

Magento 2 Database Insight

Get your hands dirty with database

Version: Magento 2.2.2

Prepared By: Gideon Babu

Presented By: Gideon & Mohan

www.linkedin.com/in/gideon-babu/

www.linkedin.com/in/mohan-sg/

Magento & Database

- Magento built on Relational Database Management System (RDBMS)
- Compatible with MySql, MariaDB 10.0+ and Percona 5.7
- Requires MySQL 5.6 and better for M2.2 installation
- Utilizes an Entity-Attribute-Value (EAV) data model
- Database information are stored in app/etc/env.php file
- M2.2.2 has 338 tables

Entity-Attribute-Model (EAV)

- Magento use EAV database model predominantly to store data.
- Entity type table name is **eav_entity_type**
- **Eav_attribute** holds all the attributes of customer, address, category, products, order, invoice and shipment
- Entity attributes are grouped in **eav_attribute_group** table
- Products have attribute set to store sets of attributes. Likewise every attribute is stores in atleast single attribute set and that is stored in **eav_attribute_set** table.
- Product swatch options are stored in **eav_attribute_option_swatch** table

Catalog Category

- Tables related to catalog category prefixed with **catalog_category**
- All categories related information are stores in tables prefixed **catalog_category**
- Category information are stores on specific data type tables. For instance category name is of type varchar and so stores in **catalog_category_entity_varchar**.
- Category int values are stored in **catalog_category_entity_int**
- Products mapped to a category is stored in **catalog_category_product**

Product Tables

- `catalog_product_entity` => main product table
- `catalog_product_entity_varchar` => Title & Description & URL
- `catalog_product_website` => Assign Product to Website
- `catalog_product_entity_decimal` => Product Price
- `catalog_product_index_price` => Assign All Group Wise Price
- `catalog_category_product` => Assign Product to Category
- `catalog_product_entity_int` => Status & Visibility
- `cataloginventory_stock_item` => For Inventory Quantity
- `url_rewrite` => For URL Rewrite
- `eav_attribute` => store ALL Attributes
- `catalog_url_rewrite_product_category` => For URL Rewrite

Catalog Product

- 68 tables related to catalog products
- Child tables are with prefix **catalog_product_entity_**
- catalog_product_entity_datetime => Product – date attribute are stored here.
Eg: New From, New To
- catalog_product_entity_media_gallery => product images path, Thumb , small, etc.,
- catalog_product_entity_media_gallery_value – Product image, product image position, title info & position of image info, etc.,
- catalog_product_entity_tier_price – Product tier price
- catalog_product_entity_varchar – Product name, URL, image path, etc.

CATEGORY ENTITY

catalog_category_entity

entity_id	int(10)	identity
entity_type_id	smallint(5)	not null
attribute_set_id	smallint(5)	not null
parent_id	int(10)	not null
created_at	datetime	not null
updated_at	datetime	not null
path	varchar(255)	not null
position	int(11)	not null
level	int(11)	not null
children_count	int(11)	not null

catalog_category_entity_datetime

value_id	int(11)	identity
entity_type_id	smallint(5)	not null
attribute_id	smallint(5)	not null
store_id	smallint(5)	not null
entity_id	int(10)	not null
value	datetime	not null

catalog_category_entity_decimal

value_id	int(11)	identity
entity_type_id	smallint(5)	not null
attribute_id	smallint(5)	not null
store_id	smallint(5)	not null
entity_id	int(10)	not null
value	decimal(12,4)	not null

catalog_category_entity_int

value_id	int(11)	identity
entity_type_id	smallint(5)	not null
attribute_id	smallint(5)	not null
store_id	smallint(5)	not null
entity_id	int(10)	not null
value	int(11)	not null

catalog_category_entity_text

value_id	int(11)	identity
entity_type_id	smallint(5)	not null
attribute_id	smallint(5)	not null
store_id	smallint(5)	not null
entity_id	int(10)	not null
value	text	not null

catalog_category_entity_varchar

value_id	int(11)	identity
entity_type_id	smallint(5)	not null
attribute_id	smallint(5)	not null
store_id	smallint(5)	not null
entity_id	int(10)	not null
value	varchar(255)	not null

