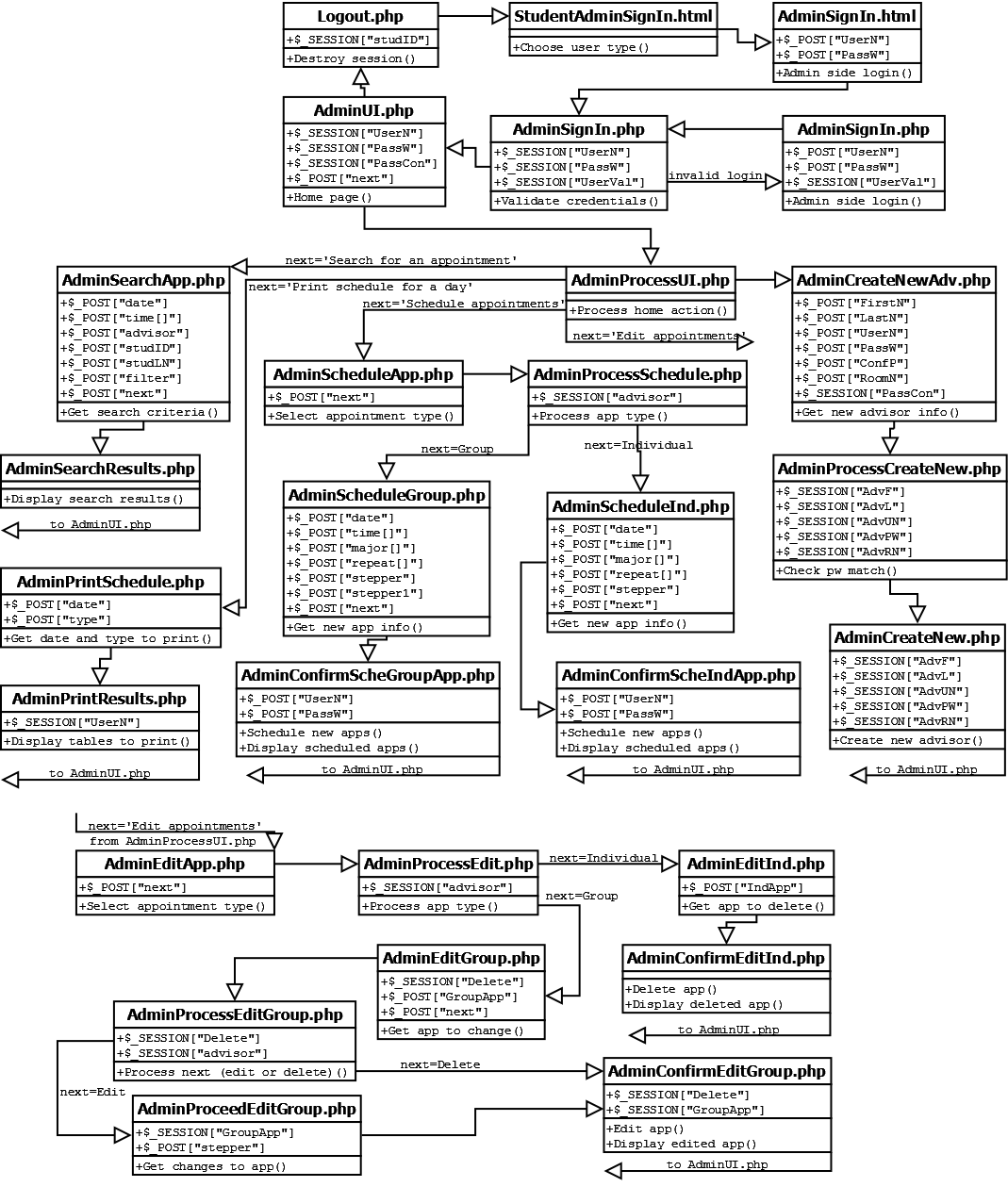
# Webpage Map



# Updated Items

## Abbreviations for Major Names

In order to save space in the database, the Proj2Appointments table was updated so that the Major column stored abbreviations for the majors instead of the full major names (Figure 1).

