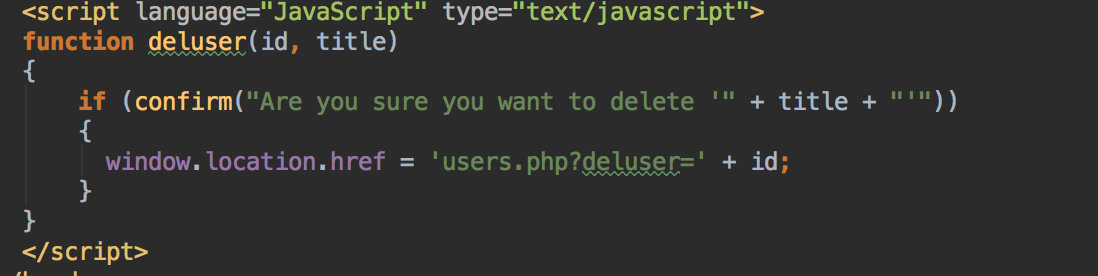
1. The project should be written using PHP and MySQL
   * Project uses PHP, HTML, MySQL, and features JavaScript and CSS elements.
2. You may add css and/or javascript, but the emphasis in this course is on database programming.

-Below is an example of a menu option made available with JavaScript.

Line 22-30 in blogsample/admin/members.php



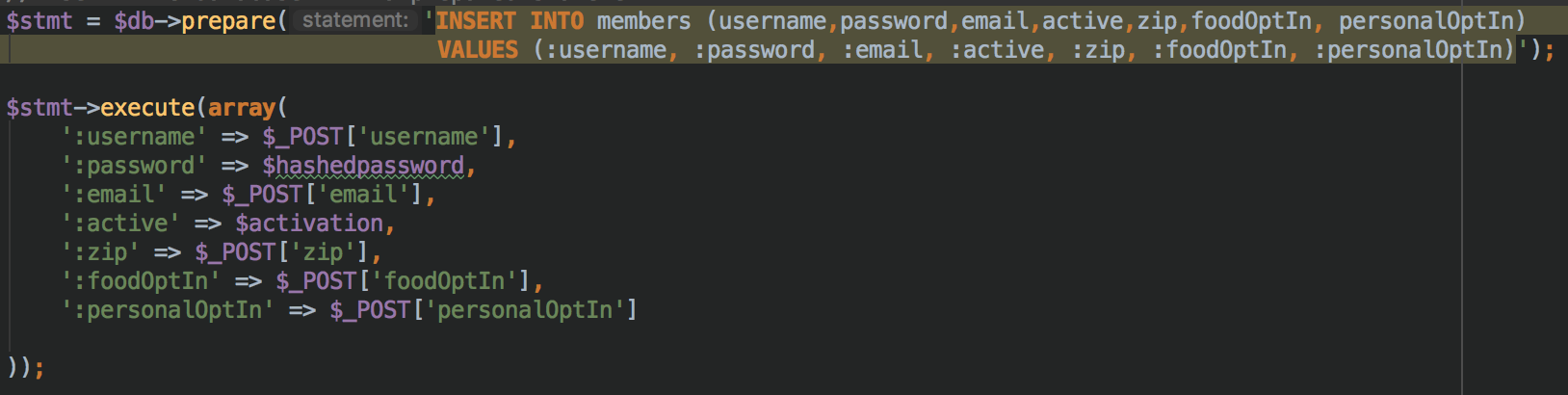
1. Include at least one structural change to the database using DDL.  (Create, Alter, Drop)

-Enclosed with the application is the DDL to create the database, which features Creates.

