Part 1: Yelp Dataset Profiling and Understanding

* 1. Profile the data by finding the total number of records for each of the tables below:

**SELECT COUNT(\*)**

**FROM table**

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.

**SELECT COUNT(DISTINCT Keys)**

**FROM Table;**

i.Business = id:

**SELECT COUNT(DISTINCT id)**

**FROM Business;**

+--------------------+

| COUNT(DISTINCT id) |

+--------------------+

| 10000 |

+--------------------+

ii. Hours = business\_id

**SELECT COUNT(DISTINCT business\_id)**

**FROM hours;**

+-----------------------------+

| COUNT(DISTINCT business\_id) |

+-----------------------------+

| 1562 |

+-----------------------------+

iii. Category = business\_id

**SELECT COUNT(DISTINCT business\_id)**

**FROM Category;**

+-----------------------------+

| COUNT(DISTINCT business\_id) |

+-----------------------------+

| 2643 |

+-----------------------------+

iv. Attribute = business\_id

**SELECT COUNT(DISTINCT business\_id)**

**FROM attribute;**

+-----------------------------+

| COUNT(DISTINCT business\_id) |

+-----------------------------+

| 1115 |

+-----------------------------+

v. Review = Primary\_key: id, Foreign\_key: business\_id, user\_id

**SELECT COUNT(DISTINCT id)**

**FROM review;**

+--------------------+

| COUNT(DISTINCT id) |

+--------------------+

| 10000 |

+--------------------+

**SELECT COUNT(DISTINCT business\_id)**

**FROM review;**

+-----------------------------+

| COUNT(DISTINCT business\_id) |

+-----------------------------+

| 8090 |

+-----------------------------+

**SELECT COUNT(DISTINCT user\_id)**

**FROM review;**

+-------------------------+

| COUNT(DISTINCT user\_id) |

+-------------------------+

| 9581 |

+-------------------------+

vi. Checkin = business\_id

SELECT COUNT(DISTINCT business\_id)

FROM Checkin;

+-----------------------------+

| COUNT(DISTINCT business\_id) |

+-----------------------------+

| 493 |

+-----------------------------+

vii. Photo = business\_id

**SELECT COUNT(DISTINCT business\_id)**

**FROM Photo;**

+-----------------------------+

| COUNT(DISTINCT business\_id) |

+-----------------------------+

| 6493 |

+-----------------------------+

viii. Tip = Foreign key” business\_id, user\_id

SELECT COUNT(DISTINCT user\_id)

FROM Tip;

+-------------------------+

| COUNT(DISTINCT user\_id) |

+-------------------------+

| 537 |

+-------------------------+

SELECT COUNT(DISTINCT business\_id)

FROM Tip;

+-----------------------------+

| COUNT(DISTINCT business\_id) |

+-----------------------------+

| 3979 |

+-----------------------------+

ix. User = id

SELECT COUNT(DISTINCT id)

FROM USER;

+--------------------+

| COUNT(DISTINCT id) |

+--------------------+

| 10000 |

+--------------------+

x. Friend = user\_id

SELECT COUNT(DISTINCT user\_id)

FROM friend;

xi. Elite\_years = user\_id

SELECT COUNT(DISTINCT user\_id )

FROM Elite\_years ;

+--------------------------+

| COUNT(DISTINCT 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: 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:

SELECT min(column), max(column), avg(column)

FROM Table;

i. Table: Review, Column: Stars min: max: avg:

**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: max: avg:

**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: max: avg:

**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: max: avg:

**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: max: avg:

**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: Copy and Paste the Result Below:

SELECT city,SUM(review\_count) AS TotalReviews

FROM business

GROUP BY city

ORDER BY TotalReviews DESC;

+-----------------+--------------+

| city | TotalReviews |

+-----------------+--------------+

| 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

SQL code used to arrive at answer: Copy and Paste the Resulting Table Below (2 columns â€“ star rating and count):

**SELECT city, stars as [Star Rating], count(stars) as [Count]**

**FROM business b**

**WHERE city = 'Avon'**

**GROUP BY stars;**

|  |
| --- |
| select |
|  | name |
|  | , stars |
|  | , review\_count |
|  | from business |
|  | where city = 'Avon'; |

