compliment_writer IS NOT NULL AND
    compliment_photos IS NOT 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) FROM Business
GROUP BY city ORDER BY SUM(review_count) DESC

    /*
    Copy and Paste the Result Below:
```

```
    */

/*
+-----------------+-------------------+
| city            | SUM(review_count) |
+-----------------+-------------------+
| 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, COUNT(stars) FROM Business
WHERE city = 'Avon' GROUP BY stars
ORDER BY stars

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

/*
+-------+--------------+
| stars | COUNT(stars) |
+-------+--------------+
|   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) FROM Business
```

```sql
WHERE city = 'Beachwood' GROUP BY stars
ORDER BY stars

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

/*
+-------+--------------+
| stars | COUNT(stars) |
+-------+--------------+
|   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:
*/

--I propose a new column that I called ratio_reviews_fans (review_count/fans),
--with this ratio we can find out a numeric value that correlates this two attributes for all
--registers and hence We can find the best.

SELECT name, review_count, fans,
    CASE review_count/fans
    --If We have a zero division case then only name ratio as zero
        --I use CAST function because I want a REAL number, not INTEGER.
        WHEN fans > 0 THEN CAST(review_count AS REAL)/fans
        ELSE 0
    END ratio_reviews_fans
FROM user
ORDER BY ratio_reviews_fans DESC
LIMIT 3

/*
+---------+--------------+------+--------------------+
| name    | review_count | fans | ratio_reviews_fans |
+---------+--------------+------+--------------------+
```

```
| Mimi    |          968 |  497 |       1.9476861167 |
| Andrew  |          115 |   60 |      1.91666666667 |
| Richard |           11 |    6 |      1.83333333333 |
+---------+--------------+------+--------------------+
*/

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

/*
    Answer:
*/

    There are more reviews with word "love".

/*
    SQL code used to arrive at answer:
*/

SELECT COUNT(text) AS reviews_with_love_word from review
--Looking for three cases
WHERE (LOWER(text) LIKE "%love%") OR
--Nested OR in order to evaluate two other cases
((LOWER(text) LIKE "%love") OR
(LOWER(text) LIKE "love%"))

/*
+------------------------+
| reviews_with_love_word |
+------------------------+
|                   1780 |
+------------------------+
*/

SELECT COUNT(text) AS reviews_with_hate_word from review
--Looking for three cases
WHERE (LOWER(text) LIKE "%hate%") OR
--Nested OR in order to evaluate two other cases
((LOWER(text) LIKE "%hate") OR
(LOWER(text) LIKE "hate%"))

/*
+------------------------+
| reviews_with_hate_word |
+------------------------+
|                    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.
*/

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

YES

/*
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.
*/
A first sight is not possible distinguish any difference between two groups because both have similar
coordinates and ZIP. However it could be possible find some insights from  this data plotting on a map the
coordinates of businesses and set all ZIP areas.

/*
SQL code used for analysis:
*/

Item i.

--With this query we can retrieve distribution of hours regarding 2-3 stars group of businesses from "Las Vegas"

SELECT name, stars, hours FROM
(
    --Retrive all registers in hours table that match with Las Vegas city
    --in business table
    --This way to retrieve a new table let save us a lot of time when we perform
    --a CROSS JOIN or CARTESIAN JOIN
    --because our query is filtered from begining
    SELECT * FROM hours
    WHERE business_id IN
    (
        SELECT id FROM business
        WHERE city = 'Las Vegas'
    )
)
--Perform a Cartesian join in order to retrieve a new table with all combinations
--with our previous table and business table with city = 'Las Vegas'
CROSS JOIN
(
    SELECT id, name, stars, city FROM business
    WHERE city = 'Las Vegas'
)
--This condition is established in order to retrive unique registers
--with 2 or 3 stars
WHERE business_id=id AND (stars BETWEEN 2 AND 3)

