

THINGS YOU DIDN'T KNOW YOU NEED TO KNOW ABOUT WORDPRESS AND DATABASES

crash course into database architecture

GABOR JAVORSZKY

- freelancer, work with clients on WooCommerce / Subscriptions / performance problems
- previously senior dev at electric studio
- previously dev at Prosspress
- blog on javorszky.co.uk
- twitter in bottom left corner

ASSUMPTIONS

1. Site is large with lots of data
2. potentially one with WooCommerce
3. InnoDB storage engine (your WordPress site most probably uses this by default)

Older versions of WordPress may run on MyISAM, but that's been deprecated. Main advantage of MyISAM used to be significantly faster read speeds, but with InnoDB catching up, that's no longer the case.

COMMON (SLOW) QUERIES

get me all the orders for user 4

*get me all subscriptions where the next payment date
is between X and Y*

get me all orders where the total is \$19.92

*get me the sum of all order totals where the order
date is in May and order by order total. or date*

basically doing anything on the admin area

All of those are going to be
...really...
...slow.

THE TALK

- indexes and how to use them
- storing data in multiple tables vs one table - foreign keys and why they're awesome
- tools to measure stuff
- won't touch caching
- won't touch ElasticSearch / Solr

Can I help existing sites? - probably not

Can I help you build better plugins / themes? - hope so

INDEXES

What are they?

Why are they important?

How do they work?

Indexable, unindexable (looking at you, postmeta value!)

Indexable: anything that you use to look things up, where values are either unique, or short. meta_value is a LONGTEXT, so feasibly you could only index the first 191 chars of it anyways. Any variation after that gets lost. Plus it would be a lot of extra data in the tablespace slowing down index lookup for the optimizer.

WHAT ARE INDEXES



Imagine your data to be the books / contents of those books in a Library. Say you want to find books by Isaac Asimov. You don't have a handy list (ignore the letters on the walls), so you check the first, then the second, etc... it'll take forever.



 Sanwal Deen

Index cards hold relevant information about the books, and are themselves ordered. You can easily find the few cards that relate to books by Isaac Asimov. The cards will tell you where you will find them in the library.

HOW DO INDEXES WORK?

From [MySQL documentation on indexes](#)

Indexes are used to find rows with specific column values quickly. Without an index, MySQL must begin with the first row and then read through the entire table to find the relevant rows. The larger the table, the more this costs. If the table has an index for the columns in question, MySQL can quickly determine the position to seek to in the middle of the data file without having to look at all the data. This is much faster than reading every row sequentially.

WHY ARE THEY IMPORTANT?

In short: well designed indexes will get you the data in a fast, efficient manner.

You can do more on the same hardware => cheaper hosting => faster site => happier customers => more income / profit

COLUMNS, DATA TYPES

- numeric: INTEGER, BIGINT, SIGNED / UNSIGNED
- string: TEXT, VARCHAR, LONGTEXT
- date type: DATE, TIMESTAMP, DATETIME
- specifying character lengths: VARCHAR(191), BIGINT(20)

Examples of signed / unsigned. "Can it have a negative sign?". Highest binary place is the sign

```
// signed -127 – +127
01111111 = +127
11111111 = -127

// unsigned 0 – 255
01111111 = 127
11111111 = 255
```

WHAT / HOW SHOULD BE INDEXED?

From [MySQL documentation on optimization and indexes](#)

Although it can be tempting to create an indexes for every possible column used in a query, unnecessary indexes waste space and waste time for MySQL to determine which indexes to use. Indexes also add to the cost of inserts, updates, and deletes because each index must be updated. You must find the right balance to achieve fast queries using the optimal set of indexes.

WHAT / HOW SHOULD BE INDEXED?

"Indexes are less important for queries on small tables, or big tables where report queries process most or all of the rows. When a query needs to access most of the rows, reading sequentially is faster than working through an index. Sequential reads minimize disk seeks, even if not all the rows are needed for the query. See Section 8.2.1.19, "Avoiding Full Table Scans" for details."

[MySQL doc / Indexes](#)

[MySQL doc / Avoiding full table scan](#)

UNIQUE / MULTI INDEX

UNIQUE: one value can only appear once in a row. Trying to insert same value will result in a database error message. Super fast

Example: post IDs

Multi index: same value can repeat. Still fast, though additional filtering required.

Example: postmeta_key

NULL VALUES IN INDEXES

From [optimizing InnoDB queries](#)

If an indexed column cannot contain any NULL values, declare it as NOT NULL when you create the table. The optimizer can better determine which index is most effective to use for a query, when it knows whether each column contains NULL values.

tldr; if it shouldn't be NULL, declare it so. It will be faster.

protip: primary keys will always be NOT NULL!

protip2: unique keys permit multiple NULL values!

PRIMARY KEY:

A unique index where all key columns must be defined as NOT NULL. If they are not explicitly declared as NOT NULL, MySQL declares them so implicitly (and silently). A table can have only one PRIMARY KEY. The name of a PRIMARY KEY is always PRIMARY, which thus cannot be used as the name for any other kind of index.

UNIQUE KEY:

A UNIQUE index creates a constraint such that all values in the index must be distinct. An error occurs if you try to add a new row with a key value that matches an existing row. For all engines, a UNIQUE index permits multiple NULL values for columns that can contain NULL.