+------+-------------+-------+

| city | Star Rating | Count |

+------+-------------+-------+

| Avon | 1.5 | 1 |

| Avon | 2.5 | 2 |

| Avon | 3.5 | 3 |

| Avon | 4.0 | 2 |

| Avon | 4.5 | 1 |

| Avon | 5.0 | 1 |

+------+-------------+-------+

ii. Beachwood

SQL code used to arrive at answer: Copy and Paste the Resulting Table Below (2 columns â€“ star rating and count):

Need to rewrite

**SELECT city, stars as [Star Rating], count(stars) as [Count]**

**FROM business b**

**WHERE city = 'Beachwood'**

**GROUP BY stars;**

+-----------+-------------+-------+

| city | Star Rating | Count |

+-----------+-------------+-------+

| Beachwood | 2.0 | 1 |

| Beachwood | 2.5 | 1 |

| Beachwood | 3.0 | 2 |

| Beachwood | 3.5 | 2 |

| Beachwood | 4.0 | 1 |

| Beachwood | 4.5 | 2 |

| Beachwood | 5.0 | 5 |

|  |  |  |
| --- | --- | --- |
|  |  |  |

7. Find the top 3 users based on their total number of reviews: SQL code used to arrive at answer: Copy and Paste the Result Below:

**SELECT name, id, Review\_Count**

**FROM user**

**ORDER BY Review\_Count DESC**

**Limit 3;**

+--------+------------------------+--------------+

| name | id | review\_count |

+--------+------------------------+--------------+

| Gerald | -G7Zkl1wIWBBmD0KRy\_sCw | 2000 |

| Sara | -3s52C4zL\_DHRK0ULG6qtg | 1629 |

| Yuri | -8lbUNlXVSoXqaRRiHiSNg | 1339 |

|  |  |  |
| --- | --- | --- |
|  |  |  |

8. Does posing more reviews correlate with more fans? Please explain your findings and interpretation of the results:

**SELECT name,Review\_Count,Fans**

**FROM User**

**Order by Review\_Count DESC;**

+-----------+--------------+------+

| 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 |

+-----------+--------------+------+

It does not look like more reviews correlate more fans. For example: You can see that Yuri has more review count than Hon but she has less fans.

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

**SELECT (select count(text)**

**from review**

**where text like "%love%") as love\_COUNT,**

**(select count(text)**

**from review**

**where text like "%hate%") as hate\_COUNT;**

+------------+------------+

| love\_COUNT | hate\_COUNT |

+------------+------------+

| 1780 | 232 |

+------------+------------+

Yes, they are more reviews with the word “love” than the word “hate” in them.

10. Find the top 10 users with the most fans: SQL code used to arrive at answer: Copy and Paste the Result Below:

**SELECT id, name,fans**

**FROM user**

**ORDER BY fans DESC**

**Limit 10;**

+------------------------+-----------+------+

| id | name | fans |

+------------------------+-----------+------+

| -9I98YbNQnLdAmcYfb324Q | Amy | 503 |

| -8EnCioUmDygAbsYZmTeRQ | Mimi | 497 |

| --2vR0DIsmQ6WfcSzKWigw | Harald | 311 |

| -G7Zkl1wIWBBmD0KRy\_sCw | Gerald | 253 |

| -0IiMAZI2SsQ7VmyzJjokQ | Christine | 173 |

| -g3XIcCb2b-BD0QBCcq2Sw | Lisa | 159 |

| -9bbDysuiWeo2VShFJJtcw | Cat | 133 |

| -FZBTkAZEXoP7CYvRV2ZwQ | William | 126 |

| -9da1xk7zgnnfO1uTVYGkA | Fran | 124 |

| -lh59ko3dxChBSZ9U7LfUw | 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 Mesa**

**Category: Shopping**

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

**Yes, they have Different distribution of hours.**

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

**Yes. The two groups that I chose have a different number of reviews.**

* 1. Are you able to infer anything from the location data provided between these two groups? Explain. SQL code used for analysis:

