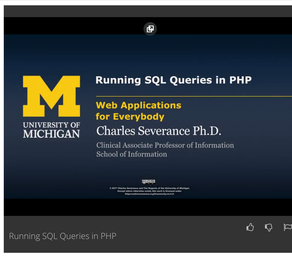
Lecture Content



Running SQL Queries in PHP

Have a question? Discuss this lecture in the week forums.

Interactive Transcript

English

0:08
So now we're going to actually insert some data. Now, I walked through these. I give you two versions of this. First, I give you kind of a lecture version of it, where it's all slides. And then I'll give you a walk through of it where I'm just writing code. So you're going to see this twice, look at it both on the slides and on the demo. The first thing we have to do is create a database and user, and we've done this before, and we've done a CREATE DATABASE. But we haven't done a GRANT command. So CREATE DATABASE is an SQL command to create a database. And then the GRANT command is effectively, you can read up on it, it's a bunch of little things. GRANT ALL, you can give different permissions, but this grants everything, this is a database name, all the tables in the database too. And then an account and then a password IDENTIFIED BY 'zap'. This next parameters basically is what incoming address will we accept fred from? And so if you basically have your laptop, and it's connected to the Internet out here somehow. When you're running a program inside the laptop, which is like PHP inside the laptop, and then you run a database, it makes a connection, and this connection is either 127.0.0.1 or localhost. So these within-laptop connections are still network connections, they're like a fake network inside the computer. It just means that by only allowing connections from these two places, some hostile person wants to come in and talk to your database server, they cannot. Even if they knew this fred and zap account. If they sneak in, they need to make the connection from within the computer, because this one will have 141.21.14.8, right? And then that'll come in, but that password only works for local connections. So this little trick here is a way to basically firewall your database server from connections coming from outside. Now if you want to have them connected from outside. Most of the times when people set up production database servers, they set them up on virtual local area networks, where you have a PHP server and a database server. There's a little LAN, but it's sort of, the outside world can talk to PHP.

2:16
But the outside world is somehow blocked from talking to your database server, and so this is all in your hosting facility. It can talk to here but you don't even allow packets to come to this database server to protect it, so people can't sort of beat the heck out of your database server. They can't even get a packet to it using something like VPN, virtual private networking, or who knows what.

2:39
Wow, that's a long way to basically say that what we're trying to do here is create an account and a password, and make it so that we can only talk from within the same box. So inside your laptop talking, the PHP inside your laptop can talk to the SQL inside your laptop. Now if you're doing all this with a command line, you have to switch to the database, but we've seen all that before. And you can change the use if you like.

3:04
We're also just going to just create a table, right, make a table with an index. I mean, by this point in time we should know how to do this stuff, it shouldn't be too particularly difficult.

3:18
And then we're going to insert some users, right? And this is from a previous lecture, we're going to insert a name, email. We've got an auto-increment field, and so we're going to get some data in there, and away you go. So that we've done, all that through phpMyAdmin, or through the command line, or some other way.

3:35
And so what we've got now, is we've got our database sitting on localhost, basically.

3:42
We've got MySQL, that's software, and we've got a couple databases. Maybe we have one named misc that we just made, we've got a table, we have some other one called sakai or whatever it is, that we use for some other thing. So that's what we've got, by typing all of that stuff in we have created that.

3:59
Now what we need to do is make it so we can send SQL commands from PHP to this database. So we have to establish an SQL connection, we call it a database connection.

4:13
And again, this is like a fake little Internet, even though it's inside of your computer. It's going to localhost or 127.0.0.1, and it's literally sending SQL commands with a little tiny special protocol. And then getting back data using that same protocol. And so what we need is doing inside PHP, when we come down inside our code and we want to make this connection, we have to know the ID and the password and the database. And we also have to know which port this is running on. So all network servers have ports that they run on so that one computer can have a mail server, a web server, a secure web server, a database server. So we have to pick the port that it is going to go on. And there's generally two ports that we run, and 3306 is the default port for MySQL on most systems. If you're on a Macintosh, you say this 8889, because this is how MAMP sets it up. And so if you are running MAMP on OS X Macintosh, this is what you do. And if you're running .Windows or Linux or XAMP or something, just change this to 3306. This'll be a pain in the neck to get right. [LAUGH] Once you've got it right, away you go. And so this basically saying, we're going to connect to a MySQL server, we're going to connect to localhost, which is within the same computer, and we're going to connect to port 8089. And we're going to go to the database misc. And we are going to present an ID and password as part of this connection, because there is a security that stops from coming in here. And so this is like logging into a computer, PHP is logging into the database. And so once this is done, you get back, it's an object, you get back an object. And we're going to store that in a variable called $pdo. I could name this $x if I want, but I just always name it pdo. It's just a mnemonic, but, it doesn't have to be named pdo. And I could have more than one connection, and I would have different variables, different objects, different instances for each one of those things. See why I taught you about objects before? Because now I just use the word object, and you don't know what object is, go to the previous lecture and learn about objects. So you get this connection, and so what the connection is, is a piece of data here that is effectively your portable that you can see the database through this variable. If you have got the wrong password, if you get this wrong or this wrong, this blows up with a trace back, blows up with a trace back. So if it works you've got a connection, and it's in variable pdo. And if it doesn't work, then you don't have a connection. And your code is blown up and probably everything else you're going to do is not going to work. So like I said, making this work is probably the hardest thing that you're going to have to do. And then two minutes after you make it work, you'll just use it over and over and you'll stop thinking about it. So it's, [LAUGH] so get some help if you get trouble with your database connection.