COMPOUND INDEXES

```
-- given these columns  
-- ID, post_name, post_date  
  
CREATE INDEX (^ID^, ^post_name^, ^post_date^)  
-- will result in a compound index across ID, post_name, post_date, in that order
```

“If the table has a multiple-column index, any leftmost prefix of the index can be used by the optimizer to look up rows.”

```
SELECT * FROM table WHERE `ID` = x, `post_name` = y      -- uses index (1,2)  
SELECT * FROM table WHERE `ID` = x                      -- uses index (1)  
SELECT * FROM table WHERE `ID` = x, `post_date` = z       -- NOT using index (1,3)  
SELECT * FROM table WHERE `post_name` = y, `post_date` = z -- NOT using index (2,3)  
SELECT * FROM table WHERE `post_name` = y                -- NOT using index (2)
```

WP DB - POSTS

Field	Type	Length	Unsigned	Zerofill	Binary	Allow Null	Key	Default	Extra	Encoding	Collation
ID	BIGINT	20	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	MUL		auto_increment	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
post_author	BIGINT	20	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	MUL	0	None	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
post_date	DATETIME		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	MUL	0000-00-00 00:00:00	None	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
post_date_gmt	DATETIME		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		0000-00-00 00:00:00	None	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
post_content	LONGTEXT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			None	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
post_title	TEXT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			None	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
post_excerpt	TEXT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			None	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
post_status	VARCHAR	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	MUL	publish	None	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
comment_status	VARCHAR	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		open	None	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
ping_status	VARCHAR	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		open	None	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
post_password	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			None	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
post_name	VARCHAR	200	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	MUL		None	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
to_ping	TEXT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			None	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
pinged	TEXT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			None	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
post_modified	DATETIME		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		0000-00-00 00:00:00	None	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
post_modified_gmt	DATETIME		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		0000-00-00 00:00:00	None	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
post_content_filtered	LONGTEXT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			None	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
post_parent	BIGINT	20	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	MUL	0	None	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
guid	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			None	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
menu_order	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		0	None	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
post_type	VARCHAR	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	MUL	post	None	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
post_mime_type	VARCHAR	100	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			None	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
comment_count	BIGINT	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		0	None	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

INDEXES									
Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Comment	
0	PRIMARY	1	ID	A	296	NULL	NULL		
1	post_name	1	post_name	A	296	191	NULL		
1	type_status_date	1	post_type	A	16	NULL	NULL		
1	type_status_date	2	post_status	A	37	NULL	NULL		
1	type_status_date	3	post_date	A	296	NULL	NULL		
1	type_status_date	4	ID	A	296	NULL	NULL		
1	post_parent	1	post_parent	A	42	NULL	NULL		
1	post_author	1	post_author	A	4	NULL	NULL		

WP DB - POSTS

Field	Type	Length	Unsigned	Zerofill	Binary	Allow Null	Key	Default	Extra	Encoding	Collation
ID	BIGINT	20	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	MUL		auto_increment	<input type="checkbox"/>	<input type="checkbox"/>
post_author	BIGINT	20	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	MUL	0	None	<input type="checkbox"/>	<input type="checkbox"/>
post_date	DATETIME		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	MUL	0000-00-00 00:00:00	None	<input type="checkbox"/>	<input type="checkbox"/>
post_date_gmt	DATETIME		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		0000-00-00 00:00:00	None	<input type="checkbox"/>	<input type="checkbox"/>
post_content	LONGTEXT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			None	<input type="checkbox"/>	<input type="checkbox"/>
post_title	TEXT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			None	<input type="checkbox"/>	<input type="checkbox"/>
post_excerpt	TEXT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			None	<input type="checkbox"/>	<input type="checkbox"/>
post_status	VARCHAR	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	MUL	publish	None	<input type="checkbox"/>	<input type="checkbox"/>
comment_status	VARCHAR	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		open	None	<input type="checkbox"/>	<input type="checkbox"/>
ping_status	VARCHAR	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		open	None	<input type="checkbox"/>	<input type="checkbox"/>
post_password	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			None	<input type="checkbox"/>	<input type="checkbox"/>
post_name	VARCHAR	200	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	MUL		None	<input type="checkbox"/>	<input type="checkbox"/>
to_ping	TEXT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			None	<input type="checkbox"/>	<input type="checkbox"/>
pinged	TEXT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			None	<input type="checkbox"/>	<input type="checkbox"/>
post_modified	DATETIME		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		0000-00-00 00:00:00	None	<input type="checkbox"/>	<input type="checkbox"/>
post_modified_gmt	DATETIME		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		0000-00-00 00:00:00	None	<input type="checkbox"/>	<input type="checkbox"/>
post_content_filtered	LONGTEXT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			None	<input type="checkbox"/>	<input type="checkbox"/>
post_parent	BIGINT	20	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	MUL	0	None	<input type="checkbox"/>	<input type="checkbox"/>
guid	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			None	<input type="checkbox"/>	<input type="checkbox"/>
menu_order	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		0	None	<input type="checkbox"/>	<input type="checkbox"/>
post_type	VARCHAR	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	MUL	post	None	<input type="checkbox"/>	<input type="checkbox"/>
post_mime_type	VARCHAR	100	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			None	<input type="checkbox"/>	<input type="checkbox"/>
comment_count	BIGINT	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		0	None	<input type="checkbox"/>	<input type="checkbox"/>

