COMP10120 Lab Session 7

Linking with MySQL

Contents

These notes are available online at
`studentnet.cs.manchester.ac.uk/ugt/COMP10120/labscripts/101lab7.pdf`

7.1 Objectives

To make sure that everyone in your tutorial group:

- Has a basic familiarity with some more of the tools and technologies that you could use to build your project web-site.
- Has begun to think about security issues with MySQL.

7.2 Configuring your Pi to access MySQL databases

Once again, you're going to be doing this exercise on your Raspberry Pi, so set it up in the usual way. In this exercise it is probably better to use it 'headless' and log into it from the desktop PC, rather than using the Pi's graphical environment.

In the last session you installed a couple of packages on your Pi that allowed the Apache server to interpret pages written in PHP. In its default state, PHP doesn't include functions to access MySQL databases (more technically to act as a **client** to a MySQL **server**), so before we do anything else we need to add another package to do this. The package you need is called `php5-mysql`, and you should by now be familiar with the command needed to get this installed on your Pi. While the package is installing, pay attention to the output that appears on the console as the installation proceeds, and see if you can make sense of what is going on.

Don't worry about understanding each and every line of output, but you should at least be able to spot the following activities:

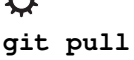
- The system recognising that in order to install `php5-mysql`, some other packages are also needed (these are called 'dependencies', that is they are things that `php5-mysql` *depends* on in order to work). You'll notice that the system automatically installs these for you so that you don't have to figure out all the dependencies yourself.
- Connecting to a remote repository to fetch the packages.
- Unpacking the contents of the files that have been retrieved, and installing this in the appropriate part of your system.
- Stopping the web server, so that it is safe to update its features.
- Automatically modifying various configuration files (in this case ones ending in `.ini` which serve similar purposes to the `php.ini` file that you edited by hand in the previous session.
- Restarting the web server with its new configuration.

Similar things have been happening every time you've installed new packages; so it's just as well that package managers exist to help automate these activities, otherwise you'd have to do all that by hand every time you wanted to modify your system!

You also need to install a package to enable you to interact with MySQL via the command line, which you will be doing later. The package name is `mysql-client`; please install it now.

Before moving on, check that the installations completed successfully and that there are no errors being reported by the package manager. If there are any problems, ask a member of lab staff for help.

You now need to set up a directory to contain this lab session's work. First, make sure that your GitLab repository for `COMP10120` is up to date and you have used `git pull` to update it on your Pi. Now create a new `ex7` directory for this session's work and, just as in the previous lab session, make a **symbolic link** from `/var/www/mysql` to `ex7` using




```
$ ln -s ~/COMP10120/ex7 /var/www/mysql
```

Now check that the file and directory permissions are set so that the web server can see the contents of this new directory.

7.3 Creating a MySQL account

In this lab session you will be using PHP to interact with a MySQL database, so you will need to activate an account on the local MySQL server. To do this we use the local **Web Dashboard** which can be found at:

`https://web.cs.manchester.ac.uk/dashboard/`

In a later lab session we will be using this dashboard to publish your web content to a 'production server', but for the moment we are interested only in the feature it has to enable you

to create a MySQL account. This process is described on the School wiki at:

`https://wiki.cs.manchester.ac.uk/index.php/Web_Dashboard/Database`

Follow the instructions there to set your password and activate your account.

Once your account is activated follow the link to **Connecting to MySQL** (in the **Web Dashboard Documentation** box at the bottom of the page). This gives some information on how to interact with MySQL in three different ways: using the web-based tool **phpMyAdmin**, via the shell on the command line, and using PHP. You will be using all of these methods.

You will notice that the dashboard refers to a file called `config.inc.php`. This file is used to store information about your username, password and other information needed to connect to MySQL.

Take a copy of the contents of this file as shown on the dashboard and create a file of the same name in your `ex7` directory, with `<?php` at the top and `?>` at the bottom of the file and the variable assignments in between. Remember to put your actual password in the file and make the file world readable. You should now be in a position to use a suitably modified version of the example PHP code given, once you have created your database. **Do not add this file to your git repository, and reflect on why doing so would be a bad idea.** Before moving on you might want to add the name of this file to `.gitignore` so that `git status` doesn't constantly remind you that it's not under `git`'s control.

git status

7.4 Using PHP to communicate with your MySQL database

For this exercise you will need to:

- Create a MySQL table to record names and email addresses, and initialise it with some test data.
- Copy the pages you created in `ex6` into your `ex7` directory, and modify them to input a name and email address, compare it with the information in your table, output a different message depending on whether the name is that of one the members of your tutorial group and depending on whether the name is new or not, and save any new name and email address into your table. Make sure that you validate input data to avoid possible problems.