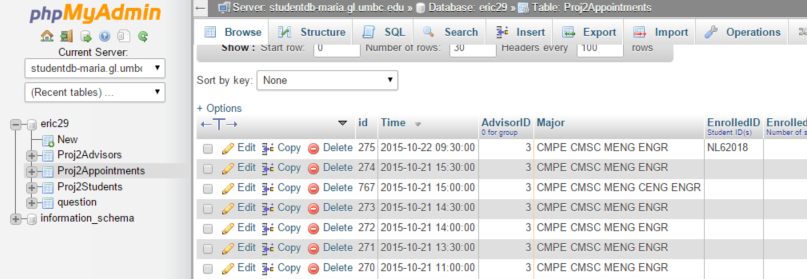
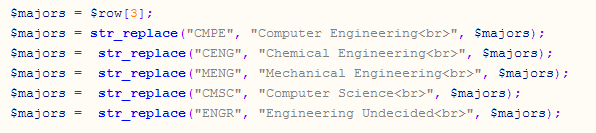
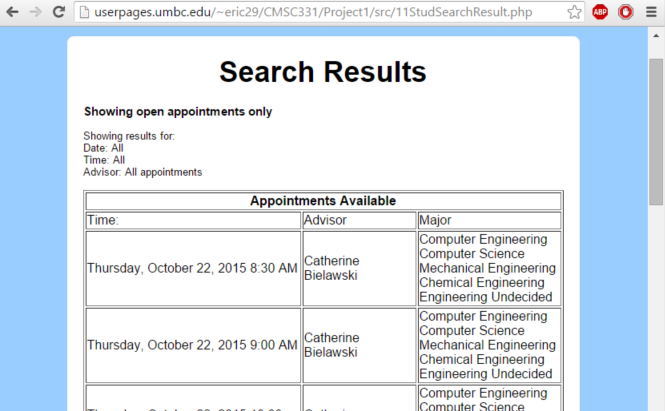


Figure : Proj2Appointments table

However, all pages on the website that display the major will still display the full name, as shown in figures 2 and 3.

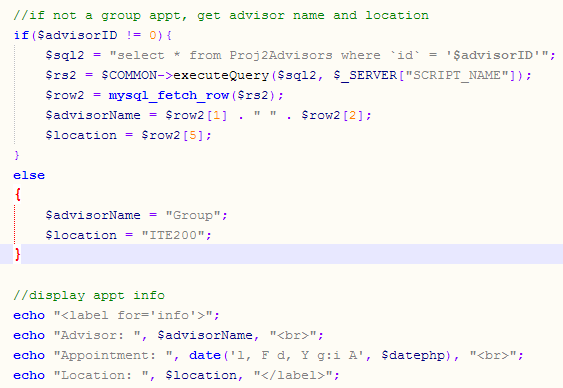


*Figure 2: 11StudSearchResult.php code*



*Figure 3: 11StudSearchResult.php web page*

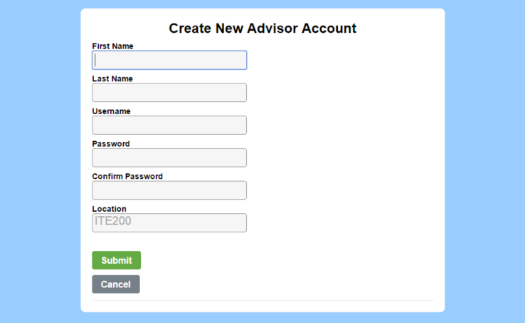
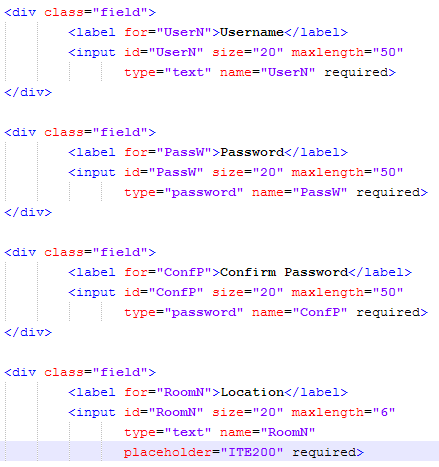
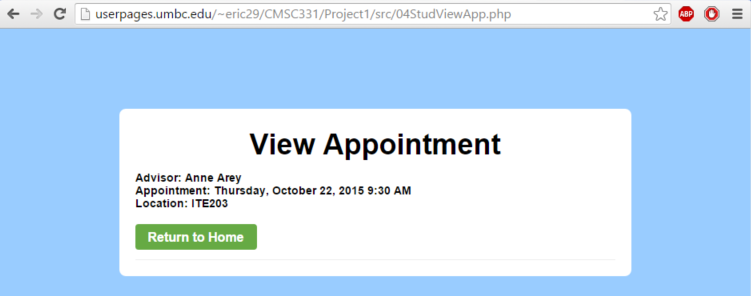
## Advisor’s Office Location

The Proj2Advisors was updated with a column containing the location of each advisor’s office (Figure 4). The location is now displayed whenever a student views, schedules, or cancels an appointment (Figures 5 and 6).

