

MySQL

Was bieten Datenbanken?

- Zentralisation von Daten
- Maschinenunterstützte Weiterverarbeitung
- Daten werden vielen Benutzern gleichzeitig zur Verfügung gestellt
- Ausschalten von konkurrierenden Zugriffen durch „Locking“
- Zentrale Zugriffsberechtigungen
- Zugriffseffizienz im Sinne von Geschwindigkeit
- Verknüpfung von Datentabellen unterschiedlicher Datenbanken

Was ist MySQL?

MySQL ist ein Datenbanksystem. Es besteht aus einem zentralen Server und aus (mehreren) Clients. Es benutzt einen Dialekt der Sprache SQL.

Was ist SQL?

SQL ist die De-facto-Standardsprache, mit der sich Daten aus relationalen Datenbanken abrufen und dort manipulieren lassen. Die Bezeichnung SQL sollte man nicht auf die Goldwaage legen. Während das S für Structured (Strukturiert) und L für Language (Sprache) ohne weiteres klar sind, stellt das Q für Query (Abfrage) einen Kompromiss dar. Man könnte annehmen, daß sich SQL ausschließlich auf die Abfrage von Datenbanken bezieht. Es lassen sich aber nicht nur Abfragen ausführen, sondern auch:

- Tabellen erzeugen,
- Daten hinzufügen,
- Daten löschen,
- Daten zusammenführen,
- Aktionen aufgrund von Änderungen an der Datenbank auslösen und
- Abfragen innerhalb eines Programms oder der Datenbank speichern.

Query steht also stellvertretend für eine ganze Reihe von Funktionen.

Relationale Datenbank:

Diese Methode hat mehrere Vorteile und viele Nachteile. Positiv ist die Tatsache, dass die physikalische Struktur der Daten auf einem Datenträger nicht von Bedeutung ist. Der Zugriff auf die Daten erfolgt einfach über Zeiger, die auf die nächste Speicherstelle verweisen. Daten lassen sich leicht hinzufügen und löschen. Nachteilig ist, daß man unterschiedliche Informationsgruppen nicht einfach verknüpfen kann, um neue Informationseinheiten zu bilden. Nachdem die Datenbank erstellt wurde, läßt sich das Format der Daten auf dem Datenträger nicht nach Belieben ändern. Bei Änderungen am Format müßte man eine neue Datenbankstruktur erzeugen.

Wie gliedert sich die MySQL-Syntax?

Die Bool'schen Operatoren werden ausgeschrieben:

and, or, not

Patzhalteoperatoren sind:

_ für ein Zeichen.

% für eine Zeichenfolge.

Alle Befehle außer use werden mit Semikolon bestätigt.

Aktivierung:

Der Use-Befehl:

```
mysql> use DATENBANK1
```

Dieser Befehl aktiviert Datenbank1.

Ausgabe:

Der Show-Befehl:

```
mysql> show databases;
```

zeigt alle vorhandenen Datenbanken an.

```
mysql> show tables;
```

zeigt die in der aktiven Datenbank vorhandenen Tabellen an.

```
mysql> show columns from TABELLE1;
```

zeigt die Spalten der TABELLE1 an.

Der Describe-Befehl:

```
mysql> describe TABELLENNAME;
```

zeigt die Struktur der Tabelle an.

Der Select-From Befehl:

```
mysql> select * from TABELLE1;
```

zeigt alle in TABELLE1 enthaltenen Spalten und alle Einträge an.

```
mysql> select SPALTE1 from TABELLE1;
```

zeigt alle in SPALTE1 enthaltenen Einträge von TABELLE1 an.

```
mysql> select SPALTE1,SPALTE2 from TABELLE1;
```

zeigt alle in SPALTE1 und SPALTE2 enthaltenen Einträge von TABELLE1 an.

```
mysql> select TABELLE1.SPALTE1, TABELLE2.SPALTE1 from  
TABELLE1,TABELLE2;
```

zeigt alle Einträge von SPALTE1 der TABELLE1 und von SPALTE1 der TABELLE2 an.