/*
+-----------+-------+----------------------+
```

```
| name      | stars | hours                |
+-----------+-------+----------------------+
| Wingstop  |   3.0 | Monday|11:00-0:00    |
| Wingstop  |   3.0 | Tuesday|11:00-0:00   |
| Wingstop  |   3.0 | Friday|11:00-0:00    |
| Wingstop  |   3.0 | Wednesday|11:00-0:00 |
| Wingstop  |   3.0 | Thursday|11:00-0:00  |
| Wingstop  |   3.0 | Sunday|11:00-0:00    |
| Wingstop  |   3.0 | Saturday|11:00-0:00  |
| Walgreens |   2.5 | Monday|8:00-22:00    |
| Walgreens |   2.5 | Tuesday|8:00-22:00   |
| Walgreens |   2.5 | Friday|8:00-22:00    |
| Walgreens |   2.5 | Wednesday|8:00-22:00 |
| Walgreens |   2.5 | Thursday|8:00-22:00  |
| Walgreens |   2.5 | Sunday|8:00-22:00    |
| Walgreens |   2.5 | Saturday|8:00-22:00  |
+-----------+-------+----------------------+
*/


--With this query we can retrieve distribution of hours regarding 4-5 stars group of businesses from "Las Vegas"

SELECT name, stars, hours, business_id, id FROM
(
    --Retrive all registers in hours table that match with Las Vegas city
    --in business table
    --This way to retrieve a new table let save us a lot of time when we perform
    --a CROSS JOIN or CARTESIAN JOIN
    --because our query is filtered from begining
    SELECT * FROM hours
    WHERE business_id IN
    (
        SELECT id FROM business
        WHERE city = 'Las Vegas'
    )
)
--Perform a Cartesian join in order to retrieve a new table with all combinations
--with our previous table and business table with city = 'Las Vegas'
CROSS JOIN
(
    SELECT id, name, stars, city FROM business
    WHERE city = 'Las Vegas'
)
--This condition is established in order to retrive unique registers
--with 4 or 5 stars
WHERE business_id=id AND (stars BETWEEN 4 AND 5)

/*
+-------------------------------+-------+----------------------+
| name                          | stars | hours                |
+-------------------------------+-------+----------------------+
| Motors & More                 |   5.0 | Monday|7:00-17:00    |
| Motors & More                 |   5.0 | Tuesday|7:00-17:00   |
| Motors & More                 |   5.0 | Friday|7:00-17:00    |
| Motors & More                 |   5.0 | Wednesday|7:00-17:00 |
| Motors & More                 |   5.0 | Thursday|7:00-17:00  |
| Motors & More                 |   5.0 | Saturday|8:00-12:00  |
| Red Rock Canyon Visitor Center |   4.5 | Monday|8:00-16:30    |
| Red Rock Canyon Visitor Center |   4.5 | Tuesday|8:00-16:30   |
| Red Rock Canyon Visitor Center |   4.5 | Friday|8:00-16:30    |
| Red Rock Canyon Visitor Center |   4.5 | Wednesday|8:00-16:30 |
| Red Rock Canyon Visitor Center |   4.5 | Thursday|8:00-16:30  |
| Red Rock Canyon Visitor Center |   4.5 | Sunday|8:00-16:30    |
| Red Rock Canyon Visitor Center |   4.5 | Saturday|8:00-16:30  |
| Sweet Ruby Jane Confections   |   4.0 | Monday|10:00-19:00   |
| Sweet Ruby Jane Confections   |   4.0 | Tuesday|10:00-19:00  |
| Sweet Ruby Jane Confections   |   4.0 | Friday|10:00-19:00   |
| Sweet Ruby Jane Confections   |   4.0 | Wednesday|10:00-19:00 |
| Sweet Ruby Jane Confections   |   4.0 | Thursday|10:00-19:00  |
| Sweet Ruby Jane Confections   |   4.0 | Saturday|10:00-19:00  |
| Vue at Centennial             |   4.0 | Monday|9:00-17:00    |
| Vue at Centennial             |   4.0 | Tuesday|9:00-17:00   |
| Vue at Centennial             |   4.0 | Friday|9:00-17:00    |
```

```
| Vue at Centennial                |   4.0 | Wednesday|9:00-17:00  |
| Vue at Centennial                |   4.0 | Thursday|9:00-17:00   |
| Vue at Centennial                |   4.0 | Sunday|10:00-16:00    |
+----------------------------------+-------+----------------------+
(Output limit exceeded, 25 of 55 total rows shown)
*/
```

Item ii.

