

Wiki / MySQL

Dieser Artikel wurde für die folgenden Ubuntu-Versionen getestet:

- **Ubuntu 14.04** Trusty Tahr
- **Ubuntu 12.04** Precise Pangolin


Zum Verständnis dieses Artikels sind folgende Seiten hilfreich:

1. **Installation von Programmen**
2. **Ein Terminal öffnen**
3. **Einen Editor öffnen**


Inhaltsverzeichnis

1. Installation
2. Konfiguration
3. Bedienung
4. Datenbank erstellen
5. MySQL-Server steuern
6. Datensicherung
7. Graphische Oberflächen
8. Links



MySQL [<http://www.mysql.de/>]  ist eine leistungsfähige Datenbank-Serveranwendung. Die Firma MySQL AB, welche hinter der Entwicklung der Datenbank steht, wurde im Jahr 2008 von Sun übernommen und ist seit der Übernahme von Sun durch Oracle nun dieser Firma zugehörig.

MySQL erfreut sich in der (Linux-)Community einer sehr großen Beliebtheit und wird sehr häufig für Webanwendungen zusammen mit dem Webserver **Apache** [<https://wiki.ubuntuusers.de/Apache/>] und **PHP** [<https://wiki.ubuntuusers.de/PHP/>] eingesetzt. Viele Projekte und Software wie z.B. **OpenStreetMap** [<https://wiki.ubuntuusers.de/OpenStreetMap/>] oder TYPO3 verwenden MySQL als Datenbank. Auch **Inyoka** [<http://ubuntuusers.de/inyoka>], die Software von ubuntuusers.de, auf der auch dieses Wiki läuft, setzt eine MySQL-Datenbank zur Speicherung der Daten ein.

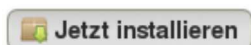
Die MySQL-Datenbank gibt es in einer freien Community-Variante (welche über die Ubuntu-Quellen verfügbar ist) und einer kostenpflichtigen Enterprise-Variante. Von der Leistungsfähigkeit unterscheiden sich die beiden Varianten nur marginal, der Hauptunterschied liegt in der Geschwindigkeit und Häufigkeit, mit der Updates und Service-Packs angeboten werden, sowie in der Verfügbarkeit von kommerziellem Support. MySQL ist neben Linux auch für Windows, MacOS, FreeBSD, Solaris und weitere Betriebssysteme **verfügbar** [<http://dev.mysql.com/downloads/mysql/5.1.html>] . Als Alternative gibt es **MariaDB** [<https://wiki.ubuntuusers.de/MariaDB/>].

Installation

MySQL-Server

MySQL kann direkt aus den Paketquellen von Ubuntu installiert werden. Dazu muss nur das Paket

- **mysql-server**



[apt://mysql-server] mit **apturl** [<https://wiki.ubuntuusers.de/apturl/>]


```
sudo apt-get install mysql-server
```

aus der Paketverwaltung heraus installiert ^[1] werden. Unter **Ubuntu 12.04** [<https://wiki.ubuntuusers.de/Precise/>] wird dabei Version 5.5 installiert.

PHP

Wer eine datenbankbasierte Webanwendung mit PHP 5 und MySQL aufsetzen möchte (z.B. ein **CMS** [https://wiki.ubuntuusers.de/CMS/] oder eine **Bildergalerie** [https://wiki.ubuntuusers.de/Bilder_verwalten/#Mit-Datenbank]), benötigt noch das Paket:

- **php5-mysql**

 **Jetzt installieren** [apt://php5-mysql] mit **apturl** [https://wiki.ubuntuusers.de/apturl/]

```
sudo apt-get install php5-mysql
```

Konfiguration

MySQL wird über die Datei **/etc/mysql/my.cnf** konfiguriert (seit 15.04 lautet die relevante Datei **/etc/mysql/mysql.conf.d/mysqld.cnf**). Es müssen üblicherweise aber keine Änderungen vorgenommen werden. Mittels eines Editors ^[3] kann man Funktionen aktivieren bzw. Einstellungen vornehmen. Die wichtigsten Optionen sollen hier kurz vorgestellt werden.

Datenpfad

MySQL speichert die Datenbanken im Verzeichnis **/var/lib/mysql** ab. Möchte man einen anderen Pfad definieren, so kann man dies über die Option

```
datadir          = /var/lib/mysql
```

eintragen. Dabei ist zu beachten, dass das Verzeichnis dem Benutzer bzw. der Gruppe **mysql** gehören sollten, damit der MySQL-Server dort auch schreiben darf. Mehr zum Thema Benutzer und Gruppen im Artikel **Benutzer und Gruppen** [https://wiki.ubuntuusers.de/Benutzer_und_Groupen/].

