

Seab@er Software AG

MySql Book



1.	Vorwort	5
2.	Installation	6
2.1	Installation.....	6
2.2	Mysql von Hand starten.....	6
2.3	Mysql automatisch starten.....	7
2.4	Herunterfahren	7
2.5	Variablen ausgeben.....	7
2.6	Datenverzeichnis ändern.....	7
3.	Konfiguration	8
3.1	Konfigurations Dateien	8
3.2	Mysql Prompt.....	8
3.3	Character Set	9
3.3.1	Abfragen	9
3.3.2	Ändern	9
4.	Allgemein.....	10
4.1	Anmelden an Mysql	10
4.2	Angemeldeten User abfragen	10
5.	Benutzerverwaltung.....	11
5.1	Benutzer Anlegen	11
5.2	Benutzer löschen.....	11
5.3	Benutzer umbenennen	11
5.4	Berechtigungen	12
5.4.1	Berechtigungen setzten	12
5.4.2	Berechtigungen entfernen	13
5.5	Berechtigungen anzeigen.....	13
5.6	Password ändern.....	14
6.	Database and more.....	15
6.1	Datenbank	15
6.1.1	Datenbanken anzeigen	15
6.1.2	Datenbank erstellen.....	15
6.1.3	Datenbank löschen	15
6.2	Tabellen	16
6.2.1	Tabellen anzeigen	16
6.2.2	Tabellenstruktur anzeigen.....	16
6.2.2	Tabellen anlegen	17
6.2.3	Tabellen Kommentar.....	19
6.2.4	Tabellen Name ändern	19
6.2.5	Tabellenfeld hinzufügen	19
6.2.6	Tabellenfeld löschen	19
6.2.7	Tabellenfeld ändern.....	19
6.3	Daten / Werte	20
6.3.1	Daten / Werte eingeben.....	20
6.3.2	Select Werte eingeben.....	20
6.3.3	Datum Werte eingeben.....	20
6.3.4	Auto Increment Werte eingeben	20
6.4	Index	21
6.4.1	Erstellen	21
6.4.2	Anzeigen	21
6.5	Abfragen	22
6.5.1	Null-Werte Abfrage	22
6.5.2	Datum / Zeit Formatieren.....	22
6.5.3	Gruppieren	23
6.6	Views	24
6.6.1	Erstellen	24
6.6.2	Anzeigen	24
6.7	Sql Script ausführen	25
7.	Backup & Restore	26
7.1	Backup.....	26
7.1.1	Backup Database.....	26

7.1.2	Backup Tabelle	26
8.	Tools	27
8.1	PHPMyAdmin	27
9.	Verweise / Links	28
10.	Copyright	29

1. Vorwort

Ein Nachschlagewerk über Mysql wollte ich schon längst mal erstellen. Da nun auch Mysql bei unserem Kunden im Einsatz ist, war die Zeit gekommen, das Nachschlagewerk zu erstellen.

Alle Konfigurationen und Befehle wurden mit der Mysql Version 5.5.25a unter OpenSuSE 12.2 ausprobiert.

Bei Fragen und Anregungen bin ich unter folgender Mail Adresse zu erreichen:

uwe@seabaer-ag.de



2. Installation

2.1 Installation

Wurde Mysql nicht mithilfe von RPM-Pakete installiert, so müssen folgende Schritte gemacht werden.

```
uws@tux>mkdir /var/log/mysql
uws@tux>chown mysql:mysql /var/log/mysql

uws@tux>mysql_install_db --user=mysql # DB installieren
uws@tux>mysqld_safe --user=mysql & # Mysql Server starten
uws@tux>mysqladmin -u root password '<newPasswd>' # root Kennwort setzten

uws@tux>mysqladmin version -u root -p
Enter password:
Mysqladmin Ver 8.42 Distrib 5.5.25a, for Linux on x86_64
Copyright © 2000, 2011, Oracle and/or ist affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or ist
Affiliates. Other names may be trademarks of their respective owners.

Server Version      5.5.25a-log
Protocol version    10
Connection          Localhost via UNIX socket
UNIX socket         /var/run/mysql/mysql.sock
Uptime              2 min 23 sec

uws@tux>mysqlshow -u root -p # Anzeige Databases
Enter password:
+-----+
|      Databases      |
+-----+
| information_schema |
| mysql              |
| performance_schema |
| test               |
+-----+
```