```
--Total number of reviews from businesses with 4 or 5 stars in Las Vegas city
SELECT SUM(review_count) FROM business
WHERE city = 'Las Vegas' AND (stars BETWEEN 4 AND 5)

/*
+-------------------+
| SUM(review_count) |
+-------------------+
|             46952 |
+-------------------+
*/

--Total number of reviews from businesses with 2 or 3 stars in Las Vegas city
SELECT SUM(review_count) FROM business
WHERE city = 'Las Vegas' AND (stars BETWEEN 2 AND 3)

/*
+-------------------+
| SUM(review_count) |
+-------------------+
|             15265 |
+-------------------+
*/
```

Items iii.

```
--Locations and ZIP's of businesses with 4 or 5 stars in Las Vegas city
SELECT city, name, stars, review_count, postal_code, latitude, longitude
FROM business
WHERE city = 'Las Vegas' AND (stars BETWEEN 4 AND 5)

/*
+-----------+-------------------------------------------------+-------+--------------+-------------+----------+-----------+
| city      | name                                            | stars | review_count | postal_code | latitude | longitude |
+-----------+-------------------------------------------------+-------+--------------+-------------+----------+-----------+
| Las Vegas | Delmonico Steakhouse                            |   4.0 |         1389 | 89109       |  36.1232 |  -115.169 |
| Las Vegas | Double Play Sports Bar                          |   4.0 |            7 | 89123       |  36.0167 |  -115.173 |
| Las Vegas | Dial Carpet Cleaning                            |   5.0 |           13 | 89102       |  36.1309 |  -115.191 |
| Las Vegas | Pinnacle Restoration                            |   5.0 |            7 | 89148       |  36.0704 |  -115.313 |
| Las Vegas | Nacho Mobile Auto Repair                        |   4.5 |           14 | 89109       |  36.1357 |  -115.176 |
| Las Vegas | Pick N Puff                                     |   5.0 |           10 | 89103       |   36.115 |  -115.237 |
| Las Vegas | Bachelor Vegas                                  |   5.0 |            8 | 89103       |  36.1119 |  -115.203 |
| Las Vegas | Bruce Trent Park                                |   4.0 |           15 | 89128       |  36.1866 |  -115.286 |
| Las Vegas | U-Haul Neighborhood Dealer                      |   4.0 |            4 | 89144       |  36.1905 |  -115.317 |
| Las Vegas | Graceland Presents Elvis                        |   4.0 |            4 | 89109       |  36.1363 |  -115.151 |
| Las Vegas | Culinary Pharmacy                               |   5.0 |            5 | 89104       |   36.149 |  -115.154 |
| Las Vegas | Eric Andrew Collection At Town Square           |   4.5 |            5 | 89119       |  36.0669 |  -115.177 |
| Las Vegas | Davis Glass and Mirror                          |   4.0 |           45 | 89118       |  36.0966 |   -115.19 |
| Las Vegas | Amazing Las Vegas Comic Con                     |   4.0 |           21 | 89109       |  36.1325 |  -115.155 |
| Las Vegas | Now and Zen Massage Therapy                     |   5.0 |           32 | 89128       |  36.1791 |  -115.256 |
| Las Vegas | Impressions Tile & Marble                       |   5.0 |            4 | 89178       |  36.0101 |  -115.293 |
| Las Vegas | Angels Carpet Care                              |   5.0 |            3 | 89112       |     36.1 |   -115.07 |
| Las Vegas | Michael Mina                                    |   4.0 |          574 | 89109       |   36.112 |  -115.177 |
| Las Vegas | Vegas Uncork'd: The Grand Tasting               |   4.0 |           20 | 89108       |  36.1154 |  -115.178 |
| Las Vegas | Premier Baseball Academy                        |   4.0 |            6 | 89148       |  36.0845 |  -115.297 |
| Las Vegas | Carquest Auto Parts                             |   4.5 |            3 | 89146       |  36.1481 |  -115.244 |
| Las Vegas | Xpand Realty & Property Management              |   5.0 |           30 | 89113       |   36.081 |  -115.242 |
| Las Vegas | Pro-Tech Transmission and Complete Auto Repair  |   4.5 |            6 | 89118       |  36.0921 |  -115.207 |
| Las Vegas | Summit Consultancy Group, CPA                   |   5.0 |            3 | 89147       |  36.1032 |  -115.297 |
| Las Vegas | Xelement Music & Dance                          |   5.0 |            4 | 89117       |  36.1455 |  -115.301 |
+-----------+-------------------------------------------------+-------+--------------+-------------+----------+-----------+
```