If you've already done something like this, use this as an opportunity to learn more about the available technologies. By the end of this exercise you should be able to demonstrate some web-pages making simple use of MySQL, PHP and HTML forms.

In order to achieve this session's goals, amongst other things you will need to:

- Find out how to use phpMyAdmin and MySQL on our server by reading the information on the School wiki at:

  `https://wiki.cs.manchester.ac.uk/index.php/Web_Dashboard/Database`.

- Find out about using MySQL and PHP, e.g. by working through some or all of an introductory tutorial. Make sure you are looking up how to use PHP's newer `mysqli` interface, rather than the older (and now deprecated) `mysql` interface.

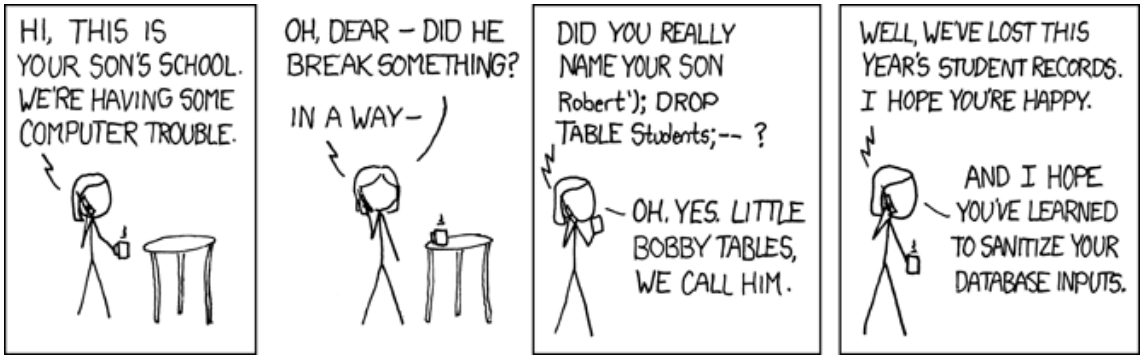


Figure 7.1
As is often the case, XKCD manages to take a complex technical concept and squash it into a few panels of black and white stick-figure loveliness. Organisations who have suffered from an SQL Injection Attack may not find this cartoon strip so amusing. Reproduced from `http://xkcd.com/327/` under Creative Commons Attribution-NonCommercial 2.5 License.

- Find out how to (re)initialise your MySQL tables by using a shell script and the mysql command line interface, to simplify testing.
- Find out about common security problems with MySQL and what to do about them. By the end of this session, you should understand the joke in XKCD's Exploits of a Mom (see Figure 7.1).

Remember that, if you copy code (e.g. from a tutorial) into your web pages, you should add comments to make it clear what you copied and where you copied it from. Remember to update your implementation page.

7.5 Knowledge

By the end of this session you should understand how to use phpMyAdmin to create, modify and delete tables in your database, and also understand how to achieve the following things using the `mysqli` PHP interface:

- Connect to a remote database server, and check whether the connection succeeded.
- Submit a query, and retrieve the results.
- Cleanly close a connection to the database once you're done with it.
- Sanitise any parameters that you are sending as part of a query (to avoid 'parameter injection' attacks on your system: no one wants to fall foul of Little Bobby Tables).

7.6 Assessment

Just run `submit` and `labprint` as usual in your `ex7` directory.

submit

You can view a copy of the marking scheme used by `submit` and `labprint` at `studentnet.cs.manchester.ac.uk/ugt/lp/ms.php?unit=COMP10120&ex=ex7.xml`

Show your web-site to a member of lab staff, and be prepared to answer any questions about what you have done. For full marks, you should have:

- a MySQL table to hold relevant information such as name and email address;
- a simple repeatable mechanism (e.g. shell script, not all done by hand) for initialising your MySQL table for testing. This shell script should be in your `ex7` directorry and have the name `init-sql-tables`;
- edited all your code by hand;
- PHP/SQL that searches your table for an input name and email address;
- PHP/SQL that inserts a new name and email address into your table;
- PHP/SQL that outputs a different message depending on whether the name is a group member or not, and whether it is new or not;
- validated your input data to avoid possible problems;
- used comments to identify code copied from tutorials etc;
- updated your implementation page with details of software and information sources used.