SHOW CREATE TABLE mysql.general\_log;

SHOW CREATE TABLE mysql.slow\_log;

I experimented with trying to create a trigger for a system table (mysql.slow\_log). It simply cannot be done. However, there is a rather unconventional method you may want to try:

Step 01) Activate the slow\_log with the output as a table and restart mysql

[mysqld]

log-output=TABLE

long-query\_time=0.0001

Step 02) Convert slow\_log to MyISAM, copy it to areva\_log, and index slow\_log on db

SET @old\_log\_state = @@global.slow\_query\_log;

SET GLOBAL slow\_query\_log = 'OFF';

ALTER TABLE mysql.slow\_log ENGINE = MyISAM;

ALTER TABLE mysql.slow\_log ADD INDEX (start\_time);

CREATE TABLE mysql.areva\_log LIKE mysql.slow\_log;

ALTER TABLE mysql.slow\_log ADD INDEX (db);

SET GLOBAL slow\_query\_log = @old\_log\_state;

Step 03) Create a Stored Procedure to Copy all Slow Log entries whose db is areva into mysql.areva\_log

DELIMITER $$

DROP PROCEDURE IF EXISTS `mysql`.`AppendToArevaLog` $$

CREATE PROCEDURE `mysql`.`AppendToArevaLog` ()

BEGIN

SET @old\_log\_state = @@global.slow\_query\_log;

SET GLOBAL slow\_query\_log = 'OFF';

INSERT INTO mysql.aerva\_log

SELECT \* FROM mysql.slow\_log WHERE db = 'areva';

SET GLOBAL slow\_query\_log = @old\_log\_state;

END $$

DELIMITER ;

Step 04) Call the stored procedure every 15 minutes using a crontab or using a mysql event

Step 05) Zap the slow log every week, month, or year:

SET @old\_log\_state = @@global.slow\_query\_log;

SET GLOBAL slow\_query\_log = 'OFF';

CREATE TABLE mysql.slow\_log\_empty LIKE mysql.slow\_log;

ALTER TABLE mysql.slow\_log RENAME mysql.slow\_log\_to\_zap;

ALTER TABLE mysql.slow\_log\_empty RENAME mysql.slow\_log;

SET GLOBAL slow\_query\_log = @old\_log\_state;

DROP TABLE mysql.slow\_log\_to\_zap;

Give it a Try !!!

[monitoring - Create a custom log table in MySQL - Database Administrators Stack Exchange](https://dba.stackexchange.com/questions/8105/create-a-custom-log-table-in-mysql)

[Redirecting MySQL errors and slow logs into another database - Database Administrators Stack Exchange](https://dba.stackexchange.com/questions/28416/redirecting-mysql-errors-and-slow-logs-into-another-database?noredirect=1&lq=1)

![Graphical user interface, text, application

Description automatically generated]()

[Finding slow and low performance queries (prisma.io)](https://www.prisma.io/dataguide/mysql/reading-and-querying-data/identifying-slow-queries)

**Showing the full process list**

To display all current operations that MySQL's processing threads are executing, type:

SHOW FULL PROCESSLIST;

**Configure MySQL to log slow queries**

Before we move on, it is important to note that while slow query logging is incredibly useful, it can potentially have an additional performance impact. MySQL must perform additional operations to time each query and to record the results to a log. This can impact performance and fill up hard drive space unexpectedly.

It may not be a good idea to log slow queries at all times. Instead, enable the functionality when you are actively investigating an issue and disable it when you are finished.

With that in mind, you can configure slow query logging by modifying the MySQL server's configuration file. You can also modify these values interactively, but setting good defaults in the configuration will make it easier to tweak interactively later.