```
(Output limit exceeded, 25 of 838 total rows shown)
*/

--Locations and ZIP's of businesses with 2 or 3 stars in Las Vegas city
SELECT city, name, stars, review_count, postal_code, latitude, longitude
FROM business
WHERE city = 'Las Vegas' AND (stars BETWEEN 2 AND 3)

/*
+-----------+-------------------------------------------+-------+--------------+-------------+----------+-----------+
| city      | name                                      | stars | review_count | postal_code | latitude | longitude |
+-----------+-------------------------------------------+-------+--------------+-------------+----------+-----------+
| Las Vegas | World Food Championships                  |   3.0 |            5 | 89109       |  36.1143 |  -115.171 |
| Las Vegas | The Coffee Bean & Tea Leaf                |   3.0 |           38 | 89162       |  36.1221 |  -115.168 |
| Las Vegas | Cancun Resort Las Vegas by Diamond Resorts |   3.0 |          298 | 89123       |   36.038 |  -115.173 |
| Las Vegas | Jody Maroni's Sausage Kingdom             |   3.0 |            5 | 89109       |   36.103 |  -115.174 |
| Las Vegas | Catholic Charities                        |   3.0 |           11 | 89145       |  36.1611 |  -115.245 |
| Las Vegas | Red Ginseng Narita Sushi & BBQ            |   3.0 |           76 | 89149       |  36.2813 |  -115.287 |
| Las Vegas | Red Rooster Antique Mall                  |   3.0 |            6 | 89102       |  36.1584 |  -115.159 |
| Las Vegas | Adore Organic Innovation                  |   2.0 |           17 | 89109       |  36.1275 |  -115.172 |
| Las Vegas | AL's Donuts                               |   2.0 |            5 | 89169       |  36.1193 |  -115.146 |
| Las Vegas | Christian Audigier The Nightclub          |   3.0 |           79 | 89109       |  36.1245 |  -115.172 |
| Las Vegas | Sam's Club Optical                        |   2.0 |            3 | 89117       |  36.1242 |  -115.248 |
| Las Vegas | T-Mobile                                  |   3.0 |            4 | 89102       |  36.1264 |  -115.193 |
| Las Vegas | WellHealth Women's Specialty Care         |   3.0 |           18 | 89149       |  36.2858 |  -115.285 |
| Las Vegas | Ticor Title At Tivoli Village             |   3.0 |            6 | 89145       |  36.1672 |  -115.286 |
| Las Vegas | Starbucks                                 |   3.0 |           33 | 89103       |  36.1179 |  -115.187 |
| Las Vegas | Huntridge Circle Park                     |   2.5 |            3 | 89104       |  36.1563 |  -115.137 |
| Las Vegas | Chinatown Foot Reflexology                |   3.0 |           24 | 89102       |  36.1264 |  -115.198 |
| Las Vegas | Fresh Buffet                              |   2.5 |          160 | 89109       |  36.1363 |  -115.151 |
| Las Vegas | Baymont Inn And Suites Las Vegas          |   3.0 |           54 | 89119       |  36.1156 |  -115.151 |
| Las Vegas | Krave Massive                             |   2.0 |            9 | 89101       |  36.1701 |  -115.141 |
| Las Vegas | Smashburger                               |   3.0 |           59 | 89019       |  36.1162 |  -115.175 |
| Las Vegas | Macy's                                    |   3.0 |           67 | 89135       |   36.149 |  -115.335 |
| Las Vegas | Spinal Care of Nevada                     |   3.0 |           15 | 89134       |  36.1944 |  -115.305 |
| Las Vegas | Coasta Cantina                            |   2.0 |           23 | 89103       |  36.1027 |  -115.202 |
| Las Vegas | Great Wraps                               |   3.0 |           23 | 89109       |   36.127 |  -115.168 |
+-----------+-------------------------------------------+-------+--------------+-------------+----------+-----------+
*/


/*
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 are more businesses open than ones are closed in 'Las Vegas'

/*
ii. Difference 2:
*/

The average reviews from open businesses is greater than the ones are closed

/*
SQL code used for analysis:
*/

item i Difference 1.