INDEXES									
Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Comment	
0	PRIMARY	1	ID	A	296	NULL	NULL		
1	post_name	1	post_name	A	296	191	NULL		
1	type_status_date	1	post_type	A	16	NULL	NULL		
1	type_status_date	2	post_status	A	37	NULL	NULL		
1	type_status_date	3	post_date	A	296	NULL	NULL		
1	type_status_date	4	ID	A	296	NULL	NULL		
1	post_parent	1	post_parent	A	42	NULL	NULL		
1	post_author	1	post_author	A	4	NULL	NULL		

Highlighted the unsigned part. Reason post_parent and ID have the same type and length is because post_parent refers to an ID. post_author refers to an ID in the users table, which is also the same type / length. And because IDs cannot be negative, unsigned makes sense.

WP DB - POSTS

The screenshot shows the 'posts' table structure and its indexes in Sequel Pro.

Table Structure:

Field	Type	Length	Unsigned	Zerofill	Binary	Allow Null	Key	Default	Extra	Encoding	Collation
ID	BIGINT	20	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	MUL		auto_increment	<input type="checkbox"/>	<input type="checkbox"/>
post_author	BIGINT	20	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	MUL	0	None	<input type="checkbox"/>	<input type="checkbox"/>
post_date	DATETIME		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	MUL	0000-00-00 00:00:00	None	<input type="checkbox"/>	<input type="checkbox"/>
post_date_gmt	DATETIME		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	MUL	0000-00-00 00:00:00	None	<input type="checkbox"/>	<input type="checkbox"/>
post_content	LONGTEXT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			None	<input type="checkbox"/>	<input type="checkbox"/>
post_title	TEXT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			None	<input type="checkbox"/>	<input type="checkbox"/>
post_excerpt	TEXT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			None	<input type="checkbox"/>	<input type="checkbox"/>
post_status	VARCHAR	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	MUL	publish	None	<input type="checkbox"/>	<input type="checkbox"/>
comment_status	VARCHAR	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			open	<input type="checkbox"/>	<input type="checkbox"/>
ping_status	VARCHAR	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			open	<input type="checkbox"/>	<input type="checkbox"/>
post_password	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			None	<input type="checkbox"/>	<input type="checkbox"/>
post_name	VARCHAR	200	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	MUL		None	<input type="checkbox"/>	<input type="checkbox"/>
to_ping	TEXT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			None	<input type="checkbox"/>	<input type="checkbox"/>
pinged	TEXT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			None	<input type="checkbox"/>	<input type="checkbox"/>
post_modified	DATETIME		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			0000-00-00 00:00:00	<input type="checkbox"/>	<input type="checkbox"/>
post_modified_gmt	DATETIME		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			0000-00-00 00:00:00	<input type="checkbox"/>	<input type="checkbox"/>
post_content_filtered	LONGTEXT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			None	<input type="checkbox"/>	<input type="checkbox"/>
post_parent	BIGINT	20	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	MUL	0	None	<input type="checkbox"/>	<input type="checkbox"/>
guid	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			None	<input type="checkbox"/>	<input type="checkbox"/>
menu_order	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			0	<input type="checkbox"/>	<input type="checkbox"/>
post_type	VARCHAR	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	MUL	post	None	<input type="checkbox"/>	<input type="checkbox"/>
post_mime_type	VARCHAR	100	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			None	<input type="checkbox"/>	<input type="checkbox"/>
comment_count	BIGINT	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			0	<input type="checkbox"/>	<input type="checkbox"/>

Indexes:

Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Comment
0	PRIMARY	1	ID	A	296	NULL	NULL	
1	post_name	1	post_name	A	296	191	NULL	
1	type_status_date	1	post_type	A	16	NULL	NULL	
1	type_status_date	2	post_status	A	37	NULL	NULL	
1	type_status_date	3	post_date	A	296	NULL	NULL	
1	type_status_date	4	ID	A	296	NULL	NULL	
1	post_parent	1	post_parent	A	42	NULL	NULL	
1	post_author	1	post_author	A	4	NULL	NULL	

There's a bug in Sequel pro in giving us the Key type. It should be PRI because: "If Key is MUL, the column is the first column of a nonunique index in which multiple occurrences of a given value are permitted within the column. If more than one of the Key values applies to a given column of a table, Key displays the one with the highest priority, in the order PRI, UNI, MUL." <https://dev.mysql.com/doc/refman/5.7/en/show-columns.html> <https://github.com/sequelpro/sequelpro/issues/2856>

```
MariaDB [dev.dev]> show columns from wp_posts;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| ID | bigint(20) unsigned | NO | PRI | NULL | auto_increment |
| post_author | bigint(20) unsigned | NO | MUL | 0 |
| post_date | datetime | NO | | 0000-00-00 00:00:00 |
| post_date_gmt | datetime | NO | | 0000-00-00 00:00:00 |
| post_content | longtext | NO | | NULL |
| post_title | text | NO | | NULL |
| post_excerpt | text | NO | | NULL |
| post_status | varchar(20) | NO | | publish |
| comment_status | varchar(20) | NO | | open |
| ping_status | varchar(20) | NO | | open |
| post_password | varchar(255) | NO | | |
| post_name | varchar(200) | NO | MUL | |
| to_ping | text | NO | | NULL |
| pinged | text | NO | | NULL |
| post_modified | datetime | NO | | 0000-00-00 00:00:00 |
| post_modified_gmt | datetime | NO | | 0000-00-00 00:00:00 |
| post_content_filtered | longtext | NO | | NULL |
| post_parent | bigint(20) unsigned | NO | MUL | 0 |
| guid | varchar(255) | NO | | |
| menu_order | int(11) | NO | | 0 |
| post_type | varchar(20) | NO | MUL | post |
| post_mime_type | varchar(100) | NO | | |
| comment_count | bigint(20) | NO | | 0 |
+-----+-----+-----+-----+-----+-----+
23 rows in set (0.00 sec)
```

