

# Different Types of Logs Supported

## MySQL query log settings

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, you see how to set up details for the connection, so that log analysis are 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 continue to grow. You can enable and disable it, as well (SQL DM for MySQL sends the appropriate `SET` of statements to MySQL).



Only DML and DDL queries are recorded in the slow query log.

## Logs written to files

First, 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.

If it is the first time you configure a server with this option, click the **Fetch query log details** button. MySQL server detects (it is stored in server variables) what logs are enabled and how logging is configured. Click **Test Path** to verify the path. SQL DM for MySQL connects and verify 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). 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.



The SSH user must have read access to the log files.

If MySQL server version is greater than 5.1.6 then all the fields mentioned in log analyzer would be editable i.e. if a user changes and saves the settings by clicking **Save** a pop up is displayed where a user can set the new value to corresponding MySQL Server.

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 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

It is supported by MySQL from version 5.1. Also, SQL DM for MySQL supports when this option is available. Here, 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.

In the Query Analyzer tab select the MySQL server, the type of log (including the 'pseudo log') you want to analyze (Slow Query Log, General Query Log, or [Sniffer](#)), and click **Analyze** to start the analysis:

TOP QUERIES BASED ON TOTAL TIME	COUNT	TOTAL TIME	AVERAGE LATENCY	USER@HOST
<code>select sleep(1000);</code>	2	28:54.263	14:27.131	root[root] @ localhost
<code>select sleep(200);</code>	1	03:20.000	03:20.000	root[root] @ localhost
<code>select sleep(20);</code>	1	20.016	20.016	root[root] @ localhost
<code>SELECT COUNT(1) AS COUNT FROM INFORMATION_SCHEMA.TABLES A LEFT JOIN INFORMATION_SCHEMA.TABL...</code>	7	16.749	02.393	root[root] @ Sandhya-P...
<code>SHOW FULL PROCESSLIST;</code>	4	10.750	02.688	root[root] @ [192.168.1....
Others	4	12.005	03.001	

  

QUERY	COUNT	TOTAL TIME	AVERAGE LATENCY ↓	USER@HOST
<code>select sleep(1000);</code>	2	28:54.263	14:27.131	root[root] @ localhost
<code>select sleep(200);</code>	1	03:20.000	03:20.000	root[root] @ localhost
<code>select sleep(20);</code>	1	20.016	20.016	root[root] @ localhost
<code>insert into `zak2`.`batchingredientsstockids` (`ID`,`BatchIngredientID`,`Stockid`,`UomQuantity`) values ('00160220-402C-4FDE-...</code>	1	07.339	07.339	root[root] @ localhost

SQL DM for MySQL includes an improved Top Query Overview to ease the query performance overtime analysis.

After getting the results of your Analysis, click a query to obtain the Query Details and Query Explain tabs with detailed information about the selected query:

**Query Details** Query Explain

Normalized Query  
`SELECT TABLE_SCHEMA AS 'Database', Table_Name AS 'Table', TABLE_ROWS AS 'Rows' FROM information_schema.TABLES`

Example Query  
`/* Requirement : MySQL V5.0+ This query will return TableName and number of records it has. */ SELECT TABLE_SCHEMA AS 'Database', Table_Name AS 'Table', TABLE_ROWS AS 'Rows' FROM information_schema.TABLES`

FIRST SEEN: May 28, 19 13:29:13 | LAST SEEN: May 28, 19 16:28:13

FULL TABLE SCAN: 180 | POOR INDEX USED: 0

**QUERY EXECUTION TIME**

TOTAL TIME: 8.52K 0.07% of total

AVERAGE LATENCY: 00.003

MAX TIME: 00.011

QUERY OCCURRENCE: 5.291

LOCK TIME: 00.002

**ROWS**

ROWS SENT: 3K

ROWS EXAMINED: 29K

**TEMP TABLES**

DISK TEMP TABLES: 180

MEMORY TEMP TABLES: 360

**SELECT**

SELECT FULL JOIN: 0

SELECT FULL RANGE JOIN: 0

SELECT RANGE: 0

SELECT RANGE CHECK: 0

SELECT SCAN: 180

**SORT**

SORT MERGE PASSES: 0

SORT RANGE: 0

SORT ROWS: 3K

SORT SCAN: 189

**Query Details** Query Explain

Query  
`SELECT COUNT(*) FROM information_schema.`PLUGINS` WHERE Plugin_Name IN ('INNOCEB','INNOCEB_LOCK_QUEUE','INNOCEB_LOCKS','INNOCEB_TSK') AND Plugin_Status='ACTIVE'`

**Explain Result**

ID	SELECT TYPE	TABLE	TYPE	POSSIBLE KEYS	KEY	KEY LEN	REF	ROWS	EXTRA
1	SIMPLE	TABLE	ALL	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	Using where

**Optimized Query**

`select count(*) as 'COUNT(*)' from 'information_schema`.`plugins` where 'information_schema`.`plugins`.`PLUGIN_NAME` in ('INNOCEB','INNOCEB_LOCK_QUEUE','INNOCEB_LOCKS','INNOCEB_TSK') and 'information_schema`.`plugins`.`PLUGIN_STATUS` = 'ACTIVE'`

Explain plan is available in Query analyzer for Slow\_log table based logging, Processlist based sniffer and performance schema based sniffer.