--Counting a total businesses are open in Las Vegas
SELECT SUM(is_open) FROM business
WHERE city = 'Las Vegas' AND (is_open = 1)

/*
+--------------+
| SUM(is_open) |
+--------------+
```

```
|         1311 |
+--------------+
*/

--Counting a total businesses are closed in Las Vegas
SELECT COUNT(is_open) FROM business
WHERE city = 'Las Vegas' AND (is_open = 0)

/*
+----------------+
| COUNT(is_open) |
+----------------+
|            250 |
+----------------+
*/

item ii Difference 2.

--Counting the avg reviews regarding businesses open in Las Vegas
SELECT avg(review_count) FROM business
WHERE city = 'Las Vegas' AND is_open = 1

/*
+-------------------+
| avg(review_count) |
+-------------------+
|     55.1212814645 |
+-------------------+
*/

--Counting the avg reviews regarding businesses closed in Las Vegas
SELECT avg(review_count) FROM business
WHERE city = 'Las Vegas' AND is_open = 0

/*
+-------------------+
| avg(review_count) |
+-------------------+
|             42.36 |
+-------------------+
*/



/*
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 f 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 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:
*/
        I decided to parse out keywords and business attributes for sentiment analysis.
        In order to response: Do users more reviews with word 'love' or 'hate' on avarage in previous defined groups
        (2-3 stars and 4-5 stars)?
        Now I pick up 'Las Vegas' city in order to get a metric for comparing these groups (2-3 stars and 4-5 stars).

        The ratios are:

        1. First ratio (I defined new columns RLove_2_3 and RLove_4_5) is:
           (total reviews with word 'love' of the group of interest)/(total reviews of the group of interest)


        2. Second ratio (I defined new column RHate_2_3 and RHate_4_5)is:
           (total reviews with word 'hate' of the group of interest)/(total reviews of the group of interest)

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

For this analysis I will count of all reviews that matched with patterns '%lovê%', 'lovê%', '%lovê', '%hate%', 'hate%' and '%hate' for each business from 'Las Vegas' in order to obtain a new columns (metrics) that I describe above.

Write another paragraph

```
/*
iii. Output of your finished dataset:
*/
```

I show one table to retrieve a new table with business names, stars, *text* (reviews) and city. With this table we can perform a process describe above, this step I called "FIRST STEP". Hence I retrieve a new table with a nested query using a table from the first step and I get the metrics that I called ratio_love and ratio_hate.

```
--"FIRST STEP"

/*
```

name	stars	review_2_3	review_4_5	love_2_3	love_4_5	hate_2_3	hate_4_5
Delmonico Steakhouse	5	0	1	0	0	0	0
Delmonico Steakhouse	5	0	1	0	0	0	0
Now and Zen Massage Therapy	5	0	1	0	0	0	0
Michael Mina	5	0	1	0	1	0	0
The Butcher Block	5	0	1	0	1	0	0
The Perfect Scoop & Boba Tea	5	0	1	0	1	0	0
Kinh Do	5	0	1	0	1	0	0
Art of Flavors	5	0	1	0	0	0	0
Hitchin Post Saloon & Steakhouse	5	0	1	0	0	0	0
Blueberry Hill Family Restaurant	5	0	1	0	0	0	0
The Cheesecake Factory	5	0	1	0	1	0	0
The Cheesecake Factory	5	0	1	0	0	0	0
My Pedi Nails & Spa	5	0	1	0	1	0	0
Orleans Spa & Fitness Center	5	0	1	0	0	0	0
Tequila Bar and Grill	5	0	1	0	0	0	0
Vintage Pools	5	0	1	0	0	0	0
Norton Jr Alexander, MD	5	0	1	0	0	0	0
Full House BBQ	5	0	1	0	1	0	0
AT&T	5	0	1	0	0	0	0
Luv It Frozen Custard	5	0	1	0	0	0	0
Tea Space	5	0	1	0	0	0	0
Vanity Nails & Spa	5	0	1	0	0	0	0
Rollin' Smoke BBQ	5	0	1	0	0	0	0
BLT Steak	5	0	1	0	0	0	0
Diablo's Cantina	5	0	1	0	0	0	0

```
(Output limit exceeded, 25 of 193 total rows shown)
*/

--"SECOND STEP" (more deep query and final step)

/*
```