Open MySQL's configuration file. On most Debian Linux-based systems, the configuration file will be located at /etc/mysql/mysql.conf.d/mysqld.conf:

sudo vim /etc/mysql/mysql.conf.d/mysqld.conf

We will want to modify or potentially add the following settings:

| **Variable** | **Setting** | **Description** |
| --- | --- | --- |
| slow\_query\_log | ON | Toggles whether slow querying is enabled. |
| slow\_query\_log\_file | /var/log/mysql/mysql-slow.log | The log file where slow queries will be recorded. |
| long\_query\_time | (time in seconds) | The threshold, in seconds, that a query must pass before being considered a "slow" query. |
| min\_examined\_row\_limit | (number of rows) | The number of rows a query must consider before it is a slow query candidate. |
| log\_slow\_admin\_statements | ON | Toggles whether administrative commands are also subject to logging. |
| log\_queries\_not\_using\_indexes | ON | Toggles whether queries will be recorded if they are not consulting an index. |
| log\_slow\_extra | ON | For MySQL servers version 8.0.14 or later, this toggles whether to log additional information about the query. |
| log\_slow\_replica\_statements | ON | For MySQL servers version 8.0.26 or later, this toggles whether to log slow statements that have been executed on the replica. This only applies to statements where binlog\_format is set to STATEMENT or MIXED. |
| log\_slow\_slave\_statements | ON | For MySQL servers version 8.0.25 or earlier, this toggles whether to log slow statements that have been executed on the replica. This only applies to statements where binlog\_format is set to STATEMENT or MIXED. |
| **Variable** | **Setting** | **Description** |
| slow\_query\_log | ON | Chuyển đổi xem truy vấn chậm có được bật hay không. |
| slow\_query\_log\_file | /var/log/mysql/mysql-slow.log | Tệp nhật ký nơi các truy vấn chậm sẽ được ghi lại. |
| long\_query\_time | (time in seconds) | Ngưỡng, tính bằng giây, mà một truy vấn phải vượt qua trước khi được coi là truy vấn "chậm". |
| min\_examined\_row\_limit | (number of rows) | Số hàng mà một truy vấn phải xem xét trước khi nó là một ứng cử viên truy vấn chậm. |
| log\_slow\_admin\_statements | ON | Chuyển đổi xem các lệnh quản trị có phải là đối tượng ghi nhật ký hay không. |
| log\_queries\_not\_using\_indexes | ON | Chuyển đổi liệu các truy vấn có được ghi lại hay không nếu chúng không tham khảo chỉ mục. |
| log\_slow\_extra | ON | Đối với máy chủ MySQL phiên bản 8.0.14 trở lên, điều này chuyển đổi xem có ghi thông tin bổ sung về truy vấn hay không. |
| log\_slow\_replica\_statements | ON | Đối với máy chủ MySQL phiên bản 8.0.26 trở lên, thao tác này sẽ chuyển đổi xem có ghi lại các câu lệnh chậm đã được thực thi trên bản sao hay không. Điều này chỉ áp dụng cho các câu lệnh mà binlog\_format được đặt thành STATEMENT hoặc MIXED. |
| log\_slow\_slave\_statements | ON | Đối với máy chủ MySQL phiên bản 8.0.25 hoặc cũ hơn, điều này chuyển đổi cho dù ghi lại các câu lệnh chậm đã được thực thi trên bản sao. Điều này chỉ áp dụng cho các câu lệnh mà binlog\_format được đặt thành STATEMENT hoặc MIXED. |

**1. MyISAM trong MySQL**

Đây là kiểu Storage Engine mặc định khi tạo bảng và được dùng phổ biến nhất. MyISAM cho phép lập chỉ mục toàn cột (Full Text Index). Do đó, Storage Engine này cho tốc độ truy suất (Đọc và tìm kiếm) nhanh nhất trong các Storage Engine. Nhược điểm của MyISAM là hoạt động theo kiểu Table Level locking nên khi cập nhật (Thêm,xóa,sửa) 1 bản ghi nào đó trong cùng 1 table thì table đó sẽ bị khóa lại, không cho cập nhật (Thêm,xóa,sửa) cho đến khi thao tác cập nhật trước đó thực hiện xong. Ngoài ra, do thiết kế đơn giản và không kiểm tra ràng buộc dữ liệu nên loại Storage Engine này dễ bị hỏng chỉ mục và dễ bị Crash. Đây là cơn ác mộng của các webmaster khi table Crash là table có dung lượng lớn, khi phục hồi rất lâu và hồi hộp

[Sự khác nhau giữa 2 storage engine : MyISAM và InnoDB (viblo.asia)](https://viblo.asia/p/su-khac-nhau-giua-2-storage-engine-myisam-va-innodb-bJzKmgVPl9N)