



PERCONA

LIVEONLINE
MAY 12 - 13th
2021

Overview of MySQL Server Plugins and What's New in MySQL 8

Sri Sakthivel



MySQL Database Admin @ Percona

Active Blogger

Living in Tamil Nadu

Love to explore new things

Twitter : @hercules7sakthi

Linkedin : <https://www.linkedin.com/in/sri-sakthivel-m-d-68abb8112/>

Agenda

- Introduction
- Overview of MySQL Server Plugins
- Different Type of Plugins
- What is New in MySQL 8 ?
- Case Study

Introduction

What is a Plugin?

- “PLUG” and “IN”
- Software Component
- Extend or Add Functionality
- Ease the Job

Plugins in MySQL

- MySQL supports Plugin API
- C , C++ (Before MySQL 8.0.13)
- C++ (From MySQL 8.0.13)
- Install/Uninstall in runtime
- Adding additional configuration parameters

Overview of MySQL Server Plugins

Plugin Directory

- BASEDIR/lib/mysql/plugin (Default)
- plugin-dir
- Plugin directory can be changed
- Need MySQL service restart

```
mysql> set global plugin_dir='/var/lib/mysql_plugins';ERROR 1238 (HY000):  
Variable 'plugin_dir' is a read only variable
```

Plugin Installation

→ INSTALL PLUGIN

- ◆ Need INSERT privilege on mysql.plugin table
- ◆ Load and initialize the plugin

```
mysql> install plugin group_replication soname 'group_replication.so';  
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> select * from mysql.plugin\G  
***** 1. row *****  
name: group_replication  
dl: group_replication.so  
1 row in set (0.00 sec)
```

Plugin Installation

- mysql.plugin
 - ◆ Needs MySQL restart after making the entry
 - ◆ Load one or more plugins after restart
 - ◆ skip-grant-tables

```
mysql> insert into mysql.plugin values ('validate_password','validate_password.so');  
Query OK, 1 row affected (0.02 sec)
```

```
[root@perconalive ~]# service mysqld restart  
Redirecting to /bin/systemctl restart mysqld.service  
[root@perconalive ~]#
```

```
| validate_password | ACTIVE |
```

Plugin Installation

- plugin-load
- plugin-load-add
 - ◆ Load plugins on startup
 - ◆ After the storage engine and built-in plugins initialization
 - ◆ Plugin-load-add option should be after plugin-load

Correct approach:

```
mysqld --plugin-load="validate_password.so" --plugin-load-add="group_replication.so"  
mysqld --plugin-load="validate_password.so;group_replication.so"
```

Wrong approach:

```
mysqld --plugin-load="validate_password.so" --plugin-load="group_replication.so"  
mysqld --plugin-load-add="validate_password.so" --plugin-load="group_replication.so"
```

Plugin Installation

- early-plugin-load
 - ◆ Plugins load before storage engine initialization
 - ◆ Plugins load before built-in plugins initialization
 - ◆ Kyring_file.so need to be loaded before storage engines/built-in plugins

```
mysqld --early-plugin-load="keyring_file.so"
```

- mysql_plugin
 - ◆ MySQL Utility
 - ◆ Removed at MySQL 8

```
mysql_plugin <plugin_name> ENABLE
```

Plugin Activation States

→ OFF



Disable the plugin
It will not impact the mysql.plugin table

```
mysql> select * from mysql.plugin\G
***** 1. row *****
name: group_replication
dl: group_replication.so
1 row in set (0.01 sec)
```

After restart,

```
mysql      2621  2587  21 09:38 pts/0    00:00:01 mysqld --group-replication=OFF --user=mysql
```

```
| group_replication          | DISABLED | GROUP REPLICATION | group_replication.so | GPL |
```

Plugin Activation States

- ON
 - ◆ Enable the plugin
 - ◆ Allow the MySQL service to start if the plugin initialization fails.
- FORCE
 - ◆ Enable the plugin
 - ◆ Will not allow the MySQL service to start if the plugin initialization fails
- FORCE_PLUS_PERMANENT
 - ◆ Enable the plugin
 - ◆ Will not allow the MySQL service to start if the plugin initialization fails
 - ◆ UNINSTALL the plugin is not possible while MySQL service running