*Figure 5: 04StudViewApp.php code*

On the Administrator side, a field was added to the Create New Advisor page to enter the new advisor’s office location (Figures 7 and 8).

*Figure 4: Proj2Advisors table*



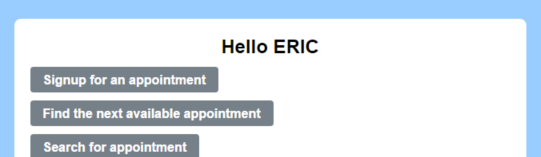
*Figure 7: AdminCreateNewAdv.php code*

*Figure 6: 04StudViewApp.php web page*

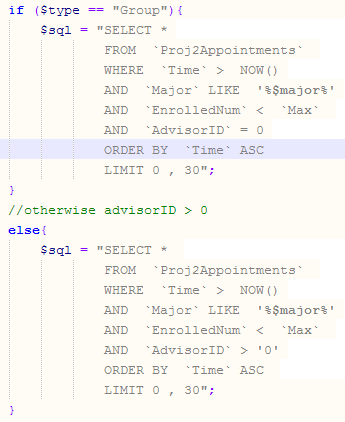
*Figure 8: AdminCreateNewAdv.php web page*

# Additional Items

## “Find Next Appointment” Button:

A “Find Next Appointment” feature was added to the Student side of the site. This feature appears on the home page when the student does not yet have a scheduled appointment (Figure 9).

*Figure 9: 02StudHome.php web page*



It will first ask the student whether they would like to sign up for a group appointment or individual appointment. Then it will find the soonest open appointment of that type for the student’s major. Figure 10 shows the MySQL query that will get the next appointment.

The function NOW() in the MySQL query will automatically fetch the current date and time in the correct format to compare with the values that are in the table. Sorting the table in ascending order by time will put the soonest appointment at the top of the list.

*Figure 10: 02StudHome.php web page*

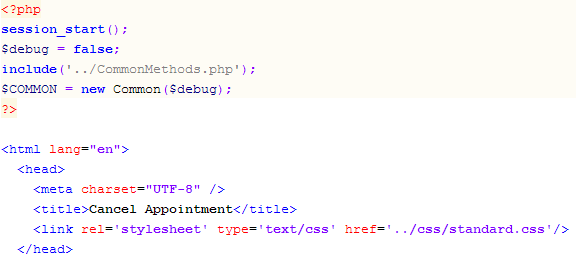
## More Coding and Documentation

*Figure 11: 11StudSearchResult.php code before commenting*

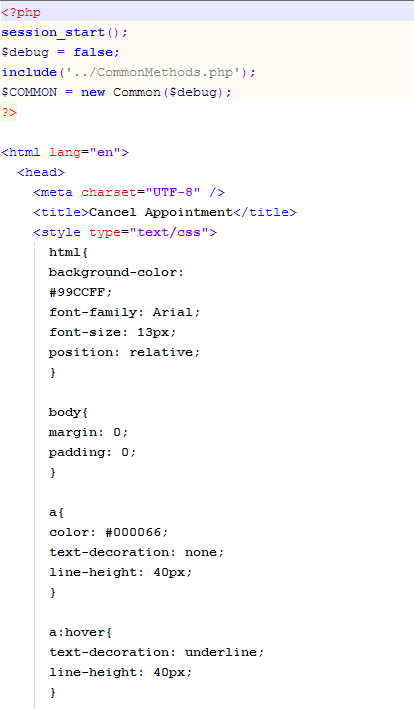
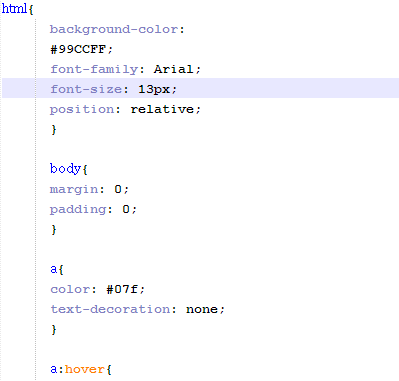
*Figure 12: 11StudSearchResult.php code after commenting*

Prior to starting the project, most of the php code had no comments. Most files are now fully commented and documented, to the extent depicted in figures 11 and 12.