WP DB - POSTS

Table structure for table `posts`:

Field	Type	Length	Unsigned	Zerofill	Binary	Allow Null	Key	Default	Extra	Encoding	Collation
ID	BIGINT	20					MUL		auto_increment		
post_author	BIGINT	20					MUL	0	None		
post_date	DATETIME						MUL	0000-00-00 00:00:00	None		
post_date_gmt	DATETIME						MUL	0000-00-00 00:00:00	None		
post_content	LONGTEXT								None	UTF-8 Unicode	utf8mb4_unicode_ci
post_title	TEXT								None	UTF-8 Unicode	utf8mb4_unicode_ci
post_excerpt	TEXT								None	UTF-8 Unicode	utf8mb4_unicode_ci
post_status	VARCHAR	20					MUL	publish	None	UTF-8 Unicode	utf8mb4_unicode_ci
comment_status	VARCHAR	20							None	UTF-8 Unicode	utf8mb4_unicode_ci
ping_status	VARCHAR	20							None	UTF-8 Unicode	utf8mb4_unicode_ci
post_password	VARCHAR	255							None	UTF-8 Unicode	utf8mb4_unicode_ci
post_name	VARCHAR	200					MUL		None	UTF-8 Unicode	utf8mb4_unicode_ci
to_ping	TEXT								None	UTF-8 Unicode	utf8mb4_unicode_ci
pinged	TEXT								None	UTF-8 Unicode	utf8mb4_unicode_ci
post_modified	DATETIME							0000-00-00 00:00:00	None		
post_modified_gmt	DATETIME							0000-00-00 00:00:00	None		
post_content_filtered	LONGTEXT								None	UTF-8 Unicode	utf8mb4_unicode_ci
post_parent	BIGINT	20					MUL	0	None		
guid	VARCHAR	255							None	UTF-8 Unicode	utf8mb4_unicode_ci
menu_order	INT	11						0	None		
post_type	VARCHAR	20					MUL	post	None	UTF-8 Unicode	utf8mb4_unicode_ci
post_mime_type	VARCHAR	100							None	UTF-8 Unicode	utf8mb4_unicode_ci
comment_count	BIGINT	20						0	None		

Indexes:

Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Comment
0	PRIMARY	1	ID	A	296	NULL	NULL	
1	post_name	1	post_name	A	296	191	NULL	
1	type_status_date	1	post_type	A	16	NULL	NULL	
1	type_status_date	2	post_status	A	37	NULL	NULL	
1	type_status_date	3	post_date	A	296	NULL	NULL	
1	type_status_date	4	ID	A	296	NULL	NULL	
1	post_parent	1	post_parent	A	42	NULL	NULL	
1	post_author	1	post_author	A	4	NULL	NULL	

WP DB - POSTMETA

The screenshot shows the 'postmeta' table structure and its indexes in MySQL Workbench.

Table Structure:

Field	Type	Length	Unsigned	Zerofill	Binary	Allow Null	Key	Default	Extra	Encoding	Collation	Comment
meta_id	BIGINT	20	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	PRI		auto_increme...	◊	◊	
post_id	BIGINT	20	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	MUL	0	None	◊	◊	
meta_key	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	MUL	NULL	None	UTF-8	◊ utf8mb4_unicode_ci	
meta_value	LONGTEXT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		NULL	None	UTF-8	◊ utf8mb4_unicode_ci	

A red arrow points from the text "no index here" to the 'Key' column for the 'meta_value' field, which is set to 'NULL'.

Indexes:

Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Comment
0	PRIMARY	1	meta_id	A	8204	NULL	NULL	
1	post_id	1	post_id	A	631	NULL	NULL	
1	meta_key	1	meta_key	A	282	191	NULL	

A red arrow points from the text "191?" to the 'Sub_part' column for the 'meta_key' index, highlighting the value 191.

Why 191? - InnoDB key prefix limit is 767 bytes. WP tables use utf8mb4 charset, that's 4 bytes per character. $767 / 4 = 191.75$, flooring that is 191. Only 191 characters fit into the key prefix.

