**General Notes**

**How Posts Work**

* Posts have a post ID, head, type, content, image, rating, username, and group ID
* The primary key for posts is an ID integer that increments automatically. The postID will be created when the post is created.
* Post ID 1 is am empty post that is used as the head for all main idea posts. (This is in the database with null value for content)
* The jsplumb notes document elaborates on how posts will be displayed in the mind map view.
* To display posts in the newsfeed, query all posts with head ID 1 (and any other criteria ie: by username or belonging to group) and display each in it's own div using a for loop.

**How Groups Work**

* Groups have an ID, creator, name, and description.
* Similarly, Group ID's 1 and 2 are designated as special. ID 1 denotes an everyone privacy setting, group 2 indicates a personal setting.
* Using if statements, any post set to group 2 can only be viewed by the user who made the post.
* We must ensure that all child posts get the same group ID as their head posts.
* The in\_group table tracks usernames and associated groups.

**How To Query the DB**

* The latest SQL file has all of the DB save to it. I recommend importing a copy of this to your MySQL Workbench and working in MAMP or XAMP as it is faster than constantly uploading in filezilla, and you can more easily watch what is happening in the DB.
* We are using PHP's PDO database functions, as they are the most secure.
* Learn more about PDO here <http://php.net/manual/en/book.pdo.php>
* Here is an example of a query

//start with a statement variable. Here, $dbh is the database name

//we are using prepared statements here. Any variables you wish to //pass into your html statement should use :var

$stmt = $dbh->prepare('SELECT \* from users WHERE username=:name');

//bind the :var to the $var name

$stmt->bindParam(':name', $username);

//execute your statment

$stmt->execute();

//check if anything was returned by the query

if ($stmt->rowCount() > 0){

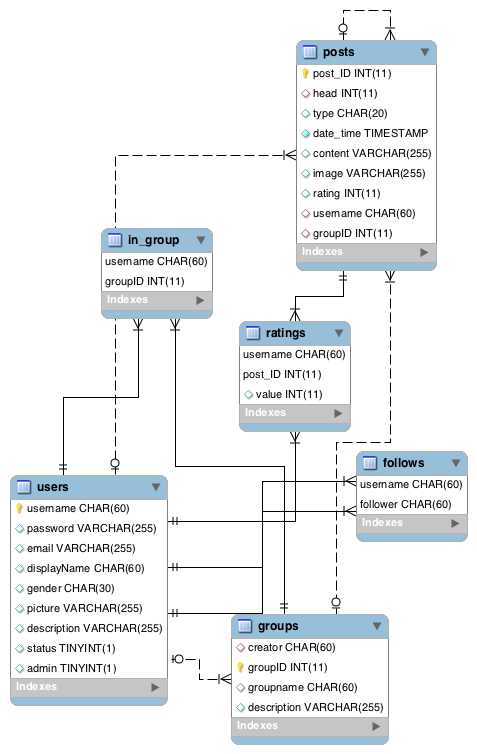
//use fetch to fill $check with an array of each db value

$check = $stmt->fetch(PDO::FETCH\_ASSOC);

$pword = $check['password'];

}

* This code runs a query in the users tale and fills an associative array named $check with the database columns and their associated values. If you are expecting more than one row from the database to return, you will need to use a loop.
* This ER diagram indicates all DB tables and attribute names. I will update it if anything is changed/added



* Inserting into the DB is similar. Always use prepared statements to query the database to prevent users from being able to insert their own strings and break things

//we hash passwords using crypt so they do not store as plain text

$hash = crypt($pass, '$6$rounds=5000$4Ds0.2.A.F\*pPi(8lxZ+H!3#l+s@wlek.!ls-$');

$blank='';

$admin = 0;

$zero = 0;

//all values must be passed as variables in prepared statements

$stmt = $dbh->prepare("INSERT INTO users (username, password, email, displayName, gender, picture, description, status, admin)

VALUES (:username, :password, :email, :displayName, :gender, :picture, :description, :status, :admin)");

$stmt->bindParam(':username', $username);

$stmt->bindParam(':password', $hash);

$stmt->bindParam(':email', $blank);

$stmt->bindParam(':displayName', $dname);

$stmt->bindParam(':gender', $gender);

$stmt->bindParam(':picture', $blank);

$stmt->bindParam(':description', $blank);

$stmt->bindParam(':status', $zero);

$stmt->bindParam(':admin', $admin);

$stmt->execute();