

# Reset root Passwort

Version ermitteln

```
dpkg -l 'mysql-server*'
```

## Variante 1: Mysql mit Schalter starten

```
/usr/sbin/mysqld --skip-grant-tables --skip-networking &
```

Client starten

```
mysql -u root  
  
FLUSH PRIVILEGES;  
  
SET PASSWORD FOR root@'localhost' = PASSWORD('password');  
  
FLUSH PRIVILEGES;
```

Danach mysql stoppen und wieder normal starten.

## Variante 2: Password mit initialer Textdatei setzen

Create a text file containing the password-assignment statement on a single line. Replace the password with the password that you want to use.

Für 5.5.x

```
SET PASSWORD FOR 'root'@'localhost' = PASSWORD('MyNewPass');
```

MYSQL Server mit der Datei initialisieren:

```
shell> mysqld_safe --init-file=/home/me/mysql-init &
```

```
root@server6:/home/thommie# mysqld_safe --init-  
file=/home/thommie/mysql_reset.txt &  
[1] 120598  
root@server6:/home/thommie# 160725 10:45:37 mysqld_safe Can't log to error  
log and syslog at the same time. Remove all --log-error configuration  
options for --syslog to take effect.  
160725 10:45:37 mysqld_safe Logging to '/var/log/mysql/error.log'.  
160725 10:45:37 mysqld_safe Starting mysqld daemon with databases from  
/var/lib/mysql  
160725 10:45:40 mysqld_safe mysqld from pid file /var/run/mysqld/mysqld.pid  
ended
```

# MySQL Replikation

[https://www.thomas-krenn.com/de/wiki/MySQL\\_Replikation](https://www.thomas-krenn.com/de/wiki/MySQL_Replikation)

Mit ssl: [https://www.thomas-krenn.com/de/wiki/MySQL\\_Verbindungen\\_mit\\_SSL\\_verschl%C3%BCsseln](https://www.thomas-krenn.com/de/wiki/MySQL_Verbindungen_mit_SSL_verschl%C3%BCsseln)

Einschränkung auf einige DBs:

```
replicate-wild-do-table=dbname1.%
```

```
replicate-wild-do-table=dbname2.%
```

Siehe auch

[http://dev.mysql.com/doc/refman/5.7/en/replication-options-slave.html#option\\_mysql\\_replicate-wild-do-table](http://dev.mysql.com/doc/refman/5.7/en/replication-options-slave.html#option_mysql_replicate-wild-do-table)

## Tuning

Quelle: <http://www.ubuntugeek.com/mysqltuner-check-your-mysql-server-performance.html>

MySQLTuner is a script written in Perl that will assist you with your MySQL configuration and make recommendations for increased performance and stability. Within seconds, it will display statistics about your MySQL installation and the areas where it can be improved.

It's key to remember that MySQLTuner is a script which can assist you with your server, but it is not the solution to a badly performing MySQL server. The best performance gains come from a thorough review of the queries sent to the server, and an evaluation of the MySQL server itself. A qualified developer in your application's programming or scripting language should be able to work with a MySQL database administrator to find improvements for your server. Once the server and application are optimized well, you may need to consider hardware upgrades to the physical server itself.

This is a really useful tool for helping to optimize MySQL performance. Understanding the various my.cnf variables and how they affect performance can seem really complicated but this tool takes some of the pain away and makes it easier to understand the effects of each variable. It is especially useful to be able to see the global memory usage, memory usage per thread and the maximum possible memory usage – that is really valuable information that is otherwise complex to calculate.

### MySQLTuner Features

- **Memory Usage:** Calculates MySQL memory usage at max load and makes recommendations for increasing or decreasing the MySQL memory footprint. Per-thread and server-wide buffer data is calculated
- **separately** for an accurate snapshot of the server's configuration.
- **Slow Queries:** Reviews the amount of slow queries relative to the total queries. Slow query time limits are also analyzed and recommendations are made.
- **Connections:** Current and historical connection counts are reviewed.
- **Key Buffer:** Takes configuration data and compares it to the actual indexes found in MyISAM

tables. Key cache hit rates are calculated and variable adjustments are suggested.

