

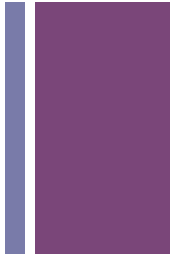


## Your New BFFs

The WordPress Core Tables + The MySQL Database  
WordCamp Pittsburgh 2017

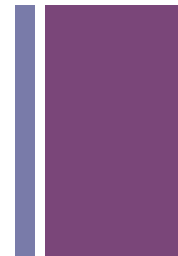


# Is this what you came for?



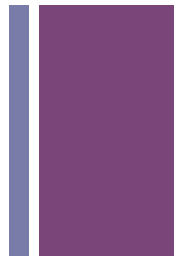
- You're a rockstar with the front end and the WordPress admin, but the back end is... a mystery. Ever wonder what happens when you create a user or save a post? Where does the data go? Let's dive in to the **MySQL database via phpMyAdmin**. We'll explore the **WordPress core tables** and **investigate the fields** to see how the data is stored. We'll also touch on **optimizing your tables** and **creating database backups** without a plugin.

+ Hey!



- Rene Morozowich
- Freelance WordPress Developer
- In former lives I've been... a database developer, a programmer and an instructor
- [renemorozowich.com](http://renemorozowich.com)
- [hello@renemorozowich.com](mailto:hello@renemorozowich.com)
- @ReneMorozowich

# + What are we doing here?

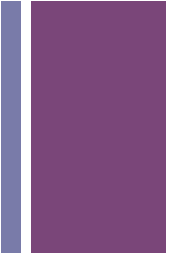


- Follow along with your computer or just watch
- Find me on Twitter to see published slides [#wcpgh](#)
- Game plan:
  - Databases are a thing
  - Let's go look at one
  - WordPress uses a database
  - Let's back up and optimize the WordPress database



# Database Basics + Lingo

# + What is a database?



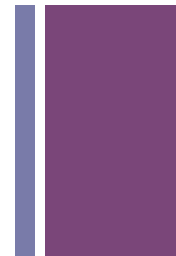
- Organized collection of data
- Data is stored in **tables**
  - Each table should be about **one specific topic**
  - Avoid repeating data
- Tables have **fields** (or columns)
  - Each piece of information that you want to collect about that topic
  - Each field has a specific **data type** (what's stored there?)
- Tables have **records** (or rows)
  - Each record is a complete set

# + Example

- Table Customer
  - All fields (columns) related to the customer
  - One field is a unique identifier called a **primary key**
  - One record (row) per customer

CustomerID	Name	Address	City	State
1	Mary Smith	123 Main St	Pittsburgh	PA
2	John Doe	444 South St	Pittsburgh	PA
3	Lisa Jones	551 North Ave	Pittsburgh	PA

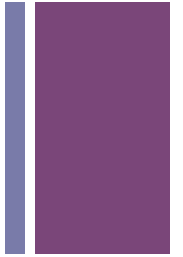
# + Relationships



- MySQL stores **relational** databases
- Remember we collect data in a table about one topic
- Other topics (tables) can be related
- Think of it like a parent/child relationship
  - Can call this **one to many** (it's the most common)
  - One parent can have many children
- We said the unique identifier in the parent table is the **primary key**
- The related field in the child table is the **foreign key**
  - Same data type and let's be cool and name them the same



# + Example

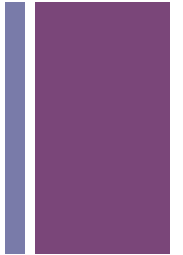


CustomerID	Name	Address	City	State
1	Mary Smith	123 Main St	Pittsburgh	PA
2	John Doe	444 South St	Pittsburgh	PA
3	Lisa Jones	551 North Ave	Pittsburgh	PA

One customer can have many orders

OrderID	CustomerID	Item	Quantity	OrderDate	Paid
1234	1	Orange shirt	2	9/1/2017	Y
1235	1	Pink shoes	1	9/2/2017	Y
1236	3	Jeans	2	9/3/2017	Y

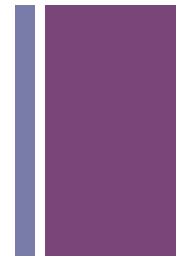
# + No no no no no no no!



