

MySQL Schulung - Zusammenfassung

Marcel Noe

9.10 - 20.10.2006

Kapitel 1

1.1 MySQL Einführung

1.1.1 Einleitung

Bei MySQL handelt es sich um einen sehr skalierbares Datenbank-Management System. MySQL wird oft als Datenbank-Backend auf Webservern verwendet. Sehr beliebt ist eine Kombination aus Linux, Apache, MySQL und PHP - auch als LAMP bezeichnet.

MySQL ist aber nicht nur ein Spielzeug für Webdesigner, sondern bietet mit der InnoDB Storage-Engine alle Features einer „richtigen“ Datenbank (z.B. MVCC-Locking, ACID-Transaktionen, Foreign Keys). Darüber hinaus ist MySQL mit der Option des Clusterings und der Replikation sehr Skalierbar. MySQL kann sowohl zum Verwalten der Playliste auf einem kleinen MP3-Player als auch zur Verwaltung von riesigen Datenbeständen auf Grossrechnern verwendet werden.

1.1.2 Storage Engines

MySQL bietet Unterstützung einer ganze Reihe verschiedener Storage-Engines. Benötigen Sie z.B. keine Transaktionen oder Foreign Keys, dann können Sie die sehr schnelle MyISAM-Engine verwenden. Haben Sie dagegen höhere Anforderungen an die Datenintegrität oder benötigen Sie MVCC-Locking, dann sind Sie mit der InnoDB-Engine sehr gut beraten.

1.1.3 Lizenzierung

MySQL ist unter zwei verschiedenen Lizenzen erhältlich. Die meisten Anwender sind mit der freien Version von MySQL – also der GPL Lizenz – sehr gut Beraten. Wenn Sie allerdings erweiterten Support benötigen oder Teile des MySQL Quellcodes in ihre eigene, kommerzielle Software einbauen möchten, müssen Sie die kommerzielle Version von MySQL kaufen.

1.2 Installation

1.2.1 Arten der Installation

Bei der Installationen haben Sie mehrere Möglichkeiten. Die einfachste Möglichkeit ist sicherlich die Installation der Pakete, die in ihrer Distribution enthalten sind. Für die meisten Anwender ist diese Art der Installation vollkommend ausreichend.

Sollten Sie ein Feature benötigen, das in der Version Ihres Distributors nicht enthalten ist, oder den Support von `MYSQ.L.COM` in Anspruch nehmen, dann müssen Sie die Binärvariante von `MYSQ.L.COM` installieren. Diese Pakete haben ausserdem den Vorteil, dass es eine Variante gibt, die mit dem Intel ICC Compiler übersetzt wurde, der auf Pentium Systemen sehr schnelle Programme erzeugt.

Sollten Sie einmal spezielle Wünsche an Ihre MySQL Installation haben, die selbst von den Binärpakete von `MYSQ.L.COM` nicht erfüllt werden, müssen Sie selbst Hand anlegen und MySQL selbst kompilieren.

1.2.2 Installation unter SuSE Linux über YaST

Unter SUSE Linux ist die Installation von MySQL sehr einfach. Starten Sie zunächst YaST und wählen Sie dort „Software installieren oder Löschen“. Tragen Sie nun in dem Feld links „mysql“ ein. Wählen Sie in der nun erscheinenden Liste die Pakete „mysql“ und „mysql-client“ aus. Klicken Sie nun auf übernehmen.

Nach der Installation werden Sie gefragt, ob Sie weitere Pakete installieren möchten. Beantworten Sie diese Frage mit „Nein“ und verlassen Sie YaST wieder. MySQL sollte nun auf ihrem System installiert sein.

1.2.3 Installation der Binärpakete von MySQL.com

Laden Sie das Binärpaket nach `/tmp` und entpacken Sie es mit dem Befehl `tar -xvzf paket.tgz` - natürlich müssen Sie hier „paket.tgz“ durch den tatsächlichen Namen ihres Paketes ersetzen.

Verschieben Sie das so ausgepacket Verzeichnis nun nach `/opt` und erstellen Sie einen symbolischen Link nach `/opt/mysql`. Linken Sie `/opt/mysql` nach `/usr/local/mysql`. Passen Sie nun die Datei `/etc/my.cnf` an (dazu später mehr). Wenn Sie diese Datei angepasst haben müssen Sie noch das Datenverzeichnis initialisieren. Dies geht am einfachsten mit dem Befehl

```
/opt/mysql/bin/mysql_install_db.
```

1.3 Konfiguration von MySQL

1.3.1 /etc/my.cnf

Die Datei `/etc/my.cnf` ist die zentrale Konfigurationsdatei aller Programme, die MySQL verwenden. Der wichtigste Abschnitt ist sicherlich die Konfiguration des Servers – allerdings kann in dieser Datei auch das Verhalten des Clients gesteuert werden.