2.2 Mysql von Hand starten

Mysql kann per Hand mit dem Befehl `/etc/init.d/mysql start` gestartet werden. Mit der Option `stop` wird die Datenbank heruntergefahren und mit `restart` erfolgt nach dem herunterfahren ein Neustart.

```
root@tux>/etc/init.d/mysql start
```

```
uws@tux>mysqld_safe --user=mysql--log &
```

2.3 Mysql automatisch starten

Damit Mysql automatisch bei jedem hochfahren des Rechners startet, wird mit `insserv` ein Dienst erstellt.

```
root@tux>cd /etc/rc.d
root@tux>insserv mysql
```

2.4 Herunterfahren

```
uws@tux>mysqladmin -u root -p shutdown
```

2.5 Variablen ausgeben

```
uws@tux>mysqladmin variables -u root -p
Enter password:
+-----+-----+
| Variable_name          | Value |
+-----+-----+
| auto_increment_increment | 1     |
| auto_increment_offset  | 1     |
| autocommit             | ON    |
| automatic_sp_privileges | ON    |
.
.
| wait_timeout           | 28800 |
+-----+-----+
```

2.6 Datenverzeichnis ändern

Standardmäßig werden die Datenbanken in dem Verzeichnis `/var/lib/mysql` abgelegt. Nach dem anlegen des neuen Verzeichnisses, wird dieses in der Konfigurationsdatei `/etc/my.cnf` in der Variable `datadir` eingetragen. Vorher ist die Datenbank herunterzufahren. Vor dem hochfahren der Datenbank muss sie mit dem Befehl `mysql_install_db` initialisiert werden.

```
uws@tux>mkdir /mysql/database
uws@tux>chown -R mysql:root /mysql
uws@tux>chmod -R 700 /mysql
uws@tux>mysqladmin -u root -p shutdown

uws@tux>vi /etc/my.cnf
#datadir = /var/lib/mysql
datadir = /mysql/database

uws@tux>mysql_install_db --user=mysql
uws@tux>/etc/rc.d/mysql start
```

3. Konfiguration

3.1 Konfigurations Dateien

Im Verzeichnis `/etc` gibt es die Datei `my.cnf`. In dieser Datei wird mit der Variable `datadir` der Pfad angegeben, wo die Datenbanken abgelegt werden. In dieser Datei wird die globale Konfiguration für Mysql vorgenommen.

Für jeden Benutzer kann man im Home Verzeichnis auch eine `.my.cnf` anlegen und diese wird bei dem starten von Mysql ausgewertet.

3.2 Mysql Prompt

Den Prompt in Mysql kann auf verschiedene Weise gesetzt werden. In einer Mysql Session geschieht das mit `\R <definition>`. Nach dem Verlassen der Session wird wieder der ursprüngliche Prompt gesetzt.

```
mysql>\R Mysql:\u@d>
PROMPT set to 'Mysql:\u@d>'
Mysql:uws@db1>
```

Ein globales setzen des Prompts erfolgt in der Datei `my.cnf` im Verzeichnis `/etc`. Hier wird im Abschnitt `[mysql]` die Variable `prompt` definiert. Hierbei ist zu achten, dass die Backslashes demaskiert werden müssen.

```
prompt='Mysql:\\u@d>'
```

Benutzer können die Variable `MYSQL_PS1` definieren und anschließend laden.

```
uws@tux>cat .zshrc
export MYSQL_PS1="Mysql:\u@d>"

uws@tux>cat .my.cnf
[mysql]
prompt='Mysql:\\u@d>'
```

Für die Konfiguration stehen folgende Optionen bereit.

<u>Option</u>	<u>Beschreibung</u>
<code>\v</code>	Serverversion
<code>\d</code>	Datenbankname
<code>\h</code>	Serverhost Name
<code>\p</code>	Aktuelle IP Adresse oder Socket Datei
<code>\u</code>	Aktueller Benutzername
<code>\U</code>	Vollständiger Benutzername incl. @hostname
<code>\R</code>	Aktuelle Uhrzeit in 24 Stunden Format
<code>\r</code>	Aktuelle Uhrzeit im 12 Stunden Format
<code>\m</code>	Aktuelle Minute
<code>\y</code>	Zweistelliges Aktuelles Jahr
<code>\Y</code>	Vierstelliges Aktuelles Jahr
<code>\D</code>	Aktuelles vollständiges Datum
<code>\s</code>	Sekunden
<code>\w</code>	Aktueller Wochentag in drei Buchstaben
<code>\P</code>	AM/PM
<code>\o</code>	Aktueller Monat numerisch
<code>\O</code>	Aktueller Monat Buchstaben