1. Provide the INSERT SQL for initially populating the database.

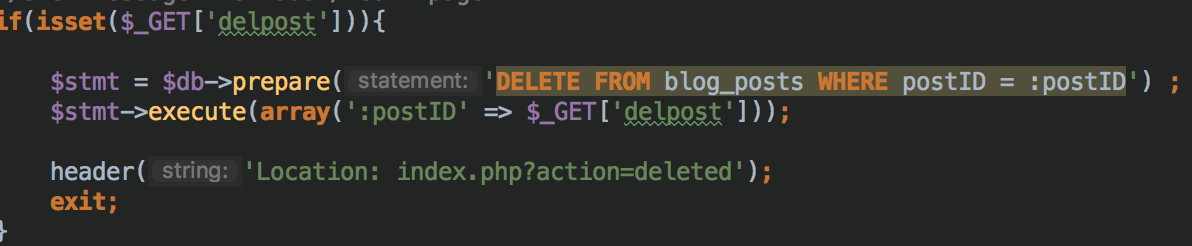
-Inserts are used in the DDL to populate fields.

1. Include at least one INSERT that will occur during the execution of the application.  This will most likely be as the result of a transaction or user that should be added to the database.

-This insert is used a new member into the members database when that members registers. It collects their username, password, email, activation status, zip, and blog preferences. Lines 70-82 of login/index.php

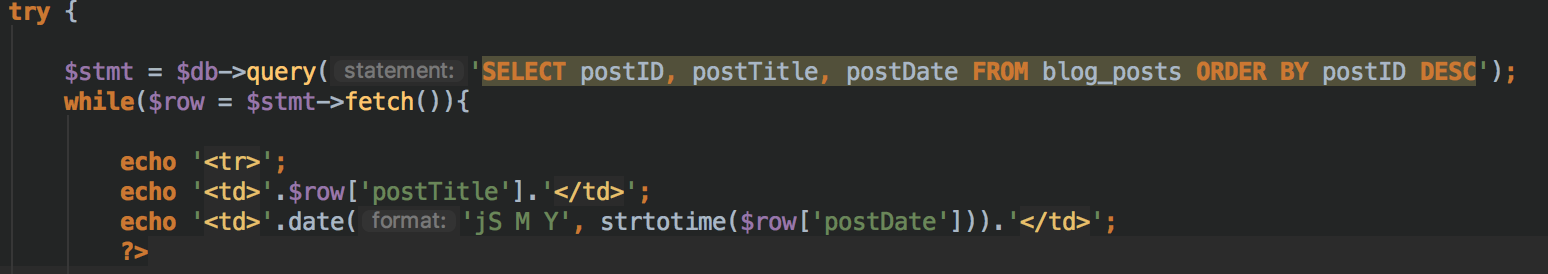
1. Include at least one DELETE.

-This query is ran when the javascript button is pressed and the delpost get is sent. It deletes the post with the associated post ID from the blog\_posts database.

Lines 9-16 in blogsample/admin/index.php

1. Include at least one simple SELECT SQL statement.

-Populates the “Blog” page which is the blogsample/admin/index.php page.

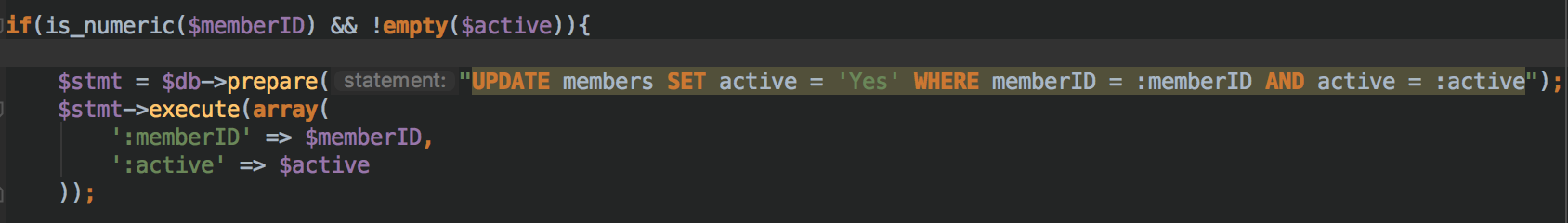
Lines 56-64

1. Include at least one SELECT using an aggregate function.

-Counts and displays the number of blog posts in the database. Lines 84-97 in blogsample/admin/index.php



