



Bericht Januar 2006 – Thema MySQL mit Linux
Prof. Witt – Berichtender: Felix Krüger (1806298)
HAW Hamburg – Fakultät DMI – Departement Technik

Inhaltsverzeichnis

1. Einführung.....	2
2. Hilfe.....	2
3. Übersicht verschaffen.....	2
4. Datenbanken anlegen und löschen.....	2
5. Mit Relationen arbeiten.....	3
6. Datentypen.....	3
6.1 Numerische Typen.....	3
6.2 Datum und Zeit.....	4
6.3 Strings.....	5
6.4 Default Values.....	6

1. Einführung

Um in Linux SQL zu benutzen, muss man die Konsole öffnen und `mysql` eintippen. Voraussetzung MySQL ist auf dem Computer richtig installiert, kann man jetzt mit den im Folgenden beschriebenen Befehlen arbeiten.

2. Hilfe

Um in die MySQL-eigene Hilfe zu kommen, kann man den Befehl `help` benutzen. Des weiteren gibt es im Internet jede Menge Seiten, um Syntax-Beispiele zu finden. Die offizielle deutsche MySQL-Seite hat die Domain www.mysql.de.

3. Übersicht verschaffen

Die wichtigsten Befehle im Überblick:

<code>show databases;</code>	Hiermit bekommt man eine Übersicht über die vorhandenen Datenbanken.
<code>use [database];</code>	Die darauf folgenden Befehle beziehen sich alle auf die Datenbank [database].
<code>show tables;</code>	Hiermit bekommt man eine Übersicht über die in der aktuell verwendeten Datenbank vorhandenen Tabellen.
<code>select * from [table];</code>	Gibt den kompletten Inhalt von [table] aus.
<code>tee [logfile]</code>	Erstellt ein Logfile der Sitzung.

4. Datenbanken anlegen und löschen

<code>create [database];</code>	Hiermit legt man eine Datenbank an.
<code>create table [table] ([spalte] [datentyp], ...);</code>	Legt eine Tabelle mit den entsprechenden Spalten und Datentypen an (siehe MySQL Datenarten).
<code>insert into [table] values([inhalt], ...);</code>	Fügt Inhalte in die Tabelle ein. Die Werte müssen mit Datentyp und Dimension der Tabelle übereinstimmen.
<code>delete from [table] where [spalte]=[wert];</code>	Löscht alle Tupel deren Wert in der [spalte] dem im Befehl angegebenen [wert] entspricht.
<code>delete from [table];</code>	Löscht alles

5. Mit Relationen arbeiten

<code>select <u>distinct</u> * from [table];</code>	Gibt alle <u>unterschiedlichen</u> Tupel aus.
<code>select * from [table] order by [spalte, ..] [desc/asc];</code>	Sortiert die Ausgabe nach den angegebenen Spalten in aufsteigender (asc) oder absteigender (desc) Reihenfolge.
<code>select * from [table] where [spalte]=[wert];</code>	Gibt alle Tupel aus, deren Wert in der [spalte] dem im Befehl angegebenen [wert] entspricht.
<code>select [spalte1], [spalte2] from [table1], [table2] where [table1].[spalte3]=[table2].[spalte3] and [spalte1]=[wert];</code>	Verbindet zwei Tabellen, in denen Die angegeben Werte übereinstimmen. In diesem Beispiel werden nur die Spalte1 und Spalte2 ausgegeben. Diese abfrage wird auch Join genannt. Zusätzlich wird die Ausgabe noch danach gefiltert, ob die Spalte1 dem [wert] entspricht.

6. Datentypen

6.1 Numerische Typen

- BIT [(M)]

Bits: M indiziert die Anzahl der Bits

- TINYINT [(M)] [UNSIGNED] [ZEROFILL]

Integer von -128 to 127 (signed) oder von 0 – 255 (unsigned).

- BOOL, BOOLEAN

Enthält entweder *false* oder *true*

- SMALLINT [(M)] [UNSIGNED] [ZEROFILL]

Integer von -32768 - 32767 (signed) oder von 0 – 65535 (unsigned).

- MEDIUMINT [(M)] [UNSIGNED] [ZEROFILL]

Integer von -8388608 – 8388607 (signed) oder von 0 bis 16777215 (unsigned).

- INT [(M)] [UNSIGNED] [ZEROFILL]

Integer von -2147483648 - 2147483647. (signed) oder von 0 - 4294967295 (unsigned).

- INTEGER [(M)] [UNSIGNED] [ZEROFILL]

Synonym für INT.

- BIGINT [(M)] [UNSIGNED] [ZEROFILL]

Integer von -9223372036854775808 - 9223372036854775807 (signed) oder von 0 - 446744073709551615 (unsigned).