PRODUCT ENTITY

catalog_product_entity

entity_id	int(10)	identity
entity_type_id	smallint(5)	not null
attribute_set_id	smallint(5)	not null
type_id	varchar(32)	not null
sku	varchar(64)	not null
category_ids	text	not null
created_at	datetime	not null
updated_at	datetime	not null
has_options	smallint(1)	not null
required_options	tinyint(1)	not null

catalog_product_entity_datetime

value_id	int(11)	identity
entity_type_id	smallint(5)	not null
attribute_id	smallint(5)	not null
store_id	smallint(5)	not null
entity_id	int(10)	not null
value	datetime	not null

catalog_product_entity_decimal

value_id	int(11)	identity
entity_type_id	smallint(5)	not null
attribute_id	smallint(5)	not null
store_id	smallint(5)	not null
entity_id	int(10)	not null
value	decimal(12,4)	not null

catalog_product_entity_int

value_id	int(11)	identity
entity_type_id	smallint(5)	not null
attribute_id	smallint(5)	not null
store_id	smallint(5)	not null
entity_id	int(10)	not null
value	int(11)	not null

catalog_product_entity_text

value_id	int(11)	identity
entity_type_id	mediumint(8)	not null
attribute_id	smallint(5)	not null
store_id	smallint(5)	not null
entity_id	int(10)	not null
value	text	not null

catalog_product_entity_varchar

value_id	int(11)	identity
entity_type_id	mediumint(8)	not null
attribute_id	smallint(5)	not null
store_id	smallint(5)	not null
entity_id	int(10)	not null
value	varchar(255)	not null

Configurable and Grouped

Configurable and grouped product information get it from below tables

- catalog_product_super_attribute
- catalog_product_super_attribute_label
- catalog_product_super_link

Bundled Product

Bundled product associated values from below table

- catalog_product_bundle_option
- Catalog_product_bundle_option_value
- catalog_product_bundle_price_index
- Catalog_product_bundle_selection
- Catalog_product_bundle_selection_price
- catalog_product_bundle_stock_index

Downloadable Product

Tables related to downloaded product information

- downloadable_link
- downloadable_link_price
- downloadable_link_purchased
- downloadable_link_purchased_item
- downloadable_link_title
- downloadable_sample
- downloadable_sample_title

Related Products

Related products information is stored in these tables:

- `catalog_product_link`
- `catalog_product_link_attribute`
- `catalog_product_link_attribute_decimal`
- `catalog_product_link_attribute_int`
- `catalog_product_link_attribute_varchar`
- `catalog_product_link_type`

Product Custom Option

Product custom option information will get from below tables

- catalog_product_option
- catalog_product_option_price
- catalog_product_option_title
- catalog_product_option_type_price
- catalog_product_option_type_title
- catalog_product_option_type_value

Inventory, website and Alerts

Product inventory information are stored in these tables:

- cataloginventory_stock
- cataloginventory_stock_item
- cataloginventory_stock_status – Currently Product is instock or out off stock

Catalog_product_website stores data of which website the product belongs to.

Product alerts are stored in product_alert_price and product_alert_stock tables

Product Ratings

Product ratings are stored in the tables below:

- rating
- rating_option
- rating_title
- rating_entity
- rating_option_vote_aggregated
- rating_option_vote
- rating_store

Product Reviews

Product reviews are stored in the tables below:

- review
- review_detail
- review_entity
- review_status
- review_store

Shopping Cart / Cart

Products in shopping carts are stored in these tables before they are converted into orders.

- Quote => each customer shopping cart is saved in a row in the quote main table
- Quote_item => each product holds each row in quote_item table
- Quote_address => billing and shipping address of the quote
- Quote_address_item => ship to multiple address as each item mapped to separate address
- Quote_item_option => Options of the product in the cart
- Quote_payment => payment information like payment method, cost, etc.,
- Quote_shipping_rate => shipping cost, method, title, code



Sales Order

These are the order tables containing all the placed order information:

- Sales_order => Main table, store ID, customer ID, status, state, price, qty, etc.,
- Sales_order_item => order products details, SKU, price, custom attributes
- Sales_order_address => billing and shipping address
- Sales_order_payment => payment method, amount and other payment info
- Sales_order_status_history => stores each process of the order in each row like order, invoice, shipping, credit memo and cancel
- Sales_order_status_label => order status and label for each store ID

Sales Order (Cntd)

- Sales_order_status => Basic order status names and labels
- Sales_order_status_state => mapped status and state
- sales_order_aggregated_created
- sales_order_aggregated_updated
- Sales_order_grid => stores order imp info to display in the grid
- Sales_order_tax => store order tax percent, amount, tax code, title, etc.,

Order Status and State

States are used for internal purpose to maintain order flow and various order actions like

States are new, pending, processing, pending payment, payment review, on hold, cancelled, closed, complete

States are not displayed anywhere and only for internal purpose

Status used for display purpose. Each status belongs to a order state. Multiple status can also be assigned to a single state.