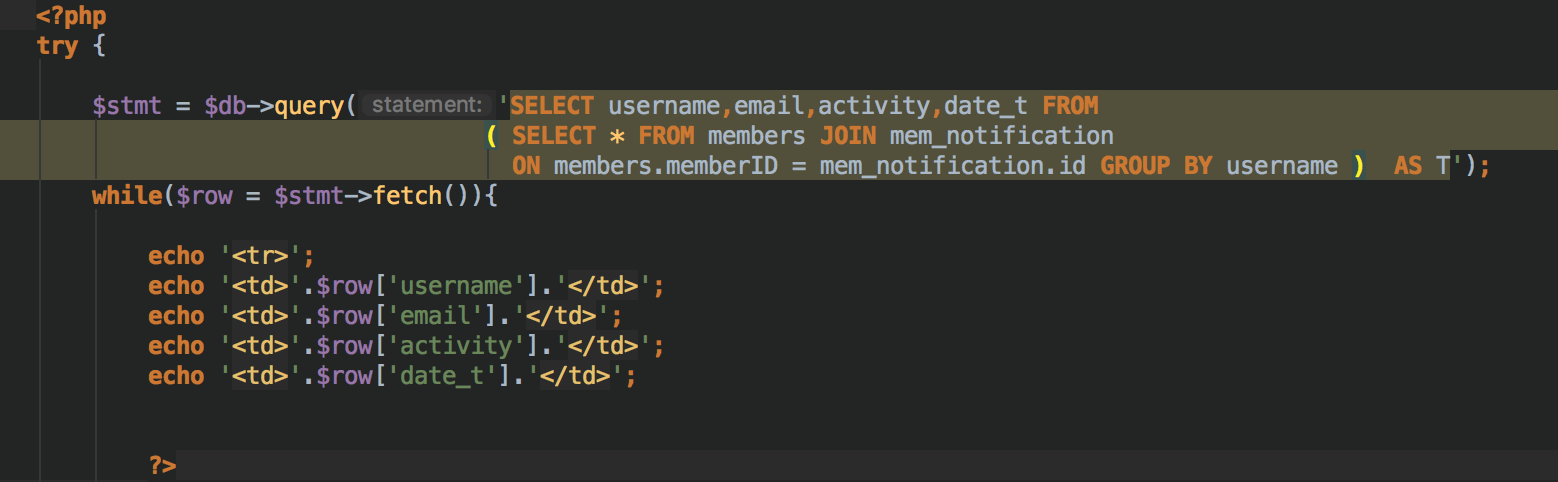
1. Include at least one SELECT using a compound condition.

-This query update the members by setting their active status to “Yes” when they click their activation link in their registration email. Lines 8-14 in login/activation.php