Lässt sich der MySQL-Server danach nicht mehr starten, liegt es wahrscheinlich an den Profil-Einstellungen von **AppArmor** [https://wiki.ubuntuusers.de/AppArmor/]. Wird das Datenbankverzeichnis in der Konfigurationsdatei geändert, aber nicht im AppArmor-Profil angepasst, kann MySQL nicht mehr auf alle Datenbanken bzw. auf Teile von bestehenden Datenbanken zugreifen. Es wird dann z.B. folgender Fehler gemeldet:

```
mysqlshow
```

```
mysqlshow: Cannot list databases: Can't read dir of '.' (errno: 13)
```

Oder wenn man auf der MySQL-Kommandozeile angemeldet ist:

```
show databases;
```

```
ERROR 1018 (HY000): Can't read dir of '.' (errno: 13)
```

Dieser Fehler wirkt sich auch auf PHPmyAdmin aus. Dort wird im linken sowie im rechten Fenster immer nur "Keine Datenbanken" angezeigt.

Abhilfe schafft die Anpassung des Ordners in der Datei **/etc/apparmor.d/usr.sbin.mysqld**:

```
1  ...
2  /var/lib/mysql/ r,
3  /var/lib/mysql/** rwk,
4  ...
```

und ein Neustart von AppArmor.

MySQL über das Netzwerk


Nach der Installation ist der MySQL-Server so konfiguriert, dass er nur via **localhost** [http://de.wikipedia.org/wiki/localhost] - also vom lokalen Rechner aus - erreichbar ist. Soll der Server über das Netzwerk ansprechbar sein, so kann man in **/etc/mysql/my.cnf** die interne IP-Adresse des Servers eintragen:

```
1  # Instead of skip-networking the default is now to listen only on
```


```
2 # localhost which is more compatible and is not less secure.
3 bind-address                = 192.168.0.10
```

So ist der MySQL-Server z.B. auch von anderen Rechnern aus dem LAN ansprechbar. Per Voreinstellung ist MySQL über den Port 3306 erreichbar. Dieser kann auch in der Konfigurationsdatei geändert werden.

Wenn von allen Netzwerk-Interfaces zugegriffen werden soll, so muss bind-address mit "bind-address = " oder mit "bind-address = 0.0.0.0" konfiguriert werden. Alternativ kann der bind-address Eintrag auch auskommentiert werden.

Weiterhin muss in den Benutzerrechten pro Benutzer hinterlegt werden, ob dieser sich über das Netzwerk anmelden darf. Weitere Hinweise hierzu findet man im **MySQL Handbuch** [<http://dev.mysql.com/doc/refman/5.0/en/grant.html>] .

Bedienung

Hier wird nur ein kurzer Einblick in die Bedienung von MySQL gegeben. Eine sehr umfangreiche Dokumentation steht in Form des **Referenzhandbuchs** [<http://dev.mysql.com/doc/refman/5.1/en/index.html>]  online zur Verfügung.


Benutzer

root

Bei der Installation des Servers wird automatisch ein root-Account angelegt. Dabei ist zu beachten, dass die Benutzerverwaltung von MySQL nichts mit den lokalen Benutzern des Linux-Systems zu tun hat - MySQL verwaltet seine Benutzer selber. Bei der Installation des MySQL-Servers wird automatisch nach einem Passwort für den Datenbank-Administrator root gefragt.

Mit dem Befehl

```
mysqladmin -u root -p password "WUNSCHPASSWORT"
```

kann das Passwort nachträglich geändert werden. Dabei wird nach dem bisherigen Passwort gefragt. Wenn noch keines hinterlegt ist, kann man hier einfach mit der Taste  bestätigen.

debian-sys-maint

Ubuntu legt noch einen weiteren Benutzer an: debian-sys-maint. Mit Hilfe dieses MySQL-Benutzers startet und stoppt Ubuntu den SQL-Server und führt weitere Wartungs-Aufgaben durch. Der Zugang ist Passwort-geschützt. Ubuntu vergibt ein voreingestelltes Passwort, das in der Datei **/etc/mysql/debian.cnf** im Klartext abgelegt wird. Für administrative Aufgaben kann man daher auch den Benutzer debian-sys-maint verwenden mit:

```
mysql -u debian-sys-maint -p
```

In **/etc/mysql/debian.cnf** befinden sich zwei identische Einträge für das Passwort: z. B. password = n4aSHUP04s1J32X5.