## Move CSS to standalone page

Many files initially had their own CSS styling that was almost identical to the standalone page, standard.css. All redundant styling was removed, and those files were linked to standard.css.

*Figure 14: 05StudCancelApp.php code linked to standard.css*



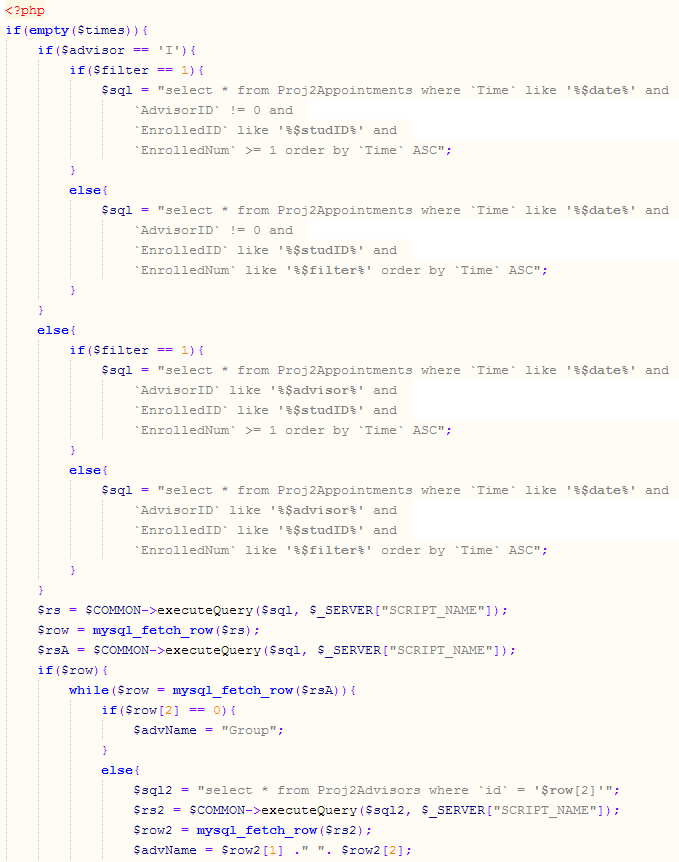
*Figure 15: 05StudCancelApp.php code original*

*Figure 13: 05StudCancelApp.php code original*

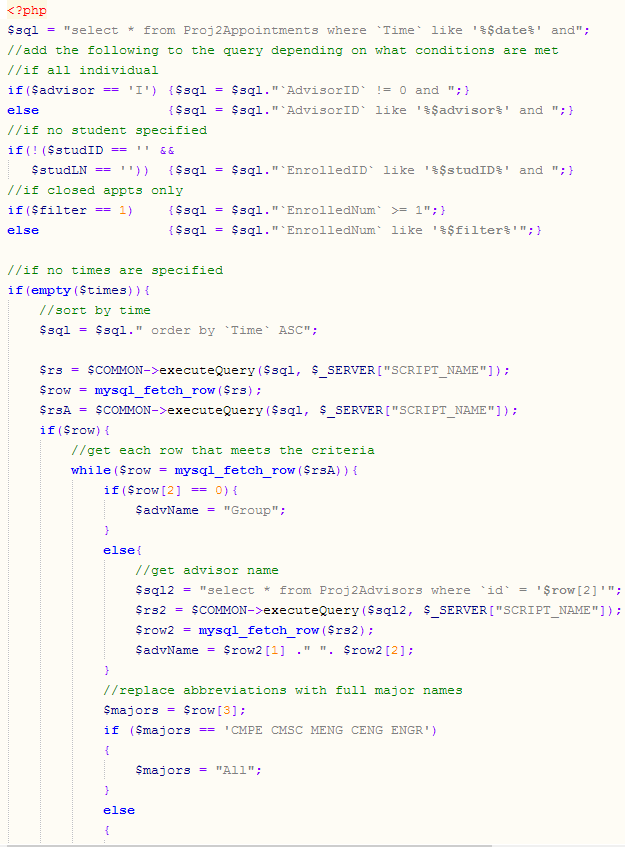
## Merge/remove redundant SQL Queries

Some files contained redundant or unnecessary code that increased line count and code complexity. One example of this was the file AdminSearchResults.php. Initially this file used nested if statements to decide which MySQL query to execute (Figure 16). It has been simplified so that the necessary criteria are concatenated with the query string when the associated condition is met (Figure 17). For example, if $advisor == ‘I’ (meaning the user specified that they want to search for individual appointments only), then the string “`Advisor` >= 0 and “ will be concatenated onto the end of the MySQL query. This made the code much more readable and reduced the line count my almost 100 lines.