\c Zähler, der bei jeder Anweisung hochgezählt wird.
 \S Semikolon
 \` Einfaches Anführungszeichen

3.3 Character Set

Von dem Character Set ist es abhängig, ob man Umlaute oder Sonderzeichen in einer Datenbank speichern kann oder nicht. Vorhandene Character Set kann man sich mit dem folgenden Befehl anzeigen lassen.

```
mysql>show character set;
+-----+-----+-----+-----+
|Charset|Description          |Default collation|Maxlen |
+-----+-----+-----+-----+
|big5   |Big5 Traditional Chinese | big5_chinese_ci | 2     |
|dec8   |Dec8 West European      | dec8_swedish_ci | 1     |
|       |                         |                 |       |
|       |                         |                 |       |
|       |                         |                 |       |
```

In der ~/.my.cnf kann man den Character Set global für alle Datenbank setzen.

```
uws@tux>cat my.cnf
default character set = utf8
```

3.3.1 Abfragen

Mit welchem Character Set die Datenbank erstellt worden ist, kann man mit dem folgenden Befehl abfragen.

```
mysql>show create database <dbname>;
+-----+-----+-----+-----+
| Database | Create Database
+-----+-----+-----+-----+
| MysqlInfo | CREATE DATABASE `MysqlInfo` /*!40100 DEFAULT CHARACTER SET
utf8 */
```

Mit show create table <table_name> kann man sich anzeigen lassen, mit welchem Character Set die Tabelle erstellt worden ist.

3.3.2 Ändern

Character Sets können für Datenbanken, Tabellen und Spalten abgeändert werden.

```
mysql>alter database <dbname> charset = utf8;
mysql>alter table <table_name> charset = utf8;
mysql>alter table <table_name> column <col_name> charset = utf8;
```

4. Allgemein

4.1 Anmelden an Mysql

Nachdem die Benutzer mit einem Kennwort eingerichtet worden sind, muss man Mysql mit dem Parameter `-p` aufrufen.

```
uws@tux>mysql -p
Enter Password:
mysql>
```

Möchte man sich direkt an einer Datenbank anmelden, so gibt man einfach den Namen der Datenbank mit an.

```
uws@tux>mysql -p <dbname>
Enter Password:
mysql>
```

Möchte man sich an die Datenbank mit einem anderen Benutzernamen anmelden, so geschieht dieses mit der Option `-u`.

```
uws@tux>mysql -u <dbuser> -p -h <dbserver> <dbname>
```

4.2 Angemeldeten User abfragen

Den angemeldeten Benutzer kann man mit einem `select` statement abfragen.

```
mysql>select current_user();
+-----+
| current_user() |
+-----+
| root@localhost |
+-----+
1 row in set (0.00 sec)
```

5. Benutzerverwaltung

5.1 Benutzer Anlegen

Ein neuer Mysql User wird mit `create user` angelegt. Jeder neuer Benutzer wird in die Tabelle `mysql.user` eingetragen. Es können auch in einer Anweisung mehrere Benutzer angelegt werden, die durch ein Komma von einander getrennt sind, angegeben werden.

```
mysql>create user '<dbuser>'@'<host>' identified by '<passwd>';
```

Wird als Host `%` angegeben, so kann man sich von allen Rechner `remote` an Mysql anmelden. Möchte man sich auch von dem `localen` Computer an Mysql anmelden, so muss zusätzlich noch der Mysql User mit `localhost` als Host angelegt werden.

5.2 Benutzer löschen

Einen Benutzer kann man mit `drop user` aus der Datenbank löschen. Auch hier lassen sich mehrere Benutzer in einer Anweisung löschen, wenn man sie durch ein Komma getrennt, angibt.

```
mysql>drop user '<dbuser>'@'<host>';
```

5.3 Benutzer umbenennen

Mit `rename user` kann man einen Datenbank Benutzer umbenennen. Mehrere Benutzer kann man auf einmal umbenennen, indem man sie mit einem Komma getrennt angibt.

```
mysql>rename user '<dbuser>'@'<host>' to '<dbuser>'@'<host>';
```

5.4 Berechtigungen

5.4.1 Berechtigungen setzen

Berechtigungen werden mit dem Befehl `grant` gesetzt. Die Angabe `identified by` ist optional. Die Syntax lautet:

```
grant priv_type [(column_list)], priv_type [(column_list)]
on [object_type] {tbl_name | * | *.* | dbname.*}
to user [identified by [password] <passwd>]
[require none | [{SSL|X509}] [cipher 'cipher' [and]] [issuer 'issuer'
[and]] [subject 'subject']]
with with_option [with_option] ...

object_type = table | function | procedure

with_option = grant option |
              max_queries_per_hour count |
              max_updates_per_hour count |
              max_connections_per_hour count |
              max_user_connections count
```

Berechtigungen können auf folgenden Ebenen gewährt werden.