Passwort zurücksetzen/neu vergeben

Hat man das Passwort für den Benutzer root vergessen, kann man mit Hilfe des Benutzers debian-sys-maint das Passwort zurücksetzen und neu vergeben. Einfach wie oben beschrieben als Benutzer debian-sys-maint am MySQL-Server anmelden und die folgende Befehlssequenz am MySQL-Prompt mysql> eingeben:

```
update mysql.user set Password=password('MyNewPass') where User='root';
flush privileges;
quit;
```

Erscheint beim (Neu-)Start des MySQL-Servers folgende Fehlermeldung:

```
sudo service mysql restart
```

```
* Stopping MySQL database server mysqld [fail]
* Starting MySQL database server mysqld [ OK ]
/usr/bin/mysql-admin: connect to server at 'localhost' failed
```

```
error: 'Access denied for user 'debian-sys-maint'@'localhost' (using password: YES)'
```

muss man das Passwort für den Benutzer *debian-sys-maint* neu setzen. Dies kann der Fall sein, wenn Ubuntu während eines Upgrades des Pakets **mysql-server** das Passwort für *debian-sys-maint* überschreibt. In diesem Fall als Benutzer *root* am SQL-Server anmelden und nachfolgende Befehlssequenz am MySQL-Prompt `mysql>` eingeben. Dabei `newpassword` durch den Passwort-Eintrag in der Datei `/etc/mysql/debian.cnf` ersetzen.

```
grant all privileges on *.* to 'debian-sys-maint'@'localhost' identified by 'newpassword' with grant
option;
flush privileges;
quit;
```

Datenbank erstellen

Zum Erstellen einer neuen Datenbank meldet man sich zuerst als Datenbank-Administrator an:

```
mysql -u root -p
```

Am MySQL-Prompt `mysql>` werden folgende SQL-Befehle benötigt. Bei erfolgreicher Ausführung erhält man eine Rückmeldung in der Form:

```
Query OK, ... rows affected (... sec)
```

- Datenbank erstellen (in der Praxis entspricht der Datenbankname häufig dem der gewünschten Anwendung):

```
create database if not exists datenbankname;
```

- Neuen Datenbank-Benutzer `www-data` mit dem Kennwort `geheim` erstellen (wenn die Datenbank im Rahmen eines **LAMP** [https://wiki.ubuntuusers.de/LAMP/] -Servers eingesetzt werden soll):

```
create user 'www-data'@'localhost' identified by 'geheim';
grant usage on *.* to 'www-data'@'localhost' identified by 'geheim';
```

- Zugriffe auf die neue Datenbank für den Benutzer `www-data` erlauben:


```
grant all privileges on datenbankname.* to 'www-data'@'localhost';
flush privileges;
```

- Abmelden mit:

```
quit;
```

Alternativ kann `exit;` benutzt werden.

Tabellenarten

MySQL kennt verschiedene Tabellenarten, die sogenannten **Storage Engines** [http://dev.mysql.com/doc/refman/5.1/en/storage-engines.html] . Diese haben jeweils spezifische Vor- und Nachteile. Die Art der Tabelle kann beim Anlegen über die Option `ENGINE=` festgelegt werden. Innerhalb einer Datenbank können verschiedene Tabellen gemischt werden.

MyISAM-Tabellen

MyISAM ist die Standardtabellenart von MySQL und - sofern nicht explizit anders angegeben - per Voreinstellung genutzt. MyISAM-Tabellen haben grundsätzlich keine Größenbeschränkung und gelten besonders bei häufigen `SELECT`- und `INSERT`-Operationen als sehr effizient und schnell. Allerdings sind MyISAM-Tabellen nicht **transaktionssicher** [http://de.wikipedia.org/wiki/Datenbank#Transaktionen] und unterstützen keine **Fremdschlüssel** [http://de.wikipedia.org/wiki/Constraint#Constraints_in_Datenbanksystemen] (sogenannte "Constraints").

InnoDB

Im Gegensatz zu den MyISAM-Tabellen sind InnoDB-Tabellen **transaktionssicher** [http://de.wikipedia.org/wiki/Transaktionssicherheit] und unterstützen Fremdschlüssel. Ist die **Datenkonsistenz** [http://de.wikipedia.org/wiki/Konsistenz_(Datenbank)] sehr wichtig, so wird in der Regel InnoDB als Engine eingesetzt.

Weitere Tabellenarten

Wie oben bereits erwähnt, unterstützt MySQL noch eine Anzahl weitere Tabellenarten, wobei MyISAM und InnoDB sicherlich die am häufigsten eingesetzten sind. Erwähnenswert sind noch Tabellen auf Basis der "Memory-Engine". Diese werden komplett im RAM des Servers gehalten und sind dadurch natürlich entsprechend schnell. Memory-Tabellen werden bevorzugt für temporäre Tabellen eingesetzt oder für Tabellenkopien mit häufigen Zugriffen, auf die nur lesend zugegriffen wird.

