Big Assignment 2024

- To complete this assignment, upload the “Yelp data.sql” database from Mycourse [“Exercises and database files” section] into HeidiSQL. The database has 16 tables containing information on reviews Yelp users have left for different businesses they have visited.
- Please read each question carefully and consider which information (attributes) you need from each table to arrive at the solution. Provide the answer in a designated space under each question and the MySQL code that generated the result by copying the code from HeidiSQL and pasting it into this document. If asked in the question, also attach screenshots of the results.
- Please check the rows of the data imported for each table (see Yelp Dataset Description.docx) and ensure you have imported the data correctly, not importing the data twice or importing insufficient data.
- If you see an error message like “the server has gone away,” please contact the teacher to update the server settings. Aalto IT might update the server settings without informing the teacher, preventing you from importing large files.
- You are free to create additional columns or tables to support your analysis.
- Please submit the assignment as a Word document, not a PDF. A Word document allows us to copy and check your code easily.
- Grading method: If your answer is wrong but the code is correct, you will still receive a substantial minus in points. A wrong answer with the correct code is still considered a failure in real-life business.
- Good luck with the assignment! :)

Question 1 (4 points, each answer for 1 point)

In the business table, assume that the businesses with a rating below 2 belong to the very poor rating category; the businesses with ratings ["stars" variable] from 2 to below 3 are considered poor ratings; the businesses with ratings from 3 to below 4.5 are considered good ratings, and the businesses that have ratings 4.5 or above are considered excellent ratings. Only the businesses that are still active ("true" in the "active" column) are considered.

Please report the average review count for the businesses in each rating level.

Rating level	Average review count
Excellent_rating	15.4550
Good_rating	31.9416
Poor_rating	8.4493
Very_poor_rating	4.6823

MySQL code that generates the result:

```
SELECT
  CASE
    WHEN stars >= 4.5 THEN 'Excellent_rating'
    WHEN stars >= 3 AND stars < 4.5 THEN 'Good_rating'
    WHEN stars >= 2 AND stars < 3 THEN 'Poor_rating'
    ELSE 'Very_poor_rating'
  END AS Rating_level,
  AVG(review_count) AS Average_review_count
FROM business
WHERE active = 'true'
GROUP BY Rating_level;
```

Question 2 (6 points)

In the table friends, we can see the Yelp users who are friends. Find the user ['name' from table users] with the most friends, where one friendship is considered only once.

Amount of friends: ____ 2032____ (3 points)
"Name" [not user_id] of the user: ____ Gabi____ (3 points)

MySQL code that generates the result:

```
SELECT u.name, f.friend_total AS Amount_of_friends
FROM users u
JOIN (
    SELECT ui.user_id, COUNT(DISTINCT ui.friend_id) AS friend_total
    FROM (
        SELECT user_id1 AS user_id, user_id2 AS friend_id FROM friends
        UNION
        SELECT user_id2 AS user_id, user_id1 AS friend_id FROM friends
    ) AS ui
    GROUP BY ui.user_id
) AS f ON u.user_id = f.user_id
ORDER BY f.friend_total DESC
LIMIT 1;
```

Question 3 (10 points)

a) In the table "business", there are different attributes for each business reviewed by users in Yelp. Consider the [city] column and find which city received the highest amount of reviews from those cities where the businesses received less than 100,000 reviews in total.

The city with the highest amount of reviews: ___ Scottsdale___ (5 points)

MySQL code that generates the result:

```
SELECT city, SUM(review_count) AS reviews_in_total
FROM business
GROUP BY city
HAVING reviews_in_total < 100000
ORDER BY reviews_in_total DESC
LIMIT 1;
```

b) In the city "Phoenix", what business category received the highest amount of reviews? Please provide the name of the business category and the number of reviews given to the category.

Business Category: __ Restaurants__ (2 points)
Number of reviews received: __ 100827__ (3 points)

MySQL code that generates the result:

```
SELECT category AS business_category, total_review
FROM (
    SELECT business_categories.category,
           SUM(business.review_count) AS total_review
    FROM business
    JOIN business_categories ON business.business_id =
business_categories.business_id
    WHERE business.city = 'Phoenix'
    GROUP BY business_categories.category
) AS category_reviews
ORDER BY total_review DESC
LIMIT 1;
```

Question 4 (10 points)

In the table "users", please find the user's name [name] for that user who ranks as the 1st among all users regarding the number of fans. Also, find this user's rank in giving funny_votes [votes_funny].