- FLOAT [(M, D)] [UNSIGNED] [ZEROFILL]

Float von -3.402823466E+38 bis -1.175494351E-38, 0 und 1.175494351E-38 bis 3.402823466E+38.

M ist die Anzahl der Stellen D sind die Nachkommastellen.

UNSIGNED verbietet negative Zahlen

- DOUBLE [(M, D)] [UNSIGNED] [ZEROFILL]

Float von -1.7976931348623157E+308 bis -2.2250738585072014E-308, 0, und 2.2250738585072014E-308 bis 1.7976931348623157E+308.

- FLOAT (p) [UNSIGNED] [ZEROFILL]

p gibt die Präzision in Bytes an. MySQL wählt danach aus, ob float oder double benutzt wird.

- DECIMAL [(M [, D])] [UNSIGNED] [ZEROFILL]

MySQL 5.0.3 und neuer:

Eine „exakte“ Zahl mit maximal 65 Stellen und 30 Nachkommastellen. Alle Berechnungen in diesem Format werden mit einer Genauigkeit von 65 Stellen erledigt.

Vor MySQL 5.0.3:

Eine „exakte“ Zahl die als CHAR gespeichert wird.

UNSIGNED, if specified, disallows negative values.

- DEC [(M[,D])] [UNSIGNED] [ZEROFILL], NUMERIC [(M[,D])] [UNSIGNED] [ZEROFILL], FIXED [(M[,D])] [UNSIGNED] [ZEROFILL]

Synonym für DECIMAL

6.2 Datum und Zeit

- DATE

Datum von '1000-01-01' bis '9999-12-31' im 'YYYY-MM-DD' Format, erlaubt sind Strings oder auch Zahlen.

- DATETIME

Datum und Zeit von '1000-01-01 00:00:00' bis '9999-12-31 23:59:59' im 'YYYY-MM-DD HH:MM:SS' Format, erlaubt sind Strings oder auch Zahlen.

- TIMESTAMP [(M)]

TIMESTAMP ist wie DATETIME, kann automatisch von MySQL in eine Spalte eingefügt werden.

- TIME

Eine Zeitangabe von '-838:59:59' bis '838:59:59' im 'HH:MM:SS' Format.

- YEAR [(2 | 4)]

Jahres Angabe im 2 bzw. 4 Stellen Format. 2 Stellen: von (19)70 bis 20(69), 4 Stellen: 1901 bis 2155.

6.3 Strings

- [NATIONAL] CHAR(M) [BINARY | ASCII | UNICODE]

Ein String mit $M = 0 - 255$ Stellen. Wird von links mit Leerzeichen ausgefüllt.

- CHAR

Synonym für CHAR (1) .

- [NATIONAL] VARCHAR (M) [BINARY]

Variable Stringlänge mit 0-255 bzw. 0-65535 (MySQL 5.0.3 und höher) Stellen

- BINARY (M)

Wie CHAR, nur dass es Binäre Daten enthält.

- VARBINARY (M)

Wie VARCHAR, nur dass es Binäre Daten enthält.

- TINYBLOB

BLOB mit maximal 255 ($2^8 - 1$) Bytes.

- TINYTEXT

TEXT mit maximal 255 ($2^8 - 1$) Stellen.

- BLOB [(M)]

BLOB mit maximal 65,535 ($2^{16} - 1$) Bytes.

- TEXT [(M)]

TEXT mit maximal 65,535 ($2^{16} - 1$) Stellen.

- MEDIUMBLOB

BLOB mit maximal 16,777,215 ($2^{24} - 1$) Bytes.

- MEDIUMTEXT

TEXT mit maximal 16,777,215 ($2^{24} - 1$) Stellen

- LONGBLOB

BLOB mit maximal 4,294,967,295 or 4GB ($2^{32} - 1$) bytes.

- LONGTEXT

TEXT mit maximal 4,294,967,295 or 4GB ($2^{32} - 1$) Stellen

- ENUM('value1', 'value2', ...)

Eine Aufzählung von maximal 65535 Werten

- SET('value1', 'value2', ...)

Ein Set von maximal 64 Werten, die einer Liste von Werten entsprechen müssen.

6.4 Default Values

Mit `DEFAULT value` kann man einen Standard-Wert für eine Spalte angeben. Diese muss immer eine konstante sein, außer man benutzt `CURRENT_TIMESTAMP` für eine `TIMESTAMP`-Spalte.

Wenn man keinen Default-Wert angibt, ist dieser normalerweise `NULL`. Wenn das nicht geht tritt Folgendes ein:

0 für Numerischen Datentypen bzw. der nächste Wert in der Zahlenfolge in einer `AUTO_INCREMENT`-Spalte

Für `BLOB` und `TEXT` können keine Default Values bestimmt werden.