

# CSCE-608 Project #1

## Leo Feng, 132003234

### March 22, 2024

#### 1. Application Description

My application is a web application that allows users to query data from a social media site like Twitter. For those unaware of Twitter (or X now), it is a platform where people can make short posts, or tweets, typically about world events, their personal lives, or popular culture. Other users can post short thoughts replying to people's tweets, as well as like and bookmark tweets. For a social media site where people post their thoughts such as Twitter, I will include the basic entities user, post, like, comment, and hashtag. Based on my entities, I decided to design my application with the five following queries:

1. Query posts with attributes such as which user made the post or the hashtag of the post.
2. Insert a new post.
3. Count how many comments and likes a post has.
4. Find a user's most liked post.
5. Retrieve all posts a user has liked.

#### 2. E/R Diagram

First, let's list the attributes of all of the entities:

**User:** A user on twitter has a user ID, public username, an email associated with the account, a short description of the person that shows up on their profile, the type of account (verified celebrities, premium members, regular members), and how many followers the user has. These are the six attributes I will include. The primary key will be the user ID.

**Post:** A post will have a post ID, the text contents of the post, the user ID of the user who made the post, and the date at which the post was created.

**Comment:** A comment will have a reply ID, the post ID of which the comment is replying to, the reply text contents, the user ID of the person making the comment, and the date at which the comment was posted.

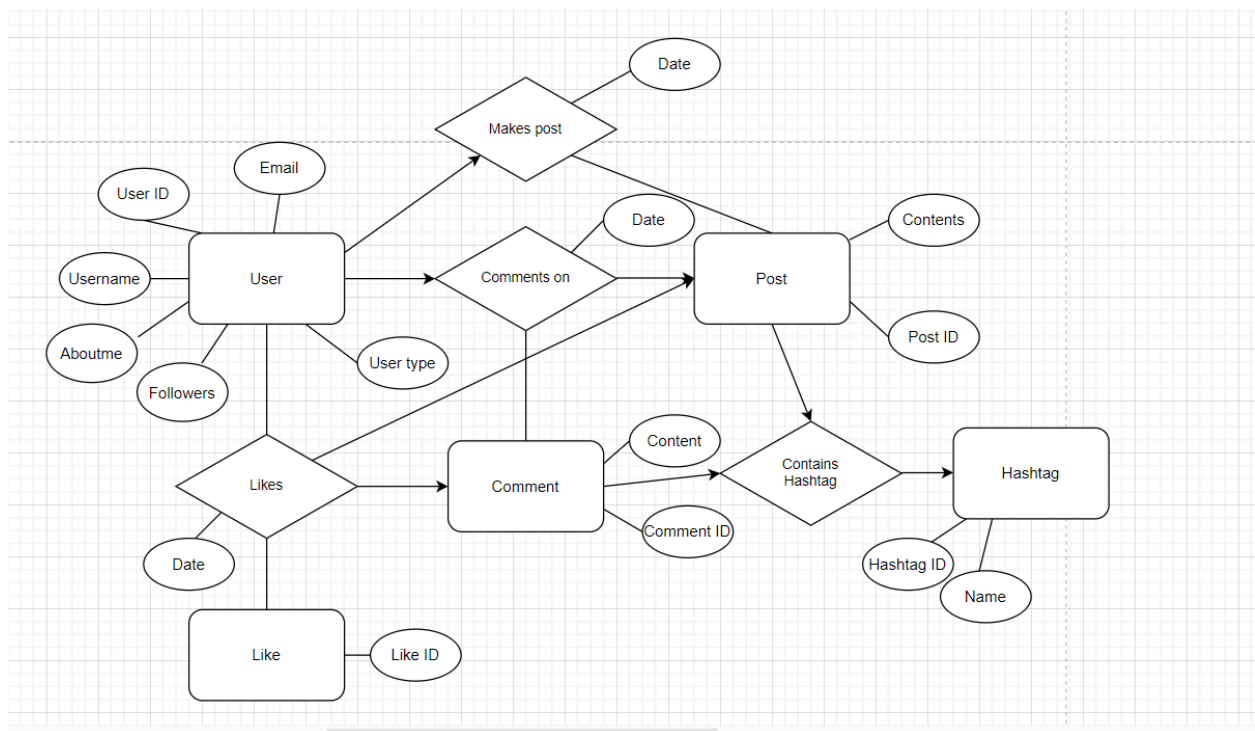
**Like:** A like will have a like ID, the user ID of the user who liked the post or comment, a bool representing whether a post or comment was liked, and the date at which the like was made.

**Hashtag:** A hashtag will have a hashtag id, whether the hashtag corresponds to a post or comment, the post or comment ID the hashtag corresponds to, and the name of the hashtag.

Next, let's list the relationships of the entities:

- User to Post: One-to-Many (A user can make many posts).
- User to Comment: One-to-Many (A user can make many comments).
- Post to Comment: One-to-Many (A post can have many comments).
- User to Like: One-to-Many (A user can like many posts/comments).
- Post/Comment to Like: One-to-Many (A post/comment can be liked many times).
- Post/Comment to Hashtag: Many-to-Many (A post/comment can have multiple hashtags and a hashtag can be associated with multiple posts/comments; the relationship is conditional based on whether the hashtag is associated with a post or comment).

Based on the above relationships, here is the E/R diagram :



### 3. Database Schema and Normalization

We have the following relations in the E/R Diagram:

1. User (user id, username, aboutme, followers, user type, email)
2. Likes (user id, comment id, post id, date)

3. Like (like id)
4. Comments on (user id, post id, comment id, date)
5. Comment (comment id, content)
6. Makes post (user id, post id, date)
7. Post (post id, contents)
8. Contains Hashtag (comment id, post id, hashtag id)
9. Hashtag (hashtag id, hashtag name)

Non-trivial FDs :

1. User : {email -> user id}
2. Likes : {{user id, post id, like id, comment id} -> date}
3. Like :  $\emptyset$
4. Comments on : {{user id, post id, comment id} -> date}
5. Comment : {comment id -> content}
6. Post : {post id -> content}
7. Makes post : {{user id, post id} -> date}
8. Contains Hashtag :  $\emptyset$
9. Hashtag : {hashtag id -> hashtag name}

For all of these FDs, they are either empty or the left hand side is all superkeys. They are all in BCNF.

We can merge our previous relations to get new ones:

1. User (user id, username, aboutme, followers, user type, email)
2. Like (like id, user id, post id, comment id, date)
3. Comment (comment id, user id, post id, content, date)
4. Post (post id, user id, content, date)
5. Hashtag (hashtag id, hashtag name)
6. Contains Hashtag (post id, comment id)

And here are the new non-trivial FDs:

1. User : {email -> user id}
2. Like : {like id -> {user id, post id, comment id}, {user id, post id, like id, comment id} -> date}
3. Comment : {comment id -> {user id, post id, content}, {comment id, user id, post id} -> date}
4. Post : {post id -> {user id, content}, {post id, user id} -> date}
5. Hashtag : {hashtag id -> hashtag name}
6. Contains Hashtag :  $\emptyset$

The FDs for Like, Comment, and Post have changed. However, these FDs are still in BCNF because like id, comment id, and post id are superkeys for their respective FDs.