

MySQL Privileges

MySQL Privileges

Whether you are running a single MySQL database server or multiple servers in a cluster, SQL DM for MySQL provides a comprehensive list of performance variables that you can monitor in real-time to check the health and performance of your MySQL server or servers.

In order to access the data for these metrics, SQL DM for MySQL requires a dedicated user on each MySQL database server it monitors, with the appropriate privileges to collect the information it needs.

Privileges

Most SQL DM for MySQL functionalities do not require any privileges at all for the user it uses in establishing client connections with MySQL. You can create it without any special global or object privileges, using the following command:

```
GRANT USAGE ON *.*  
TO 'user'@'host';
```

Fully enabling all SQL DM for MySQL functionality requires that you grant the user some additional privileges, ensuring that it can access all the data it needs.



For an example of granting full functionality to the SQL DM for MySQL user, see Database Configuration <db-config> below.

Security Counters

SQL DM for MySQL user requires `SELECT` privileges to the `mysql.user` table to enable SQL DM for MySQL security counters.

Additionally, if the server was started with the `skip_show_databases` configuration variable, you must also explicitly grant the `SHOW DATABASES` privilege.

Replication Counters

In order to collect replication metrics from a slave, the MySQL user on the slave requires the `REPLICATION_CLIENT` privilege. You can also enable this functionality using the `SUPER` privilege.

When registering MySQL servers with SQL DM for MySQL, there is an option **Is this a Replication Slave?** to include replication information. You must set this to **Yes**, when registering the server. It defaults to **No**, when set to No, SQL DM for MySQL ignores replication data.

When testing the connection from the server registration page, SQL DM for MySQL displays the error message:

```
Access denied. You need the SUPER/REPLICATION CLIENT  
privilege for retrieving REPLICATION details!
```

You can ignore this error if the given MySQL server is not a replication slave or if you do not want to monitor replication.

InnoDB Deadlock Monitoring

Depending on the MySQL server version, the `SUPER` or `PROCESS` privileges allow you to monitor for deadlocks with the InnoDB storage engine.



Beginning in the 5.1.24 version of MySQL, you can enable this feature using `PROCESS`. Older versions require `SUPER`.

Processlist Feature

In order to collect data on the MySQL server processes for all users, you must enable the `PROCESS` privilege. Additionally, when granted the `SUPER` privilege, you can use SQL DM for MySQL to kill running processes.

In order to use the `EXPLAIN` option on the processlist, you need to grant `SELECT` and `SHOW VIEW` to the objects accessed by the statements you want to explain, or simply, grant the global `SELECT` privilege.



SQL DM for MySQL displays N/A on counters where the user lacks the required privileges and logs a record of the MySQL server error for every attempt to retrieve those counters from the server. Be aware the log may grow considerably if this is the case.

Performance Schema-based Sniffer

Collecting data on the Performance schema requires `SELECT`, `DROP` and `UPDATE` privileges.

SQL DM for MySQL uses the `SELECT` privilege to read the Performance Schema tables. The `DROP` privilege allows it to truncate these tables. The `UPDATE` privilege enables the statement digest and statement history log in the performance schema tables.

Log Retrieval

In order to retrieve log information when stored in a table, (which is supported in version 5.1 and newer of MySQL), the SQL DM for MySQL user requires the `SELECT` privilege on the log tables.

Flush Status

In order to execute `FLUSH STATUS` commands, the SQL DM for MySQL user also requires the `RELOAD` privilege.

[IDERA](#) | [Products](#) | [Purchase](#) | [Support](#) | [Community](#) | [Resources](#) | [About Us](#) | [Legal](#)