

ECE356 Project - Social Network

Tian Lun Liu (tliu) #20683297

Rishi Srivastava (r9srivas) #20657657

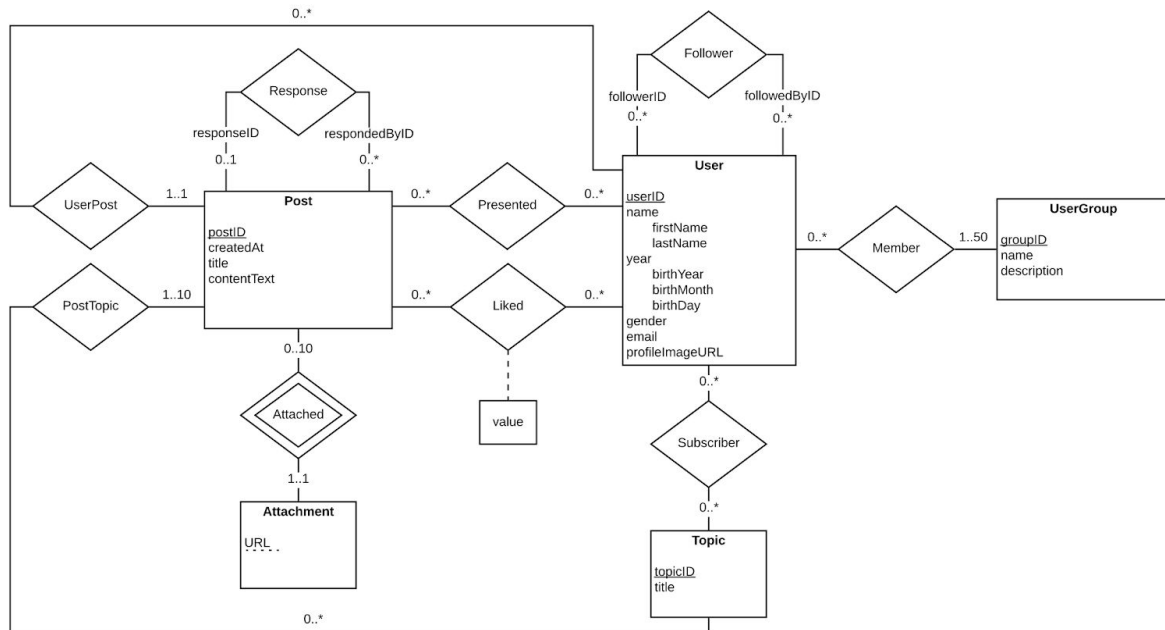
Project Setup

To setup the social network, which includes creating the database, populating its data and initializing the API, run the following command:

```
source setup.sh
```

[Part 1] Schema Design

Entity-Relationship Diagram



Note: Also see ER.png.

Entity Sets

User, Post, UserGroup, Topic

Represents users, posts, groups, and topics on the social network, respectively. Each of these sets' attributes provide a primary key, so each entity may be uniquely specified without reference to another.

Attachment

Represents an attachment on a post, in the form of an URL linking to an image or some other resource. This is a weak entity set with 'Post' as its identifying entity set, since an attachment's existence depends on the existence of the post it's attached to. Its discriminator is the URL attribute, and its primary key is (postID, URL).

Relationship Sets

Member

A user can be a member of any number of groups. On the other hand, a group must have at least 1 member user (total participation), and may have up to 50.

Follower

A user acting in the 'followerID' role may follow any number of other users. A user acting in the 'followedByID' role may also be followed by any number of other users.

Subscriber

A user may subscribe to any number of topics, and a topic may be subscribed to by any number of users.

Presented

A post may have been presented to any number of users, and a user may have been presented any number of posts.

Liked

A post may be liked or disliked by any number of users, and a user may like or dislike any number of posts. In this relationship, 'value' is a descriptive attribute that contains either 1 or -1 to indicate a like or a dislike respectively.

Response

A post acting in the 'responseID' role may be a response to either 1 or 0 posts. That is, it may or may not be a response. A post acting in the 'respondedByID' role may have any number of other posts responding to it.

Attached

This is an identifying relationship relating the weak entity set 'Attachment' to its identifying entity set 'Post'. An attachment must be attached to exactly one post (total participation), but a post may have between 0 to 10 attachments, inclusive.

UserPost

A post must be created by exactly one user (total participation), but a user can create any number of posts.

PostTopic

A post must be about at least one topic (total participation), and may be about up to 10 topics. A topic, however, can have any number of posts.

[Part 2] Database Setup

See `setup.sql`.

[Part 3] Data Generation

Data was generated in the form of SQL insert statements. Please see `gen_data.py` as well as `insertData.sql`.

[Part 4] API

Usage

The methods available in the API are listed in the table below. They can be accessed from the project directory as follows:

```
social-api <api-method> <args>
```

NOTE: List arguments must be passed in in the form `[item1,item2,item3]`, with no whitespace outside of the actual list item.

NOTE: The first time the API is used successfully, your password and socket will be cached.

NOTE: You can use the default socket by pressing enter when prompted to input the socket. If you experience difficulty with the default setting, you may try entering the socket `/tmp/mysql.sock` when prompted.

See `demo.mov` and `api.py`.

Method	Input
post <i>Adds a post from a user</i>	userID: int, topicIDs: [int], title: string, contentText: string, attachmentURLs: [string]
respond <i>A user responds to a post with a post, RespondedByID is the postID to be responded to.</i>	userID: int, respondedByID: int, topicIDs: [int], title: string, contentText: string, attachmentURLs: [string]
subscribe_topic <i>A user starts subscribing to a topic</i>	userID: int, topicID: int
follow_user <i>A user starts following another user</i>	followerID: int, followedByID: int
join_group <i>A user joins a group</i>	userID: int, groupID: int
like <i>A user responds to a post with a value of +1 or -1</i>	userID:int, postID:int, value: int
read <i>Mark a post as read</i>	userID: int, postID: int
get_unread_posts <i>Fetches unread post from a user's followed users and subscribed topics</i>	userID:int, maxResults:int
get_likes <i>Returns userIDs of users who reacted to a post and the value of the reaction (+1/-1)</i>	postID: int
get_post <i>Returns Post record for a given postID</i>	postID: int
get_responses <i>Returns responses as Post records for a given post</i>	postID: int
get_followers <i>Returns userIDs of followers of a user</i>	userID: int
get_members <i>Returns userIDs of members of a group</i>	groupID: int
get_subscribers <i>Returns userIDs of subscribers to a topic</i>	topicID:int

get_attachments <i>Returns attachment URLs for a post</i>	postID: int
get_user <i>Returns User record for a given userID</i>	userID: int