



Lesson #41

phpMyAdmin

By

Lynette Chandler
Tech Based Training

<http://TechBasedTraining.com/>

**NOTICE: You Do NOT Have the Right
to Reprint or Resell this Report!**

You Also MAY NOT Give Away, Sell or Share the Content Herein

If you obtained this report from anywhere other than **Tech Based Training.com** , you have a pirated copy.

Help stop internet piracy by letting me know. Send email to
customerservice@techbasedsupport.com

© 2010 Copyright Lynette Chandler

Legal Disclaimer: Keeping things simple here, I have to say that I cannot promise you success. I can give you direction and advice based on my experiences and good internet marketing practices. What you do with this information is up to you. As a Tech Based Training member you agree to not hold me responsible for your results.

Introduction

Since you've been working with WordPress for a while now you undoubtedly have run into or used phpMyAdmin. If you're serious in your WordPress business, you cannot ignore this. Technically, it's like stepping up to the next level.

Knowing how to work with databases directly is what truly separates the dabblers from the real powerhouse users and players. I understand that you signed up as a designer not as a developer.

Yet, as I've mentioned in the beginning of the course, we are a little bit of both. When it comes to working with WordPress or any database driven content management system, the line sometimes blur between a developer and designer. You'll always be stronger in one area and that's OK.

As time goes by and your skills develop, your clients will look to you for more complex work and if you can do it, your clients' confidence will also grow and more work often follows – that is in my experience.

Maybe you don't want that. Maybe you want to be able to tell your clients that is beyond the scope of your services and forward the job to others. That's perfectly OK too. If that's what you prefer, you might want to point them to WebTechRelief.com and you're free to *skip this lesson entirely*.

If you want to learn *just a little bit* in order to equip yourself to be able to quickly help your clients, this lesson will be helpful.

Why PhpMyAdmin, What Is It Anyway?

PhpMyAdmin is a web based tool to manage the MySQL database. It is not the database itself. Without PhpMyAdmin, you'll have to manage your database through commands – imagine DOS prompt you know, black screen, white text, everything you do is in code. Not fun.

PhpMyAdmin gives the database a nicer graphical layout and a much easier way to work on the database directly.

I've actually hesitated several times while writing this lesson, whether to proceed with it or not. The reason being, **there is no Undo**. Working directly with the database gives you *ultimate* power over the data. You must be precise and very sure when executing any commands or you can lose the data and its consequences are final. Definitely not for the faint hearted.

Even though the warning is severe, I'd like to encourage you even if you feel a little scared and uncertain, yet want to advance your skills. I was there once too and my magic pill? The backup.

Simple, solution but super effective. Never do anything to the database until you've backed up. As you improve, you'll also get more comfortable with it.

Another reason I hesitated is also because, it is always a good idea to have **some** basic knowledge how to use MySQL without phpMyAdmin. And this is one thing that is completely out of our bootcamp topic. It'll probably take another course to do so.

But as you see, you're reading this lesson and I am going ahead to teach you **specific** commands to do **specific** tasks that I believe you may encounter along the way. These are not the only things you can do with the database but they will certainly help get you help your client out of a bind quickly.

Beyond these examples, I suggest you consider sub-contracting it out.

Now that the housekeeping is done, let's get into the scenarios and what you can do to fix it using phpMyAdmin.

Basic MySQL Jargon

To move on, it's best you understand a little bit of the jargon so here goes. It might help you imagine a normal spreadsheet.

Database – this is the complete set of data for a site/blog. A database consists of a set of tables and fields. This is like a collection of different sheets inside your spreadsheet document.

Tables – these are sets of data inside the database. In each table is the data organized in columns and rows. This would be comparable to a single spreadsheet in a file.

Fields – these are the labels for each data piece. In your spreadsheet, these would be like the column headings.

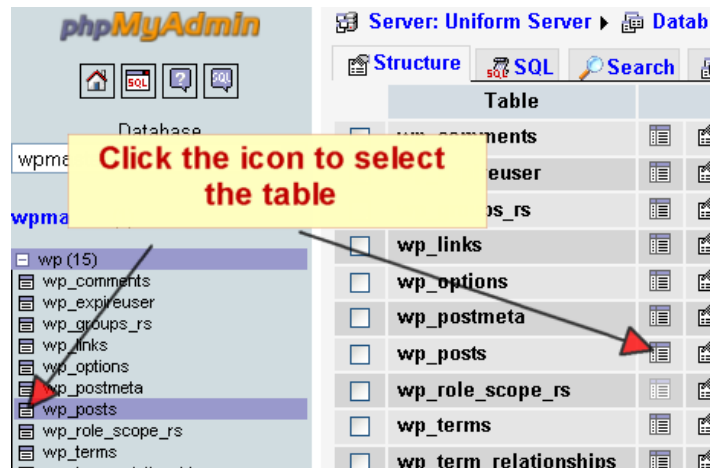
Rows (or ID's) – these tell you exactly where a single piece of data is. This would be the same as the rows in your spreadsheet.