With the General Query Log there are few specific problems:

1. With multi-line queries only record the first line of the statement. The reason is that, as the log does not record the statement DELIMITERS, there is no way to tell where a multi-line statement ends. Even the option to 'Show full' is not displayed more than one line as SQL DM for MySQL has only stored one line. Refer to [FAQ 23](#).
2. It is not always possible to tell what user executed a specific query. When this is the case the User column displays empty in the Query Analyzer output. It is not a bug in SQL DM for MySQL but a limitation with the general log itself.

You can sort the display by clicking on the column header. Note that statement grouping/counting and sorting is case insensitive.

## Filter settings

There is an option to **Replace literals from the query**. The purpose of this option is to eliminate small differences between almost identical queries. Currently, quoted strings and numbers are replaced with the dummy string '?' only. The filtering settings are stored for that particular session which is not permanent.

For example:

```
SELECT * FROM customer_master  
WHERE cust_id = 23 AND address = 'r;#23 fleet street';
```

becomes,

```
SELECT * FROM customer_master  
WHERE cust_id = ? AND address = ?;
```

## Settings - Slow Query Log

Replace literals in queries with '?'

### FILTER USERS

Include



Supports regex character \*

### FILTER HOSTS

Include

Supports regex character \*

Include Users executing the queries with Host names

Read All

### READING LIMIT FROM FILE

1

MB



APPLY

The reading limit **All** is selected it considers the whole file for analyzing but if the option is **Last**, it reads the last specified chunk in KB, MB, or bytes out of the whole log file. Also, you can define a timeframe to be analyzed and the size of the 'log chunk' (in KB, MB and Bytes for file based logs and in rows for table-based logs) to be transferred to SQL DM for MySQL.

If **All** is selected in the list, is not considering any timeframe and just displays all queries within the specified size/chunk. Also note, it is the smallest of those two settings that have effect for the analysis. For analyzing the sniffer pseudo log there is no chunk size to be defined as the complete pseudo log as stored in the SQL DM for MySQL database that is considered. The selected log chunk needs to have statements for the selected period. If not, then SQL DM for MySQL of course only display data from the first log record available.

**Include User and Host Information:** If this option is selected it displays the User and Host of that particular query and it groups the query analyzer table based on `user@host` and query.



If SQL DM for MySQL is already installed in the machine and Sniffer is enabled, it only displays the User info in Query Analyzer table because old SQL DM for MySQL never used to store the Host information in sniffer.data table. Also note, that this option is not supported by MySQL Proxy. In General query log, if connect string is not included in the specified chunk, it is not display the `user@host` information which is just left as an empty space.

## Export As CSV

The option to define the field delimiter is provided because some localized Windows programs for LOCALEs where comma ", " is used as a decimal sign and requires a semicolon "; " as field separator. This includes Microsoft Office programs (Excel and Access) and Microsoft text-ODBC driver on such localized Windows. On Linux, the situation is more non-uniform but also such localized OpenOffice Calc (spreadsheet) requires semicolon "; " as field separator.

