



○ Code Samples Overview

concept.php

I had noticed with a lot of my projects that each piece of the software tends to have this one thing in common: There's a table, with an id, and multiple other columns. That statement most likely describes 75% of all tables ever created. Maybe it's a blog post, a user, or a customer order. The problem is that sometimes I would only need the users name, and other times just an email address, and I wanted a solution.

There are comments in this file that give usage examples. I had called a previous version "entity", however this one works with my PDO wrapper (see below). It's important to note that obviously not every table is worthy of being a "concept" or "entity". Many to Many relationship tables are a prime example, and wouldn't be built with the above in mind. While I would make a `users` and `accounts` table a concept, the table that connects them (`user_accounts` if you will) would just be handled either within one of the two existing concepts as a function, or it would get its own model, depending on the complexity.

db.php

This is my PDO wrapper. I have abandoned it in favor of a stable PHP framework, however I think it was mostly complete at this stage. It was built while I was learning about PDO, and future versions would have likely extended the PDO class instead of relying on an existing PDO connection. At the time, my "roll your own" framework was using mysqli, and I wanted to make the change to something more up to date. Obviously CodeIgniter and other frameworks have better ways of handling this.

Both of these php files will give you a good idea of my documentation style. I'm a big believer in respecting future developers, including myself, who have to decipher code years later.

js.mauve_table.js (see also: *mauve.phtml*, *mauve_orders.png*, *mauve_repro.png*)

This is a MooTools JavaScript class I wrote. It makes an ajax call based on a table header when something changes. The two images are examples of something the class would control (specifically the header pieces). The PHTML is one of the View's that generates the base HTML. An AJAX call would generate the <tbody> content. I liked using the phtml extension as an extra means of categorizing it as a view.



Nicholas Wright

3748 Hertland Dr, Kettering, OH 4549 • 937.214.2129 • tickthokk@gmail.com

biggest query ever

This INSERT ... SELECT statement is one of the biggest and most complex queries I've ever written. It is over 70 lines long, and looking at it now, kind of messy (which you'll have to forgive, it's several years old). Documentation existed above it, however I still feel that it wasn't enough. If I were to do it again I think I would have placed the documentation in between the lines for this one.

I only show this to prospective employers because it really highlights my proficiency with relational databases and MySQL in general. It contains a dozen joins, utilizes nested sets, mysql variables, logic statements (IF, IFNULL, etc), and miscellaneous math functionality.

buttons.scss and buttons.css

I really like SASS. The .scss file is the SASS version, which generates the CSS. I made the CSS pretty for viewing, but normally when I tell SASS to compile, it compresses it as well, taking it down to one line and removing unnecessary white spaces. My approach to CSS (and code in general) is to code for readability, while keeping in mind that I'm not a computer, so I shouldn't act like one. If you look at the previous PNG files again, the green "Filter" button is generated thanks to these CSS files.

I would be more than happy to answer any questions regarding these files. I am most easily reached via email at tickthokk@gmail.com any time of the day, or by phone at 937.214.2129.

Thank you,

Nick Wright