Cust omer ID	Name	Address	City	State	OrderID	Item	Qua nti ty	OrderDate	Paid
1	Mary Smith	123 Main St	Pittsburgh	PA	1234	Orange shirt	2	9/1/2017	Y
1	Mary Smith	123 Main St	Pittsburgh	PA	1235	Pink shoes	1	9/2/2017	Y
2	John Doe	444 South St	Pittsburgh	PA					
3	Lisa Jones	551 North Ave	Pittsburgh	PA	1236	Jeans	2	9/3/2017	Y



# I know I just said...



- Each table should be about one specific topic
- But... there are other types of tables (the meta tables) in WordPress
- These tables contain:
  - An ID unique to the table
  - An ID that corresponds back to the table they're related to
  - A key
  - A value
- And they can hold almost anything
- There are drawbacks, however this allows for flexibility!!

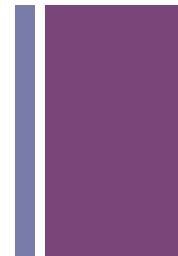


+

MySQL + phpMyAdmin

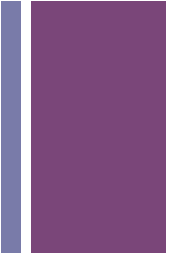


# About MySQL



- WordPress supports MySQL 5.0.15 or higher and any version of MariaDB
- “The world’s most popular open source database”
- From Oracle
- Relational Database Management System (RDBMS)
- Uses Structured Query Language (SQL)

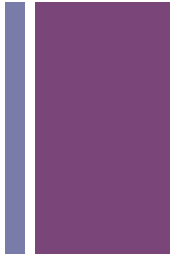
# + Getting to phpMyAdmin



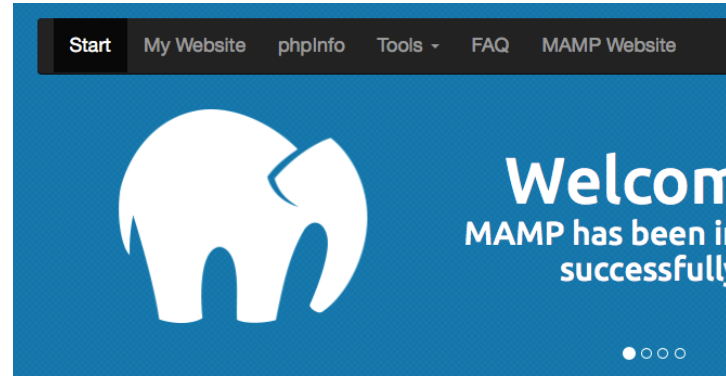
- Every place is different!
- I'll give you four examples and none are the same
- Look around your hosting environment



# Working locally, using MAMP



- Start MAMP
- Once the servers start
- <http://localhost:8888/MAMP/> opens
- See link for phpMyAdmin under MySQL



**PHP**  
phpinfo shows the current configuration of PHP.

**MySQL**  
MySQL can be administered with [phpMyAdmin](#).  
To connect to the MySQL server from your own scripts use the following connection parameters:

**M/**  
3.5:

**Ne**  
Mobi

# + Hosting, Siteground

- Log in and go to the cPanel
- In the section “Databases” choose phpMyAdmin





# + Hosting, Pair

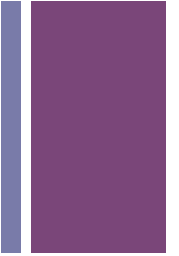
- Log in, go to Databases, choose Manage Your Databases
- Choose the desired database, then click phpMyAdmin
- You will then have to log in with the database username/  
password



A screenshot of the phpMyAdmin interface for a database. At the top, there is a navigation bar with icons and labels for 'phpMyAdmin', 'Optimize', 'Back Up', 'Change Password', 'Purge', and 'Delete'. Below this, the server information is displayed: 'Server: db154.pair.com'. The users section shows 'Users: 1031380\_2 (full access), 1031380\_2\_r (read access), 1031380\_2\_w (readwrite access)'. The disk usage is shown as 'Disk Usage: 89.47 M'. A blue button labeled 'View History' is located at the bottom right of the interface.

phpMyAdmin	Optimize	Back Up	Change Password	Purge	Delete
Server:	db154.pair.com				
Users:	1031380_2 (full access), 1031380_2_r (read access), 1031380_2_w (readwrite access),				
Disk Usage:	89.47 M				<a href="#">View History</a>