Trev_2_3	Tlove_2_3	Thate_2_3	RLove_2_3	RHate_2_3	Trev_4_5	Tlove_4_5	Thate_4_5	RLove_4_5	RHate_4_5
31	5	0	0.16	0.0	139	32	2	0.23	0.01

```
*/

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

--"FIRST STEP"

SELECT name, stars, --text, city,
--I intensionally omit to retrieve text and city columns in order to improve visualization
--Building review_2_3_stars, If review exists 1, otherwise 0
CASE
    WHEN stars = 2 THEN 1
    WHEN stars = 3 THEN 1
```

```
    ELSE 0
END review_2_3,
--Building review_4_5_stars, If review exists 1, otherwise 0
CASE
    WHEN stars = 4 THEN 1
    WHEN stars = 5 THEN 1
    ELSE 0
END review_4_5,
--Building love_2_3_stars, If love word appears 1, otherwise 0
CASE
    WHEN (stars IN (2,3)) AND (text LIKE "%love%" OR ((text LIKE "%love") OR (text LIKE "love%"))) THEN 1
    ELSE 0
END love_2_3,
--Building love_4_5. If love word appears 1, otherwise 0
CASE
    WHEN (stars IN (4,5)) AND (text LIKE "%love%" OR ((text LIKE "%love") OR (text LIKE "love%"))) THEN 1
    ELSE 0
END love_4_5,
--Building hate_2_3_stars. If hate word appears 1, otherwise 0
CASE
    WHEN (stars IN (2,3)) AND (text LIKE "%hate%" OR ((text LIKE "%hate") OR (text LIKE "hate%"))) THEN 1
    ELSE 0
END hate_2_3,
--Building hate_4_5_stars. If hate word appears 1, otherwise 0
CASE
    WHEN (stars IN (4,5)) AND (text LIKE "%hate%" OR ((text LIKE "%hate") OR (text LIKE "hate%"))) THEN 1
    ELSE 0
END hate_4_5
FROM
(
    SELECT name, stars, text, city FROM
    (
        SELECT * FROM
        (
            --Retrive all registers in review table that match with Pitsburg city
            --in business table
            --This way to retrieve a new table let save us a lot of time when we perform
            --a CROSS JOIN or CARTESIAN JOIN
            --because our query is filtered from begining
            SELECT r.business_id, r.stars, r.text FROM review AS r
            WHERE r.business_id IN
            (
                SELECT id FROM business
                WHERE city = 'Las Vegas'
            )
        )
        --Perform a Cartesian join in order to retrieve a new table with all combinations
        --with our previous table and business table with city = 'Las Vegas'
        CROSS JOIN
        (
            SELECT id, name, city FROM business
            WHERE city = 'Las Vegas'
        )
    )
    --This condition is established in order to retrive unique registers
    WHERE business_id=id
    ORDER BY stars DESC
)



--"SECOND STEP"

SELECT
--In order to improve visualization I had to trim final variable names
SUM(review_2_3) AS Trev_2_3, --Total Reviews in group of businesses with 2-3 stars
SUM(love_2_3) AS Tlove_2_3, --Total times that love word appears in group of businesses with 2-3 stars
SUM(hate_2_3) AS Thate_2_3, --Total times that hate word appears in group of businesses with 2-3 stars
CASE --Computing ratio between Tlove_2_3 and Trev_2_3
    WHEN CAST(SUM(review_2_3) AS REAL) = 0 THEN 0 --Avoiding ZERO division
    ELSE ROUND(SUM(love_2_3)/CAST(SUM(review_2_3) AS REAL), 2)
END RLove_2_3,
CASE --Computing ratio between Thate_2_3 and Trev_2_3
```

```sql
    WHEN CAST(SUM(review_2_3) AS REAL) = 0 THEN 0 --Avoiding ZERO division