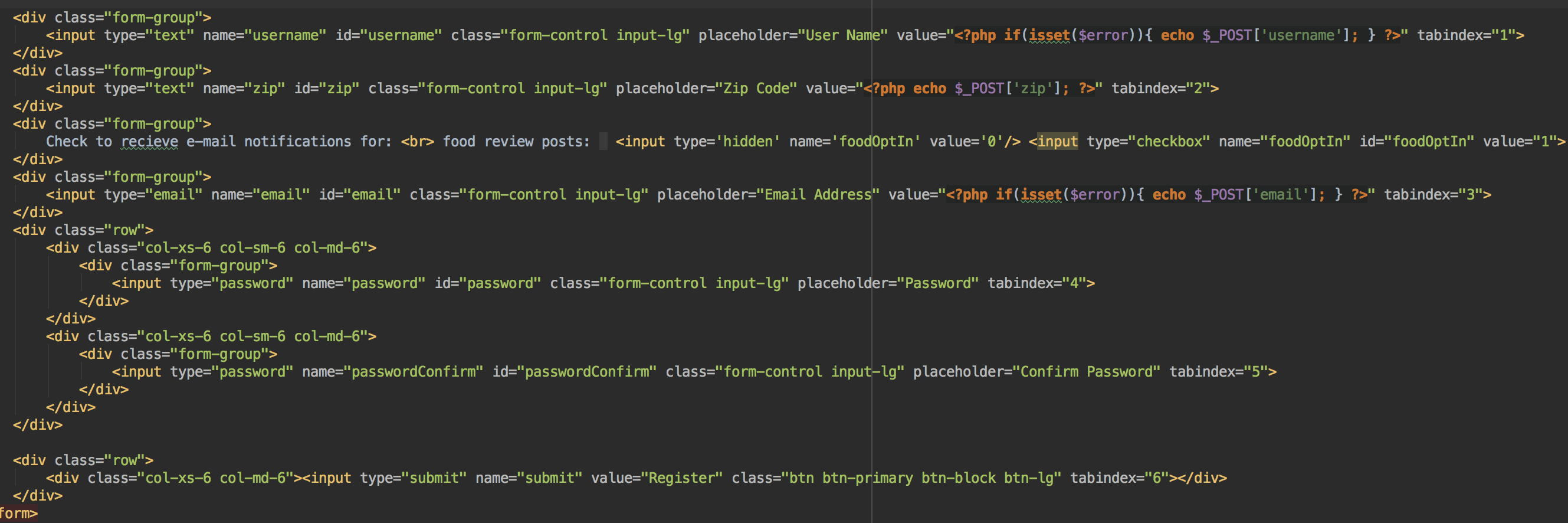
1. And 12.

Include at least one JOIN query.  This can be an inner or outer join.  It will be probably be an inner join. Include a subquery.

-This query contains both a join and a subquery. This query selects username, email, activity status, and date from a subquery. The subquery selects all from the members table, and is joined on members.id = mem\_notification.id. These columns, while in different tables, share the same member id information and reference one another via this column. The results are grouped by username to get rid of duplicates. Duplicate usernames are not allowed in the members database. Entries in the mem\_notification table are created via a trigger that “fires” when a new user registers. The table tracks the member id, date, and holds a message that “A new user has registered!”

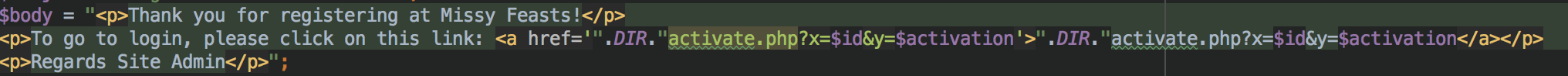
Lines 48-62 of blogsample/admin/members.php

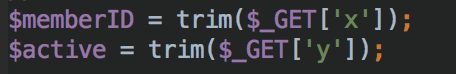
13. Use a form to collect user data.  Pass variables to the next page (or postback) using POST.

-This form (Lines 146-175 in login/index.php) is where the user can enter information in order to register for an account. The information here is collected and entered into the database by the query from number 5 above.

14. Use one GET to pass data to another page.

-When a user is sent their registration email, the user id and activation code is put into the url of their activation link. When the click the link, a GET is used to pull that information from the url and the information is checked with the database, if correct they become active.

Lines 90-92 and Lines 5-6 of login/index.php and login/activate.php respectively.



15. Populate a field on a form or table from the database.

-When the admin clicks edit post, they are brought to the post form, and the title, description, and contents of that post are filled into their associated forms from the database.

