

WooCommerce Database Cleanup

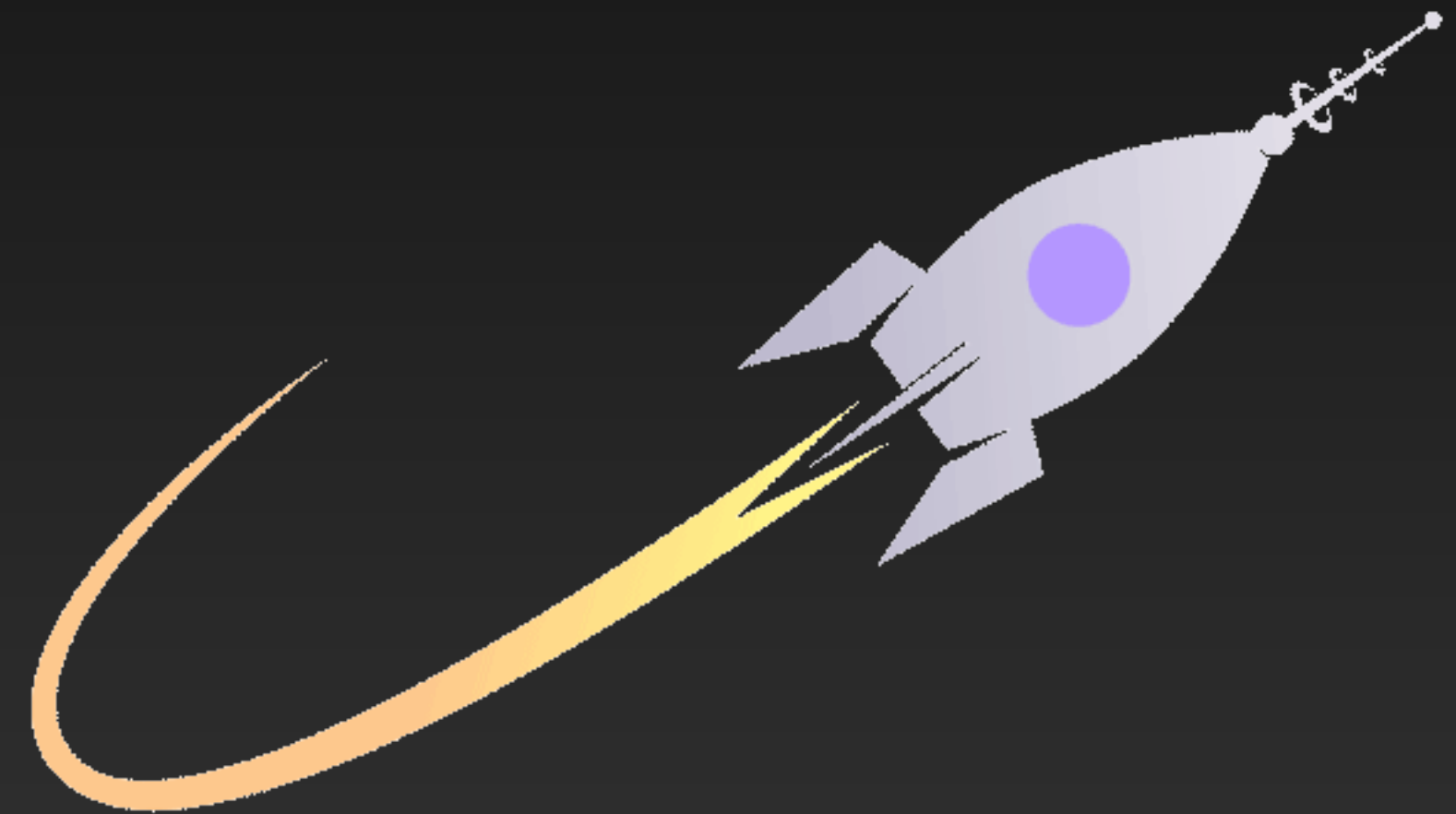
Cleaning legacy theme and plugin cruft
to free up server resources for growth