WP DB - POSTS - EXPLAIN

```
2
3 EXPLAIN SELECT * FROM wp_posts
4 WHERE ID = 88;
5
```

Query Favorites ▾ Query History ▾

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	wp_posts	const	PRIMARY	PRIMARY	8	const	1	

WP DB - POSTS - EXPLAIN

```
9
10 EXPLAIN SELECT * FROM wp_posts
11 WHERE post_status = 'wc-processing';
12
```

ID	Select Type	Table	Type	Possible Keys	Key	Key Len	Ref	Rows	Extra
1	SIMPLE	wp_posts	ALL	NULL	NULL	NULL	NULL	298	Using where

WP DB - POSTS - EXPLAIN

```
16
17 EXPLAIN SELECT * FROM wp_posts
18 WHERE post_type = 'shop_order'
19 AND post_status = 'wc-processing';
20
```

The screenshot shows the MySQL Workbench interface with the 'Query' tab selected. The query editor contains the following code:

```
EXPLAIN SELECT * FROM wp_posts
WHERE post_type = 'shop_order'
AND post_status = 'wc-processing';
```

Below the editor is the results grid. The top row of the grid contains column headers: id, select_type, table, type, possible_keys, key, key_len, ref, rows, and Extra. The data for the single row is as follows:

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	wp_posts	ref	type_status_date	type_status_date	164	const,const	54	Using index condition

WP DB - POSTS - EXPLAIN

```
24 EXPLAIN SELECT * FROM wp_posts
25 WHERE post_type IN ('shop_order', 'shop-subscription')
26 AND post_status = 'wc-processing';
27
```

The screenshot shows a MySQL Workbench interface with the following details:

- Query Editor:** Contains the EXPLAIN query for selecting all columns from wp_posts where post_type is either 'shop_order' or 'shop-subscription' and post_status is 'wc-processing'. The query is numbered 24, 25, 26, and 27.
- Toolbar:** Includes buttons for Query Favorites, Query History, and Run Current.
- Result Set:** A table showing the execution plan. The table has the following columns: id, select_type, table, type, possible_keys, key, key_len, ref, rows, and Extra.
- Table Data:** One row is shown, corresponding to the query number 27:

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	wp_posts	range	type_status_date	type_status_date	164	NULL	55	Using index condition

WP DB - POSTS - EXPLAIN

```
31
32 EXPLAIN SELECT *| FROM wp_posts
33 WHERE post_status = 'wc-processing'
34 AND post_date < '2017-08-03 12:00:00';
35
```

The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Displays the SQL query:

```
EXPLAIN SELECT *| FROM wp_posts
WHERE post_status = 'wc-processing'
AND post_date < '2017-08-03 12:00:00';
```
- Result Set:** Shows the EXPLAIN output for the query. The table has the following columns: id, select_type, table, type, possible_keys, key, key_len, ref, rows, and Extra. The single row of data is:

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	wp_posts	ALL	NULL	NULL	NULL	NULL	298	Using where

WP DB - POSTS - EXPLAIN

```
40 EXPLAIN SELECT * FROM wp_posts
41 WHERE post_type = 'shop_order'
42 AND post_status = 'wc-processing'
43 AND post_date < '2017-08-03 12:00:00';
```

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	wp_posts	range	type_status_date	type_status_date	169	NULL	51	Using index condition

Note that the found rows is 51 here! That means that when using the index, MySQL returned 51 rows that matched the index.

WP DB - POSTS - QUERY

```
76  
77  
78 SELECT * FROM wp_posts  
79 WHERE post_type = 'shop_order'  
80 AND post_status = 'wc-processing'  
81 AND post_date < '2017-08-03 12:00:00';  
82  
83
```

The screenshot shows a database query interface with the following details:

- Query Editor:** Displays the SQL query with line numbers 76 through 83.
- Results Table:** Shows the results of the query. The table has columns: ID, post_author, post_date, post_date_gmt, and post_content. The data is as follows:

ID	post_author	post_date	post_date_gmt	post_content
9	1	2017-06-04 13:46:44	2017-06-04 13:46:44	
20	1	2017-06-16 14:40:18	2017-06-16 12:40:18	
23	1	2017-06-16 14:49:42	2017-06-16 12:49:42	
26	1	2017-06-16 14:50:33	2017-06-16 12:50:33	
29	1	2017-06-16 14:52:16	2017-06-16 12:52:16	
33	1	2017-06-24 22:34:00	2017-06-24 20:34:00	
35	1	2017-07-05 22:48:52	2017-07-05 20:48:52	

- Status Bar:** Shows "No errors; 51 rows affected, taking 1.3 ms".

Note that the ACTUAL returned rows is also 51 in this case.

WP DB - POSTS - EXPLAIN

```
48 EXPLAIN SELECT * FROM wp_posts
49 WHERE post_type = 'shop_order'
50 AND post_status = 'wc-processing'
51 AND post_date < '2017-08-03 12:00:00'
52 AND ID = 88;
```

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	wp_posts	const	PRIMARY,type_status_date	PRIMARY	8	const	1	

Using all parts of the compound index, including the primary query. Because using the primary query would yield less results over using the compound index, that's the one being used. Because it's an equivalence, there is only one row returned.

WP DB - POSTS - EXPLAIN

```
58 EXPLAIN SELECT * FROM wp_posts
59 WHERE post_type = 'shop_order'
60 AND post_status = 'wc-processing'
61 AND post_date < '2017-08-03 12:00:00'
62 AND ID < 88;
```

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	wp_posts	range	PRIMARY,type_status_date	PRIMARY	8	NULL	69	Using where

Here we're using ID as a range comparison. ID is also a primary key, so that's the one being used because it's faster. It shows that there would be 69 rows returned.