Lines 112-126 in blogsample/admin/edit-post.php

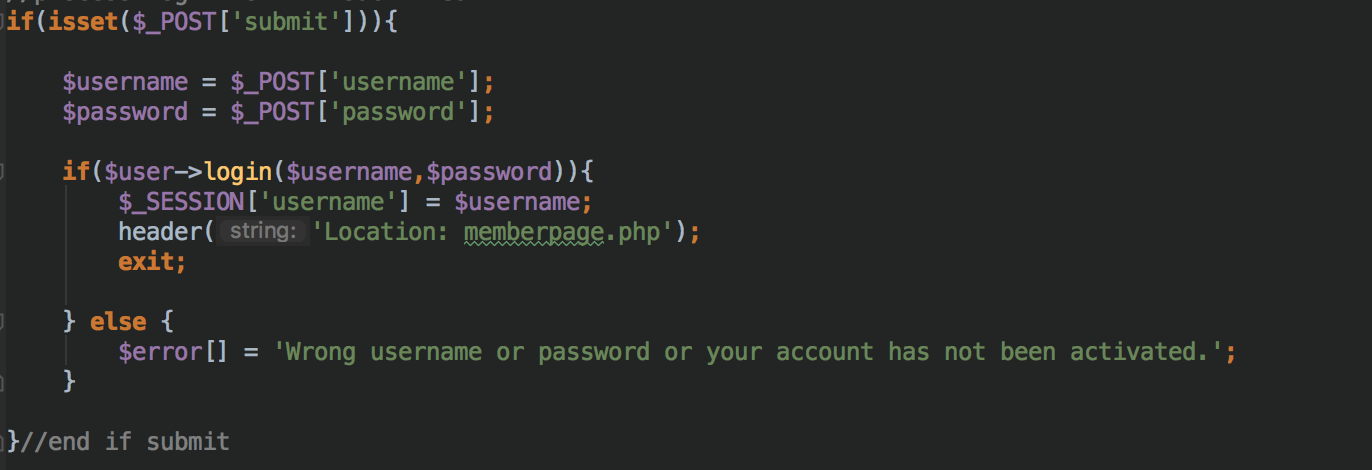
16. Login and password.  Extra credit if you use bcrypt.

Login/phpmailer.user.php

In this file the password is encrypted and hashed with bcrypt.

User enters username and password to login.