# + Hosting, 1&1



- Setup as Managed by 1&1, you will not be able to see or access your database
- You must switch to Standard mode
- Your installation is moved from their system database to your own database
- Access under My Products, MySQL database, phpMyAdmin



# Walking around the screen



The screenshot shows the phpMyAdmin interface for a MySQL server at localhost:8889. The left sidebar contains a tree view of databases, with an orange box highlighting the list. The main content area is divided into several sections: General Settings, Appearance Settings, Database server, Web server, and phpMyAdmin. The Database server section lists server details like type, version, and user. The Web server section lists installed modules and versions. The phpMyAdmin section provides version information and links to documentation and support.

**Databases here on the left**

- New
- accelerate
- information\_schema
- mysql
- passionDXpb6J
- performance\_schema
- rene\_wordpress
- rm\_FEGDvmEvUd58LUMW
- test

**General Settings**

Server connection collation: utf8mb4\_unicode\_ci

**Appearance Settings**

Language: English

Theme: Original

Font size: 82%

[More settings](#)

**Database server**

- Server: Localhost via UNIX socket
- Server type: MySQL
- Server version: 5.5.42 - Source distribution
- Protocol version: 10
- User: root@localhost
- Server charset: UTF-8 Unicode (utf8)

**Web server**

- Apache/2.2.29 (Unix) mod\_wsgi/3.5 Python/2.7.10 PHP/7.0.8 mod\_ssl/2.2.29 OpenSSL/0.9.8zh DAV/2 mod\_fastcgi/2.4.6 mod\_perl/2.0.9 Perl/v5.22.0
- Database client version: libmysql - mysqlnd 5.0.12-dev - 20150407 - \$Id: 241ae00989d1995ffcbbf63d579943635faf9972 \$
- PHP extension: mysqli
- PHP version: 7.0.8

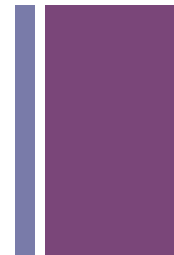
**phpMyAdmin**

- Version information: 4.4.10
- [Documentation](#)
- [Wiki](#)
- [Official Homepage](#)
- [Contribute](#)
- [Get support](#)
- [List of changes](#)

phpMyAdmin



# And around



Server: localhost:8889 » Database: test

Structure SQL Search Query Export Import Operations Privileges Routines More

Table	Action	Rows	Type	Collatic
<input type="checkbox"/> wp_commentmeta	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_general
<input type="checkbox"/> wp_comments	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_general
<input type="checkbox"/> wp_links	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_general
<input type="checkbox"/> wp_options	Browse Structure Search Insert Empty Drop	186	InnoDB	utf8_general
<input type="checkbox"/> wp_postmeta	Browse Structure Search Insert Empty Drop	254	InnoDB	utf8_general
<input type="checkbox"/> wp_posts	Browse Structure Search Insert Empty Drop	52	InnoDB	utf8_general
<input type="checkbox"/> wp_termmeta	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_general
<input type="checkbox"/> wp_terms	Browse Structure Search Insert Empty Drop	18	InnoDB	utf8_general
<input type="checkbox"/> wp_term_relationships	Browse Structure Search Insert Empty Drop	25	InnoDB	utf8_general
<input type="checkbox"/> wp_term_taxonomy	Browse Structure Search Insert Empty Drop	18	InnoDB	utf8_general
<input type="checkbox"/> wp_usermeta	Browse Structure Search Insert Empty Drop	28	InnoDB	utf8mb4_uni
<input type="checkbox"/> wp_users	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_uni
<b>12 tables</b>	<b>Sum</b>	<b>583</b>	<b>InnoDB</b>	<b>utf8_bi</b>

Check All With selected: [dropdown]