User name: __ Connie__ (5 points)
Ranking in votes_funny: ___7___ (5 points)

MySQL code that generates the result:

```
SELECT
    u.name,
    (SELECT COUNT(*)
     FROM users
     WHERE votes_funny > u.votes_funny) + 1 AS funny_rank
FROM users u
WHERE u.fans = (SELECT MAX(fans) FROM users);
```

Question 5 (8 points)

a) Table "users" contains information on reviews users have left for businesses. From those users who started Yelping between September 2010 and May 2011 [yelping_since_year] [yelping_since_month], what is the number of users who have written reviews but received no votes in funny [votes_funny], useful [votes_useful], and cool [votes_cool] categories in the "**reviews**" table?

Number of users: ___ 7946___(4 points)

MySQL code that generates the result:

```
SELECT COUNT(DISTINCT u.user_id) AS number_of_users
FROM users u
JOIN reviews r ON u.user_id = r.user_id
WHERE
  (
    (u.yelping_since_year = 2010 AND u.yelping_since_month >= 9)
    OR (u.yelping_since_year = 2011 AND u.yelping_since_month <= 5)
  )
  AND r.votes_funny = 0
  AND r.votes_useful = 0
  AND r.votes_cool = 0;
```

b) From the users that have reviewed businesses located in the city “Phoenix”, please identify the user with the highest total amount of reviews written [“review_acount” in the “users” table], and give the name of the user [name] and the total number of reviews written [“review_acount” in the “users” table].

The name of the user: __ Bruce__(2 points)
The number of reviews: __ 3286__(2 points)

MySQL code that generates the result:

```
SELECT u.name, u.review_count

FROM users u

JOIN reviews r ON u.user_id = r.user_id

JOIN business b ON r.business_id = b.business_id

WHERE b.city = 'Phoenix'

ORDER BY u.review_count DESC

LIMIT 1;
```

Question 6 (8 points)

Explore “business”, “business_attribute_goodfor” and “business_hours” tables, and answer the following question:

What is the business name [business_name], the opening time [opening_time] on Sundays of a business that is currently **active**, has been reviewed as good for “breakfast” in the table “business_attribute_goodfor” and has received the star rating 5 among other similar businesses?

Note: you can find the data about the “active” and “stars” of a business in the “business” table; “opening_time_hours” at “business_hours” table; “subattribute” and true or false “value” at “business_attributes_goodfor” table.

The name of the business [NOT business_id]: _ Rayner's Chocolate & Coffee Shop_(4 points)
Opening time: __ 06:00:00__(4 points)

MySQL code that generates the result:

```
SELECT b.business_name, bh.opening_time
FROM business b
JOIN business_attributes_goodfor bag ON b.business_id = bag.business_id
JOIN business_hours bh ON b.business_id = bh.business_id
WHERE bh.day_of_week = 'Sunday' AND b.active = 'true'
  AND bag.subattribute = 'breakfast'
  AND bag.value = 'true' AND b.stars = 5;
```

Question 7 (8 points)

Use the tables "reviews" and "business" to answer this question. Please find the business name [business_name] that has received the highest proportion of star ratings [stars] **above** 4 out of all their reviews. Consider only those businesses which have over 800 reviews in total.

The name of the business [NOT business_id]: __ Cibo__ (4 points)
The ratio of the ratings for the business: __ 0.5174__ (4 points)

MySQL code that generates the result:

```
SELECT b.business_name,
       (SUM(CASE WHEN r.stars > 4 THEN 1 ELSE 0 END) / COUNT(r.review_id))
AS ratio
FROM business b
JOIN reviews r ON b.business_id = r.business_id
GROUP BY b.business_id
HAVING COUNT(r.review_id) > 800
ORDER BY ratio DESC
LIMIT 1;
```

Question 8 (8 points)

The table "checkin" represents a simplified report on the amounts of customers checking in per each hour from Mondays to Sundays. Which are the top 3 busiest hotels in terms of checkins on Mondays that are currently **active**? Consider those businesses that have a business category only in "**Hotels**" [see "business_categories" table, whether a business is of "Hotels & Travel" category or other similar is not relevant here] and the number of checkins only during the evening hours between 18 pm and 23 pm [time_18] – [time_23]. Provide the name of the hotel, star rating, and total amount of customers checking for that business, which had the highest star rating among the three busiest hotels on Monday evenings.