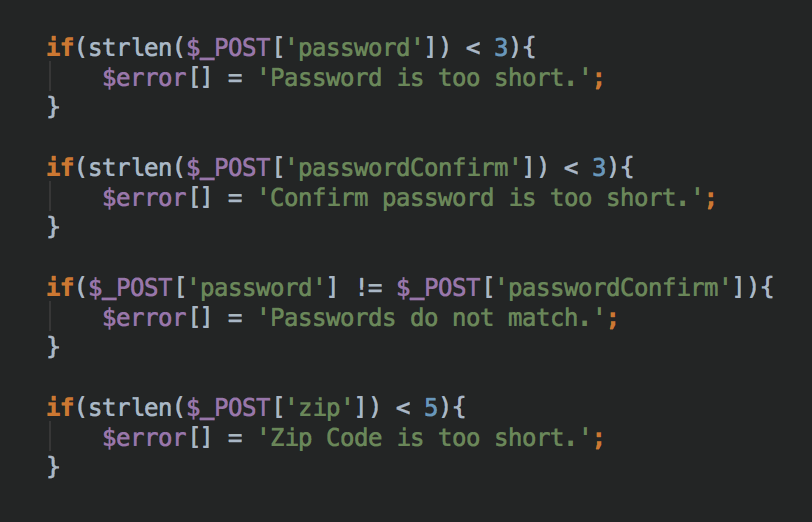
Lines 6-23 in login/login.php



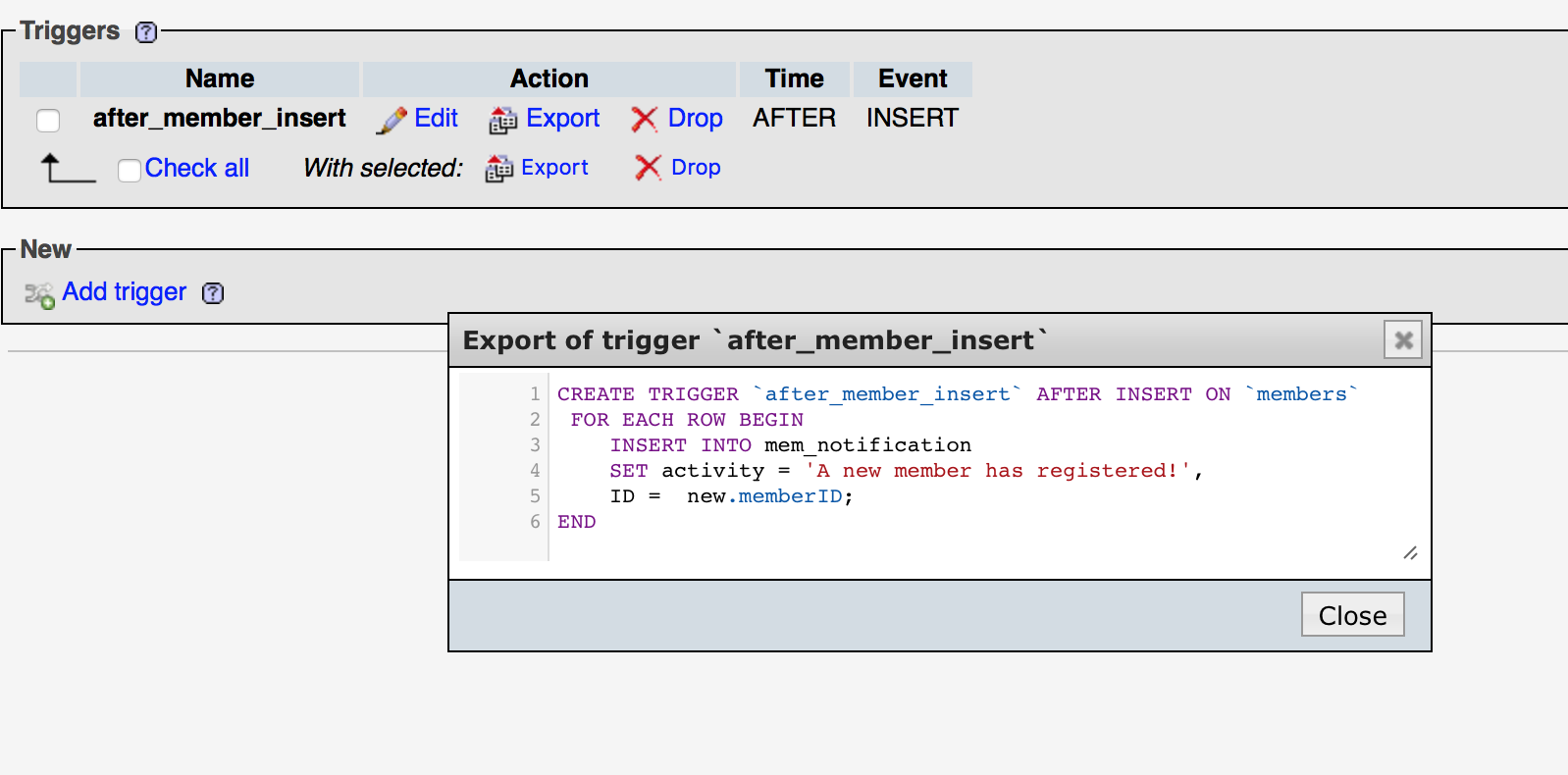
17. Check for empty data fields.

-An example of some of the checks used to ensure that user enters data when registering, and receives an error if they miss a field.

Lines 25-39 of login/index.php



18. Implement one trigger.  Demonstrate what happens when it is violated.

-Trigger that is used as mentioned previously. This trigger “fires” after a new user is inserted into the members table.

19. Implement referential integrity.  Demonstrate what happens when it is violated.

-This foreign key constraint ensures that if a member in the member table is deleted then the field with the row with the related member id in mem\_notification is deleted as well.