For instance **processing** and **suspected fraud** are two status can be assigned to state **processing**



Order Status

Assign Status to State

Create New Status

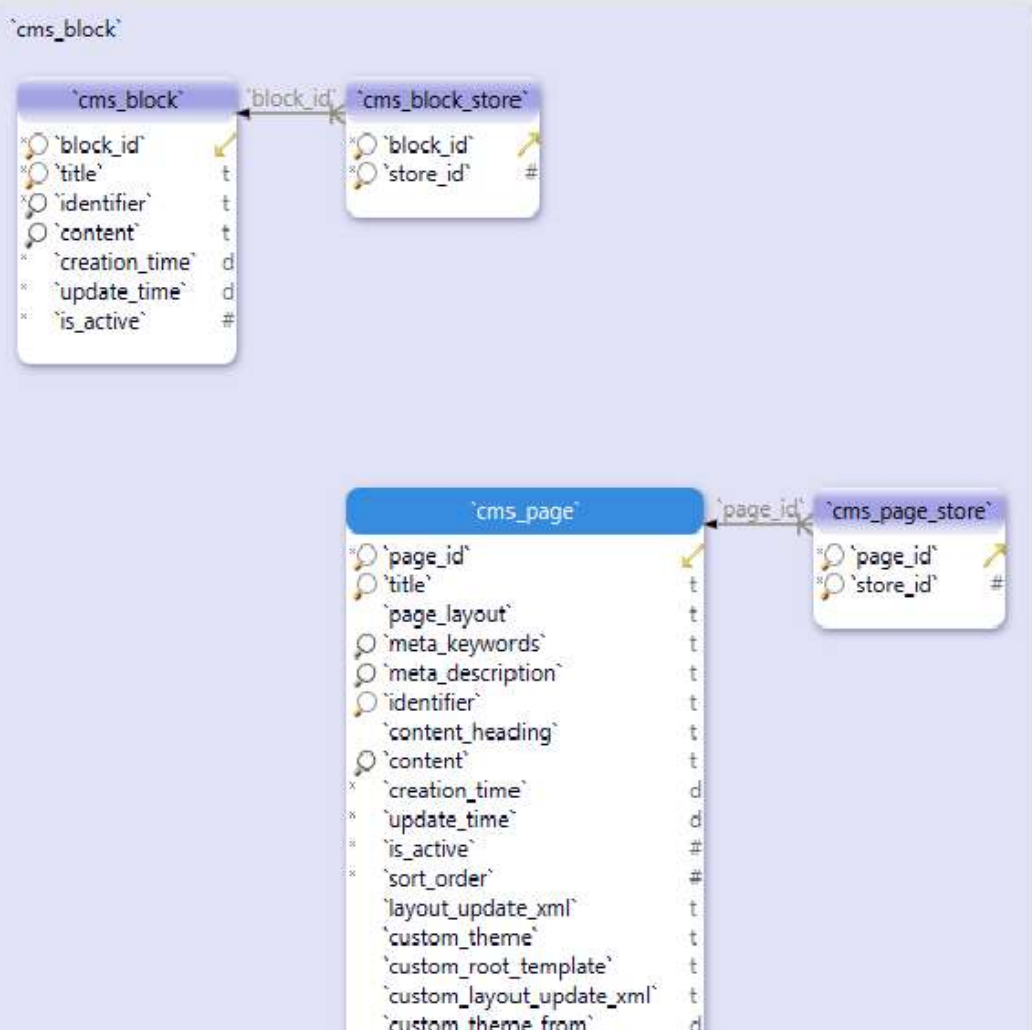
Status	Status Code	Default Status	Visible On Storefront	State Code and Title	Action
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
Processing	processing	Yes	Yes	processing[Processing]	Unassign
Suspected Fraud	fraud	No	Yes	processing[Suspected Fraud]	Unassign
Pending Payment	pending_payment	Yes	No	pending_payment[Pending Payment]	Unassign
Payment Review	payment_review	Yes	Yes	payment_review[Payment Review]	Unassign
Suspected Fraud	fraud	No	Yes	payment_review[Suspected Fraud]	Unassign
Pending	pending	Yes	Yes	new[Pending]	Unassign
On Hold	holded	Yes	Yes	holded[On Hold]	Unassign
Complete	complete	Yes	Yes	complete[Complete]	Unassign
Closed	closed	Yes	Yes	closed[Closed]	Unassign
Canceled	canceled	Yes	Yes	canceled[Canceled]	Unassign
PayPal Reversed	paypal_reversed				
PayPal Canceled Reversal	paypal_canceled_reversal				
Pending PayPal	pending_paypal				