The name of the business [NOT business_id]: _ Hotel Valley Ho_ (2 points)
Star rating: ___ 4___ (3 points)
The total number of customers checking in (18 pm-23 pm): ____53____ (3 points)

MySQL code that generates the result:

```
SELECT b.business_name, b.stars,
       (SUM(c.time_18) + SUM(c.time_19) + SUM(c.time_20) + SUM(c.time_21) +
SUM(c.time_22) + SUM(c.time_23)) AS total_checking
FROM business b
JOIN business_categories bc ON b.business_id = bc.business_id
JOIN checkins c ON b.business_id = c.business_id
WHERE bc.category = 'Hotels'
  AND c.day_of_week = 'Monday'
  AND b.active = 'true'
GROUP BY b.business_name, b.stars
ORDER BY total_checking DESC, b.stars DESC
LIMIT 3;
```

Question 9 (8 points, each answer for 2 points)

Assuming that when a hair salon specializes only in certain areas rather than providing all different kinds of services, the quality tends to be better. Your task is to determine if the star ratings for businesses in the table "business_attribute_hairtypesspecializedin" tend to be better or worse when their areas of specialization increase or decrease.

To answer, please create four specialization categories based on the amount of true values in the subattribute categories each business_id in the "business_attribute_hairtypesspecializedin" has. For the categorization, the businesses that specialize in 8 or 7 categories belong to the full_specialities category; the businesses with 6 or 5 categories belong to the multiple_specialities category; the businesses with 4 or 3 categories belong to some_specialities category, and the businesses with 2 or 1 categories belong to few_specialities category. You can obtain "stars" values from the "business" table.

Please report the average stars for the businesses in each specialty category.

Speciality category	Average stars
Full_specialities (8 – 7)	4.041666666666667
Multiple_specialities (6 – 5)	4.65
Some_specialities (4 – 3)	4.5
Few_specialities (2 – 1)	4.875

MySQL code that generates the result:

```
SELECT Speciality_category, AVG(stars) AS Average_stars
FROM (
    SELECT b.business_id, b.stars,
        CASE
            WHEN COUNT(subattribute) BETWEEN 7 AND 8 THEN
'Full_specialities (8 - 7)'
            WHEN COUNT(subattribute) BETWEEN 5 AND 6 THEN
'Multiple_specialities (6 - 5)'
            WHEN COUNT(subattribute) BETWEEN 3 AND 4 THEN
'Some_specialities (4 - 3)'
            ELSE 'Few_specialities (2 - 1)'
        END AS Speciality_category
    FROM business_attributes_hairtypesspecializedin bah
    JOIN business b ON bah.business_id = b.business_id
    WHERE bah.value = 'true'
    GROUP BY bah.business_id
) AS specialities
GROUP BY Speciality_category;
```

Question 10 (10 points)

In the "tip" table in the Yelp database, based on the column "tip_text" that have at least two words:

a) Was the frequency of the word "food" as the first or second word more popular? Please ignore whether the letter in the word is an upper or lower case. Please give frequencies:

The word "Food" as 1st word: __ 346__(2 points)
The word "Food" as 2nd word: __ 1358__(3 points)

b) What is the most popular word when food was the second word?

The most popular first word with the word "food" as second word: __ great__(2 points)
Frequency of the word: __477__(3 points)

Please remember to drop seven punctuation marks from the title, including:

“	”	"	-	,	!	`

We do not consider removing more punctuation marks to obtain consistent results. Also, please remove empty spaces from both sides of the title.

MySQL code that generates the result:
a)

```
SELECT
    'food1' AS categories,
    COUNT(*) AS counts
FROM
    tips
WHERE
    tip_text LIKE '% %'
    AND LOWER(SUBSTRING_INDEX(

TRIM(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(tip_text, '"',
''), '”', ''), '"', ''), '-', ''), ',', ''), '!', ''), '`', '')),
        ' ', 1)) = 'food'

UNION ALL

SELECT
    'food2' AS categories,
    COUNT(*) AS counts
FROM
    tips
WHERE
    tip_text LIKE '% %'
    AND LOWER(SUBSTRING_INDEX(
        SUBSTRING_INDEX(

TRIM(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(tip_text, '"',
''), '”', ''), '"', ''), '-', ''), ',', ''), '!', ''), '`', '')),
            ' ', 2), ' ', -1)) = 'food';
```

b)

```
SELECT
LOWER(SUBSTRING_INDEX(tip_text1, ' ', 1)) AS PopularWord,
COUNT(*) AS frequency
FROM (
SELECT
TRIM(BOTH ' ' FROM
REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(tip_text, '^', ''),
'”', ''), '"', ''), '-', ''), ',', ''), '!', ''), '`', '')
) AS tip_text1
FROM tips
) AS tips2
WHERE LOWER(SUBSTRING_INDEX(SUBSTRING_INDEX(tip_text1, ' ', 2), ' ', -1)) =
'food'
GROUP BY PopularWord
ORDER BY frequency DESC
LIMIT 1;
```

Question 11 (10 points)

Based on the “review” table, please compare the total amounts of reviews for each business for 2013 and 2012, respectively, and report the name of the business [business_name] and the difference in reviews for the business that had the highest decrease in number of reviews received from 2012 to 2013.