**No. This is because even the store has more stars rating they can have less reviews.**

**SELECT**

**B.NAME AS Store\_Name,**

**C.CATEGORY,**

**B.ADDRESS,**

**B.CITY,**

**B.STARS,**

**B.REVIEW\_COUNT,**

**H.HOURS**

**FROM BUSINESS B INNER JOIN CATEGORY C**

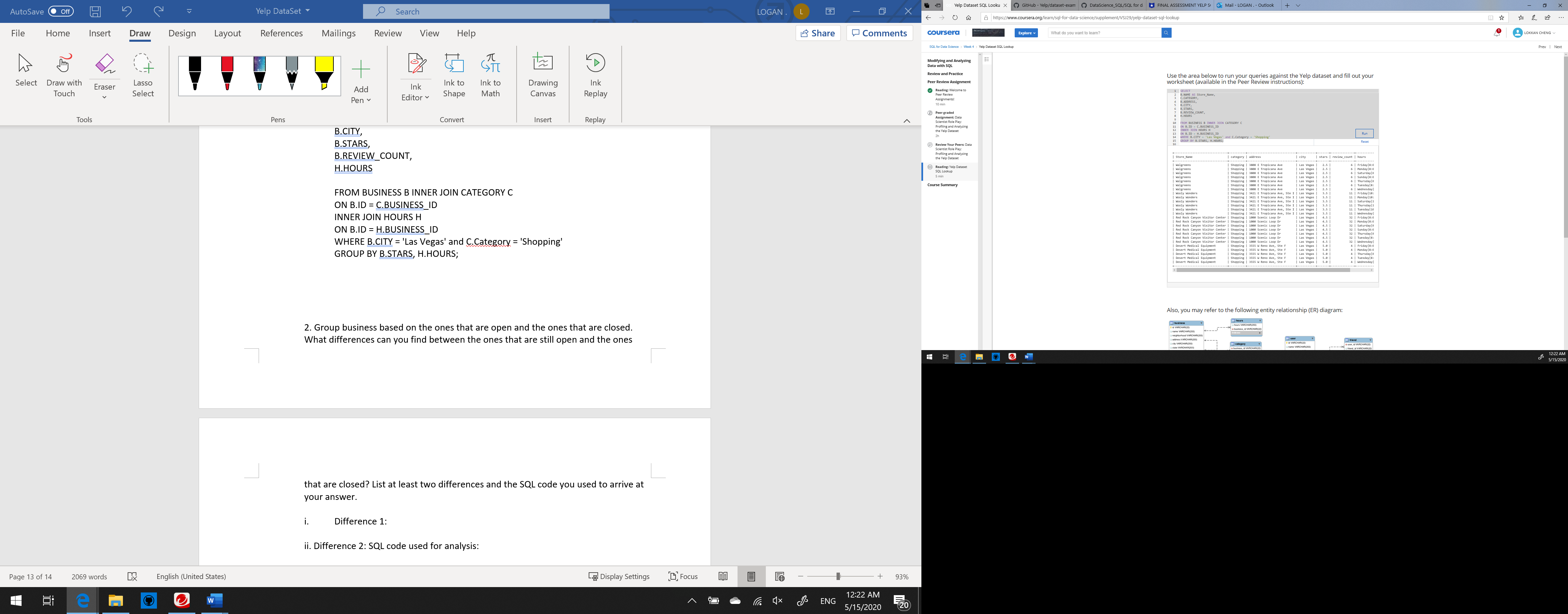
**ON B.ID = C.BUSINESS\_ID**

**INNER JOIN HOURS H**

**ON B.ID = H.BUSINESS\_ID**

**WHERE B.CITY = 'Las Vegas' and C.Category = 'Shopping'**

**GROUP BY B.STARS, H.HOURS;**



**SELECT**

**B.NAME AS Store\_Name,**

**C.CATEGORY,**

**B.ADDRESS,**

**B.CITY,**

**B.STARS,**

**B.REVIEW\_COUNT,**

**H.HOURS**

**FROM BUSINESS B INNER JOIN CATEGORY C**

**ON B.ID = C.BUSINESS\_ID**

**INNER JOIN HOURS H**

**ON B.ID = H.BUSINESS\_ID**

**WHERE B.CITY = 'Mesa' and C.Category = 'Shopping'**

**GROUP BY B.STARS, H.HOURS;**

+------------------------+----------+---------------------+------+-------+--------------+-----------------------+

| Store\_Name | category | address | city | stars | review\_count | hours |

+------------------------+----------+---------------------+------+-------+--------------+-----------------------+