Activate Windows
Go to Settings to activate Windows.

Pages and Static Blocks

Content pages and blocks are stored in this tables:

- Cms_block
- Cms_block_store
- Cms_page
- cms_page_store



Admin

Admin user information are stored in these tables:

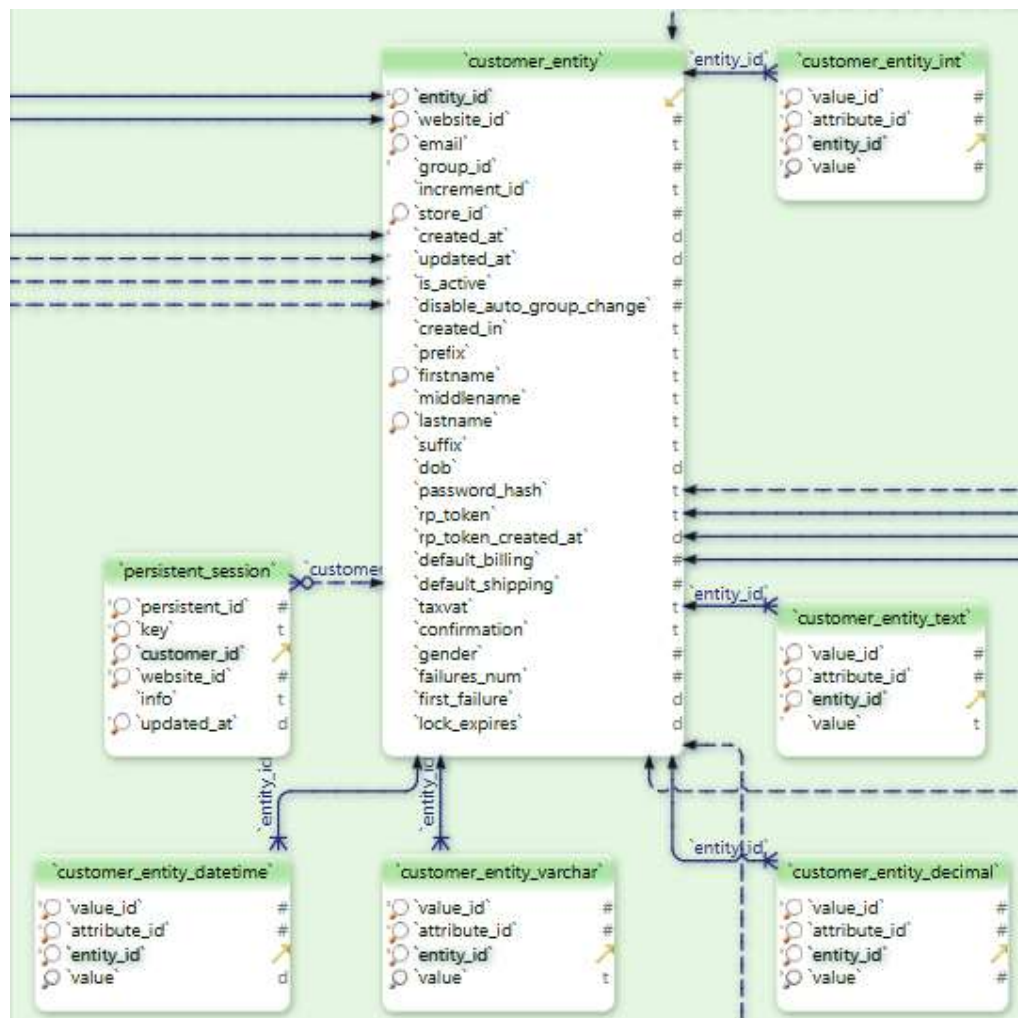
- Admin_user
- Admin_user_session
- Admin_passwords
- Admin_system_messages
- adminnotification_inbox tables
- When new admin user is created from admin or while installation, the admin user details are stored in this tables.
- Authorization_rule and authorization_role are two tables used to store access level for admin users

