

MySQL Query Log

SQL DM for MySQL retrieves (completely or partially) the General query log and the Slow query log from the MySQL servers it connects to, and analyzes them. Here, we see how to set up details for the connection, so that log analysis is available with SQL DM for MySQL. You have to set up details for the general query log and the slow query log independently. Enabling **Slow Query Log** 'log queries not using indexes' instead needs SUPER privilege. Refer to the MySQL documentation on how to enable and configure logging. MySQL server logs can be written to files on the server machine or to tables in the MySQL database itself.

The MySQL server (since version 5.0) has an option to log (in the slow log) queries that do not use an index. Such queries need not be slow if there are only a few hundred or few thousand records in the table(s) involved. But they are 'potentially slow' and should be identified if they access tables, which will continue to grow. You can enable and disable this from here too (SQL DM for MySQL will send the appropriate SET of statements to MySQL) Note: Only DML and DDL queries are recorded in the slow query log.

Test Server

CONFIGTAGSNOTIFICATIONSADVANCED

System Metrics

Data Collection

Replication

Galera

MySQL Error Log

MySQL Query Log

Audit Log

Sniffer

Deadlock

Monitors

Real-Time

Connection Settings

Monitor MySQL Query Log

Slow Query Log

FILE PATH

/var/log/mysql/slow.log

Make sure Monyog has privilege to read it if the file is on a shared network drive

LONG QUERY TIME

10

Second(s)

Log queries not using indexes

General Query Log

FILE PATH

/var/log/mysql/general.log

Make sure Monyog has privilege to read it if the file is on a shared network drive

LOGGING IN

FILE

SAVE

Logs written to files:

First, let's consider the situation where server logs are stored as files on the server machine. This is the most common situation and the only one available with MySQL servers before version 5.1. First time you configure a server with this option, click the **Fetch query log details** button. The MySQL server knows (it is stored in server variables) what logs are enabled and how logging is configured. Click **Test Path** to verify that the path SQL DM for MySQL connects and verifies the existence of the file (but not its content).

The log files can be accessed from the local file system (if SQL DM for MySQL and MySQL is running on the same computer) or by using SFTP (if SQL DM for MySQL and MySQL is running on different computers) or by using RDS/Aurora (Using API) if you are using a RDS/Aurora instance. Note that you must use the file and path syntax of the machine where the logs are.

If the log files can be accessed from a shared drive, over a network, or from a network enabled file system (like NFS on Linux), then SQL DM for MySQL can access them as if they were local files. No additional SSH/SFTP configuration is required in this case: the operating system takes care of the file transfer transparently.

When **via SFTP** option is chosen, then SSH server details as defined in SSH server details settings are used to read the file from the remote system. Note that the SSH user must have read access to the log files.

When RDS/Aurora (Using API) option is chosen, make sure that you have the required Access credentials with you to fetch the log files. The Access credentials can be generated from the AWS Management Console. You can refer to the documentation, on how to generate credential keys here: [Getting Your Access Key ID and Secret Access Key](#).

By default, MONyog(SQL DM for MySQL) service runs under Local System Account. If you have Slow query or General query logs in a Mapped Network Drive, SQL DM for MySQL is not be able to reach it. You need to use UNC notation for SQL DM for MySQL to be able to access them. See [FAQ 31](#) for details.

Logs written to MySQL tables:

This option is supported by MySQL from version 5.1. Also, SQL DM for MySQL supports when this option is available. Here, you click the **Fetch Log Details From MySQL** button. When this option is used there is no file path to configure and no SSH details to consider. SQL DM for MySQL can retrieve the server log by sending simple `SELECT` statements. Only the MySQL user used by SQL DM for MySQL to connect to MySQL must have `SELECT` privileges to the tables.