Der Select-From-Befehl mit Where-Erweiterung:

```
mysql> select * from TABELLE1 where SPALTE1="WERT1";
```

zeigt alle Zeilen von TABELLE1 an, in denen der Wert von SPALTE1 dem WERT1 entspricht.

```
mysql> select SPALTE1,SPALTE3 from TABELLE1 where SPALTE2="WERT2";
```

zeigt die SPALTE1 und SPALTE3 derjenigen Zeilen an, in denen der Wert von SPALTE2 dem WERT2 entspricht.

```
mysql> select TABELLE1.SPALTE1,TABELLE1.SPALTE2,TABELLE2.SPALTE9 from  
TABELLE1,TABELLE2 where TABELLE1.SPALTE1=TABELLE2.SPALTE1 and  
TABELLE2.SPALTE9="WERT5";
```

zeigt SPALTE1 und SPALTE2 der TABELLE1 sowie SPALTE9 der TABELLE2 derjenigen Zeilen an, in denen SPALTE1 der TABELLE1 den gleichen Wert wie SPALTE1 der TABELLE2 und in denen der Wert von SPALTE9 der TABELLE2 WERT5 entspricht.

Ergänzung mit Where Like-Befehl:

```
mysql> select * from HAUSTIER where name like "MI%";  
findet Minka, Mieze und andere Tiere mit Mi- beginnend.
```

```
mysql> select * from HAUSTIER where name like "%fisch";  
findet Goldfisch, Silberfisch und andere Tiere, die mit -fisch aufhören.
```

```
mysql> select * from HAUSTIER where name like "G_ldfisch";  
findet Goldfisch, Geldfisch, etc.
```

Ergänzung mit Order By-Befehl:

Zeilen sortieren aufsteigend:

```
mysql> select GEWÜNSCHTE_SPALTEN from TABELLE order by SPALTENNAME;  
mysql> select NAME,BESITZER,RASSE from HAUSTIER order by BESITZER;
```

Zeilen sortieren absteigend:

```
mysql> select GEWÜNSCHTE_SPALTEN from TABELLE order by SPALTENNAME  
desc;  
mysql> select NAME,BESITZER,RASSE from HAUSTIER order by BESITZER  
desc;
```

Zeilen nach mehreren Spalten hierarchisch sortieren

```
mysql> select GEWÜNSCHTE_SPALTEN from TABELLE order by  
SPALTENNAME1,SPALTENNAME2;  
mysql> select NAME,BESITZER,RASSE from HAUSTIER order by  
BESITZER,NAME;
```

Erstellung:

Der Create-Befehl:

```
mysql> create database DATENBANK1;  
erstellt DATENBANK1
```

```
mysql> create table TABELLE1 (SPALTE1 OPTION1,SPALTE2 OPTION2,SPALTE3  
OPTION3);
```

erstellt TABELLE1 mit SPALTE1, SPALTE2 und SPALTE3 in der aktiven Datenbank. Einige Optionen der Spalten sind:

int	in Form einer Zahl von -2147483648 bis 2147483647
date	in der Form von 1999-08-06
time	in der Form von 23:59:59
year	in der Form von 1999
char(N)	wobei N eine Zahl zwischen 1 und 255 ist
varchar(N)	wobei N eine Zahl zwischen 1 und 255 ist (ist aber variabel)

Der Alter-Befehl:

Name der Tabelle ändern:

```
mysql> alter table NAME1 rename NAME2;  
mysql> alter table HAUSTIER rename ALLEHAUSTIERE;
```

Typ einer Spalte ändern:

```
mysql> alter table TABELLENNAME modify SPALTENNAME TYP_NEU;  
mysql> alter table ALLEHAUSTIERE modify BESITZER varchar(50);
```

Umbenennen einer Spalte:

```
mysql> alter table TABELLENNAME change NAME_ALT NAME_NEU TYP_NEU;
```

```
mysql> alter table ALLEHAUSTIERE modify BESITZER HERRCHEN
```

```
varchar(50);
```

hier ist eine gleichzeitige änderung des typs möglich.