- Query Cache: Query cache hit rates and usage percentages are used to make recommendations for the query cache configuration variables.
- Sorting & Joins: Per-thread buffers that affect sorts and joins are reviewed along with the statistics from the queries run against the server.
- Temporary Tables: Variable recommendations are made to reduce temporary tables that are written to the disk.
- Table Cache: Compares total tables opened to the currently open tables. Calculates the table cache hit rate in order to make suggestions.
- Open Files: Determines if the server will approach or run into the open file limit set by the operating system or the MySQL server itself.
- Table Locks: Finds table locking that forces queries to wait and makes suggestions for reducing locks that require a wait.
- Thread Cache: Calculates how many times MySQL must create a new thread to respond to a query.
- Aborted Connections: Finds applications that are not closing connections to MySQL properly.
- Read/Write Ratios: Calculates the percentage of read and write operations on your MySQL installation.

Download MySQLTuner using the following command

<https://github.com/major/MySQLTuner-perl/tarball/master>

## Usage example

```
perl mysqltuner.pl --host localhost --user root --password xxxx
```

## MARIADB

Login direkt aus der Shell mit root möglich, wenn das unix\_socket plugin aktiv ist (default). Ein User, der nur über die lokale mysql DB authentifiziert werden soll, muss in der plugin Spalte mysql\_native\_password und ein Passwort gesetzt haben.

User anlegen

```
CREATE USER foo2@test IDENTIFIED BY 'mariadb';
```

Datenbank anlegen

```
CREATE DATABASE passbolt;
```

User volle rechte auf die DB geben

```
GRANT ALL privileges ON `passbolt`.* TO 'passbolt';:
```

Kontrolle

```
MariaDB [(none)]> show grants for passbolt;
```

```
+-----+
+-----+
| Grants for passbolt@%
|
+-----+
+-----+
| GRANT USAGE ON *.* TO 'passbolt'@'%' IDENTIFIED BY PASSWORD
'|*44BF0BD050389323145AA5E7F8D7549C194AE951' |
| GRANT ALL PRIVILEGES ON `passbolt`.* TO 'passbolt'@'%'
|
+-----+
+-----+
```

Alle Spalten einer Tabelle zeigen

```
SHOW COLUMNS FROM user FROM mysql;
```

Inhalt von Spalten zeigen

```
MariaDB [mysql]> SELECT host, user, password, plugin FROM user;
```

```
+-----+-----+-----+-----+
+-----+
| Host      | User   | Password                                     | plugin
|
+-----+-----+-----+-----+
+-----+
| localhost | root   | invalid                                     |
mysql_native_password |
| localhost | mysql  | invalid                                     |
mysql_native_password |
| %         | joomla | *7CFBEFC5A73BBC76917C58894D45FB09D7E3F59B |
mysql_native_password |
+-----+-----+-----+-----+
+-----+
```

Privilegien eines Users zeigen:

```
MariaDB [mysql]> show grants for joomla;
```

```
+-----+
+-----+
| Grants for joomla@%
|
+-----+
+-----+
| GRANT USAGE ON *.* TO 'joomla'@'%' IDENTIFIED BY PASSWORD
'|*7CFBEFC5A73BBC76917C58894D45FB09D7E3F59B' |
| GRANT ALL PRIVILEGES ON `joomla2`.* TO 'joomla'@'%'
|
| GRANT ALL PRIVILEGES ON `joomla`.* TO 'joomla'@'%'
|
```

```
|  
| GRANT ALL PRIVILEGES ON `joomla`.`joomla` TO 'joomla'@'%'  
|  
+-----+  
-----+
```

## SHell Login und sql Query Test

From:

<https://wiki.netzwissen.de/> - netzwissen.de Wiki

Permanent link:

[https://wiki.netzwissen.de/doku.php?id=mariadb\\_mysql&rev=1617051724](https://wiki.netzwissen.de/doku.php?id=mariadb_mysql&rev=1617051724)

Last update: **05/03/2024 - 10:52**

