

CSCU9B3 Practical 1: Introduction to MySQL and phpMyAdmin

Computing Science, University of Stirling

In all the practicals a pen and paper will be very useful so that you can make notes as you go. Things you do in this session will be useful later, so all notes will have further use.

Finding and Logging in to MySQL

MySQL is a database management system (DBMS). Unlike MS Access (with which you might be familiar), most databases do not have an integrated user interface. They communicate with software at a level that is generally hidden from the user. However, there is usually software available to make accessing the database easier. In this practical we will learn about one such program, **phpMyAdmin**.

phpMyAdmin is a browser-based interface to the MySQL database engine. It is written in the scripting language, PHP and allows you to manage a database via a web browser.

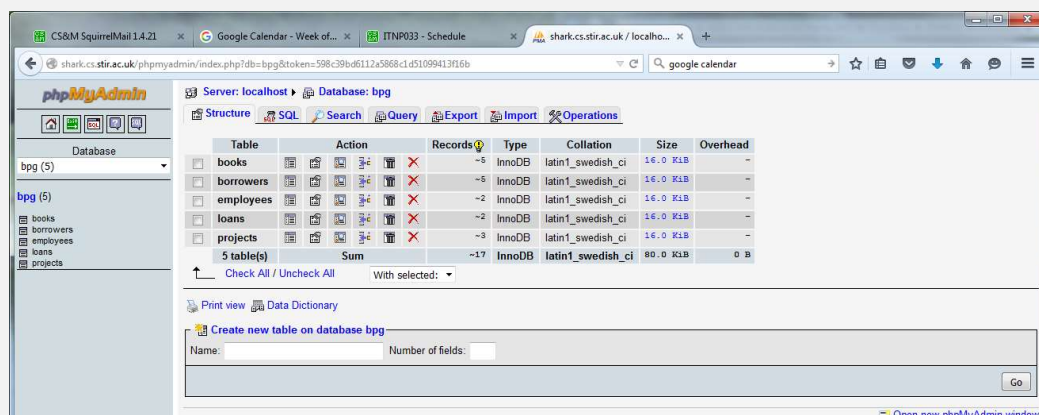
Each student has a space on the database that is separate from all other students' space.

You can see the phpMyAdmin interface for your space in a web browser by opening up Chrome or another browser and going to the web address:

<http://wamp0.cs.stir.ac.uk/phpmyadmin/>

You will be asked to log in with your MySQL username and password: your username is either your normal STUDENTS user ID or your 3 letter CS username (if you have one), with the initial password set to the same. Once you have phpMyAdmin running, you should change your password to something more secure, via the General Settings tab on the Home screen. Note when changing your password, just use the Enter and Re-type fields to change to a password of your choice; DO NOT use "Generate password".

Select your database (the one called by your username; NOT information_schema) from the panel on the left. The phpMyAdmin interface now shows the tables in your database in a panel on the left of the screen and allows you to explore the currently selected table via a set of tabs on the right of the screen (though initially you will not have any tables):



In this first practical session, we will explore the MySQL interface, create some tables and enter some data. The data is from a simple company database containing two tables:

Table: Employees


Name	Employee Number	Date of Birth	Salary	Full Time (%)
John Jones	234621	1972-07-04	45345.95	100
Sam Smith	374837	1970-08-23	44000.50	50

Table: Projects

Project Name	Manager	Description	Full Days Worked
Thrust	234621	Build a rocket to the moon	100
Gold	374837	Turn lead into gold	200
Win	234621	Win Wimbledon	20

Create a Table

You should see a box like this on the main page.

 Create new table on database **prn**

Name:

Number of fields:

Use this to create your first table. Call it **Employees** and enter the correct number of columns (also called fields). Click **Go** to move to the next step.

Define the Columns

You should see a new web page with a row for each column in your new table. This is where you tell the database about your columns. Today, we will only look at the first three attributes: the column name and its type and length. Appropriate names are given in the table in this document, above (Name, Employee Number ...).