Print view Data Dictionary

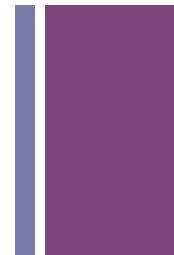
Create table Number of columns: 4

Select database on left

See tables on right



# And around...



Server: localhost:8889 » Database: test

Structure SQL Search Query Export Import Operations Privileges Routines More

Table	Action	Rows	Type	Collation
<input type="checkbox"/> wp_commentmeta	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_general
<input type="checkbox"/> wp_comments	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_general
<input type="checkbox"/> wp_links	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_general
<input type="checkbox"/> wp_options	Browse Structure Search Insert Empty Drop	186	InnoDB	utf8_general
<input type="checkbox"/> wp_postmeta	Browse Structure Search Insert Empty Drop	254	InnoDB	utf8_general
<input type="checkbox"/> wp_posts	Browse Structure Search Insert Empty Drop	52	InnoDB	utf8_general
<input type="checkbox"/> wp_termmeta	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_general
<input type="checkbox"/> wp_terms	Browse Structure Search Insert Empty Drop	18	InnoDB	utf8_general
<input type="checkbox"/> wp_term_relationships	Browse Structure Search Insert Empty Drop	25	InnoDB	utf8_general
<input type="checkbox"/> wp_term_taxonomy	Browse Structure Search Insert Empty Drop	18	InnoDB	utf8_general
<input type="checkbox"/> wp_usermeta	Browse Structure Search Insert Empty Drop	28	InnoDB	utf8mb4_uni
<input type="checkbox"/> wp_users	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_uni
12 tables	Sum	583	InnoDB	utf8_bi

Check All With selected: [dropdown]

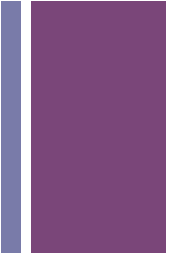
Print view Data Dictionary

Create table Number of columns: 4

For each table:  
Browse,  
Structure,  
Rows

Size and  
Overhead,  
too (not  
pictured)

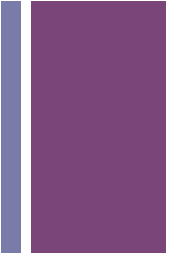
# + What are we looking at?



- The WordPress core tables!!
- There are 12 of them currently
  - A bit of a mystery with wp\_termmeta
- Last updated with version 4.4 (we're on 4.8.1)
- [https://codex.wordpress.org/Database\\_Description](https://codex.wordpress.org/Database_Description)
- Let's discuss what they are, how they're related and the fields in each table



# But first... a note about prefixes



- The default naming for WordPress tables is `wp_table_name`
- There is debate on whether using a different prefix makes your database more secure
- Some hosting companies also require you to use a prefix other than `wp_`



# WordPress Core Tables

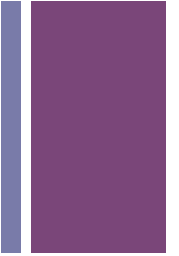


# + Posts!



- **wp\_posts**
  - The most important!!
  - More than just posts
  - Also stores pages, menu items, media attachments and custom types
- **wp\_postmeta**
  - Holds extra information about individual items above
  - The first of our key/value pair tables
- 1 to many
  - For every one post (primary key ID), there can be many meta records (foreign key post\_id)

# + Fields in wp\_posts



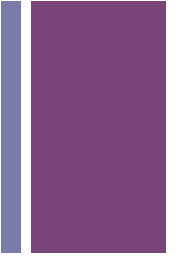
- **ID** – unique number assigned to each post
- **post\_author** – the user ID who created it (**wp\_users**)
- **post\_date** – time and date of creation
- **post\_date\_gmt** – GMT time and date of creation (no dependency on a site's timezone in the future, so brilliant)
- **post\_content** – holds all the content for the post, including HTML, shortcodes and other content
- **post\_title** – title of the post

# + Fields in wp\_posts



- **post\_excerpt** – custom intro or short version of the content
- **post\_status** – status of the post such as *draft*, *pending*, *private*, *publish*
- **comment\_status** – if comments are allowed
- **ping\_status** – if the post allows ping and trackbacks
- **post\_password** – optional password used to view the post
- **post\_name** – URL friendly slug of the post title

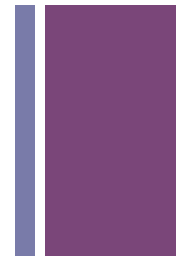
# + Wait, what's a ping?