Customer

Tables related to customer/visitor prefixed with “**customer_**”

All the information about the customer is stored in these tables:

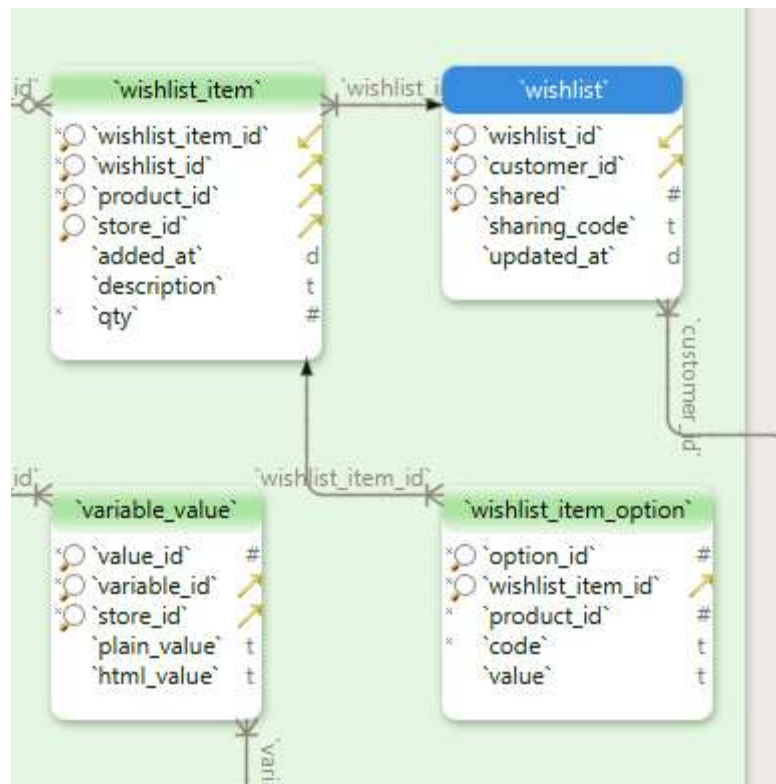
- Customer_entity => main customer table
 - customer_entity_datetime
 - customer_entity_decimal
 - customer_entity_int
 - customer_entity_text
 - customer_entity_varchar
 - Customer_visitor => to store website visitor data like session ID, last visit time and customer ID if the visitor is customer
- Customer entity attribute tables



Customer Address

- `customer_address_entity`
- `Customer_address_entity_datetime`, `customer_address_entity_decimal`, `customer_address_entity_int`, `customer_address_entity_text`, `customer_address_entity_varchar` => Customer address entity attribute tables
- `Customer_eav_attribute` => Has all the eav attribute data for customer
- `customer_eav_attribute_website`
- `Customer_form_attribute` => define in which form which attributes are required.
Eg. which fields to values in checkout shipping method and admin customer form
- `Customer_grid_flat` => used to display customer information in the admin grid view
- `customer_group`
- `Customer_log` => customer login and logout timestamp stored here

Wishlist



Store Configuration

Core_config_data is a main table has the store configuration values.

Independent table with some values are loaded from the XML files.

On update add entry in the table.

Cache and Index

Indexing details are stored in **indexer_state** table.

Useful to know when indexing stuck in the middle.

Just update the status to valid and run the reindex command

```
php bin/magento indexer:reindex
```

Cache configurations are loaded from app/etc/env.php file

Indexer Optimization

replica -> prevent dead and wait locks caused by read/write collisions

Concurrent full reindex and navigating to category, search pages and user layered navigation filters with price filters.