MySQL-Server steuern

Wie alle anderen Dienste bringt MySQL-Start/Stop-Skripte mit. Diese Skripte lassen sich natürlich auch zum Kontrollieren des Servers gebrauchen. Aus einem Terminal ^[2] heraus kann man mit

```
sudo service mysql {start|stop|restart|reload|force-reload|status}
```

Beispiel:

```
sudo service mysql reload
```

den Server steuern. Die einzelnen Optionen dabei bedeuten:

- start - Startet den MySQL-Server
- stop - Stoppt den Server
- restart - Startet den Server neu, bestehende Verbindungen auf den Server werden gekappt
- reload - Lädt die Konfigurationsdateien neu, ohne dass Verbindungen getrennt werden
- force-reload - Lädt die Konfigurationsdateien neu, auch wenn dabei Verbindungen verloren gehen
- status - Zeigt Status-Informationen und den Zustand des MySQL-Servers an

Mehr dazu im Wiki unter **Dienste** [https://wiki.ubuntuusers.de/Dienste/].

Achtung!

Für die einwandfreie Funktion von MySQL ist es entscheidend, dass der Daemon stets korrekt heruntergefahren und nicht bspw. bei einem Neustart des Systems abgeschossen wird. Wenn man ihn nur gelegentlich zu Entwicklerzwecken braucht, sollte man deshalb am besten nicht - wie in **Dienste** beschrieben - mit dem Kommando update-rc.d alle Start- und Stop-Links aus den Runleveln entfernen, sondern nur den Startlink aus dem Runlevel 2:

```
sudo rm /etc/rc2.d/S20mysql
```

Dadurch wird der Dienst beim Shutdown immer korrekt heruntergefahren, auch wenn man vergessen hat, dass er noch läuft. Eine Raute # vor dem Eintrag start on in **/etc/init/mysql.conf** verhindert den automatischen Start:

1	#start on ...
---	---------------


Datensicherung


Hat man kritische, und/oder sensible, Daten in der Datenbank, ist es - wie üblich - ratsam, regelmäßig ein Backup der Daten zu machen. Dazu gibt es unterschiedliche Möglichkeiten wie etwa die direkte Kopie der Daten, das Auslesen der Datenbanken und Speichern des Inhalts aber auch den Aufbau von Redundanzen (Replikation). Details zum Thema sind im Artikel **MySQL/Backup** [https://wiki.ubuntuusers.de/MySQL/Backup/] näher beschrieben.

Graphische Oberflächen

Es gibt eine Reihe von graphischen Benutzeroberflächen, die den Umgang mit MySQL erleichtern. Mehr Informationen findet man im Artikel **MySQL/Werkzeuge** [https://wiki.ubuntuusers.de/MySQL/Werkzeuge/].

Links

- **Offizielle Homepage** [http://www.mysql.de/] 

- **MySQL** [http://de.wikipedia.org/wiki/MySQL] - Wikipedia
- **Offizielle Community-Webseite** [http://forge.mysql.com/] 
- **vergessenes Root-Passwort** [http://www.zeroathome.de/wordpress/mysql-root-passwort-vergessen/] zurücksetzen 
- **Pro-Linux-Buchvorstellung: »High Performance MySQL«** [http://www.pro-linux.de/news/2009/14256.html] Optimierung, Backups, Replikation und Lastverteilung (06/2009) 
- **OpenBook Praktischer Einstieg in MySQL mit PHP, 2. Auflage** [http://www.oreilly.de/german/freebooks/einmysql2ger/] 
- **MySQL Workbench** [https://wiki.ubuntuusers.de/MySQL_Workbench/] - Modellierung, Planung und Wartung von (MySQL) Datenbanken

Diese Revision [https://wiki.ubuntuusers.de/MySQL/a/revision/829228/] wurde am 1. August 2015 21:19 von **aasche** erstellt.

Die folgenden Schlagworte wurden dem Artikel zugewiesen: **Netzwerk** [https://wiki.ubuntuusers.de/wiki/tags/Netzwerk/], **Server** [https://wiki.ubuntuusers.de/wiki/tags/Server/], **Internet** [https://wiki.ubuntuusers.de/wiki/tags/Internet/], **Datenbank** [https://wiki.ubuntuusers.de/wiki/tags/Datenbank/]

Inhalte von ubuntuusers.de lizenziert unter Creative Commons, siehe <https://ubuntuusers.de/lizenz/>.