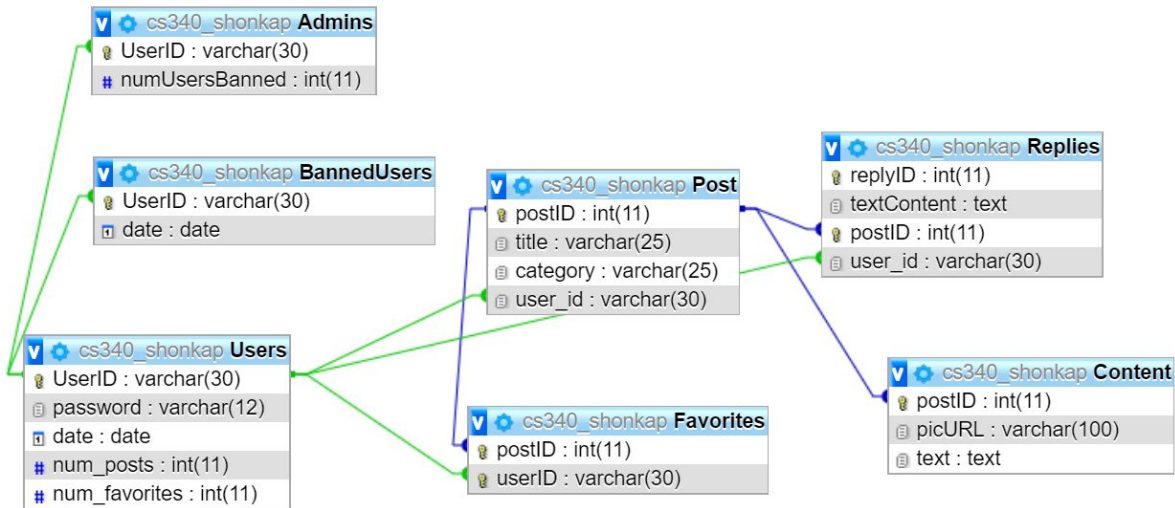## Structure of Tables:

Queries:

Users can click on a category and display the posts in that category with
SELECT title FROM Post WHERE category='$category_selected'
and $category_selected is passed in from the php.

SELECT * FROM Users WHERE UserID='$inputUser' AND password='$inputPass'
Used to login a user. If a tuple is returned, they can proceed, if not then there will be an error.

Var = (SELECT postID FROM Post WHERE postID > ALL(SELECT postID FROM Post))
+ 1;
INSERT INTO Post VALUES (var, '$title', '$category')
Used to create a new post and assign a new post id based on passed in values title and
category.

UPDATE Admins
    SET numUsersBanned = numUsersBanned + 1
    WHERE UserID = admin;
Used to update the admins count of user banned when they ban another user.

Triggers:

```
CREATE FUNCTION banUser(admin VARCHAR(12), user VARCHAR(12) )
RETURNS INT(1)
AS BEGIN
      INSERT INTO BannedUsers (UserID, date)
      VALUES (admin, CURDATE());

      UPDATE Admins
      SET numUsersBanned = numUsersBanned + 1
      WHERE UserID = admin;

      DELETE FROM Favorites WHERE UserID = user;
END

CREATE TRIGGER addBanned
AFTER INSERT
ON BannedUsers
BEGIN
UPDATE Users
SET UserID = SELECT CONCAT('[Banned-User]', UserID) AS
ConcatenatedString
WHERE UserID = new.UserID;
END
```

The above function and trigger add the functionality for banning users from the site. When an admin wants to ban a user, the function will be called. This adds the user to the banned user list, increases the admins banned user count and deletes the banned users favorites list. When the user is added to the banned user table, the trigger updates the post to show their name as 'Banned-User'.

How SQL is used in the website:

Home Page

Users can search through categories and posts on this home page so the SQL here needs to add that functionality:

`SELECT * FROM Posts WHERE title LIKE '%$search_word%'`

With this functionality, the user can only search for keywords within the title of the post. For example a user could search for 'CS160' and would be shown a list of posts with 'CS160' in the title.

`SELECT title FROM Posts` is used to display all the posts.

Users can click on a category and display the posts in that category with

`SELECT title FROM Posts WHERE category='$category_selected'`

and $category_selected is passed in from the php. This will be the most used post retrieval method as the homepage shows posts grouped by category. And the easiest method of searching will be selecting a category from the list.