- In the early days, blog A could notify blog B automatically when blog A linked to blog B's content
- The pingback would appear in blog B's comment moderation queue with a link to blog A's website
- Kinda cool, but now often just used by spammers
- Turn it off under Settings, Discussion



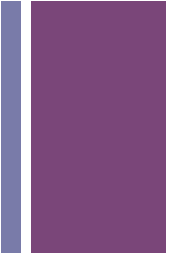
# Back to it. More fields in wp\_posts



- **to\_ping** – a list of URLs WP should send pingbacks to when updated
- **pinged** – a list of URLs WP has sent pingbacks to when updated
- **post\_modified** – time and date of last modification
- **post\_modified\_gmt** – GMT time and date of last modification
- **post\_content\_filtered** – used by plugins to cache a version of post\_content
- **post\_parent** – used to create a relationship between this post and another when this post is a revision, attachment or another type



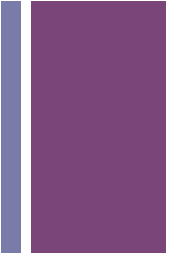
# Oh so many fields in wp\_posts



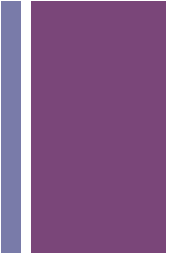
- **guid** – global unique identifier, the permanent URL to the post, not the permalink version
- **menu\_order** – holds the display number for pages and other non-post types
- **post\_type** – the content type identifier (posts, revisions, pages, menu items, media attachments and custom post types)
- **post\_mime\_type** – only used for attachments, the MIME type of the uploaded file
- **comment\_count** – total number of comments, pingbacks and trackbacks

# + Fields in wp\_postmeta

- **meta\_id** – unique number assigned to each row of the table
- **post\_id** – ID of the related post (**wp\_posts**)
- **meta\_key** – an identifying key for the piece of data
- **meta\_value** – the actual piece of data



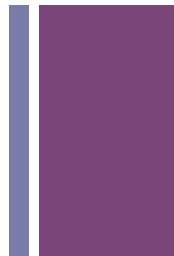
# + Terms and taxonomies



- Four term tables, but before we get to them...
- Terms
  - Any descriptive words
- Taxonomies
  - Category
  - Tag
  - Other custom ones
    - Cars could be a taxonomy with terms like Toyota, Honda and Audi
    - Fruits a taxonomy with terms like strawberries, apples and grapes

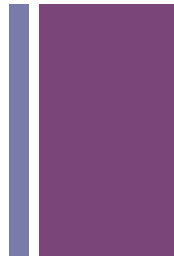


# + Term tables



- **wp\_terms**
  - Holds our generic terms (descriptive adjectives)
- **wp\_termmeta**
  - Holds additional data about terms
- 1 to many
  - For every one term (primary key term\_id), there can be many meta records (foreign key term\_id)

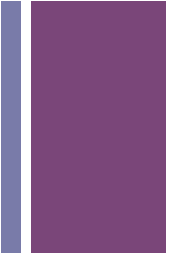
# + Fields in wp\_terms



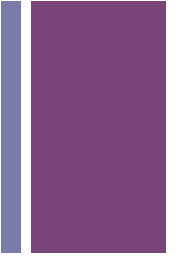
- **term\_id** – unique number assigned to each term
- **name** – the name of the term
- **slug** – the URL friendly slug of the name
- **term\_group** – ability for themes or plugins to group terms together to use aliases; not populated by WordPress core itself

# + Fields in wp\_termmeta

- **meta\_id** – unique number assigned to each row of the table
- **term\_id** – ID of the related term (**wp\_terms**)
- **meta\_key** – an identifying key for the piece of data
- **meta\_value** – the actual piece of data



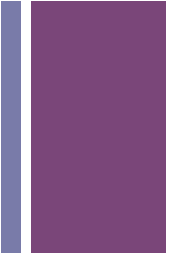
# + Now with the taxonomy



- Mark each term with an appropriate taxonomy
- What's a taxonomy again?
- Remember categories and tags
- Creating a term/taxonomy pair



# Fields in wp\_term\_taxonomy



- **term\_taxonomy\_id** – unique number assigned to each row of the table
- **term\_id** – the ID of the related term (**wp\_terms**)
- **taxonomy** – the slug of the taxonomy (category, post\_tag, etc.)
- **description** – description of the term in this taxonomy
- **parent** – ID of a parent term; used for hierarchical taxonomies like categories
- **count** – number of post objects assigned the term for this taxonomy