WP DB - POSTS - QUERY

```
66
67 SELECT * FROM wp_posts
68 WHERE post_type = 'shop_order'
69 AND post_status = 'wc-processing'
70 AND post_date < '2017-08-03 12:00:00'
71 AND ID < 88;
72
```

ID	post_author	post_date	post_date_gmt	post_content
9	1	2017-06-04 13:46:44	2017-06-04 13:46:44	
20	1	2017-06-16 14:40:18	2017-06-16 12:40:18	
23	1	2017-06-16 14:49:42	2017-06-16 12:49:42	
26	1	2017-06-16 14:50:33	2017-06-16 12:50:33	
29	1	2017-06-16 14:52:16	2017-06-16 12:52:16	
33	1	2017-06-24 22:34:00	2017-06-24 20:34:00	
35	1	2017-07-05 22:48:52	2017-07-05 20:48:52	

No errors; 14 rows affected, taking 0.7 ms

However when we run the ACTUAL query, there are only 14 rows given. That's because first MySQL evaluates the ID < 88 query, and only then evaluates the rest of the where clauses on the results returned by the ID < 88 query.

USING WP_QUERY

By the way the previous is why WP Query forces you to declare a `post_type`, otherwise defaults to `post`.

```
// in class-wp-query.php line 2281 in version 4.8.1
// method get_posts()
} else {
    $where .= " AND {$wpdb->posts}.post_type = 'post'";
    $post_type_object = get_post_type_object( 'post' );
}
```

If there was no `post_type` declaration, lots of queries would skip using the index.

ORDER BY

Think carefully what to use for order by. The full list is on [MySQL docs on ORDER BY optimization](#).

Tips:

- The larger the data to order, the slower it'll be.
- Indexes can be used to order by if "if the ORDER BY does not match the index exactly, as long as all unused portions of the index and all extra ORDER BY columns are constants in the WHERE clause."

[MySQL doc / ORDER BY optimization](#)

ORDER BY

These will use index:

```
-- same compound key, consecutive, no WHERE clause
SELECT * FROM t1
    ORDER BY key_part1, key_part2;

-- key_part1 in WHERE with constant (not column), key_part2 in order by
SELECT * FROM t1
    WHERE key_part1 = constant
        ORDER BY key_part2;

-- key_part1 is compared to constant, same key used in order by
SELECT * FROM t1
    WHERE key_part1 > constant
        ORDER BY key_part1 ASC;

-- key parts compared to constants, order by uses same key part
SELECT * FROM t1
    WHERE key_part1 = constant1 AND key_part2 > constant2
        ORDER BY key_part2;
```

MySQL doc / ORDER BY optimization

ORDER BY

These will not use index though:

```
-- Query uses two different indexes
SELECT * FROM t1 ORDER BY key1, key2;

-- non-consecutive compound index parts in order by
SELECT * FROM t1 WHERE key2=constant ORDER BY key_part1, key_part3;

-- mixes ASC and DESC
SELECT * FROM t1 ORDER BY key_part1 DESC, key_part2 ASC;

-- key in WHERE and key in ORDER BY are different
SELECT * FROM t1 WHERE key2=constant ORDER BY key1;

-- order by is an expression other than the index (needs calculation)
SELECT * FROM t1 ORDER BY ABS(key);
SELECT * FROM t1 ORDER BY -key;
```

[MySQL doc / ORDER BY optimization](#)

GROUP BY

Either creates a temporary table, or uses indexes in VERY SPECIFIC cases.

Douglas Adam's words apply to Group By too:

The Encyclopedia Galactica, in its chapter on Love states that it is far too complicated to define. The Hitchhiker's Guide to the Galaxy has this to say on the subject of love: Avoid, if at all possible. Unfortunately, Arthur Dent has never read the Hitchhiker's Guide to the Galaxy.

[MySQL doc / GROUP BY optimization](#)

JOINS

most efficient if indexes on columns are declared same type and size.
Which is why all ID fields are BIGINT(20) in WP.

VARCHAR and CHAR are considered the same if their sizes match.
VARCHAR(10) == CHAR(10).

Use same charset (this is usually taken care of for you)

Do not use columns that need type juggling though. eg numeric 1 and
string '1', '00001', ' 1', '01.e1', etc

[MySQL doc on indexes](#)

SAME TABLE / NEW REL TABLE?

Hypothetical WooCommerce Orders new table structure

- Is data required, and there's only one per order? => Same table, new column! eg order total
- Is data optional, but there can only be one per order? => Same table, allow NULL
- Is data required, but there can be multiple per order? => new rigid table, for example list of coupons, list of line items
- Is data optional, is it arbitrary? => new general table, key-value pair, much like postmeta

DELETING RECORDS

delete things from wp_posts

then need to clean up in wp_postmeta

... and terms

... and comments

... and taxonomy relationships...

The solution is (would be)

Foreign keys!!

FOREIGN KEYS

foreign keys are constraints imposed on the database that keep data across multiple tables consistent

```
CREATE TABLE parent (
    id INT NOT NULL,
    PRIMARY KEY (id)
) ENGINE=INNODB;

CREATE TABLE child (
    id INT,
    parent_id INT,
    INDEX par_ind (parent_id),
    FOREIGN KEY (parent_id)
        REFERENCES parent(id)
        ON DELETE CASCADE
) ENGINE=INNODB;
```

FOREIGN KEYS

You can't insert data that doesn't belong to something that already exists
in the main table.