Presented by Sean Conklin
Coded Commerce, LLC

Boosting performance

Getting the most out of your WordPress powered websites

- Back-end
 - Plugin reduction
 - Upgrading hosting
 - Database cleanup
- Front-end
 - CDN w/compression & caching
 - Image & media optimization
 - See Google PageSpeed Insights



Modern websites are databases

Content Management Systems contain:

- **Content**
 - Users, pages, posts, metadata, taxonomies, menus, widgets
 - eCommerce: products, orders, customers, metadata, indexes
 - Custom post types, product types and fields
- **Design**
 - Theme, child theme, customizer settings
- **Functionality**
 - Core features
 - Plugins and all of the settings



About the database

Databases are the most valuable parts of your sites

- If only given the database one can usually restore theme, plugins, images
- High volatility – Backups are critical, recommend daily if not hourly
- Hosting tiers are customarily quoted in traffic, but data tells more
- Data uses precious RAM on the server, especially global settings, caches
- Database queries can be numerous, large, duplicative, or slow to run
- Previously installed themes and plugins leave stuff behind

Cleanup goals

Measure, backup, implement, re-measure:



- Speed-up your website
 - TTFB time-to-first-byte, response
 - Meaningful paint time
 - Fully loaded time
- Prevent backup file bloat, overage fees
- Save hosting costs, get the most out of it
- Make room for future growth
 - Websites serve current visitor needs, typically not a historic archive

Cleanup steps

1. Check built-in WordPress & WooCommerce tools
2. Prune outdated stuff within WordPress Admin
3. Run a database cleanup plugin
4. Key queries to check in PhpMyAdmin
5. Understand the Query Monitor plugin

Cleanup step 1 of 5

Built-in WordPress & WooCommerce tools

- Review *WP Admin > Tools > Site Health*
- Review *WP Admin > WooCommerce > Status*
- **WP Admin > WooCommerce > Status > Tools**
 - *WooCommerce & expired transients*
 - *Orphaned product variations*
 - *Verify & update database tables*



Cleanup step 2 of 5

Prune outdated stuff within WordPress Admin

- Users, especially old staff accounts
- Media library items
- Post types and ACF custom field groups
- Posts (pages, posts, products, etc.)
- Categories (blog, products)
- Tags (blog, products)
- WooCommerce attributes global & product levels
- WooCommerce order history
 - completed, cancelled, refunded, failed, pending



Cleanup step 3 of 5

Run your database cleanup plugin of choice



WP Optimize



- Post revisions
- Orphaned post meta
- Comments: Unapproved and SPAM
- Custom tables from legacy plugins
- Clear transients manually
 - After major operations
- Defragmenting table overhead

Cleanup step 4 of 5

Key queries to check in PhpMyAdmin ([SQL here](#))

- *wp_options*
 - Query top entries by size
 - Especially where *autoload=yes*
- *wp_postmeta*
 - Review all meta_keys and counts
- *wp_usermeta*
 - Review all meta keys & counts

```
1 SELECT
2     'Autoload KB' as name, ROUND( SUM( LENGTH( option_value ) ) / 1024 ) as size, null as autoloa
3     FROM wp_options
4     WHERE autoload = 'yes'
5 UNION
6     SELECT 'Autoload count', count( * ), null as autoload
7     FROM wp_options
8     WHERE autoload = 'yes'
9 UNION (
10    SELECT option_name, length( option_value ), autoload
11    FROM wp_options
12    WHERE autoload = 'yes'
13    ORDER BY length( option_value ) DESC
14    LIMIT 20
15 )
16 UNION (
17    SELECT option_name, length( option_value ), autoload
18    FROM wp_options
19    WHERE autoload = 'no'
20    ORDER BY length( option_value ) DESC
21    LIMIT 20
22 );
```

report-wp-options.sql hosted with ❤ by GitHub [view raw](#)

Cleanup step 5 of 5

Run the Query Monitor plugin

- Admin bar menu
- Check all pages
 - Duplicate queries
 - Error queries
 - Slow queries
 - Filter by plugin or function