To display the category list as shown above we will use

`SELECT DISTINCT category FROM Posts`

Login Page

The SQL for this page will retrieve the tuple for the user specified from the login. If there isn't one then the javascript will handle alerting the user. The username and password will have to be passed in to the SQL. Because username is a primary key and has to be unique, there should never be more than one tuple returned from this query.

```
SELECT * FROM User WHERE username='$inputUser' AND password='$inputPass'
```

Sign Up Page

CheggButFree.com
Sign Up!

User ID: [   ]

Password: [   ]

Submit

Users are added to the User table with

```
INSERT INTO User (UserID, password, date,numberPosts,NumFavorites) VALUES
('$user_id', '$password', '$date','$numberPosts','$numFavorites')
```

The $user_id and $user_pw are variables passed in from the php file. The date joined is automatically taken from the current date. The num posts and num favorites are initialized to 0.

Account Page

The information is displayed with

```
"SELECT UserID, date, num_posts, num_favorites FROM Users WHERE
UserID='$currentId' "
```

Where the current user is passed in.

Admin Account page

This page will retrieve the admin account info for the current signed in user. The current user will need to be passed in to the SQL. The info on the users banned will be displayed in a table for the admin to see. This page is the same page as the normal user account page but if they are an admin (the query to Admins is not empty) it shows the table of banned users.
http://web.engr.oregonstate.edu/~ulepk/cs340_hw_db/UserAccount.php

```
SELECT * FROM Users WHERE UserID='$currentId'
```

```
SELECT * FROM Admins WHERE UserID = '$currentId'
```

A user is banned by using the function created BanUser.
```
"SELECT BanUser('$admin', '$userToBan') AS BanUser"
```

$admin, $userToBan is passed in from the current user and a text input.

```
DELIMITER $$
CREATE DEFINER=`cs340_shonkap`@`%` FUNCTION `BanUser`(`admin` VARCHAR(30), `user` VARCHAR(30))
RETURNS int(11)
    NO SQL
BEGIN
      UPDATE Admins
      SET numUsersBanned = numUsersBanned + 1
      WHERE UserID = admin;

      DELETE FROM Favorites WHERE UserID = user;

    INSERT INTO BannedUsers (UserID, date)
VALUES (user, CURDATE());
```

```
      RETURN 1;
END$$
DELIMITER ;
```

Create Post

This page will be used submit a new post which will be sent to the database. The title, category, content and the image path will have to be passed in. This will likely be done as a trigger so the numpost value can be updated.

http://web.engr.oregonstate.edu/~garciros/cs340/cs340_hw_db/newPost.php

```
Var = (SELECT postID FROM Posts WHERE postID > ALL( SELECT postID FROM
Posts)) + 1;
INSERT INTO Posts (postID, title, category) VALUES (var, '$title',
'$category')
INSERT INTO Content (postID, picURL, text) VALUES (var, '$imgPath',
'$content')
INSERT INTO Replies (replyID, textContent, postID) VALUES (var,
'$textcontent', var_postID)
```

Reply To a Post

       Favorites   My Account   Logout

(Title:) Question about SQL Homework   By: User45

Question 4: some questions about junk

Question 3: more questions about some more junk

User2: some words and stuff   6   7

    User3: some more writing stuff   6   7

Reply to User2:

Reply

This page will be used submit a new comment which will be sent to the database. The original post ID and the content will have to be passed in.

```
Var = (SELECT replyID FROM Replies WHERE replyID > ALL( SELECT replyID FROM
Replies)) + 1;
INSERT INTO Replies VALUES (var, $content, $postID)
```

View a Post

This functionality will display a single post and its replies. The post ID will have to be passed in.

```
SELECT * FROM Posts WHERE postID='$postID'
SELECT * FROM Replies WHERE postID='$postID'
```

Favorites

The user is also able to add posts to their favorite lists by clicking a button on the post. The username of the current user and the post id will have to be passed in.

```
SELECT p.title, u.UserID FROM Posts p, Users u, favorites f where (f.User ==
u.UserID) and (p.postID == f.postID);
```

```
INSERT INTO Favorites VALUES('$postID', '$userID')
```

A user can view their favorited post titles and their posters with:

```
SELECT Post.user_id, Post.title FROM Favorites, Post
        WHERE Favorites.postID = Post.postID
        AND Favorites.UserID = '$currentId'
```

$currentId is passed in as the current user logged in.