Can't drop table that has other tables depending on it.

FOREIGN KEYS

ON DELETE CASCADE

if I delete the main record in parent, everything else that had its foreign key constrained to THAT id will automatically be deleted with it.

Automatic cleanup! No more hunting for orphan data!

Of course it's not implemented in WordPress, mostly due to backwards compat reasons. ([Trac 19207](#))

TOOLS

- NewRelic APM configured to give you slow queries
- MySQL Workbench / Seqel Pro / mysql cli for the EXPLAIN queries.
Please don't use phpmyadmin for this even though it COULD.
- Blackfire.io

TOOLS - NEW RELIC

The screenshot shows the New Relic APM interface. The top navigation bar includes links for APPS, BROWSER, SYNTHETICS, MOBILE, PLUGINS, INSIGHTS, INFRASTRUCTURE, SERVERS, Maps, Alerts New, Tools, Help, and a user profile icon.

The main dashboard displays monitoring metrics for an application named "APPS". It includes sections for MONITORING (Overview, Service maps), Transactions (WordPress, Hooks, Plugins and themes, Databases, External services), and EVENTS (Error analytics, Errors, Alerts, Deployments). A "TIME PICKER" indicates "Last 30 minutes ending now".

A central modal window titled "Transaction trace" shows a trace for a transaction on Aug 10, '17 at 11:44 am. The trace details are: 14,500 ms total time, 4,580 ms (31.5%) RESP. TIME, 1,160 ms (7.95%) USER CPU BURN, and 1,160 ms (7.95%) SYSTEM CPU BURN. A note states: "This is a partial trace. For performance reasons, the agent limited the size of the trace that it reported." The "Database queries" tab is selected, displaying two queries:

Total duration	Call count	Database instance	Query
43 ms	3	pod-13016 5:3306	SELECT t.*, tt.* FROM wp_terms AS t INNER JOIN wp_term_taxonomy AS tt ON t.term_id = tt.term_id INNER JOIN wp_term_relationships AS tr ON tr.term_taxonomy_id = tt.term_taxonomy_id WHERE tt.taxonomy IN (?) AND tr.object_id IN (?) ORDER BY t.name ASC ?
16 ms	1	pod-13016 5:3306	SELECT t.*, tt.*, tm.meta_value FROM wp_terms AS t INNER JOIN wp_term_taxonomy AS tt ON t.term_id = tt.term_id INNER JOIN wp_term_relationships AS tr ON tr.term_taxonomy_id = tt.term_taxonomy_id LEFT JOIN wp_termmeta AS tm ON (t.term_id = tm.term_id AND tm.meta_key = ?) WHERE tt.taxonomy IN (?) AND tr.object_id IN (?) GROUP BY t.term_id ORDER BY tm.meta_value+? ASC, t.name ASC ?

New Relic

TOOLS - BLACKFIRE

The screenshot shows the Blackfire.io dashboard for a WordPress application. The main interface includes a sidebar with navigation links like 'Functions', 'Recommendations', 'Assertions', and 'Support'. The central area displays a table of SQL queries and a detailed call graph.

SQL Queries: 59.4 ms / 89 rq

Calls	↑ I/O Wait	SQL
~7x	20.8 ms	update wp_options set option_value = ? where option_name = ?
1x	6.55 ms	other
37x	6.17 ms	select ? from wp_options where option_name = ? limit ?
~1x	3.92 ms	select ...sum(?) as order_item_count from wp_posts as posts join wp_postmeta as meta_refund_amount o...
1x	3.23 ms	select sum(?) as qty, ? from wp_posts as posts join wp_woocommerce_order_items as order_items on post...
~1x	2.75 ms	select ... from wp_posts as posts join wp_postmeta as parent_meta_order_total on (posts.post_parent = pa...
1x	2.3 ms	select t.*, tt.* , ? from wp_terms as t join wp_term_taxonomy as tt on t.term_id = tt.term_id join wp_term_rel...
~1x	2.22 ms	select ?sum(?) as discount_amount, ? from wp_posts as posts join wp_woocommerce_order_items as order...
~1x	1.56 ms	select sum(?) as total_sales, sum(?) as total_shipping, sum(?) as total_tax, sum(?) as total_shipping_tax, ? fr...
6x	1.15 ms	select * from wp_posts where id = ? limit ?
~1x	868 µs	select sum(?) as order_item_count, ? from wp_posts as posts join wp_woocommerce_order_items as order...
3x	837 µs	select ?count(*) as num_posts from wp_posts where post_type = ? group by post_status
1x	793 µs	select ... from wp_options whereautoload = ?
2x	748 µs	select ? from wp_comments where ((comment_approved = ? or comment_approved = ?)) and comment_typ...
1x	571 µs	select count(distinct ?) as count from wp_posts as wcorder join wp_postmeta as meta_subscription_renew...
~6x	464 µs	SET SESSION SQL_BIG_SELECTS=?
1x	449 µs	SELECT posts.ID, posts.post_title, comments.comment_author, comments.comment_ID, comments.comm...
1x	414 µs	select count(distinct ?) as count from wp_posts as wcsubs join wp_posts as wcorder on wcsubs.post_paren...
1x	371 µs	select count(*) from wp_posts where post_type = ? and post_status = ?
1x	368 µs	select sum(?) as order_item_count from wp_posts as posts join wp_woocommerce_order_items as order_it...
~1x	315 µs	select count(distinct ?) as count, ? from wp_posts as posts where posts.post_type in (?) and posts.post_st...