Query Monitor					
Page generation time	Peak memory usage	Database query time	Database queries	Object cache	
0.0321 0.0% of 0s limit	2,048 kB 0.0% of 16,777,216 kB limit	0.0022	SELECT: 3	99.9% hit rate External object cache: true	
\$wpdb Queries					
	Query	Caller	Component	Rows	Time
1	SELECT user_id, meta_key, meta_value FROM wp_usermeta WHERE user_id IN (1) ORDER BY umeta_id ASC	update_meta_cache() wp-includes/meta.php:826	Plugin: coming-soon-builder	39	0.0004
2	SELECT wp_posts.* FROM wp_posts WHERE 1=1 AND (wp_posts.ID = '2') AND wp_posts.post_type = 'page' ORDER BY wp_posts.post_date DESC	WP_Query->get_posts() wp-includes/query.php:3655	Core	1	0.0017
3	SELECT * FROM wp_posts WHERE (post_type = 'page' AND post_status = 'publish') AND post_parent = 2 ORDER BY wp_posts.post_title ASC LIMIT 0,1	get_pages() wp-includes/post.php:4598	Core	0	0.0001
Total Queries: 3					0.0022

Site	Generation	Memory (MB)	DB time	DB queries	Dupes	Cache	Notes
	0.80	80	0.08	128	1	93.50%	Genesis
	0.95	76	0.05	103	2	97.60%	Storefront
	0.81	63	0.06	115	2	93.60%	Elementor
	0.94	59	0.17	154	2	94.20%	Custom
	1.24	89	0.17	116	13	87.60%	Custom
	0.55	54	0.03	66	2	96.90%	Storefront
	0.46	47	0.04	73	0	95.90%	Storefront
	0.73	68	0.03	69	1	98.50%	Beaver Builder
	0.91	67	0.06	121	4	94.10%	Elementor
	0.81	74	0.11	168	2	92.90%	Elementor
	0.56	57	0.06	73	2	95.00%	Pagelines
	1.87	113	0.10	166	4	98.00%	Divi
	0.64	59	0.05	88	2	97.50%	Storefront
	1.51	75	0.29	270	2	96.80%	WPML
	0.22	26	0.02	30	0	94.90%	Storefront
	1.02	81	0.07	141	10	95.50%	Custom
	0.62	57	0.03	69	2	95.70%	Elementor
	1.33	103	0.05	100	3	98.90%	Divi
	1.45	90	0.09	198	5	96.10%	WPBakery
	1.70	83	0.08	93	0	97.20%	Custom
	0.60	60	0.05	83	1	97.60%	Custom
	1.33	87	0.06	117	3	97.60%	Elementor
	0.33	39	0.03	64	1	94.10%	Storefront
	1.29	231	0.50	394	3	88.50%	Custom
	0.54	56	0.05	71	1	97.70%	Storefront
	0.60	42	0.08	156	3	95.50%	Genesis
	0.85	72	0.08	168	2	92.70%	Genesis
	0.56	53	0.04	55	1	95.30%	Storefront
Thresholds	1.00	85	0.10	150	4	93.00%	

Bonus step 1 of 2

Tuning your database for performance

- Switch from *MyISAM* to the *InnoDB* storage engine on all tables
- Use *utf8mb4* charset in *wp-config.php* file and all columns (since WordPress v4.2 in 2015)
- Use *utf8mb4_unicode_520_ci* collation in *wp-config.php* file and all columns
 - Based on the official Unicode rules v5.2 for universal sorting & comparison, which sorts accurately in a wide range of languages
- *my.cnf* database config file ([instructions](#))
 - *innodb_buffer_pool_size=3072M*
 - *Keep 80% of your working set in memory*
 - *innodb_log_file_size=768M*
 - *max_allowed_packet=16M*

Bonus step 2 of 2

Server & Application Performance Monitoring (APM)

- See realtime data directly from the server with an APM tool like New Relic
 - sitedistrict.com offers free migration and speed testing
 - patheon.io offers free access to New Relic via their dev sites
- If you have access to your server configuration files:
 - Disable unneeded [Apache modules](#) and [PHP libraries](#)
 - Ensure you're running the latest PHP v7.4.x (2020)
 - *php.ini*
 - Set *memory_limit* to 128M, 256M or 512M
 - Set *max_execution_time* to 60s

Q & A

Presented by Sean Conklin,
Coded Commerce, LLC

Slides:

<https://codedcommerce.com/2020dbcleanup>