Uninstall Plugin

- UNINSTALL PLUGIN <plugin_name>
 - ◆ Unload the plugin
 - ◆ Remove the entry from mysql.plugin table
 - ◆ Need DELETE privilege

```
mysql> uninstall plugin group_replication;  
Query OK, 0 rows affected (0.00 sec)
```

Uninstall Plugin

Fails on:

→ Built-in plugins

```
mysql> uninstall plugin MyISAM;  
ERROR 1619 (HY000): Built-in plugins cannot be deleted
```

→ Plugin loaded with FORCE_PLUS_PERMANENT

```
mysql> uninstall plugin group_replication;  
ERROR 1702 (HY000): Plugin 'group_replication' is force_plus_permanent and can not  
be unloaded
```

Obtain Plugin Information

- SHOW PLUGINS
- INFORMATION_SCHEMA.PLUGINS
- MYSQL.PLUGIN
- MYSQL.USER
 - ◆ plugin name
 - ◆ plugin status
 - ◆ plugin type
 - ◆ plugin library
 - ◆ plugin author
 - ◆ plugin license

Different Types of Plugins

Built-in vs Non-Built-in

→ Built-in plugins

- ◆ Static plugins
- ◆ Compiled with server
- ◆ Uninstall is not possible on runtime
- ◆ `plugin_library` is null

```
select plugin_name, plugin_status, plugin_library from  
information_schema.plugins where plugin_library is null;
```

```
mysql> uninstall plugin MyISAM;  
ERROR 1619 (HY000): Built-in plugins cannot be deleted
```

Built-in vs Non-Built-in

→ Non-built-in plugins

- ◆ Dynamic plugins
- ◆ `INSTALL PLUGIN / plugin-load /plugin-load-add / mysql.plugin`
- ◆ Uninstall is possible on runtime
- ◆ `plugin_library` with library name

```
select plugin_name, plugin_status, plugin_library from  
information_schema.plugins where plugin_library not is null
```

```
mysql> uninstall plugin clone;  
Query OK, 0 rows affected (0.03 sec)
```

Types

- Storage engines
- Full-text parsers
- Daemons
- INFORMATION_SCHEMA tables
- Semisynchronous replication
- Auditing
- Authentication
- Password validation and strength checking
- Protocol tracing
- Query rewriting
- Keyring plugins

What's New in MySQL 8 ?

What's New in MySQL 8 ?

- The `mysql_plugin` utility has been removed (MySQL 8.0.0)
- Includes a component-based infrastructure for improving server extensibility (MySQL 8.0.1)

```
[root@testMySQL plugin]# pwd
/usr/lib64/mysql/plugin
[root@testMySQL plugin]# ls -lrth | grep -i component
-rwxr-xr-x. 1 root root 268K Dec 11 08:33 component_audit_api_message_emit.so
-rwxr-xr-x. 1 root root 152K Dec 11 08:33 component_log_sink_syseventlog.so
-rwxr-xr-x. 1 root root 195K Dec 11 08:33 component_log_filter_dragnet.so
-rwxr-xr-x. 1 root root  39K Dec 11 08:33 component_query_attributes.so
-rwxr-xr-x. 1 root root 459K Dec 11 08:33 component_log_sink_json.so
-rwxr-xr-x. 1 root root 908K Dec 11 08:33 component_reference_cache.so
-rwxr-xr-x. 1 root root 528K Dec 11 08:33 component_mysqlbackup.so
-rwxr-xr-x. 1 root root 500K Dec 11 08:33 component_validate_password.so
```

What's New in MySQL 8 ?

```
mysql> install component 'file://component_log_sink_json';
```

```
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> select * from mysql.component where component_urn like '%json%'\G
```

```
***** 1. row *****
```

```
    component_id: 4
```

```
component_group_id: 1
```

```
    component_urn: file://component_log_sink_json
```

```
1 row in set (0.01 sec)
```

```
mysql> uninstall component 'file://component_log_sink_json';
```

```
Query OK, 0 rows affected (0.01 sec)
```

What's New in MySQL 8 ?

→ Caching_sha2_password is available (MySQL 8.0.3)

```
mysql> select * from information_schema.plugins where plugin_name='caching_sha2_password'\G
***** 1. row *****
      PLUGIN_NAME: caching_sha2_password
      PLUGIN_VERSION: 1.0
      PLUGIN_STATUS: ACTIVE
      PLUGIN_TYPE: AUTHENTICATION
      PLUGIN_TYPE_VERSION: 2.0
      PLUGIN_LIBRARY: NULL
      PLUGIN_LIBRARY_VERSION: NULL
      PLUGIN_AUTHOR: Oracle Corporation
      PLUGIN_DESCRIPTION: Caching sha2 authentication
      PLUGIN_LICENSE: GPL
      LOAD_OPTION: FORCE
1 row in set (0.00 sec)
```

What's New in MySQL 8 ?