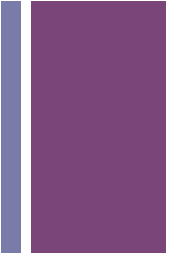
# Finally with the relationship



- This is a junction table!
- Many to many
  - Each post can have many term/taxonomy pair records
  - Each term/taxonomy pair can be used by many posts



# Fields in wp\_term\_relationships



- **object\_id** – the ID of the post object (**wp\_posts**)
- **term\_taxonomy\_id** – the ID of the term / taxonomy pair (**wp\_term\_taxonomy**)
- **term\_order** – allow ordering of terms for an object, not fully used



# Some additional reading

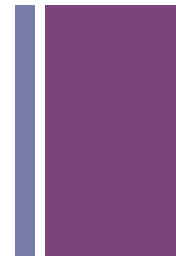


- <https://codex.wordpress.org/Taxonomies>
- [https://codex.wordpress.org/WordPress\\_Taxonomy](https://codex.wordpress.org/WordPress_Taxonomy)
- [https://en.wikipedia.org/wiki/Associative\\_entity](https://en.wikipedia.org/wiki/Associative_entity)



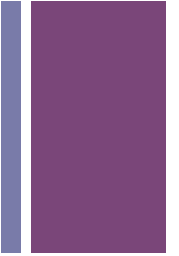


# Comments



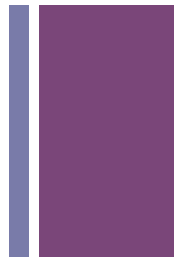
- **wp\_comments**
  - Each comment of a post
- **wp\_commentmeta**
  - Any additional information needed about comments; Akismet anti-spam plugin uses this for example
- 1 to many
  - For every one comment (primary key comment\_ID), there can be many meta records (foreign key comment\_id)

# + Fields in wp\_comments



- **comment\_ID** – unique number assigned to each comment
- **comment\_post\_ID** – ID of the post this comment relates to (**wp\_posts**)
- **comment\_author** – Name of the comment author
- **comment\_author\_email** – Email of the comment author
- **comment\_author\_url** – URL for the comment author
- **comment\_author\_IP** – IP Address of the comment author
- **comment\_date** – Time and date the comment was posted
- **comment\_date\_gmt** – GMT time and date the comment was posted

# + Fields in wp\_comments



- **comment\_content** – the actual comment text
- **comment\_karma** – unused by WordPress core; can be used by plugins to help manage comments
- **comment\_approved** – if the comment has been approved
- **comment\_agent** – where the comment was posted from (browser, operating system, etc.)
- **comment\_type** – type of comment (comment, pingback or trackback)
- **comment\_parent** – refers to another comment when this comment is a reply
- **user\_id** – ID of the comment author if registered user on the site ([wp\\_users](#))



# Fields in wp\_commentmeta



- **meta\_id** – unique number assigned to each row of the table
- **comment\_id** – the ID of the related comment (**wp\_comments**)
- **meta\_key** – an identifying key for the piece of data
- **meta\_value** – the actual piece of data

# + Users



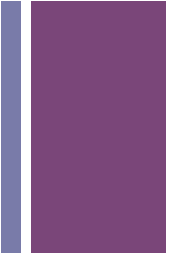
- **wp\_users**
  - Each WordPress user
- **wp\_usermeta**
  - Any additional information needed about users
  - For example, first\_name and last\_name
- 1 to many
  - For every one user (primary key ID), there can be many meta records (foreign key user\_id)

# + Fields in wp\_users

- **ID** – unique number assigned to each user
- **user\_login** – unique username
- **user\_pass** – hash of the user's password
- **user\_nicename** – user display name
- **user\_email** – email address of the user



# + Fields in wp\_users



- **user\_url** – user's URL
- **user\_registered** – time and date the user registered
- **user\_activation\_key** – used for resetting passwords
- **user\_status** – was used in Multisite pre WordPress 3.0 to indicate a spam user
- **display\_name** – desired name to be used publicly in the site (user\_login, user\_nicename or first\_name/last\_name from wp\_usermeta)

# + Fields in wp\_usermeta

- **umeta\_id** – unique number assigned to each row of the table
- **user\_id** – ID of the related user (**wp\_users**)
- **meta\_key** – an identifying key for the piece of data
- **meta\_value** – the actual piece of data