    ELSE ROUND(SUM(hate_2_3)/CAST(SUM(review_2_3) AS REAL), 2)
END RHate_2_3,
SUM(review_4_5) AS Trev_4_5, --Total Reviews in group of businesses with 4-5 stars
SUM(love_4_5) AS Tlove_4_5, --Total times that love word appears in group of businesses with 4-5 stars
SUM(hate_4_5) AS Thate_4_5, --Total times that hate word appears in group of businesses with 4-5 stars
CASE --Computing ratio between Tlove_4_5 and Trev_4_5
    WHEN CAST(SUM(review_4_5) AS REAL) = 0 THEN 0 --Avoiding ZERO division
    ELSE ROUND(SUM(love_4_5)/CAST(SUM(review_4_5) AS REAL), 2)
END RLove_4_5,
CASE --Computing ratio between Thate_4_5 and Trev_4_5
    WHEN CAST(SUM(review_4_5) AS REAL) = 0 THEN 0 --Avoiding ZERO division
    ELSE ROUND(SUM(hate_4_5)/CAST(SUM(review_4_5) AS REAL), 2)
END RHate_4_5
FROM
--This is our column shown it in previous step (FIRST STEP)
(
    SELECT name, stars, text, city,
    --Building review_2_3_stars, If review exists 1, otherwise 0
    CASE
        WHEN stars = 2 THEN 1
        WHEN stars = 3 THEN 1
        ELSE 0
    END review_2_3,
    --Building review_4_5_stars, If review exists 1, otherwise 0
    CASE
        WHEN stars = 4 THEN 1
        WHEN stars = 5 THEN 1
        ELSE 0
    END review_4_5,
    --Building love_2_3_stars, If love word appears 1, otherwise 0
    CASE
        WHEN (stars IN (2,3)) AND (text LIKE "%love%" OR ((text LIKE "%love") OR (text LIKE "love%"))) THEN 1
        ELSE 0
    END love_2_3,
    --Building love_4_5_stars. If love word appears 1, otherwise 0
    CASE
        WHEN (stars IN (4,5)) AND (text LIKE "%love%" OR ((text LIKE "%love") OR (text LIKE "love%"))) THEN 1
        ELSE 0
    END love_4_5,
    --Building hate_2_3_stars. If hate word appears 1, otherwise 0
    CASE
        WHEN (stars IN (2,3)) AND (text LIKE "%hate%" OR ((text LIKE "%hate") OR (text LIKE "hate%"))) THEN 1
        ELSE 0
    END hate_2_3,
    --Building hate_4_5_stars. If hate word appears 1, otherwise 0
    CASE
        WHEN (stars IN (4,5)) AND (text LIKE "%hate%" OR ((text LIKE "%hate") OR (text LIKE "hate%"))) THEN 1
        ELSE 0
    END hate_4_5
    FROM
    (
        SELECT name, stars, text, city FROM
        (
            SELECT * FROM
            (
                --Retrive all registers in review table that match with Pitsburg city
                --in business table
                --This way to retrieve a new table let save us a lot of time when we perform
                --a CROSS JOIN or CARTESIAN JOIN
                --because our query is filtered from begining
                SELECT r.business_id, r.stars, r.text FROM review AS r
                WHERE r.business_id IN
                (
                    SELECT id FROM business
                    WHERE city = 'Las Vegas'
                )
            )
            --Perform a Cartesian join in order to retrieve a new table with all combinations
            --with our previous table and business table with city = 'Las Vegas'
            CROSS JOIN
            (
                SELECT id, name, city FROM business
```

```
            WHERE city = 'Las Vegas'
        )
    )
    --This condition is established in order to retrive unique registers
    WHERE business_id=id
    ORDER BY stars DESC
  )
)
```