- `validate_password` plugin has been reimplemented to use the component infrastructure (MySQL 8.0.4)
- `INSTALL UNINSTALL PLUGIN` is deprecated

```
[root@perconalive plugin]# ls -lrth | grep -i validate
-rwxr-xr-x. 1 root root 431K Dec 11 08:33 validate_password.so
-rwxr-xr-x. 1 root root 500K Dec 11 08:33 component_validate_password.so
```

```
mysql> INSTALL COMPONENT 'file://component_validate_password';
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> UNINSTALL COMPONENT 'file://component_validate_password';
Query OK, 0 rows affected (0.01 sec)
```

What's New in MySQL 8 ?

- For RHEL, SLES, and Fedora RPMs, plugin directory changed for debug builds from /usr/lib64/mysql/plugin to /usr/lib64/mysql/plugin/debug (MySQL 8.0.4)

```
[root@testMySQL plugin]# ps -ef | grep -i mysql[d]
mysql      16864 14186  3 02:02 pts/0      00:00:02 mysqld-debug --user=mysql
[root@testMySQL plugin]# mysql -e "select @@plugin_dir\G"
***** 1. row *****
@@plugin_dir: /usr/lib64/mysql/plugin/debug/
```

- Default_authentication_plugin = caching_sha2_password (MySQL 8.0.4)

```
[root@testMySQL plugin]# mysql -e "select @@default_authentication_plugin\G"
***** 1. row *****
@@default_authentication_plugin: caching_sha2_password
```

What's New in MySQL 8 ?

- Query rewriter plugin now include support for INSERT, REPLACE, UPDATE, and DELETE. (MySQL 8.0.12)

```
[root@testMySQL mysql-8.0]# cat install_rewriter.sql | grep -i  
'create\\|install\\|drop'  
CREATE DATABASE IF NOT EXISTS query_rewrite;  
CREATE TABLE IF NOT EXISTS query_rewrite.rewrite_rules (  
INSTALL PLUGIN rewriter SONAME 'rewriter.so';  
CREATE FUNCTION load_rewrite_rules RETURNS STRING  
CREATE PROCEDURE query_rewrite.flush_rewrite_rules()
```

What's New in MySQL 8 ?

- MySQL header files used by plugins now contain C++ code (MySQL 8.0.13)
- MySQL Enterprise Edition now provides data masking and de-identification capabilities, implemented as a plugin library containing a plugin and a set of user-defined functions (MySQL 8.0.13)
- Introduced ddl_rewriter plugin (MySQL 8.0.16)
 - ◆ Modifies the CREATE TABLE statements

```
mysql> install plugin ddl_rewriter soname 'ddl_rewriter.so';  
Query OK, 0 rows affected (0.04 sec)
```

What's New in MySQL 8 ?

- Sha256_password is deprecated (MySQL 8.0.16)
- Clone plugin introduced (MySQL 8.0.17)

```
mysql> INSTALL PLUGIN clone SONAME 'mysql_clone.so';  
Query OK, 0 rows affected (0.14 sec)
```

- MySQL Enterprise Edition now includes a keyring_oci plugin (MySQL 8.0.22)

Case Study

Situation

- UPDATE query on a transaction table
- UPDATE based on the column “name”
- The column “name” have the index
- The UPDATE was mistakenly deployed with LOWER function
- UPDATE not using the index
- The table has millions of rows
- Concurrent UPDATES are killing the performance
- MySQL version is 8.0.17

Example

```
Create Table: CREATE TABLE `transactions` (  
  `id` int NOT NULL AUTO_INCREMENT,  
  `name` varchar(16) DEFAULT NULL,  
  `dob` date DEFAULT NULL,  
  `account_no` varchar(16) DEFAULT NULL,  
  PRIMARY KEY (`id`),  
  KEY `idx_name` (`name`)  
) ENGINE=InnoDB AUTO_INCREMENT=390280 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci  
1 row in set (0.00 sec)
```

```
mysql> select count(*) from transactions;  
+-----+  
| count(*) |  
+-----+  
| 6983488 |  
+-----+  
1 row in set (4.18 sec)
```