| Ghost Armor SS Springs | Shopping | 6555 E Southern Ave | Mesa | 2.0 | 3 | Friday|10:00-21:00 |

| Ghost Armor SS Springs | Shopping | 6555 E Southern Ave | Mesa | 2.0 | 3 | Monday|10:00-21:00 |

| Ghost Armor SS Springs | Shopping | 6555 E Southern Ave | Mesa | 2.0 | 3 | Saturday|10:00-21:00 |

| Ghost Armor SS Springs | Shopping | 6555 E Southern Ave | Mesa | 2.0 | 3 | Sunday|11:00-18:00 |

| Ghost Armor SS Springs | Shopping | 6555 E Southern Ave | Mesa | 2.0 | 3 | Thursday|10:00-21:00 |

| Ghost Armor SS Springs | Shopping | 6555 E Southern Ave | Mesa | 2.0 | 3 | Tuesday|10:00-21:00 |

| Ghost Armor SS Springs | Shopping | 6555 E Southern Ave | Mesa | 2.0 | 3 | Wednesday|10:00-21:00 |

+------------------------+----------+---------------------+------+-------+--------------+-----------------------+

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.

1. Difference 1:

The one that is not open has lower star but higher review

1. Difference 2: SQL code used for analysis:

The one that is still open has different store hours

SELECT

b.NAME,

b.IS\_OPEN,

c.CATEGORY,

b.STARS,

h.HOURS,

b.REVIEW\_COUNT,

b.POSTAL\_CODE

FROM BUSINESS b INNER JOIN CATEGORY c

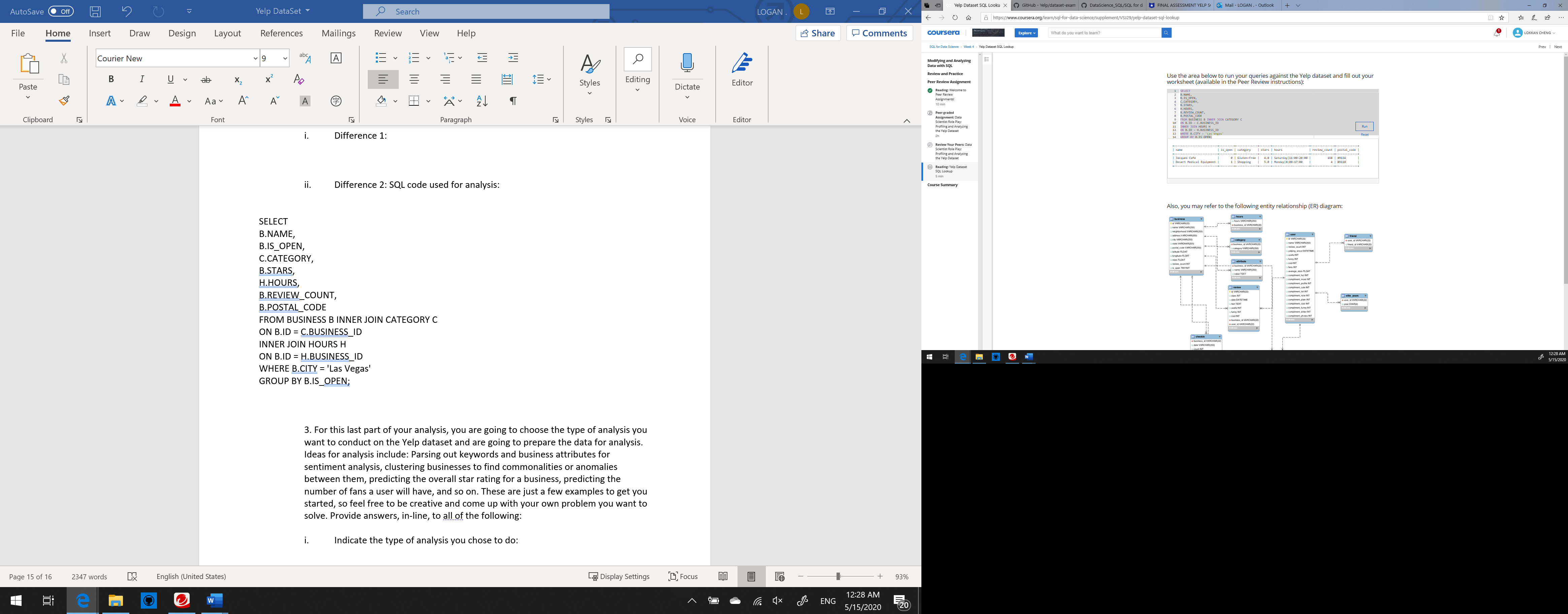
ON B.ID = C.BUSINESS\_ID

INNER JOIN HOURS h

ON b.ID = h.BUSINESS\_ID

WHERE b.CITY = 'Las Vegas'

