

## CIS 1152 - Lab #7 SQL via PHP

S. Ruegsegger

Modified by Peter Chapin with permission

Last Revised: 2023-03-14

### Objective

Write PHP code to create HTML using data in your MySQL DBMS.

### Task 1 – Enhance the course schedule database

Consider the table you made in the previous lab that contains information about your personal class schedule. Enhance that table by first creating a second table, called `student`, that lists student ID numbers and the corresponding student names. I suggest separating the first and last names and don't worry about names that don't fit that pattern. Declare the student ID numbers as the primary key of that table by adding `PRIMARY KEY` to the end of the field definition for the ID numbers (after the type, but before the semicolon). Add several students to this table (e.g., three or four students).

Next, create a course table that contains the same information as in the previous Lab #6. Also include an ID field to be the primary key of that table. Add several courses to this table.

Finally create a schedule table that associates a student ID with a course ID. You can create yet another ID value to be the primary key of this table (or it might make sense to define the combination of student ID and course ID as a composite primary key). Include foreign key references at the bottom of the table to ensure that the only student IDs and course IDs that are allowed are those that already exist in the other tables.

### Task 2 – Display your course schedule

This task is to make a web page that lets users look up student schedules. Since you have just a few students and not 100's, make a *pull-down menu* in your form where the user can select a student's name. To obtain the names, do a `SELECT` on the student table. When the form is submitted, retrieve the appropriate course schedule, and display it in a nice HTML table. A *nice* HTML table has column headers with a different background color, font, etc. to make it stand out.

Your input form might look like the picture on the left. And the output table might look like the table on the right.

## Request Course Schedule

Select the student's name then click Submit

Get schedule for:

## Course Schedule Results

Below is the Schedule for *steve*

instructor	course	days	room
Prof C Damon	CIS1152 - Adv Web Dev	MWF	401A
Prof Merat	CIS2300 - Intro to Computers	MWF	401B
Prof Poa	MATH100 - Linear Algebra	TR	401A
Prof Grizzle	EECS4500 - Control Theory	TR	202

There are two ways to deal with the two tables.

1. Use SELECT on the student table to look up the ID of the student. Then use SELECT again on the schedule table to look up all rows associated with that ID.
2. JOIN the two tables together and SELECT the rows based on the student names (in a WHERE clause).

### Task 3 – Add a class

Create a new form that accepts information about a class and adds it to the schedule for a specified student. If the student does not exist yet, a new entry should be created in the student table automatically (with a unique ID number). You should be able to use the form you created above to view the new schedule for the student and see the new class.

Modify your script so that it checks to ensure there is no conflict between the new class and the other classes in the student's existing schedule. If there is a conflict, the script should produce an error message.

### Task 4 – Remove a schedule

Create a third form that allows the user to delete the entire schedule for the selected student. For purposes of this lab, we will not provide a way to delete or modify just part of a student schedule or to remove students (although these things could be done in the database back-end, if necessary).