Total	A	B	C	D	E	F	G	H	I	J	K	L	M	N
Total	Avg	Max	locktime	count	success	Query Occurrer	errcount	warningcount	firstseen	lastseen	query	orgquery	db	
28.035	28.035	28.035	28.035	28.035	1	0	0	1	0	Jan 14, 17 20:2	Jan 14, 17 20:2	SELECT 'sleep'	select sleep(3)	information
01:10.660	5.435	12	2	13	11	0	2	0	0	Jan 06, 17 20:0	Jan 16, 17 20:0	SELECT 'sleep'	select sleep(3)	
7.251	0.196	3.046	0.009	37	37	0	0	0	0	Jan 04, 17 17:0	Jan 16, 17 22:1	SELECT 'SCHE'	SELECT SCHEMATA.SCHE	
2:41:27	0.146	01:00.109	0.001	66201	66201	0.676	0	0	0	Jan 04, 17 14:1	Jan 16, 17 22:4	SELECT * FROM	SELECT * FROM (SEL	
40.198	0.137	18.479	0.001	294	294	0.003	0	0	0	Jan 04, 17 15:5	Jan 16, 17 22:3	SELECT COUN'	SELECT COUNT(*) FROM ir	
03:05.078	0.072	15.857	0.001	2568	2568	0.026	0	0	0	Jan 04, 17 14:1	Jan 14, 17 16:0	SELECT EVENT'	SELECT EVENT_NAME, MA	
0.602	0.067	0.223	0.006	9	9	0	0	0	0	Jan 06, 17 17:2	Jan 11, 17 17:1	SET GLOBAL ge	SET GLOBAL general_log_fi	
0.113	0.057	0.113	0	2	2	0	0	0	0	Jan 06, 17 17:3	Jan 06, 17 17:3	UPDATE 'perfor	UPDATE 'performance_sche	
0.213	0.053	0.207	0.002	4	4	0	0	0	0	Jan 12, 17 21:3	Jan 14, 17 20:0	SHOW FULL TA	show full tables from 'axon_r	
12.341	0.047	1.903	0	262	0	0.003	262	0	0	Jan 04, 17 16:0	Jan 11, 17 17:2	SHOW ALL	show all slaves status	
0.057	0.029	0.057	0	2	2	0	0	0	0	Jan 12, 17 21:3	Jan 14, 17 20:0	SHOW CREATE	show create procedure 'sakila	
0.098	0.024	0.095	0.001	4	4	0	0	0	0	Jan 12, 17 21:3	Jan 14, 17 20:0	SHOW FULL TA	show full tables from 'sakila'	
0.02	0.02	0.02	0.02	1	1	0	0	0	0	Jan 11, 17 18:1	Jan 11, 17 18:1	EXPLAIN EXTE'	EXPLAIN /*140100 EXTEND'	
0.018	0.018	0.018	0.018	1	1	0	0	0	0	Jan 12, 17 21:3	Jan 12, 17 21:3	SELECT 'versio	select version()	
0.34	0.015	0.057	0	23	3	0	20	3	3	Jan 05, 17 17:1	Jan 16, 17 21:2	EXPLAIN EXTE'	EXPLAIN /*140100 EXTEND'	
0.412	0.015	0.046	0.001	27	26	0	1	26	26	Jan 06, 17 20:3	Jan 16, 17 20:2	EXPLAIN EXTE'	EXPLAIN /*140100 EXTEND'	
0.03	0.015	0.028	0.002	2	2	0	0	0	0	Jan 06, 17 17:2	Jan 15, 17 18:0	SHOW TABLE S	show table status from 'axon	
34.996	0.014	6.429	0	2487	2487	0.025	0	0	0	Jan 04, 17 14:1	Jan 16, 17 22:4	SELECT 'ENGIN'	SELECT 'ENGINES':SUPPI	
0.052	0.013	0.05	0.001	4	4	0	0	0	0	Jan 12, 17 21:3	Jan 14, 17 20:0	SHOW FULL TA	show full tables from 'mysql'	
0.013	0.013	0.013	0.013	1	0	0	1	0	0	Jan 14, 17 20:0	Jan 14, 17 20:0	USE 'new_trail_	use 'new_trail_new'	
44.445	0.013	8.745	0	3446	3446	0.035	0	0	0	Jan 04, 17 14:1	Jan 16, 17 22:4	SHOW GLOBAL	SHOW /*150002 GLOBAL*/ S	
45.201	0.012	10.494	0	3854	3854	0.039	0	0	0	Jan 04, 17 16:0	Jan 11, 17 17:2	SHOW ENGINE	SHOW ENGINE INNODB ST	
0.023	0.012	0.012	0.011	2	2	0	0	0	0	Jan 05, 17 14:1	Jan 05, 17 14:2	SELECT SUM (	'DATA_LENGTH' ) 'Data_jeng	
0.112	0.012	0.027	0.001	9	9	0	0	0	0	Jan 06, 17 17:2	Jan 11, 17 17:1	SET GLOBAL sli	SET GLOBAL slow_query_lo	
01:35.594	0.011	7.645	0	9090	9090	0.093	0	0	0	Jan 04, 17 14:1	Jan 16, 17 22:4	SET SESSION v	SET SESSION wait_timeout=	
0.136	0.011	0.136	0	12	12	0	0	0	0	Jan 06, 17 17:2	Jan 11, 17 17:1	SET GLOBAL sli	SET GLOBAL slow_query_lo	
15:16.269	0.011	16.393	0.001	80445	80445	0.822	0	0	0	Jan 04, 17 14:1	Jan 16, 17 22:4	SHOW GLOBAL	SHOW /*150002 GLOBAL*/ S	