Backing Up Specific Tables Only

There will be some clients you work with who have huge web sites. To backup their database it takes forever for you to download. This can be quite a pain especially when you want to get into the database to do just as simple and quick fix up. The good news is, you can only backup just a specific table, the ones you are working with. That way you have a safety net and can get to work quickly.

Please note this is not the same thing as the backup I've introduced you to in lesson #76. You should always do that regularly. I have a regular backup going on. In addition, when working on the files or database, I also backup as I work.

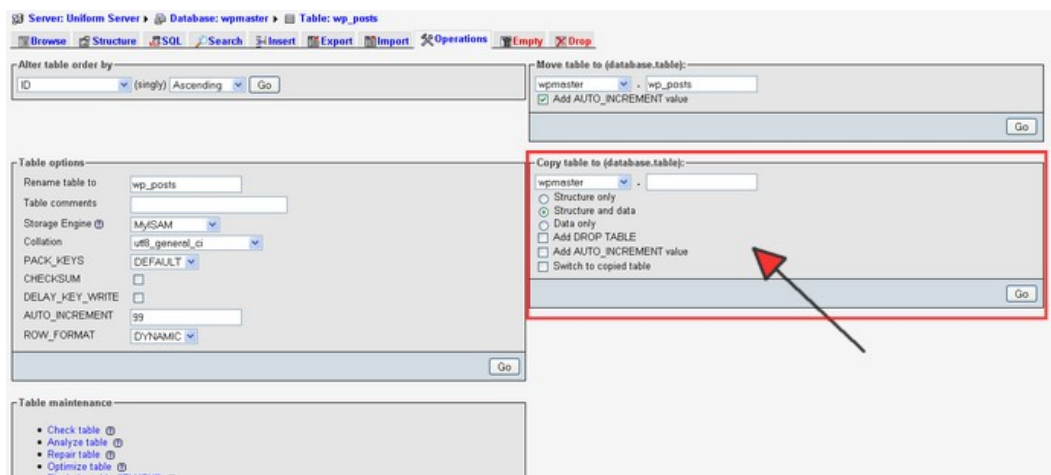
In order to backup just one table only, log into your web host or you can practice on your own server on your computer – this is safer. Find phpMyAdmin and launch it. Select the database you want to work with, then click on the table you want to backup.



Now find the Operations tab found on the top of the screen - **after** you select the table.



Find the box that says Copy table to (database.table):



In the box next to the drop down, type the name of the table you are in then add “_YYMMDD” just like the way we backup other files, without quotes. This way you know which was the most recent backup. Your drop down field may look different from mine. wpmaster is my database name. Most of the time this is selected already to leave it where it is, no change.

Select structure and data – or no change, usually selected this already. Then click Go.

Copy table to (database.table):

wpmaster . wp_posts_090611

☐ Structure only

☒ Structure and data

☐ Data only

☐ Add DROP TABLE

☐ Add AUTO_INCREMENT value

☐ Switch to copied table

Go

If successful, you will see a new table in your database. You will also see some messages on the top of the right pane – that just tells you what it has copied over.

phpMyAdmin

Database: wpmaster (16)

Success!

wp (16)

- wp_comments
- wp_expireuser
- wp_groups_rs
- wp_links
- wp_options
- wp_postmeta
- wp_posts
- wp_posts_090611
- wp_role_scope_rs
- wp_terms
- wp_term_relationships

Server: Uniform Server Database: wpmaster Table

Table `wpmaster`.`wp_posts` has been copied to `wpmaster`.`wp_posts_090611`.

```
CREATE TABLE `wpmaster`.`wp_posts_090611` ( `ID` bigint( 20 ) NOT NULL DEFAULT '0', `post_author` bigint( 20 ) NOT NULL DEFAULT '0', `post_date` datetime NOT NULL DEFAULT '0000-00-00 00:00:00', `post_date_gmt` datetime NOT NULL DEFAULT '0000-00-00 00:00:00', `post_content` longtext NOT NULL, `post_title` text NOT NULL, `post_category` int( 4 ) NOT NULL DEFAULT '0', );
```

Showing rows 0 - 84 (85 total, Query took 0.0058 sec)

```
SELECT * FROM `wp_posts` LIMIT 0 , 300
```

Changing Username Or Password

Sometimes people forget their admin password or something happens to the blog that the admin is somehow locked out and attempts to retrieve the password fails. You can use the following steps to change the admin or any user's password from the database.