6:58
So let's take a look at some PHP code that's going to take a look at this, right? So here's a PHP script, it's got an echo statement in it, I'm using a pre tag, so everything looks pretty and doesn't get wrapped. I can say, hey, let's make a PDO connection to the database with fred and zap, yadda yadda. If this works I get pdo, otherwise this will trace back, blow up. And then I'm going to basically send a query. And so you say it's pdo, and arrow means go find the method query within the pdo object, and SELECT * FROM users, that's just a string. That's like you type it on a keyboard, SELECT * FROM users. We are literally sending strings of SQL to the database server, okay? And then we get back this thing called the statement. And then we can loop through whatever records were selected by doing this while statement, while ($row = $stmt->fetch(PDO::FETCH_ASSOC)), print row. [SOUND] Until this becomes false, and then it pops out. And in this case, because they put two records in, this print_r runs twice. And it just shows you the data, and it shows it to you in a key-value array. That's what the FETCH_ASSOC, that says, give me back rows in an associative array. You don't want the other one, the other one gives you back an ordering. You have to remember that the first one is user_id and the second one is name. No, you don't want that. So you can look this up by user_id. And I'm just doing a print_r, so this loop runs twice. And out comes the stuff, and it gives me an array for each of the rows in the database. And so it's the way we can see in PHP, super simple code. Understand every single line of this. Don't try to do fancy stuff until you know what's going on here.

8:43
Okay, now we don't usually want to just print with print_r. We want to do something that sort of makes sense. And so we're going to make a data base connection, we're going to do a SELECT name, email, and password FROM users. And we're going to then print out a table tag. And so that the table tag will come out, then we're going to loop through the rows, and we're going to print out a table tr, td, that's this little bit right here. Then we're going to print the name out, then we're going to print out td, we're just printing out HTML now, right? And then at the end of it, we print out /tr, right? And then we're going to do it again, and we get a second row, and then we print out /table at the end. And then the output of this, the collective output of this, is what we see on our browser, which is this little bit of output right there. So usually we're not just dumping this data out to show the user, usually we're formatting it, we're selecting which data we want. And we're formatting it in a way that's pleasant to the user. And this is a very simple formatting, so it's all three. You go, here comes some PHP, it gets some data from the database, it pulls this data in. And then it formats it before it sends it back out, so it chooses to format that as HTML. So it's really, in a sense, transforming the database data into HTML. Now you saw at the beginning of these files, the first thing that I did was make a database connection. And in each of those files, I have the host name, I have the database name, I have the account and the ID. And so you don't want to put that in too many files, and so we just put it in a separate file and include it all over the place. And so this is that pattern. So if you look at this one, we just put the database connection in the one file, pdo.php. This is exactly that same line, it leaves it in $pdo. And we're going to set ERRMODE, and we'll talk a little bit about this a bit later. There are different error modes for PDO, and it depends on whether you're in production or not. And so we're telling this to be very talkative when things blow up, because it's really important to be talkative when you're getting started. Otherwise you'll see these blank screens and it's like what did I do wrong, it's just a blank screen. And then you realize I didn't turn on ERRMODE. It's kind of like when you have PHP and you don't turn error modes on, and you've got syntax errors and you don't see them. So I'm going to basically suggest that while you're developing, especially in this class, you set ERRMODE to be loudest, blow up, die, don't continue. Okay, out of these residual we get $pdo, so we require that, require once, it's a good reason to use require once. And then we just run the code, and we've got this pdo that came out of there. The naming convention of $pdo and the name of this file being pdo is just convention. It doesn't have to be that, the only thing it has to be this is this. because that's the name of the PDO class, capital PDO is the name of the PDO class that we're going to instantiate, that's given to us by PHP. But after that, every other name and variable that I've chosen is something that I have chosen.

11:43
Okay, so now we're going to talk about how you don't just select from a PHP app, but we're going to actually insert data in a PHP app

Downloads

Lecture Video mp4
Subtitles (English) WebVTT
Transcript (English) txt

Would you like to help us translate the transcript and subtitles into additional languages?