# + Other



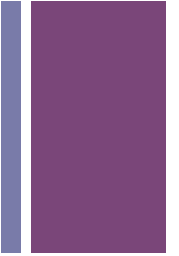
## ■ wp\_options

- Table that stores site configuration settings
- Data about themes, active plugins, widgets, temporary cached data
- Where other plugins and themes store settings

## ■ wp\_links

- Many sites used to have a blogroll (list of links to other sites)
- This table held those links!
- Removed from the admin UI
- But table remains for backwards compatibility

# + Fields in wp\_options



- **option\_id** – unique number assigned to each row of the table
- **option\_name** – an identifying key for the piece of data
- **option\_value** – the actual piece of data (often serialized)
- **autoload** – controls if the option is automatically loaded by the function `wp_load_alloptions()` (puts options into object cache on each page load)

# + Browse wp\_options

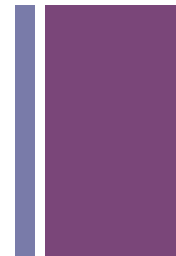
The screenshot shows the phpMyAdmin interface for a MySQL database. The left sidebar shows a tree view of databases, with 'test' selected and 'wp\_options' highlighted. The main area displays the 'Browse' view for the 'wp\_options' table. The table has 15 rows and 4 columns: option\_id, option\_name, option\_value, and autoload. The data is as follows:

option_id	option_name	option_value	autoload
1	siteurl	http://localhost:8888/test	yes
2	home	http://localhost:8888/test	yes
3	blogname	Test Blog	yes
4	blogdescription	A test blog for a WordCamp talk on databases.	yes
5	users_can_register	0	yes
6	admin_email	rene@renemorozowich.com	yes
7	start_of_week	1	yes
8	use_balanceTags	0	yes
9	use_smilies	1	yes
10	require_name_email	1	yes
11	comments_notify	1	yes
12	posts_per_rss	10	yes
13	rss_use_excerpt	0	yes
14	mailserver_url	mail.example.com	yes
15	mailserver_login	login@example.com	yes

Each record  
in our  
options table



# Serialized Data



- Some rows store serialized data, like the active\_plugins row

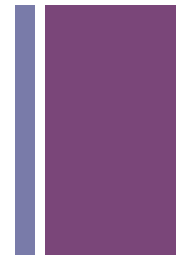
```
a:3:{i:0;s:31:"query-monitor/query-monitor.php";i:1;s:57:"accesspress-instagram-feed/accesspress-instagram-feed.php";i:2;s:19:"akismet/akismet.php";}
```

```
$array = array(  
'0' => 'query-monitor/query-monitor.php'  
'1' => 'accesspress-instagram-feed/accesspress-instagram-feed.php'  
'2' => 'akismet/akismet.php'  
);
```

- Read more at <https://wpengine.com/support/wordpress-serialized-data/>



# Look! There it is!



Server: localhost:8889 » Database: test » Table: wp\_options

Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

Showing rows 30 - 59 (186 total, Query took 0.0003 seconds.)

```
SELECT * FROM `wp_options`
```

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP Code ] [ Refresh ]

<< < 2 > >> |  Show all | Number of rows: 25 | Filter rows: Search this table

Sort by key: None

+ Options

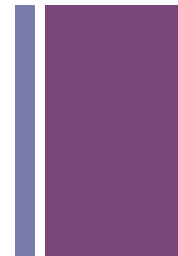
			option_id	option_name	option_value	autoload
<input type="checkbox"/>	Edit	Copy	Delete	32	moderation_keys	no
<input type="checkbox"/>	Edit	Copy	Delete	33	active_plugins	a:6:{i:0;s:30:"advanced-custom-fields/acf.php";i:1...
<input type="checkbox"/>	Edit	Copy	Delete	34	category_base	yes
<input type="checkbox"/>	Edit	Copy	Delete	35	ping_sites	http://rpc.pingomatic.com/
<input type="checkbox"/>	Edit	Copy	Delete	37	comment_max_links	2
<input type="checkbox"/>	Edit	Copy	Delete	38	gmt_offset	0



Backup + Optimize



# A note on backups



- Before making a change, back up your database!
- Also backup regularly
- Do it. No seriously. Do it.