- catalog_category_product_index_replica
- catalog_product_index_eav_decimal_replica
- catalog_product_index_eav_replica
- catalog_product_index_price_replica
- cataloginventory_stock_status_replica
- catalogrule_group_website_replica
- catalogrule_product_price_replica
- catalogrule_product_replica

_idx and _tmp suffixed tables

Used for indexation to prepare new data and are truncated afterwards.

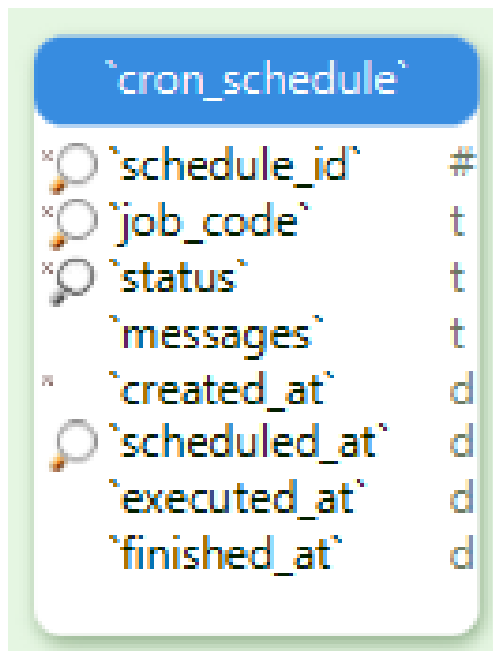
Modules

- Installed modules names and its schema and data versions are stored in **setup_module** table.
- Module enable and disable information is stored in app/etc/config.php file.
- All the modules are declared in this config file.

Crons

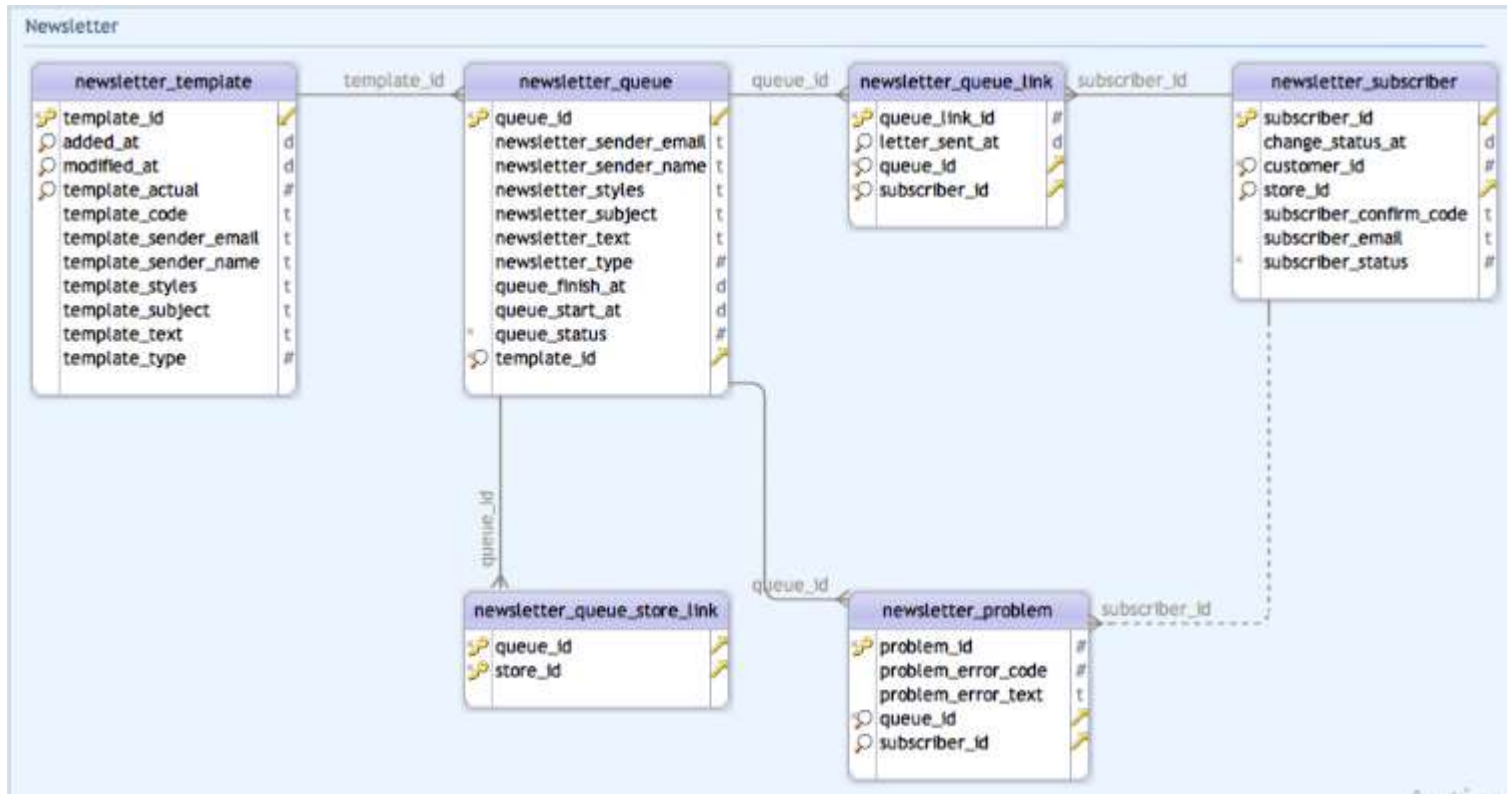
Cron_schedule table lists all the cron, which ran before with executed and response messages.