Die Datei besteht aus einzelnen Abschnitten. Ein neuer Abschnitt wird durch den Namen des Abschnitts, der in eckige Klammern eingeschlossen ist, eingeleitet. So wird der Konfigurationsabschnitt des Servers beispielsweise mit `[mysqld]` eingeleitet.

Die einzelnen Abschnitte als solches bestehen aus einfachen Wertzuweisungen. Links steht der Name der Variablen, der der Wert zugewiesen soll. Danach folgt ein Gleichheitszeichen. Nach dem Gleichheitszeichen steht der Wert, der der Variablen zugewiesen werden soll. Die Zeile `port = 3306` weist z.B. der Variablen „port“ den Wert „3306“ zu.

1.3.2 Konfigurationsdatei im Homeverzeichnis des Users

Weiterhin besteht die Möglichkeit, dass sich jeder User in seinem Homeverzeichnis eine eigene Konfigurationsdatei speichert. Wenn man sich z.B. regelmässig zu einem bestimmten MySQL Server verbindet, kann man diesen in die `[client]` Sektion seiner eigenen Konfigurationsdatei eintragen.

1.3.3 my.cnf im Datenverzeichnis des MySQL Servers

Auch im Datenverzeichnis des MySQL Servers kann eine `my.cnf` Datei erstellt werden. Dies ist besonders sinnvoll, wenn Sie auf einem Rechner mehrere MySQL Server betreiben möchten. Sie weisen einfach jeden Server ein eigenes Datenverzeichnis vor und konfigurieren jeden Server über die `my.cnf` Datei in dessen Datenverzeichnis individuell.

1.4 Vorlagen

MySQL bringt standardmässig vier Vorlagen für die `my.cnf` Datei mit. Diese Dateien haben die Namen `my-small.cnf`, `my-medium.cnf`, `my-large.cnf` und `my-huge.cnf`. Der Name entspricht der Größe der Maschine, auf dem der MySQL-Server laufen soll. `my-small.cnf` ist so z.B. für den Einsatz auf kleinen Rechnern und `my-huge.cnf` für den Einsatz auf grossen Rechnern gedacht. Diese Dateien können Sie als Ausgangsbasis für Ihre eigene `my.cnf` Datei verwenden.

1.5 Serverstart

1.5.1 Manuelles Starten über den service Befehl

So, nun wird es aber Zeit, ihren MySQL Server das erste mal zu starten. Unter SUSE LINUX geht dies ganz einfach über den `service` Befehl. Geben Sie nun folgendes ein:

```
service mysql start
```

Sollte alles glatt gelaufen sein, sollte nun ein MySQL Server auf ihrem Rechner laufen. Versuchen wir doch mal, ob wir diesen Server pingen können:

```
mysqladmin -u root ping
```

Wenn Sie nun „mysql is alive“ als Ausgabe erhalten haben ist alles glatt gelaufen, und ihr Server läuft. Erhalten Sie stattdessen eine Fehlermeldung, liegt dies wahrscheinlich daran, dass Ihre Datenverzeichnisse nicht korrekt initialisiert wurden. Dies können Sie nachträglich einfach mit dem Befehl

```
mysql_install_db
```

nachholen. Wiederholen Sie nun obige Prozedur und überprüfen Sie, ob Ihr Server nun läuft. Sollte Ihr Server immer noch nicht laufen, dann könnte dies daran liegen, dass auf das Datenverzeichnis (unter SUSE liegt dies Standardmässig unter `/var/lib/mysql`) falsche Zugriffsrechte gesetzt sind. Versuchen Sie dies mit dem Befehl

```
chown -R mysql:mysql /var/lib/mysql
```

zu beheben. Wiederholen Sie nun obige Prozedur. Sollte dies immer noch nicht zum Erfolg führen, ist wohl ein größerer Fehler aufgetreten. Schauen Sie sich in diesem Fall die Datei `/var/lib/mysql/mysql.log` an, um weitere Hinweise zu erhalten.

Sobald Sie den Server gestartet haben können Sie sich testweise mittels des Befehls `mysql` zu diesem Verbinden. Sie sollten nun einen MySQL Prompt (`mysql>`) erhalten. Lassen Sie sich nun erstmal alle Datenbanken anzeigen:

```
mysql> show databases;
```

Dannach können Sie den MySQL Client mit dem Befehl `exit` wieder beenden.

