Q1 [5 pt] write a SQL query to find name of all the business in the ‘fast food’ category.

SELECT Name

FROM Business

WHERE Category = ‘fast food’;

Q2 [10 pt] write a SQL query to find the top 3 users based on total number of fans.

SELECT Name

FROM User

ORDER BY Fans DESC

LIMIT 3;

Q3 [10 pt] write a SQL query to find the number of users registered after 2020.

SELECT COUNT(\*)

FROM User

WHERE Yelping\_since > 2020;

Q4 [10 pt] write a SQL query to list the cities with the most reviews in descending order.

SELECT City

FROM Business

ORDER BY Review\_count DESC;

Q5 [10 pt] write SQL to update stars and review\_counts for business named " yami sushi".

UPDATE Business

SET Stars = 5, Review\_count =1000

WHERE Name=" yami sushi";

Q6 [10 pt] write a SQL query to find the distribution of star ratings to the business in the Pittsburgh (return stars and corresponding counts).

SELECT Review.Stars, COUNT(\*)

FROM Review, Business

WHERE Business.ID=Review.Business\_id

AND Business.City=" Pittsburgh"

GROUP BY Review.Stars;

Q7 [10 pt] for the restaurant named “yami sushi” write a SQL query to find if there are more reviews with the word ‘love’ or with the word ‘hate’.

SELECT IF(R1.LOVE\_REVIEW > R2.HATE\_REVIEW, ‘more reviews with the word ‘love’’, ‘more reviews with the word ‘hate’’)

FROM (SELECT Business.Name AS R1\_NAME, COUNT(\*) AS LOVE\_REVIEW

FROM Review, Business

WHERE Business.ID=Review.Business\_id

AND Business.Name=" yami sushi"

AND Review.Text LIKE ‘%love%’

GROUP BY Business.Name, Review.Text

) AS R1,

(SELECT Business.Name AS R2\_NAME, COUNT(\*) AS HATE\_REVIEW

FROM Review,Business

WHERE Business.ID=Review.Business\_id

AND Business.Name=" yami sushi"

AND Review.Text LIKE ‘%hate%’

GROUP BY Business.Name, Review.Text

)AS R2

Where R1.R1\_NAME = R2.R2\_NAME;

Q8 [10 pt] for each user, calculate average star, average number of “useful”, “funny” and “cool” for all the reviews this user wrote.

SELECT User.Name, AVG(Review.Stars), AVG(Review.Useful), AVG(Review.Funny), AVG(Review.Cool)

FROM User, Review

WHERE User.Id= Review.User\_id

GROUP BY User.Name;

Q9 [10 pt] Find the user who has the largest number of fans, return the name and amount for the top 5 businesses he tipped.

SELECT User.Name, Tip.amount

FROM User, Tip, (SELECT User.Id AS User\_Id, MAX(User.Fans)

FROM User) AS R1

WHERE User.Id= R1.User\_Id

AND Tip.User\_Id = R1.User\_Id

ORDER BY Tip.amount DESC

LIMIT 5;

Q10 [15 pt] Group business based on the one that are open and the ones that are closed, what differences you can find between those two groups (you can use different SQL to answer different sub questions)

1. Are there more reviews written for the business that are still open.

SELECT IF(R1.OPEN\_REVIEW > R2.CLOSED\_REVIEW, ‘more reviews for the business that are still open’, ‘more reviews for the business closed’)

FROM (SELECT COUNT(\*) AS OPEN\_REVIEW

FROM Business

WHERE Business.Is\_open=1

) AS R1,

(SELECT SELECT COUNT(\*) AS CLOSED\_REVIEW

FROM Business

WHERE Business.Is\_open=0

)AS R2;

1. Are the average star rating higher for business that are open than business that are closed.

SELECT IF(R1.OPEN\_avgstar > R2.CLOSED\_avgstar, ‘the average star rating is higher for business that are open’, ‘the average star rating is higher for business that are closed’)

FROM (SELECT AVG(Business.Stars) AS OPEN\_avgstar

FROM Business

WHERE Business.Is\_open=1

) AS R1,

(SELECT AVG(Business.Stars)AS CLOSED\_avgstar

FROM Business

WHERE Business.Is\_open=0

)AS R2;

3. Are there more tips for business that are open than business that are closed.

SELECT IF(R1.OPEN\_tips > R2.CLOSED\_tips, ‘there are more tips for business that are open’, ‘there are more tips for business that are closed’)

FROM (SELECT SUM(Tip.amount) AS OPEN\_tips

FROM Business, Tip

WHERE Business.Id=Tip.Business\_id

AND Business.Is\_open=1

) AS R1,

(SELECT SUM(Tip.amount) AS CLOSED\_tips

FROM Business

WHERE Business.Id=Tip.Business\_id

AND Business.Is\_open=0

)AS R2;