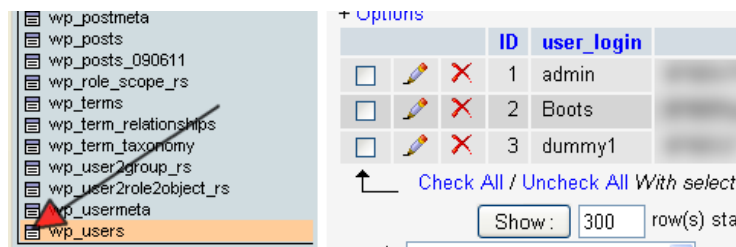
Changing Passwords

WordPress passwords are saved in an encrypted format. Which means you can't just enter a new password in the password field for a user and be done. You must first get the encrypted version of your new password then paste it in the database.

I have created a password encrypter for you to use. Be very careful about using those that you find through search engines. There has been sites known to track the password that you enter. The one I created for you does nothing of the sort and is 100% safe to use.

Step 1: Enter your new password [here](#) to generate a new encrypted password. Copy the encrypted password string.

Step 2: Find your way to phpMyAdmin, select your WordPress database, find the wp_users table, click on the icon so it will display all the users in the blog.



Step 3: Find the user you want to work with, click the icon pencil on the same row.

+ Options

	ID	user_login	user_pass	user_nicename	
<input type="checkbox"/>	1	admin		admin	lynette@
<input type="checkbox"/>	2	Boots		boots	lynette.
<input type="checkbox"/>	3	dummy1		dummy1	lynette.

Step 4: In the field where the password is, delete it and paste your new encrypted password. Click Go.

Field	Type	Function	Null	Value					
ID	bigint(20) unsigned	<input type="text"/>		<input type="text" value="2"/>					
user_login	varchar(60)	<input type="text"/>		<input type="text" value="Boots"/>					
user_pass	varchar(64)	<input type="text"/>		<div>Paste new password here</div>					
user_nicename	varchar(50)	<input type="text"/>		<input type="text" value="boots"/>					
user_email	varchar(100)	<input type="text"/>		<input type="text" value="lynette.chandler@gmail.com"/>					

You're done. Now the user can log in with the new password – the human readable version, not the encrypted version.

Changing Usernames

To change usernames, the process is the same as before, except you don't need an encrypted version of the username and you enter the new user name in the user_login field. Repeat steps 2 – 3. In Step 4, this is where you'd change the username.

Server: Uniform Server

Database: wpmaster

Table: wp_users

Browse

Structure

SQL

Search

Insert

Export

Import

Operations

Empty

Field	Type	Function	Null	Value
ID	bigint(20) unsigned			2
user_login	varchar(60)			Enter New Username Here
user_pass	varchar(64)			

This method is super useful because you cannot change your username through the WordPress dashboard. It is also good to use this method to change the admin username as a security precaution since WordPress default admin user name is 'admin' so this makes it easier for people to attempt to gain access to your admin account.

Search And Replace

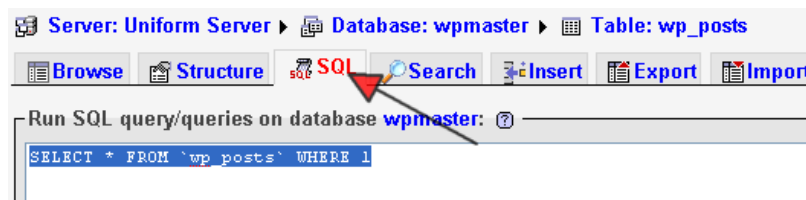
There will be times you need to mass edit text within posts. Maybe the client has decided to move the physical blog installation from one folder to another or they are trying to merge two WordPress's together. After importing the content from one WordPress to the other, the imported posts containing pictures or links to media may break.

I've also experienced once where the imported posts contained strange characters and I had no way to find out which posts among thousands these characters showed up.

If the blog is small changing them individually is not a problem but when the blog is big with thousands of posts this can be a big issue. Surely you won't be going through each post. To do that, you can do a simple search and replace on the database.

WARNING! There is no undo when you set this in motion so practice in your sandbox until you're comfortable.

Here's how. Go to phpMyAdmin, navigate to select the wp_posts table. Click on the SQL tab.



In the big box, delete the code that's already in there, paste the following – please make sure there are no hard line breaks.

```
update `wp_posts` set `post_content` =  
replace(post_content, 'TEXT_TO_BE_REPLACED', 'TEXT_  
TO_REPLACE_WITH');
```

Change out TEXT_TO_BE_REPLACED with the exact text that you want to change. e.g. <http://clientblog.com/wp/wp-content/uploads>

Change out TEXT_TO_REPLACE_WITH with the new text. e.g. <http://clientblog.com/wp/wp-content/uploads>

Click Go. MySQL will get to work and you'll see some messages on the top of your screen about how many lines or records have been changed or affected by your command. Check your work.

WARNING! - this code above only works on WordPress posts. Do not use it to search and replace other things like user data, or post titles. It won't work and will likely mess up your database.

Next Lesson...

Is all about WordPress on the go. No, we're not going anywhere but we are going to make our WordPress site mobile friendly.

Lynette Chandler

Lynette Chandler

Tech Based Training