1.5.2 Manuelles Starten über das `mysqld_safe` Skript

Alternativ können Sie MySQL auch über den Befehl `mysqld_safe` starten. Dies ermöglicht es Ihnen, beim Serverstart weitere Optionen anzugeben, wie z.B. ein weiteres Datenverzeichnis. Um dies einmal zu testen müssen Sie den laufenden MySQL Server zunächst einmal beenden:

```
service mysql stop
```

Starten Sie nun den Server mittels `mysqld_safe` neu. Einen so gestarteten Server beenden Sie am besten mittels `mysqladmin shutdown`.

1.5.3 Starten mehrere MySQL Server auf dem gleichen Rechner

Als nächstes versuchen wir mehrere Server mit jeweils eigenen Datenverzeichnissen zu starten. Beenden Sie zunächst alle noch laufenden MySQL Server:

```
service mysql stop; killall mysqld
```

Dannach müssen wir das bestehende Datenverzeichnis kopieren:

```
cp -ar /var/lib/mysql /var/lib/mysql2
```

Nun existiert unter `/var/lib/mysql2` ein weiteres MySQL Datenverzeichnis. Wir wollen nun zwei Datenbankserver so starten, dass der erste Server auf `/var/lib/mysql` und der zweite Server auf `/var/lib/mysql2` zugreift. Ausserdem möchten wir, dass der erste Server über den Socket `/tmp/my1.sock` und der zweite Server über den Socket `/tmp/my2.sock` erreichbar ist. Vom Netzwerk aus soll Server eins über Port 3306 und Server zwei über Port 3307 erreichbar sein.

Starten wir nun den ersten Server: `mysqld_safe --pid-file=/tmp/my1.pid --socket=/tmp/my1.sock --datadir=/var/lib/mysql --port=3306`

Wir sollten nun folgende Ausgabe erhalten: `Starting mysqld daemon with databases from /var/lib/mysql`.

Nun können wir den zweiten Server starten: `mysqld_safe --pid-file=/tmp/my2.pid --socket=/tmp/my2.sock --datadir=/var/lib/mysql2 --port=3307`

und erhalten folgende Ausgabe: `Starting mysqld daemon with databases from /var/lib/mysql2`.

Sie können sich nun mit `mysql -S /tmp/my1.sock` zum ersten und `mysql -S /tmp/my2.sock` zum zweiten Server verbinden. Möchten Sie die Server beenden, so tun Sie dies am einfachsten über `mysqladmin -S /tmp/my1.sock shutdown` und `mysqladmin -S /tmp/my2.sock shutdown`.

1.5.4 Starten des Servers beim Systemboot

Mittels `chkconfig -add mysql` können Sie einrichten, dass ihr MySQL Server beim Booten von SuSE Linux automatisch gestartet wird.

1.6 Einrichten des MySQL Servers

1.6.1 Passwort für den Benutzer Root setzen

Sicherlich ist Ihnen aufgefallen, dass wir bisher nie ein Passwort eingeben mussten. Dies liegt einfach daran, dass für den User „root“ noch gark kein Passwort auf der Datenbank gesetzt ist. Dies wollen wir nun ändern. Grundsätzlich gibt es mehrere Möglichkeiten, um das Passwörter zu setzen – wir gehen an dieser Stelle den Weg über die SQL-Console.

Stellen Sie zunächst sicher, dass Ihr MySQL Server gestartet ist:

```
service mysql start
```

Nun ändern wir das Passwort des Users „root“ auf den Wert „StengGeheim“:

```
mysql -u root
mysql> USE mysql;
mysql> UPDATE user
-> SET password = password('StengGeheim')
-> WHERE user = 'root'
-> ;
Query OK, 2 rows affected (0.00 sec)
Rows matched: 2 Changed: 2 Warnings: 0

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)
```

Der Befehl `FLUSH PRIVILEGES` führt dazu, dass die Berechtigungstabellen von MySQL neu geladen werden. Ihre Passwortänderung wird erst nach Eingabe dieses Befehls aktiv.

Wichtig ist nun noch, dass Sie die Datei `.mysql_history` in ihrem Homeverzeichnis löschen, da diese das obige Passwort im Klartext enthält.

1.6.2 Ein bisschen aufräumen

Zum Schluß räumen wir noch ein bisschen unseren MySQL Server auf und löschen alle Benutzer ohne Passwort:

```
mysql -u root
mysql> USE mysql;
mysql> DELETE * FROM user WHERE password = '';
Query OK, 1 row affected (0.00 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)
```