Globale Ebene	Globale Berechtigungen gelten für alle Datenbanken (<code>grant all *.*</code>).
Datenbankebene	Berechtigungen gelten nur für die angegebene Datenbank (<code>grant all on db_name</code>).
Tabellenebene	Tabellenberechtigungen gelten für alle angegebenen Tabellen (<code>grant all on db_name.table_name</code>).
Spaltenebene	Berechtigungen gelten nur für die einzelnen Spalten einer Tabelle.
Routineebene	Die Berechtigungen gelten für die Routinen (Functions und Procedure).

Folgende `priv_types` gibt es:

<u>Berechtigung</u>	<u>Beschreibung</u>
ALL [PRIVILEGES]	Setzt alle Berechtigungen außer die GRANT OPTION.
ALTER	Die Verwendung von ALTER TABLE ist erlaubt.
ALTER ROUTINE	Ändern oder Löschen von gespeicherten Routinen.
CREATE	Die Verwendung von CREATE TABLE ist erlaubt.
CREATE ROUTINE	Erstellen von Routinen.
CREATE TEMPORARY TABLES	Das erstellen von temporären Tabellen ist mit CREATE TEMPORARY TABLE erlaubt
CREATE USER	Das anlegen / ändern und löschen eines Benutzers ist erlaubt. CREATE USER, DROP USER, RENAME USER, REVOKE ALL PRIVILEGES.
CREATE VIEW	Das anlegen von Views, CREATE VIEW, ist erlaubt.
DELETE	Das Anwenden des Befehls DELETE ist erlaubt.
DROP	DROP TABLE ist erlaubt.
EVENT	Das erstellen von Ereignissen für den Ereignisplaner ist erlaubt.
EXECUTE	Das ausführen von Routinen ist erlaubt.
FILE	Die Verwendung von SELECT ... INTO OUTFILE und LOAD DATA INFILE ist erlaubt.
INDEX	Das anlegen von Indexen, CREATE INDEX und DROP INDEX, ist erlaubt.
INSERT	Erlaubt das einfügen von Werten (INSERT).
LOCK TABLES	Die Verwendung von LOCK TABLES wird für alle Tabellen erlaubt, die die Berechtigung SELECT haben.

PROCESS	Die Verwendung von <code>SHOW FULL PROCESSLIST</code> wird erlaubt.
REFERENCES	Nicht implementiert.
RELOAD	Die Verwendung von <code>FLUSH</code> wird erlaubt.
REPLICATION CLIENT	Die Abfrage der Positionen von Slave- oder Master-Servern ist erlaubt.
REPLICATION SLAVE	Für Replikationsslaves erforderlich.
SELECT	Erlaubt <code>SELECT</code> Abfragen.
SHOW DATABASES	Das Anzeigen aller Datenbank ist mit <code>SHOW DATABASES</code> erlaubt.
SHOW VIEW	Das Anzeigen von Views mit <code>SHOW CREATE VIEW</code> ist erlaubt.
SHUTDOWN	Die Verwendung von <code>MYSQLADMIN SHUTDOWN</code> ist erlaubt.
SUPER	Die Verwendung der Anweisungen <code>CHANGE MASTER</code> , <code>KILL</code> , <code>PURGE MASTER LOGS</code> , <code>SET GLOBAL</code> und <code>MYSQLADMIN DEBUG</code> ist erlaubt. Ferner wird die einmalige Verbindung für den Fall erlaubt, dass die <code>max_connections</code> ereicht wurde.
TRIGGER	Das Erstellen und Löschen von Triggern ist erlaubt.
UPDATE	Der Befehl <code>UPDATE</code> ist erlaubt.
USAGE	Synonym für keine Berechtigungen.
GRANT OPTION	Die Gewährung von Berechtigungen wird erlaubt.