The data types might be new to you. Just as variables in a programming language like Java have types, so do columns in a database. You must pick the right type for each entry in your table. There are many choices in the drop down list, but we will just look at the following:

- VARCHAR
- INT
- DECIMAL
- DATE

You can read about these types at either of the sites (the first, official site is very detailed):

<https://dev.mysql.com/doc/refman/5.7/en/data-types.html>

https://www.w3schools.com/sql/sql_datatypes.asp

Spend some time reading about each of the types above. Ask if you do not understand any of them. Once you have understood them, choose the right type for each column in your table. Note that Salary needs exactly 2 decimal places and has a maximum value of 99999.99.

When you have chosen the types and their lengths (if necessary), click the **Save** button to create your table. You will likely find that lots of warnings / errors appear. Check if anything seems to relate to your columns, and if not, simply select "Ignore all" (some of the messages are warnings of things that we will worry about later on, but not just now).

Now do the same to create the **Projects** table.

Preparing to Enter or Change Data in a Table

To edit a table, first you need to make one column of each table designated as "Unique":

1. Select the table you want to edit e.g. Employees
2. Select the Structure tab
3. Click the checkbox for the "Employee number" and select "Unique" from the list underneath the column definitions.
4. Now if you "Browse" this table you will see that you can edit each row of data.

You can do the same for Projects, perhaps choosing "Project Name" to be unique.

If you ever need to **delete a table** and start again, first select that table in your database list and then go to the Operations tab, where you will find a "drop table" command.

Entering Data

Now you need to put some data into your tables.

1. Select the **Employees** table
2. Click on the **Insert** tab and use the form to enter the data given above
3. Try to enter data of the wrong type to see what happens
4. When the data is entered, you can view it in the **Browse** tab.

Once you have the Employees data inserted, do the same for the **Projects** table. Make sure you have entered the data exactly as it is on this sheet, otherwise the questions later may not work properly.

Searching the Tables

Your tables are very small, but for larger tables, you would need to search for specific data rather than browsing. So let's try some searching...

1. Select the table you want to search from the left panel – **Employees** in this case.
2. Click on the **Search** tab for the search page to bring up:

Or Do a "query by example" (wildcard: "%")

Field	Type	Collation	Operator	Value
Name	varchar(100)	utf8_unicode_ci	LIKE	
Employee Number	int(11)		=	
Date of Birth	date		=	
Salary	decimal(7,2)		=	
Full Time	tinyint(1)		=	

Go

3. Every column in your table has an operator and a value for the search.
4. You'll see there is a list of options for the **operator** in a search. Operators work differently depending on the data type. For example, you'll see > and < for numeric data types, but not for character strings.
5. The **value** field is where you tell the database what you want to search for.
6. Try a few simple searches:
 - a. Search for the details of Sam Smith
 - b. Search for all the full time employees
 - c. Search for all employees born after 1967-01-01
 - d. Search for all the employees with a salary over 40000.
7. You can also search for entries that are similar to a given search value using the LIKE operator. Use the '%' as a wild card. For example **Name LIKE % Smith** will search for all the names ending in Smith. You try:
 - a. Search for all the names ending in Smith
 - b. Search for all the names with 'Jon' somewhere in their name.

CHECKPOINT 1: demonstrate the final search above (step 7b)

More Searching

If you click "Options" on the Search page you find some extra ways of specifying and refining a search. You can:

- Choose which fields are shown in the search results (use Shift or Ctrl to select more than one when clicking on column names).
- Select only distinct rows (only one example of any given set of values)
- Specify how many results are shown per page
- Sort the results in ascending or descending order
- Add further clauses to the search term (this can be ignored for now).

Now try the following searches:

1. List the projects that employee number 234621 is manager for.
2. Show just the name of all the projects with more than 150 full days worked.
3. List the employee numbers of any employee working on a project with fewer than 400 days worked. Only show each employee number once in the list.
4. List all the projects in order of time taken, with the longest running project first.

Next time...

In the next practical you will write SQL statements directly to create and search tables, rather than using the “forms” interface provided by phpMyAdmin.