/*

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 analysis 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
```

```
SELECT count(*) FROM Attribute;
```

```sql
SELECT count(*) FROM Business;

SELECT count(*) FROM Category;

SELECT count(*) FROM Checkin;

SELECT count(*) FROM elite_years;

SELECT count(*) FROM friend;

SELECT count(*) FROM hours;

SELECT count(*) FROM photo;

SELECT count(*) FROM review;

SELECT count(*) FROM tip;

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.

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

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

SELECT count(DISTINCT id) FROM Business;

SELECT count(DISTINCT business_id) FROM hours;

SELECT count(DISTINCT business_id) FROM Category;

SELECT count(DISTINCT business_id) FROM Attribute;

SELECT count(DISTINCT id) FROM review;
SELECT count(DISTINCT business_id) FROM review;

SELECT count(DISTINCT user_id) FROM review;

SELECT count(DISTINCT business_id) FROM Checkin;

SELECT count(DISTINCT id) FROM photo;

SELECT count(DISTINCT business_id) FROM photo;

SELECT count(DISTINCT user_id) FROM tip;

SELECT count(DISTINCT business_id) FROM tip;

SELECT count(DISTINCT id) FROM user;

SELECT count(DISTINCT user_id) FROM friend;

SELECT count(DISTINCT user_id) FROM elite_years;

-- 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(*)
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


--   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

SELECT MIN(Stars),MAX(Stars),AVG(Stars)
FROM Review;

SELECT MIN(Stars),MAX(Stars),AVG(Stars)
from Business;

SELECT MIN(Likes),MAX(Likes),AVG(Likes)
from Tip;

SELECT MIN(Count),MAX(Count),AVG(Count)
from Checkin;

SELECT MIN(Review_count),MAX(Review_count),AVG(Review_count)
from User;
```

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

-- SQL code used to arrive at answer:

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

-- Copy and Paste the Result Below:

```
 ----------------- + ------- +
| city              | NUM    |
+ ----------------- + ------- +
| Las Vegas         | 82854 |
| Phoenix           | 34503 |
| Toronto           | 24113 |
| Scottsdale        | 20614 |
| Charlotte         | 12523 |
| Henderson         | 10871 |
| Tempe             | 10504
| Pittsburgh        | 9798 |
| Montreal          | 9448 |
| Chandler          | 8112 |
| Mesa              | 6875 |
| Gilbert           | 6380 |
| Cleveland         | 5593 |
| Madison           | 5265 |
| Glendale          | 4406 |
| Mississauga       | 3814 |
| Edimburgo         | 2792 |
| Peoria            | 2624 |
| North Las Vegas   | 2438 |
| Markham           | 2352
| Champaign         | 2029
| Stuttgart         | 1849 |
| Surpresa          | 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 SUM(review_count) AS Numbers, stars
FROM business
WHERE city == "Avon"
GROUP BY stars;
```

```
-- Copy and Paste the Resulting Table Below (2 columns â€‟ star rating and count):
+---------+-------+
| Numbers | stars |
+---------+-------+
| 10      | 1.5   |
| 6       | 2.5   |
| 88      | 3.5   |
| 21      | 4.0   |
| 31      | 4.5   |
| 3       | 5.0   |
+---------+-------+

-- ii. Beachwood

-- SQL code used to arrive at answer:
SELECT SUM(review_count) AS Numbers, stars
FROM business
WHERE city == "Beachwood"
GROUP BY stars;

-- Copy and Paste the Resulting Table Below (2 columns â€‟ star rating and count):
+---------+-------+
| Numbers | stars |
+---------+-------+
| 8       | 2.0   |
| 3       | 2.5   |
| 11      | 3.0   |
| 6       | 3.5   |
| 69      | 4.0   |
| 17      | 4.5   |
| 23      | 5.0   |
+---------+-------+

-- 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:

Not necessarily correlated. Some that have more fans, and have less ratings. Others have fewer fans but have the third highest number of ratings.