Nach dem setzen von Berechtigungen müssen die Berechtigungen einmal neu eingelesen werden. Dieses geschieht mit dem Befehl `flush privileges`.

In dem nachfolgenden Beispiel werden alle Privilegien für die Datenbank `my_db` gesetzt.

```
mysql>grant all privileges on my_db.* to '<dbuser>'@'<host>' [identified by '<passwd>'];
mysql>flush privileges;
```

5.4.2 Berechtigungen entfernen

Berechtigungen können mit `revoke` dem Datenbank Benutzer wieder entzogen werden. Die Syntax hierzu ist:

```
revoke priv_type [(column_list)] [, priv_type [(column_list)]
on [object_type] {tbl_name | * | *.* | db_name.*}
from user [, user] ...
```

In dem Bespielt werden all Berechtigungen dem Benutzer für die Datenbank `my_db` entzogen.

```
mysql>revoke all privileges on my_db.* to '<dbuser>'@'<host>';
mysql>flush privileges;
```

5.5 Berechtigungen anzeigen

Berechtigungen können mit `show grants` angezeigt werden. Die Angabe `for` ist optional, ohne die Angabe werden die Grants für den angemeldeten Benutzer angezeigt.

```
mysql>show grants for '<dbuser>'@'<host>';
+-----+-----+-----+-----+
| Grants for uws@localhost |
+-----+-----+-----+-----+
| grant usage on *.*to 'uws'@'localhost' identified by password 'hfdhp90' |
| grant all privileges on 'my_db.*' to 'uws'@'localhost' |
+-----+-----+-----+-----+
```

Globale Berechtigungen werden in der Tabelle `mysql.user` gespeichert.

```
mysql>select user, host, select_priv, insert_priv from mysql.user;
+-----+-----+-----+-----+
| user | host      | select_priv | insert_priv |
+-----+-----+-----+-----+
| root | localhost | Y           | Y           |
| root | 172.0.0.1 | Y           | Y           |
| uws  | localhost | Y           | N           |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Berechtigungen auf Datenbank Ebene werden in den Tabellen `mysql.db` und `mysql.host` abgespeichert.

```
mysql>grant all on <dbname>.* to '<dbuser>';
```

Tabellen Berechtigungen werden in der Tabelle `mysql.tables_priv` gespeichert.

```
mysql>
```

Berechtigungen für die Spalten einer Tabelle werden in `mysql.columns_priv` gespeichert.

```
mysql>
```

Die Routinen Berechtigungen werden in der Tabelle `mysql.procs_priv` gespeichert.

```
mysql>
```

5.6 Password ändern

Das Kennwort wird in Mysql mit `set password` geändert. Das eigene Passwort kann man ohne die Angabe von `for` setzen.

```
mysql>set password = password('<new_passwd>');
mysql>flush privileges;
mysql>set password for '<dbuser>'@'<host>' = password('<new_passwd>');
mysql>flush privileges;
```

Anstelle von `set password for` kann man auch den Befehl `update` nehmen.

```
mysql>update mysql.user set password=password('<new_passwd>')
  where user='<dbuser>' and host='<host>';
mysql>flush privileges;
```

6. Database and more

6.1 Datenbank

6.1.1 Datenbanken anzeigen

Die vorhandenen Datenbanken kann man sich mit dem Befehl `show databases` anzeigen lassen.

```
mysql>show databases;
+-----+
| Database           |
+-----+
| Information_schema |
| mysql              |
| test               |
+-----+
3 rows in set (0.00 sec)
```

Mit `mysqlshow` kann man sich auch die Datenbanken anzeigen lassen.

```
uws@tux>mysqlshow -u root -p
Enter Password:
+-----+
| Database           |
+-----+
| Information_schema |
| mysql              |
| test               |
+-----+
```

6.1.2 Datenbank erstellen

Eine Datenbank wird mit dem Befehl `create database` erstellt.

```
mysql>create database <dbname>
Query OK, 1 row affected (0.00 sec)
```

6.1.3 Datenbank löschen

Mit `drop database` wird eine vorhandene Datenbank gelöscht.

```
mysql>drop database <dbname>
Query OK, 0 rows affected (0.00 sec)
```

6.2 Tabellen

6.2.1 Tabellen anzeigen

Vorhandene Tabellen werden mit dem Befehl `show tables` angezeigt. Auch definierte Views werden hier ausgegeben.