GROUP BY b.IS\_OPEN;



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:

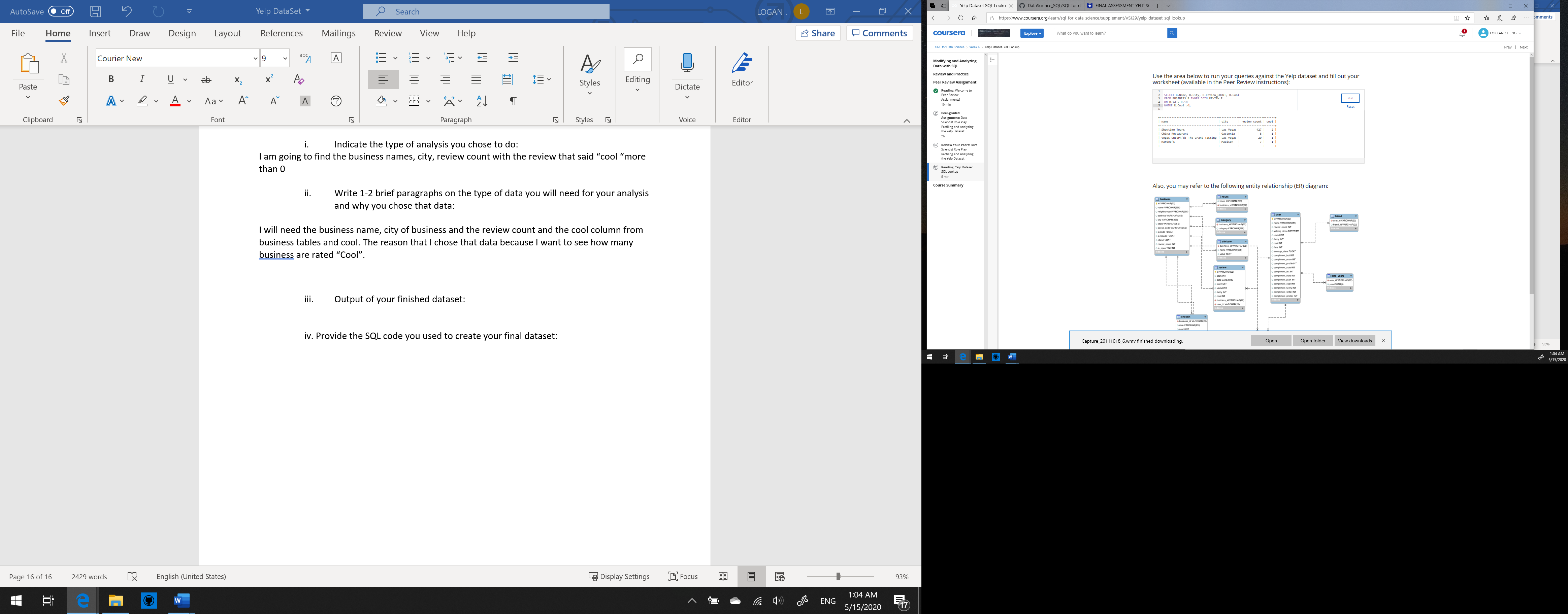
1. Indicate the type of analysis you chose to do:

I am going to find the business names, city, review count with the review that said “cool “more than 0

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

I will need the business name, city of business and the review count and the cool column from business tables and cool. The reason that I chose that data because I want to see how many business are rated “Cool”.

1. Output of your finished dataset:



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

**SELECT B.Name, B.City, B.review\_COUNT, R.Cool**

**FROM BUSINESS B INNER JOIN REVIEW R**

**ON B.id = R.id**

**WHERE R.Cool >0;**