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

```
+-----------+--------------+------+
| 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 |
| Mark      |          861 |  115 |
| Tiffany   |          408 |  111 |
| bernice   |          255 |  105 |
| Roanna    |         1039 |  104 |
| Angela    |          694 |  101 |
| .Hon      |         1246 |  101 |
| Ben       |          307 |   96 |
| Linda     |          584 |   89 |
| Christina |          842 |   85 |
| Jessica   |          220 |   84 |
| Greg      |          408 |   81 |
| Nieves    |          178 |   80 |
| Sui       |          754 |   78 |
| Yuri      |         1339 |   76 |
| Nicole    |          161 |   73 |
+-----------+--------------+------+
(Output limit exceeded, 25 of 10000 total rows shown)
```

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

-- Answer:
Love has 1780, while hate only has 232

-- SQL code used to arrive at answer:

```
SELECT COUNT(*)                                                  SELECT COUNT(*)
          FROM review                                                      FROM review
          WHERE text LIKE "%love%"                              WHERE text LIKE "%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.

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

The 4-5 star group seems to have shorter hours then the 2-3 star group.

Please note the query returned only three businesses so not a great sample size.

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

Yes and no, one of the 4-5 star group has a lot more reviews but then the other 4-5 star group has close to the same number of reviews as the 2-3 star group.

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

No, every business is in a different zip-code.

-- SQL code used for analysis:

```
SELECT
business.name
, business.city
, category.category
, business.stars
,hours.hours,
business.review_count,
business.address,
business.postal_code
FROM (business INNER JOIN category ON business.id =
category.business_id) INNER JOIN hours ON hours.business_id =
business.id
WHERE business.city = 'Toronto' AND category.category = "Food"
GROUP BY business.stars;
```

-- 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 ones that are still open have more reviews on average than ones that are closed.

-- ii. Difference 2:

There are more business that are still open listed as “useful” or “funny”.

-- SQL code used for analysis:

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

```
+---------------------+-------------------+------------------+---------------+-------
--+
| COUNT(DISTINCT(id)) | AVG(review_count) | SUM(review_count) |    AVG(stars) |
is_open |
+---------------------+-------------------+------------------+---------------+-------
--+
|                1520 |     23.1980263158 |            35261 | 3.52039473684 |
0 |
|                8480 |     31.7570754717 |           269300 | 3.67900943396 |
1 |
+---------------------+-------------------+------------------+---------------+-------
--+
```

-- 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:
    Here I chose to study the preference among different types of food on yelp.

-- 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 pick several types of food including "Chinese","Mexican","Korean","French","Italian","Japanese" and "Indian". Then I will analyze their star ratings and number of reviews so that I can get some insights on which type of food is popular on yelp.

-- iii. Output of your finished dataset:

```
+----------+----------------------+---------------+-------------------+-----------+
| category | Number_Of_Resturants |    AVG(stars) | AVG(review_count) | city      |
+----------+----------------------+---------------+-------------------+-----------+
| Korean   |                    7 |           4.5 |               8.0 | Toronto   |
| French   |                   12 |           4.0 |     135.083333333 | Las Vegas |
| Chinese  |                   13 | 3.76923076923 |     423.230769231 | Las Vegas |
| Mexican  |                   28 |         3.625 |              73.0 | Edinburgh |
| Italian  |                   13 | 3.53846153846 |     78.2307692308 | Montréal  |
| Japanese |                   20 |         3.475 |             22.85 | Toronto   |
+----------+----------------------+---------------+-------------------+-----------+
-- iv. Provide the SQL code you used to create your final dataset:

SELECT c.category,COUNT(b.name) AS
Number_Of_Resturants,AVG(stars),AVG(review_count),b.city
FROM (business b INNER JOIN hours h ON b.id = h.business_id)
INNER JOIN category c ON c.business_id = b.id
WHERE c.category IN
("Chinese","Mexican","French","Italian","Korean","Japanese","Indian")
GROUP BY c.category
ORDER BY AVG(stars) DESC;
```