Name of the business [NOT business_id]: __ Matt's Big Breakfast__ (3 points)
The decrease in reviews from 2012 to 2013: __ 107__ (7 points)

My SQL code that generates the result:

```
SELECT b.business_name,
       (r2012.review_count - r2013.review_count) AS review_decrease
FROM (SELECT business_id, COUNT(*) AS review_count
      FROM reviews
      WHERE YEAR(review_date) = 2012
      GROUP BY business_id) r2012
JOIN (SELECT business_id, COUNT(*) AS review_count
      FROM reviews
      WHERE YEAR(review_date) = 2013
      GROUP BY business_id) r2013
  ON r2012.business_id = r2013.business_id
JOIN business b ON r2012.business_id = b.business_id
ORDER BY review_decrease DESC
LIMIT 1;
```

Question 12 (10 points)

Analyze the revisiting customer's satisfaction on their next visit to a business branch. You will only need the table "reviews" for this task.

Users[user_id] can leave a lower star rating [stars] on their first visit [review_date] to a business [business_id], but give a higher star rating on their next visit to the same business, indicating that the customer revisited the business and is more satisfied on their next visit. Please count the number of users who rated a business higher on their second [or third, et.] visit than their first visit.

Please consider only the reviews from **2012** and thus it is recommended to filter only rows with reviews written in 2012 first to reduce the run time of the SQL commands.

Note 1: If one user visits different businesses, all the visits count, and the same user could be counted several times in the analysis.

Note 2: If a user wrote two or more reviews for a business within the same day, this user's reviews for the specific business are considered untrustworthy. Thus, we drop this user's reviews for this specific business. However, the user's reviews on other businesses will be retained if no more untrustworthy reviews are found.

Amount of users who rated a business higher on their second visit: __ 305__

MySQL code that generates the result:

```
CREATE TEMPORARY TABLE temp_review_2012 AS
SELECT
    review_id,
    business_id,
    user_id,
    stars,
    review_date
FROM reviews
WHERE YEAR(review_date) = 2012;


CREATE TEMPORARY TABLE temp_untrustworthy AS
SELECT
    business_id,
    user_id,
    review_date
FROM temp_review_2012
GROUP BY business_id, user_id, review_date
HAVING COUNT(*) > 1;


DELETE FROM temp_review_2012
WHERE (business_id, user_id, review_date) IN (
    SELECT business_id, user_id, review_date
    FROM temp_untrustworthy
);


CREATE TEMPORARY TABLE temp_first_visits AS
SELECT
    r.business_id,
    r.user_id,
    MIN(r.review_date) AS first_visit_date
```

```sql
FROM temp_review_2012 r
GROUP BY r.business_id, r.user_id;


CREATE TEMPORARY TABLE temp_first_visit_details AS
SELECT
    f.business_id,
    f.user_id,
    f.first_visit_date,
    r.stars AS first_visit_stars
FROM temp_first_visits f
JOIN temp_review_2012 r
    ON f.business_id = r.business_id
    AND f.user_id = r.user_id
    AND f.first_visit_date = r.review_date;


CREATE TEMPORARY TABLE temp_follow_up_visits AS
SELECT
    r.business_id,
    r.user_id,
    r.review_date AS follow_up_date,
    r.stars AS follow_up_stars
FROM temp_review_2012 r
JOIN temp_first_visit_details f
    ON r.user_id = f.user_id
    AND r.business_id = f.business_id
WHERE r.review_date > f.first_visit_date;


SELECT COUNT(f.user_id) AS improved_ratings_count
FROM temp_first_visit_details f
JOIN temp_follow_up_visits fv
    ON f.user_id = fv.user_id
    AND f.business_id = fv.business_id
WHERE fv.follow_up_stars > f.first_visit_stars;


DROP TEMPORARY TABLE IF EXISTS temp_review_2012, temp_untrustworthy,
temp_first_visits, temp_first_visit_details, temp_follow_up_visits;
```