**CS411, Fall 2023, Homework 4 – Simple SNS Database, due by midnight March 14**

Refer to the SQL script sns\_data.sql. The file contains a simplified social networks database schema. It contains the following tables: users, photos, comments, follows, likes, phooto\_tags, and tags. The

* Users: (id, username, created\_at)
* Photos: (id, image\_url, user\_id, created\_at)
* Comments: (id, comment\_text, photo\_id, user\_id, created\_at)
* Likes: (user\_id, pnoto\_id, created\_at)
* Follows: (follower\_id, followee\_id, created\_at)
* Tags: (id, tag\_name, created\_at)
* Pnoto\_tags: (photo\_id, tag\_i)

Users can post photos, and post comments on photos posted by other users. Users can follow other users. Users can like photos posted by other users. Photos contain image URLs and the time the photo was posted. Tags describe the photo. Once photos are tagged, the photos can be organized with the tags. A photo can have multiple tags.

Create the SNS database from the given script and perform the following tasks.

1. Create an ER diagram by using Workbench. Arrange the generated diagram so that no lines or boxes intersect. Refer to the Workbench note about how to create an ER diagram from database.

Diagram

Description automatically generated

1. Write a query to find the least popular signup dates for users. The answer should be in Day, that is, Monday, Tuesday, …, and the number of days. If there is a tie, show all tied days.

use sns;

SELECT DATE\_FORMAT(created\_at, '%W') AS day\_of\_week, COUNT(DATE\_FORMAT(created\_at, '%W')) AS numOfDays

FROM users

GROUP BY day\_of\_week

HAVING numOfDays <= ALL(

SELECT COUNT(DATE\_FORMAT(created\_at, '%W')) AS numOfDays

FROM users

GROUP BY DATE\_FORMAT(created\_at, '%W')

)

1. Write a query to find the users who have not posted any photos.

SELECT username

FROM users

WHERE username NOT IN (

SELECT username

FROM users u JOIN photos p

ON u.id = p.user\_id

)

Graphical user interface, application

Description automatically generated



1. Write a query to find five most popular photos and users who created them. Most popular photos are those being liked most. Show the user id, user name, the photo URL, and the number of likes.

USE sns;

SELECT u.id AS userID, username, image\_url, COUNT(photo\_id) AS numOfLikes

FROM users u

JOIN photos p ON u.id = p.user\_id

JOIN likes l ON l.photo\_id = p.id

GROUP BY l.photo\_id

ORDER BY COUNT(l.photo\_id) DESC LIMIT 5

Graphical user interface, text, application

Description automatically generated



1. Write a query to find the average number of photos posted by users who posted at least one photo.

use sns;

SELECT AVG(t.posted) as Average

FROM (

SELECT COUNT(user\_id) as posted

FROM photos

GROUP BY user\_id

HAVING posted > 0

)t



1. Write a query to find the most followed users and the most following users. The answer should like the following.

|  |  |  |  |
| --- | --- | --- | --- |
| Most followed or following | Count | ID | Name |
| Most followed | … | … | … |
| Most following | … | … | … |

USE sns;

(SELECT 'Most followed' As 'Most followed or following', COUNT(\*) AS 'Count', f.follower\_id AS 'ID' , u.username AS 'Name'

FROM follows f

JOIN users u ON f.follower\_id = u.id

GROUP BY follower\_id

ORDER BY COUNT(\*) DESC

LIMIT 1)

UNION (

SELECT 'Most following' As 'Most followed or following', COUNT(\*) AS 'Count', f.followee\_id AS 'ID' ,u.username AS 'Name'

FROM follows f

JOIN users u ON f.followee\_id = u.id

GROUP BY followee\_id

ORDER BY COUNT(\*) DESC

LIMIT 1)

Table

Description automatically generated

1. Write a query to find the user who has posted most photos, and the photos’ ids and the tag names. Show the user name, photo id, and tag name.

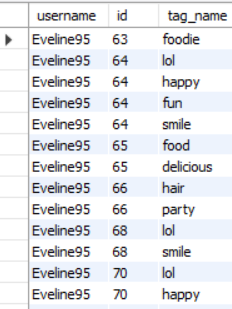
SELECT username, p.id AS photoID, tag\_name

FROM (

SELECT user\_id, COUNT(id)

FROM photos

GROUP BY user\_id

HAVING COUNT(id) = (

SELECT MAX(posts)

FROM(

SELECT user\_id, COUNT(id) posts

FROM photos

GROUP BY user\_id)t

)

)y

JOIN users ON y.user\_id = id

JOIN photos p ON y.user\_id = p.user\_id

JOIN photo\_tags ON p.id = photo\_id

JOIN tags t ON t.id = tag\_id

ORDER BY photo\_id



1. Write a query to find the five most popular tag names and the total number of tags for each tag name. Show the tag id, tag name, and the number of tags.

USE sns;

SELECT tag\_id, tag\_name, numOfTags

FROM tags JOIN (

SELECT tag\_id, COUNT(tag\_id) numOfTags

FROM photo\_tags

GROUP BY tag\_id

) t ON t.tag\_id = id

ORDER BY numOfTags DESC

LIMIT 5

Table

Description automatically generated

1. Bots (or fake user ids) can be created and the bots like every photo being posted. Write a query to find such bots. Show the bot user names and user ids.

Graphical user interface, application

Description automatically generated

USE sns;

SELECT username, user\_id

FROM likes l JOIN users u

ON u.id = l.user\_id

GROUP BY user\_id

HAVING COUNT(\*) =

(SELECT COUNT(\*) FROM photos)



1. With the bot accounts removed, show the result of #6**. I know I have something wrong but I cannot find it and I am going crazy**

USE sns;

SELECT mostFollowed AS 'Most Followed or Following', count AS 'Count', id AS ID, name AS 'Name'

FROM(

(SELECT 'Most followed' As 'mostFollowed', COUNT(\*) AS count, f.follower\_id AS ID , u.username AS name

FROM follows f

JOIN users u ON f.follower\_id = u.id

GROUP BY follower\_id

ORDER BY COUNT(\*) DESC

LIMIT 1)

UNION (

SELECT 'Most following' As 'mostFollowed', COUNT(\*) AS count, f.followee\_id AS ID ,u.username AS name

FROM follows f

JOIN users u ON f.followee\_id = u.id

GROUP BY followee\_id

ORDER BY COUNT(\*) DESC

LIMIT 1)

)t

WHERE ID NOT IN (

SELECT username, user\_id

FROM likes l JOIN users u

ON u.id = l.user\_id

GROUP BY user\_id

HAVING COUNT(\*) =

(SELECT COUNT(\*) FROM photos)

)

All the query problems should be written with one query. Subqueries or Unions are allowed, but multiple queries will not get full credits.

Provide all query answers in TEXT and the result in images in Word file. For the ER diagram, capture the image and put in in Word file. Submit the Word file.