Example

```
mysql> explain update transactions set name='sakthi' where
lower(name)='sakthi_4027621'\G
***** 1. row *****
      id: 1
  select_type: UPDATE
        table: transactions
   partitions: NULL
         type: index
possible_keys: NULL
          key: PRIMARY
       key_len: 4
         ref: NULL
      rows: 6984129
   filtered: 100.00
      Extra: Using where
1 row in set, 1 warning (0.00 sec)
```

Example

```
mysql> update transactions set name='sakthi' where lower(name)='sakthi_4027621'\G
```

Query OK, 1 row affected (8.44 sec)

Rows matched: 1 Changed: 1 Warnings: 0

```
mysql> select table_schema, table_name, column_name,  
character_set_name,collation_name from information_schema.columns where  
column_name='name' and table_name='transactions'\G
```

```
***** 1. row *****
```

```
TABLE_SCHEMA: percona
```

```
TABLE_NAME: transactions
```

```
COLUMN_NAME: name
```

```
CHARACTER_SET_NAME: utf8mb4
```

```
COLLATION_NAME: utf8mb4_0900_ai_ci
```

```
1 row in set (0.00 sec)
```

Challenge

- Code change is not possible at that time
- Need to fix that query without changing the code

Solution

→ Need to remove the LOWER function

```
mysql> explain update transactions set name='sakthi' where name='sakthi_4027621'\G
***** 1. row *****
      id: 1
  select_type: UPDATE
        table: transactions
  partitions: NULL
         type: range
possible_keys: idx_name
          key: idx_name
       key_len: 67
         ref: const
        rows: 1
   filtered: 100.00
      Extra: Using where; Using temporary
1 row in set, 1 warning (0.00 sec)
```

Solution

→ Implemented the query rewrite plugin

```
mysql> select @@lc_messages_dir;
```

```
+-----+  
| @@lc_messages_dir      |  
+-----+  
| /usr/share/mysql-8.0/  |  
+-----+  
1 row in set (0.00 sec)
```

```
[root@testMySQL ~]# cd /usr/share/mysql-8.0/
```

```
[root@perconalive mysql-8.0]# mysql -vv < install_rewriter.sql | grep -i  
'create\\|install\\|drop'
```

```
CREATE DATABASE IF NOT EXISTS query_rewrite  
CREATE TABLE IF NOT EXISTS query_rewrite.rewrite_rules (  
INSTALL PLUGIN rewriter SONAME 'rewriter.so'  
CREATE FUNCTION load_rewrite_rules RETURNS STRING  
CREATE PROCEDURE query_rewrite.flush_rewrite_rules()
```

Solution

→ Added the rules for that UPDATE by removing the function LOWER()

```
mysql> insert into query_rewrite.rewrite_rules
      -> (id,pattern_database,pattern,replacement) values
      -> (1,'percona','update transactions set name = ? where LOWER(name) = ?',
      -> 'update transactions set name = ? where name = ?');
Query OK, 1 row affected (0.01 sec)

mysql> call query_rewrite.flush_rewrite_rules();
Query OK, 1 row affected (0.01 sec)

mysql> select id,pattern_database,pattern,replacement from query_rewrite.rewrite_rules\G
***** 1. row *****
      id: 1
pattern_database: percona
      pattern: update transactions set name = ? where LOWER(name) = ?
      replacement: update transactions set name = ? where name = ?
1 row in set (0.00 sec)
```

Result

→ Problem Fixed

```
mysql> update transactions set name = 'sakthi' where LOWER(name) =  
'sakthi_4027621';
```

```
Query OK, 1 row affected, 1 warning (0.01 sec)
```

```
Rows matched: 1  Changed: 1  Warnings: 1
```

```
mysql> show warnings\G
```

```
***** 1. row *****
```

```
Level: Note
```

```
Code: 1105
```

```
Message: Query 'update transactions set name = 'sakthi' where LOWER(name)  
= 'sakthi_4027621'' rewritten to 'update transactions set name = 'sakthi' where  
name = 'sakthi_4027621' by a query rewrite plugin
```

```
1 row in set (0.00 sec)
```

THANK YOU!



PERCONA
LIVEONLINE
MAY 12 - 13th
2021