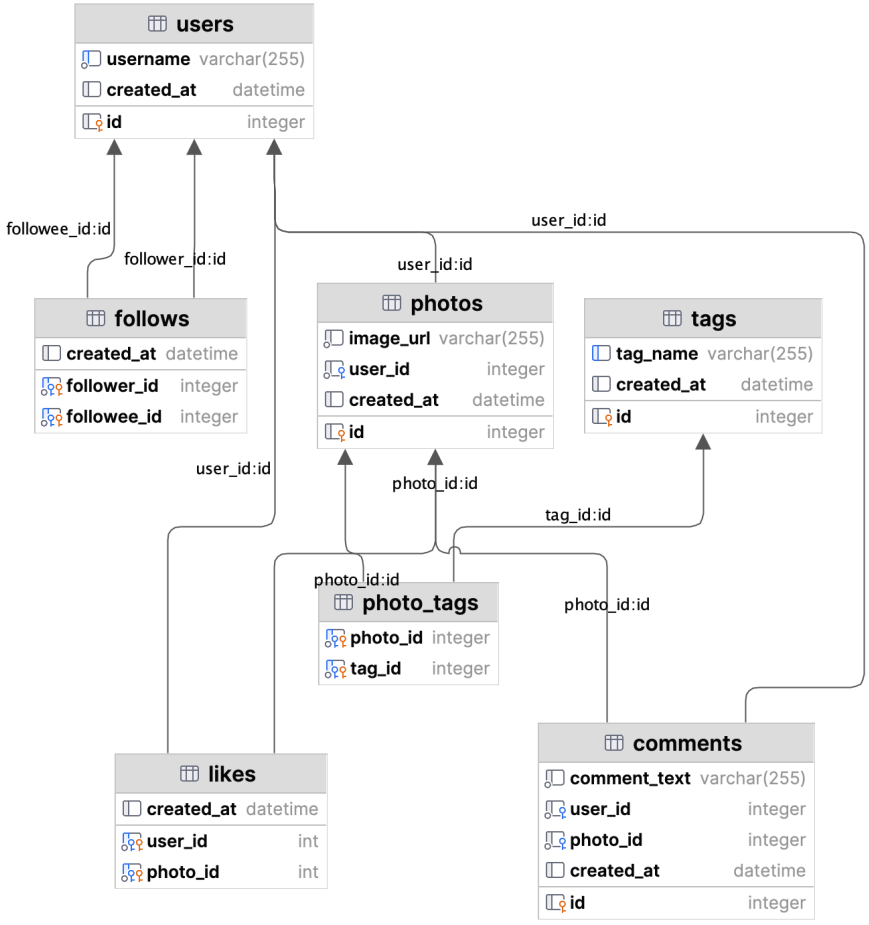
CS 3200 - Introduction to Databases

SQL Practice Problems

These practice problems will be based on the HuskyGram data model below. It represents a database schema for a small image-sharing social media platform like Instagram. Notes on the model:

- Users can upload photos and add tags like *#huskylife* or *#cs3200rocks* to each photo.
    - A tag may appear in the tags table but not be associated with any photos
    - A photo may have many tags or none at all.
- Users (including whoever posted the photo) may make comments on/about the photo. A photo may have no comments, however.
- Users may also add a like to a photo, but no more than 1 time
- Users follow other users, but following is not automatically reciprocal
    - For clarity, with respect to the **follows** table, if Sam follows the posts of Julie, Sam would be considered the follower (the one doing the following), and Julie would be considered the followee (the one being followed). If they both follow each other, there would be two separate rows in the **follows** table.



Provide SQL SELECT queries for each of the following questions.

Level 1 Problems:

1. Give the image url of all images uploaded by user *SillyCat123*.

2. Which users follow user *CooperTheDoggo*? Give the names of their followers.

3. Give the user names of any user that has tagged a photo with *#huskylife*.

4. Provide a list of tag names associated with any photo that has been liked by *SillyCat123.*

5. List all the comments on any photo tagged with #huskylife uploaded by users *CooperTheDoggo*, *WinstonPupper,* or *BusterBeagle* in ascending chronological order of when the comment was made (created).

Level 2 Problems:

1. Provide a list of all users with all of each users' followers. The result should contain two columns: one named userName and the other named followerName, and it should be sorted by userName then by followerName for each user.

2. Provide a count of the number of photos that have more than 3 comments and 2 or more tags. Your query should return a single column named numPhotos with a single row containing the count.

3. What are the three most popular tags? A tag's popularity is defined as the number of photos to which a particular tag is applied. The result of your query should contain the tag name (aliased as tagName) and number of photos that it has been applied to (aliased as numPhotos) sorted in descending order based on numPhotos (most popular tag at the top of the list).
   Simplifying assumptions: Assume that all tags are applied to a different number of photos.

4. How many comments have been written about each photo? The resulting table should contain the photo's id number, URL of the photo, and number of comments. The results should be in reverse chronological order based on when the photo was uploaded (created_at attribute).

5. Which users have uploaded photos that have not yet been "liked" but have received comments? The resulting table should include a list of usernames sorted alphabetically.

6. How many photos have been tagged with both #NEU and #BU? The resulting table should have a single column named "num_photos".

7. Provide a list of all comments that contain the substring "college". The results should contain the name of the commenter and the text of the comment, and the data should be sorted in ascending order by the date the comment was made. (Note: don't worry about case-sensitivity in comments).

8. Which users have not uploaded any photos? The resulting table should contain a list of user names sorted alphabetically as well as when they joined the platform (create_at field in the users table). Use appropriate column names in the result.