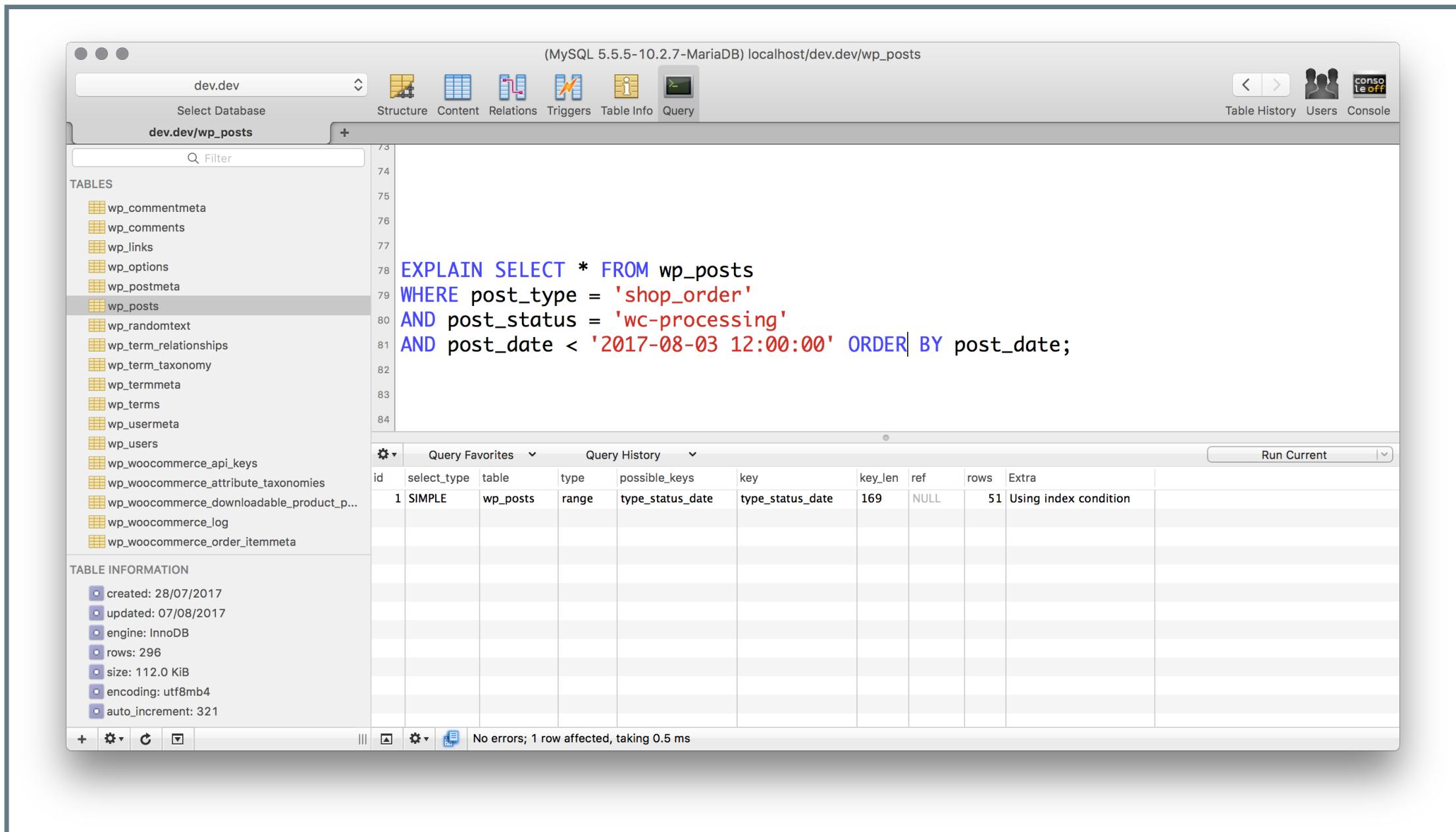
Call Graph: 871 x

```
graph TD; update_option[update_option 5.48%] --> WP_Hook[WP_Hook: do_action@1 4.87%]; WP_Hook --> get_option[get_option 7.36%]
```

The call graph highlights three main components: 'update_option' (5.48%), 'WP_Hook: do_action@1' (4.87%), and 'get_option' (7.36%). Arrows indicate the flow of calls between them, with numerical values representing the number of times each call occurs.

Blackfire

TOOLS - SEQUEL PRO



Sequel Pro aka "Pancakes"

TIPS

- You can't do much to WordPress database schema without breaking something
- rethink your theme / plugin / customisation code to make sure your queries use indexes
- rethink your theme / plugin's data storage schema to make sure you're not wasting resources
- learn more about MySQL. Their documentation is A++ 5-7 would recommend
- use profiling tools to find where the bottlenecks are
- you can always hire someone who knows how to database

終わり

(OWARI – THE END)

QUESTIONS?

@JAVORSZKY