```
mysql>show tables;
+-----+
| Tables_in_test |
+-----+
| employees      |
| countries      |
| state          |
+-----+
3 rows in set (0.00 sec)
```

Die Tabellen lassen sich auch mit `mysqlshow` abfragen.

```
uws@tux>mysqlshow <dbname> -u <dbuser> -p
Enter Password:
+-----+
| Tables_in_test |
+-----+
| employees      |
| countries      |
| state          |
+-----+
```

6.2.2 Tabellenstruktur anzeigen

Die Tabellenstruktur kann man sich mit `describe` oder `show create table` sich anzeigen lassen.

```
mysql>describe <table_name>;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id    | int(11)       | NO   | PRI | NULL    |       |
| city  | varchar(15)  | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

```
mysql>show create table <table_name>;
+-----+-----+-----+-----+-----+-----+
| Table | Create Table
+-----+-----+-----+-----+-----+-----+
| city  | CREATE TABLE `city` (
  `id` int(11) NOT NULL,
  `city` varchar(15) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=Test DEFAULT CHARSET=utf8 |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.01 sec)
```


6.2.2 Tabellen anlegen

Eine neue Tabelle wird mit dem Befehl `create table` angelegt. Die Syntax hierfür lautet:

```
create table <table_name> (<name1> <typel> <attribut1>, <name2>, <type2>);
```

```
mysql>create table contact(
-> id int not null auto_increment primary key,
-> vorname char(30),
-> nachname char(30),
-> city char(50),
-> alter int(2),
-> size double(3,2) # Groesse in m
->);
```

Folgende Datentypen gibt es:

Zeichenketten

<u>Typ</u>	<u>Beschreibung</u>	<u>Wertebereich</u>	<u>Note</u>
Char	Feste Länge	255 Zeichen	Keine Speicherung führende Leerzeichen
Varchar	Variable Länge	255 Zeichen	Keine Speicherung führende Leerzeichen
Text	Große Länge	65.535 Zeichen	Speichert auch führende Leerzeichen
Tinytext		255 Zeichen	Speichert auch führende Leerzeichen
Mediumtext		16.777.215 Zeichen	Speichert auch führende Leerzeichen
Longtext		4.294.967.295 Zeichen	Speichert auch führende Leerzeichen

Zahlentypen

<u>Typ</u>	<u>Beschreibung</u>	<u>Wertebereich</u>	<u>Wertebereich +-</u>
Int	Ganze Zahl	0 – 4.294.967.295	-2.147.483.648 ... 2.147.483.647
Float	Gebrochene Zahl	Wertabhängig	
Double	Gebrochene Zahl	Wertabhängig	
Decimal	Gebrochene Zahl	Wertabhängig	
Tinyint	Ganze Zahl	0 – 255	-128 ... 127
Smallint	Ganze Zahl	0 – 65.535	-32.768 ... 32.767
Bigint	Ganze Zahl	0 – 2(64)	

Genauigkeit für Gebrochene Zahlen

<u>Fließkommazahl</u>	<u>Genauigkeit</u>	<u>Bereich</u>
12,3	2	1
359,2	3	1
936.524,73592	6	5
3,1411591654	1	9

Zeit und Datumstypen

<u>Typ</u>	<u>Beschreibung</u>	<u>Format</u>
Date	Datum	'JJJJ-MM-TT'
Time	Zeit	'HH:MM:SS'
Datetime	Datum & Zeit	'JJJJ-MM-TT HH:MM:SS'
Year	Jahr, vierstellig	'JJJJ'
Timestamp	Zeitstempel	'JJJJMMTTHHMMSS'

Binärtypen

<u>Typ</u>	<u>Beschreibung</u>
Tinyblob	255 Bytes
Blob	65.535 Bytes
Mediumblob	16.777.215 Bytes
Longblob	4.294.967.295 Bytes

Aufzählungstypen

<u>Typ</u>	<u>Beschreibung</u>
Enum	Speichert einen Wert aus einer definierten Liste
Set	Speichert mehrere Werte aus einer definierten Liste

Nach den Datentypen können noch dem Feld Attribute mitgegeben werden. Nachfolgend ist eine Liste, in der eine Auswahl der zu Verfügung stehenden Attribute beschrieben ist.