# Creating a database backup

- Choose the **database**, click **Export** and choose **Custom**

The screenshot shows the phpMyAdmin interface. On the left, a tree view shows the database structure, with the 'test' database selected. The main panel displays the 'Export' dialog for the 'test' database. The 'Export Method' section has two radio buttons: 'Quick - display only the minimal options' (unselected) and 'Custom - display all possible options' (selected). Below this, the 'Table(s):' section shows a list of tables in the 'test' database, including 'wp\_commentmeta', 'wp\_comments', 'wp\_links', 'wp\_options', 'wp\_postmeta', 'wp\_posts', 'wp\_termmeta', 'wp\_terms', 'wp\_term\_relationships', and 'wp\_term\_taxonomy'. The 'Export' button is visible in the top navigation bar.



# + Options

- Save output to a file

**Output:**

Rename exported databases/tables/columns

Save output to a file

File name template:   use this for future exports

Character set of the file:

Compression:

View output as text

Skip tables larger than  MiB

# + More options

- Check **Add DROP TABLE / VIEW / PROCEDURE / FUNCTION / EVENT / TRIGGER statement**
- If the tables (etc.) are already there, they will be dropped and recreated

## Object creation options

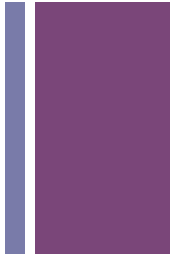
---

Add statements:

- Add CREATE DATABASE / USE statement
- Add DROP TABLE / VIEW / PROCEDURE / FUNCTION / EVENT / TRIGGER statement



# Done!



- Click **Go**
- This downloads a .sql file for you
- Feel free to check it out in your text editor



# What does optimizing mean?



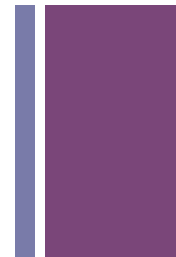
- Your database will grow!
- Anytime you add/change/remove data
- A large database can affect performance
- It takes awhile for your site to fetch the data from the database
  - When there's so much to look through
  - When data isn't stored efficiently

# + It's like...



- Cleaning out your closet
  - You put things in there day after day
  - It gets cluttered
  - Harder and harder to find things
  - Space isn't used efficiently over time
- Defragmenting your hard drive
- And maybe a little like that 3,000 mile oil change

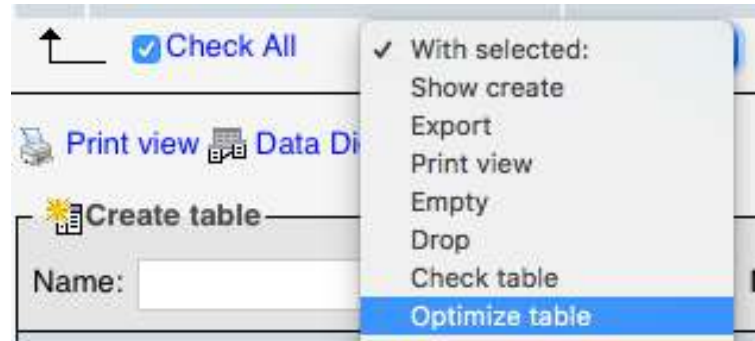
# + What does it actually do?



- “Reorganizes the physical storage or table data and associated index data, to reduce storage space and improve I/O efficiency when accessing the table”
- Got that?
- Plain English:
  - Reorganizes it
  - Frees up space
  - Now data can get **in** and **out** more quickly (I/O)

# + How to optimize tables

- In phpMyAdmin, choose the database on the left
- On the right at the bottom of the table list, check “Check all”
- In the drop down “With selected:” choose Optimize table



## + Also...

- You may also want to purge old post revisions!
- You can do this with the database, however I don't recommend it
  - Running insert/update/delete SQL statements can be scary!
- Try a plugin like [WP-Optimize](#) or [Better Delete Revision](#)
- You can also limit the number of post revisions you keep
- Add/modify in your wp-config.php file:

```
define( 'WP_POST_REVISIONS' , 5 );
```



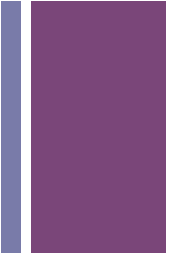


Questions?

Thanks for attending!



# Credits



- Field descriptions from the tables were used from <https://deliciousbrains.com/tour-wordpress-database/>