Project video: <https://youtu.be/Q724sEPPjMc>



*Figure 16: AdminSearchResults.php code original*



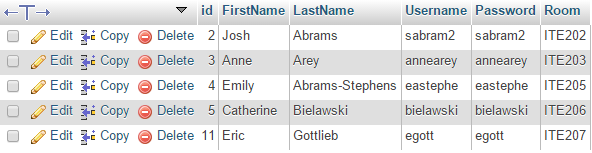
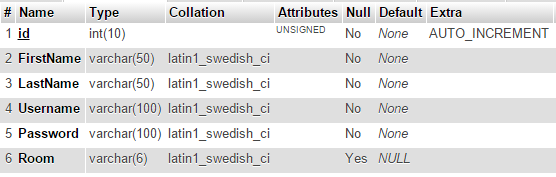
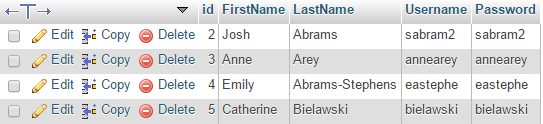
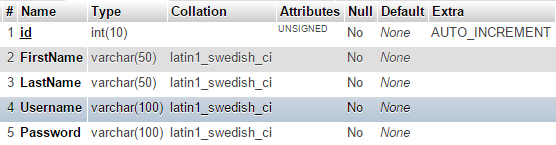
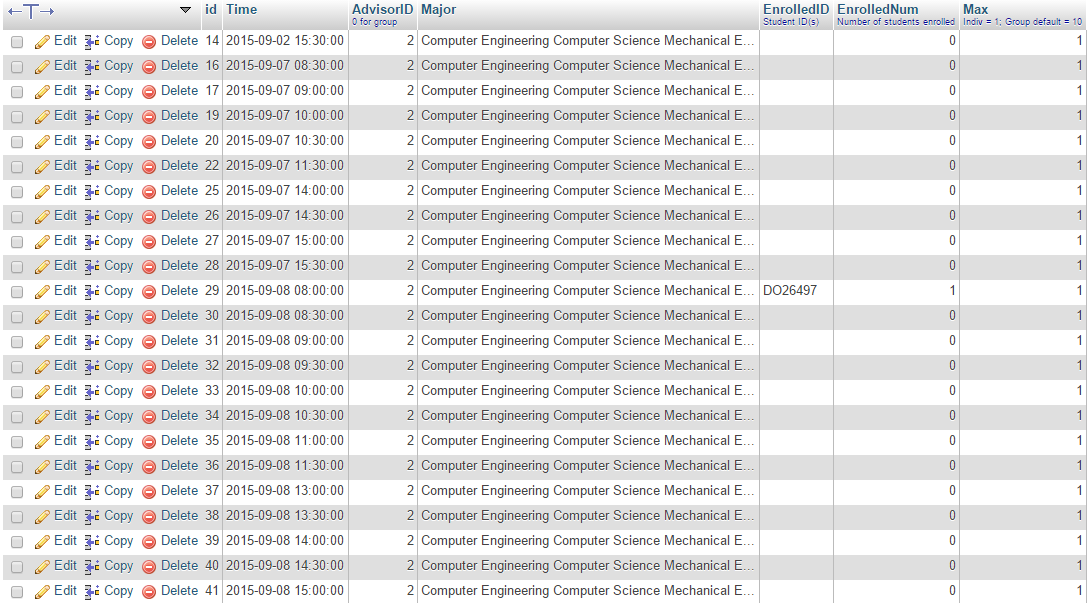
*Figure 17: AdminSearchResults.php code after simplification*

# MySQL Table Setup

The following section shows the changes that were made to the MySQL database tables to support the updates and additional features that were made to the website. The biggest challenge I encountered was implementing the code to convert the major abbreviations to their full names and back correctly and efficiently. I wanted to keep the code simple and avoid loops. In most places, I used the str\_replace function to replace each major abbreviation with its full name while keeping the rest of the string unchanged.