<u>Attribut</u>	<u>Beschreibung</u>
Unsigned	
Null/not null	Bei <i>not null</i> muss ein Wert eingegeben werden.
Default	Das Feld wird mit einem Standardwert gefüllt.
Binary	
Zero fill	
Unique	Der eingegeben Wert gibt es nur ein mal.
Auto_increment	Automatisches Hochzählen.
Primary key	
Index	

6.2.3 Tabellen Kommentar

Einer Tabellen kann man ein Kommentar mit der Option `comment` mitgeben. Um sich einen Kommentar anzeigen zu lassen, dafür wird der Befehl `show create table` benutzt.

```
mysql>alter table <tableName> comment 'Version 1.0';
Query OK, 0 rows affected (0.01 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql>show create table <tableName>;
```

6.2.4 Tabellen Name ändern

Eine Tabelle kann man mit `rename` einen neuen Namen vergeben.

```
mysql>alter table <table_name> rename to <new_table_name>;
Query OK, 0 rows affected (0.02 sec)
```

6.2.5 Tabellenfeld hinzufügen

Einer vorhandenen Tabelle kann man mit `add column` ein neues Tabellenfeld hinzugefügt werden.

```
mysql>alter table <table_name> add column <feld_name> <feld_attribute>;
Query OK, 0 rows affected (0.02 sec)
```

6.2.6 Tabellenfeld löschen

Ein Tabellenfeld kann mit `drop column` aus einer Tabelle löschen.

```
mysql>alter table <table_name> drop column <feld_name>;
Query OK, 0 rows affected (0.02 sec)
```

6.2.7 Tabellenfeld ändern

Mit `modify column` kann man die Eigenschaften eines Datenbank Feldes ändern.

```
mysql>alter table <table_name> modify column <feld_name> <feld_type>
-> not null;
Query OK, 0 rows affected (0.02 sec)
```

```
mysql>alter table <table_name> change <column_name> <new_column_name>
-> <feld_type>;
Query OK, 24 rows affected (0.09 sec)
Records: 24 Duplicates: 0 Warnings: 0
```

6.3 Daten / Werte

6.3.1 Daten / Werte eingeben

Werte werden mit dem `insert into` Befehl in der Tabelle eingegeben. Dieses kann auf zwei Arten erfolgen.

```
mysql>insert into contact (vorname, nachname) values ('Hans','Im Glück');
Query OK, 1 row affected (0.00 sec)

mysql>insert into contact set vorname='Harry', nachname='Hirsch';
Query OK, 1 row affected (0.00 sec)
```

6.3.2 Select Werte eingeben

Möchte man Werte aus einer anderen Tabelle einfügen, so geschieht das mit einem `select` Statement auf die andere Tabelle.

```
mysql>insert into <table_name> (<field1>, <field2>, <field3>)
->select 'text1','text2', id from <table_name> where <bedingung>;
```

6.3.3 Datum Werte eingeben

Ein Datum wird mit `yyyy-mm-dd` in einer Tabelle eingefügt. Das aktuelle Datum wird mit `curdate` in der Tabelle geschrieben.

```
mysql>insert into <table_name> values (1,'2012-05-23');
Query OK, 1 row affected (0.00 sec)

mysql>insert into <table_name> values (2, curdate());
Query OK, 1 row affected (0.00 sec)
```

6.3.4 Auto Increment Werte eingeben

Ist in einer Tabelle ein Feld definiert, das per `auto_increment` hochgezählt wird, so gibt man `null` als Wert an.

```
mysql>insert into <table_name> values
->(null,'Hilden'),
->(null,'Ratingen '),
->(null,'Essen ');
Query OK, 3 row affected (0.00 sec)
Records: 3 Duplicates: 0 Warnings:0
```

6.4 Index

6.4.1 Erstellen

Einen Index auf einem Tabellenfeld wird mit `create index` angelegt.

```
mysql>create [unique, fulltext] index incident_n on incident (incno(10));  
Query OK, 0 rows affected (0.04 sec)  
Record: 0 Duplicates: 0 Warnings: 0
```

6.4.2 Anzeigen

Erstellte Indexe kann man sich mit `show index` anzeigen lassen..

```
mysql>show index from <table_name>;
```

6.5 Abfragen

6.5.1 Null-Werte Abfrage

Möchte man sich nur die Daten anzeigen lassen, die keine Werte enthalten, so wird hierzu die Bedingung `is [not] null` genommen. .

```
mysql>select * from contact where city is null;
+----+-----+-----+-----+
| id | vorname | nachname | city |
+----+-----+-----+-----+
| 1 | Hans | Im Glück | NULL |
| 2 | Harry | Hirsch | NULL |
| 3 | Erwin | Lottermann | NULL |
+----+-----+-----+-----+
3 rows in set (0.00 sec)
```