Cron table also has information about the cron scheduled for future.

A screenshot of a database table structure for 'cron_schedule'. The table name is in a blue header. Below it, the columns are listed with their data types and some have search icons. The columns are: schedule_id (integer, primary key), job_code (text), status (text), messages (text), created_at (datetime), scheduled_at (datetime), executed_at (datetime), and finished_at (datetime).

`cron_schedule`		
* 🔍	`schedule_id`	#
* 🔍	`job_code`	t
* 🔍	`status`	t
	`messages`	t
*	`created_at`	d
🔍	`scheduled_at`	d
	`executed_at`	d
	`finished_at`	d

Newsletter



url_rewrite

	url_rewrite_id	entity_type	entity_id	request_path	target_path	redirect_type	store_id	description	is_autogenerated
<input type="checkbox"/>	1	cms-page	1	no-route	cms/page/view/page_id/1	0	1	(NULL)	1
<input type="checkbox"/>	2	cms-page	2	home	cms/page/view/page_id/2	0	1	(NULL)	1
<input type="checkbox"/>	3	cms-page	3	enable-cookies	cms/page/view/page_id/3	0	1	(NULL)	1
<input type="checkbox"/>	4	cms-page	4	privacy-policy-cookie-restriction-modal	cms/page/view/page_id/4	0	1	(NULL)	1
<input type="checkbox"/>	5	category	3	gear.html	catalog/category/view/id/3	0	1	(NULL)	1
<input type="checkbox"/>	6	category	4	gear/bags.html	catalog/category/view/id/4	0	1	(NULL)	1
<input type="checkbox"/>	7	category	5	gear/fitness-equipment.html	catalog/category/view/id/5	0	1	(NULL)	1
<input type="checkbox"/>	8	category	6	gear/watches.html	catalog/category/view/id/6	0	1	(NULL)	1
<input type="checkbox"/>	9	category	7	collections.html	catalog/category/view/id/7	0	1	(NULL)	1
<input type="checkbox"/>	10	category	8	collections/yoga-new.html	catalog/category/view/id/8	0	1	(NULL)	1
<input type="checkbox"/>	11	product	1	joust-duffle-bag.html	catalog/product/view/id/1	0	1	(NULL)	1
<input type="checkbox"/>	12	product	1	gear/joust-duffle-bag.html	catalog/product/view/id/1/c	0	1	(NULL)	1
<input type="checkbox"/>	13	product	1	gear/bags/joust-duffle-bag.html	catalog/product/view/id/1/c	0	1	(NULL)	1
<input type="checkbox"/>	14	product	2	strive-shoulder-pack.html	catalog/product/view/id/2	0	1	(NULL)	1

Activate Windows

Search, tablerate & widget

Search_query table stores the search terms, number of results, store ID, etc.,

Shipping_tablerate table has the information on table rates, destination country ID, region ID, dest_zip

Widget related tables are:

- widget
- Widget_instance
- Widget_instance_page
- widget_instance_page_layout