*Figure 19: Modified Proj2Advisors table data*

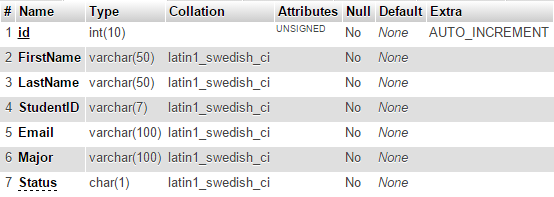
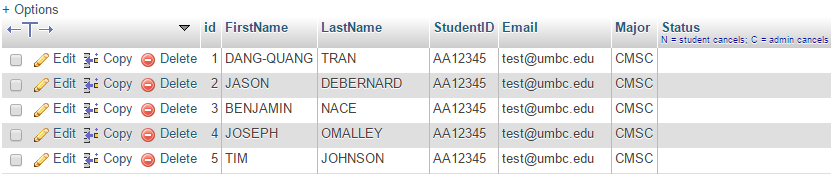
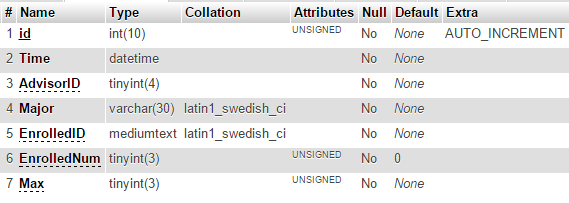
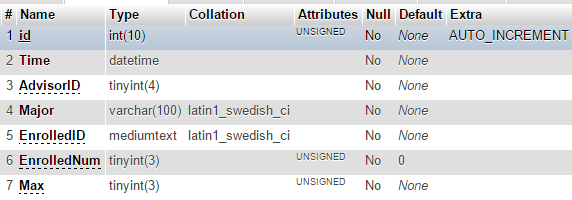
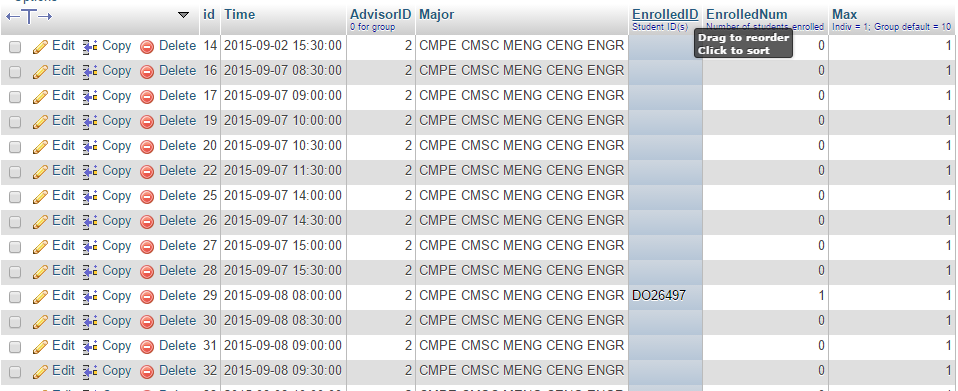
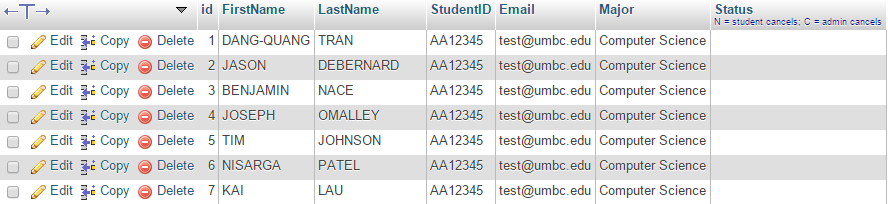
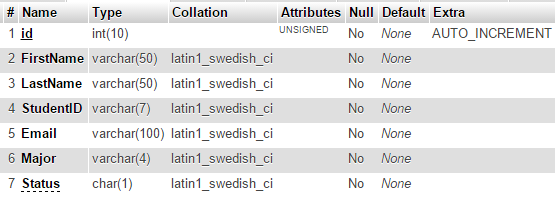
*Figure 18: Original Proj2Advisors table data*



*Figure 22: Original Proj2Appointments table data*

*Figure 21: Modified Proj2Advisors table structure*

*Figure 20: Original Proj2Advisors table structure*



*Figure 28: Modified Proj2Students table structure*

*Figure 26: Original Proj2Students table data*

*Figure 25: Modified Proj2Appointments table data*

*Figure 23: Original Proj2Appointments table structure*

*Figure 24: Modified Proj2Appointments table structure*

*Figure 27: Original Proj2Students table structure*

*Figure 29: Modified Proj2Students table data*