6.5.2 Datum / Zeit Formatieren

Das Datum wird in Mysql im Format `yyyy-mm-dd` eingegeben und abgespeichert. Möchte man eine andere Ausgabe erhalten, so erfolgt dieses mit einer `date_format` Abfrage.

```
mysql>select id, date_format(MailDate, '%d.%m.%y') MailDatum
->from tracking_inc;
+----+-----+
| ID | MailDatum |
+----+-----+
| 1 | 02.09.2011 |
| 2 | 03.09.2011 |
+----+-----+
2 rows in set (0.00 sec)
```

Folgende Formate stehen bereit:

<u>Format:</u>	<u>Beschreibung:</u>
%d	Datum zweistellig
%D	Datum englisch (2nd, 16th)
%m	Monat zweistellig
%M	Monat ausgeschrieben
%y	Jahr vierstellig
%Y	Jahr zweistellig
%H	Stunde
%i	Minute
%S	Sekunde

6.5.3 Gruppieren

Eine Abfrage wird mit `group by` definiert.

```
mysql>select sum(intWert) '€'. intYear from t_uws where strName = 'uws'
->group by intYear;
+-----+-----+
| € | intYear |
+-----+-----+
| 20 | 2012 |
| 30 | 2013 |
+-----+-----+
2 rows in set (0.00 sec)
```

6.6 Views

6.6.1 Erstellen

Ein View wird mit dem Befehl `create view` erstellt.

```
mysql>create view <view_name> as
->select a.incno, a.owner, a.description,
->date_format(b.sendmail, '%d.%m.%y'), b.grund
->from incident a, mail b where
->a.closed is null and b.incident_id = a.id
->order by a.incno, b.grund;
```

6.6.2 Anzeigen

Views werden wie Tabellen verwaltet, deshalb werden sie auch bei einem `show tables` aufgelistet. Möchte man das `select` statement angezeigt bekommen, so wird hierzu der Befehl `show create table` genommen.

```
mysql>show create table <table_name>;
```


6.7 Sql Script ausführen

Ein Sql Script kann innerhalb von Mysql mit dem Befehl `source` ausgeführt werden.

```
mysql>source <sqlscript>
```

Ausserhalb von Mysql wird das Sql Script mit `<` eingelesen.

```
uws@tux>mysql -u <db_user> -p <dbname> < meinscript.sql  
Enter Password:
```

7. Backup & Restore

7.1 Backup

7.1.1 Backup Database

Um eine Datenbank zu sichern, gibt es den Befehl `mysqldump`. Man kann einzelne Datenbanken oder auch alles in einer Sql-Datei schreiben lassen.

```
uws@tux>mysqldump -p --databases <db1> > backupdb1.sql
Enter Password:

uws@tux>mysqldump -p -all-databases > complete.sql
Enter Password:
```

7.1.2 Backup Tabelle

Das sichern von Tabellen in einer Datenbank geschieht mit dem Befehl `backup table`. Hierbei werden die DB-Files in ein angegebenes Verzeichnis kopiert. Die MYD Datei enthält die Daten der Tabelle, die FRM Datei enthält die Struktur der Datenbank. Den Index-Baum wird in der Datei MZI gespeichert, diese braucht wiederum nicht gesichert zu werden, da sie durch die zwei vorherigen Dateien neu erstellt werden kann.

```
mysql>backup table <tableName> to '/var/mysql/backup/test';
```

8. Tools

8.1 PHPMysqlAdmin

Kommt bei der Anmeldung an den Mysql Server folgende Fehlermeldung: Error #1045 Access denied, so muss in der Datei config.inc.php die Zeile `$cfg['servers'][$i]['auth_type']='cookie'` suchen und den Wert `coockie` durch `config` ersetzen. Sind die nachfolgenden zwei Zeilen auskommentiert (`//`), so Sind diese zu entfernen.

9. Verweise / Links

Mysql

<http://www.mysql.de>

10. Copyright

Dieses Dokument ist urheberrechtlich geschützt. Das Copyright liegt bei Uwe Schimanski.

Das Dokument darf gemäß der GNU *General Public License* verbreitet werden. Insbesondere bedeutet dieses, daß der Text sowohl über elektronische wie auch physikalische Medien ohne die Zahlung von Lizenzgebühren verbreitet werden darf, solange dieser Copyright Hinweis nicht entfernt wird.