Spalte hinzufügen:

```
mysql> alter table TABELLENNAME add SPALTE_NEU TYP_NEU;
```

```
mysql> alter table ALLEHAUSTIERE add LIEBLINGSFUTTER varchar(10);
```

spalte löschen:

```
mysql> alter table TABELLENNAME drop column SPALTE_ALT;
```

```
mysql> alter table ALLEHAUSTIERE drop column LIEBLINGSFUTTER;
```



Der Drop-Befehl:

```
mysql> drop database DATENBANK1;
```

löscht DATENBANK1.

```
mysql> drop table TABELLE1;
```

löscht TABELLE1.

Eingabe:

Manuell:

Der Insert-Befehl:

```
mysql> insert into TABELLE1 values ('WERT1','WERT2','WERT3');
```

fügt TABELLE1 eine Zeile mit den Werten WERT1 in der ersten, WERT2 in der zweiten und WERT3 in der dritten Spalte hinzu.

Der Update-Befehl:

```
mysql> update table set WERT1=WERT2 where SPALTE1="WERT5";
```

verändert den Wert von WERT1 in WERT2 in den Zeilen, in der der Wert von SPALTE1 WERT5 entspricht.

Der Delete-Befehl:

```
Mysql> delete * from TABELLE1;
```

Löscht den gesamten Inhalt der TABELLE1.

```
Mysql> delete * from TABELLE1 where SPALTE1="WERT1";
```

Löscht die Zeilen von TABELLE1, in denen der Wert von SPALTE1 WERT1 entspricht.

Automatisch:

Der Load-Data-Local-Infile-Befehl:

```
mysql> load data local infile "DATEI.TXT" into table TABELLE1;
```

läd die Daten der Datei DATEI.TXT in die TABELLE1. Im Standard sollte jede Zeile der ASCII-Datei einer Tabellenzeile entsprechen und die Spaltenbrüche durch TABULATOREN gekennzeichnet sein.

Sonstiges:

Mathematische Operatoren:

= > < != <= >=

Ausgabe in Datei:

```
mysql> select * from TABELLE1 into outfile "DATEI1.TXT";
```

schreibt alle Werte der TABELLE1 in die Datei DATEI1.TXT.

Wie komme ich auf dem Server zur Mysql-Eingabeaufforderung?

ssh -l benutzername servername
 passwort
 mysql -u benutzername
 use DATENBANKNAME

Eine kleine Übung:

Erstellen Sie folgende Tabellen in einer neuen Datenbank mit Ihrem Namen.

Haustier1

<i>Name</i>	<i>Besitzer</i>	<i>Rasse</i>	<i>Geschlecht</i>	<i>Geburtstag</i>
Minka	Susanne	Katze	F	1998-05-08
Boxer
Fischli	Susanne
...				

Haustier2

<i>Name</i>	<i>Nächster Tierarzt- besuch am:</i>	<i>Letzte Impfung</i>	<i>Lieblingsfutter</i>	<i>Unterhaltskosten in DM pro Jahr</i>
Minka	1999-12-06	1999-01-04	Mäuse	200
Boxer				
Fischli				
...				

Suchen Sie jeweils die Haustiere eines Besitzers (z. B. Susannes Minka und Fischli) über eine Suchanfrage an die Datenbank.

Suchen Sie alle weiblichen Katzen in der Datenbank. Ergänzen Sie die Ausgabe um deren Geburtstag.

Suchen Sie alle Tiere, die weniger als 500DM im Jahr Unterhalt kosten. Ergänzen Sie die Ausgabe um deren Geburtstag, Besitzer und Lieblingsfutter (in dieser Reihenfolge).

Variante:

Quellenbelege epistemische Modalverben im Frühneuhochdeutschen (oder anderer Name)

<i>Verfasser</i>	<i>Werk</i>	<i>Modalverb</i>	<i>Kontext</i>	<i>Jahr</i>

Frühneuhochdeutsche Bibliographie (oder anderer Name)

<i>Verfasser</i>	<i>Werk</i>	<i>Ausgabe</i>	<i>Herausgeber</i>	<i>Signatur UB</i>
...
...
...

Suchen Sie alle epistemischen Modalverben, in deren Kontext "goetz"(oder sonstiges) vorkommt.

Suchen Sie alle Verfasser, die im Jahr 1540 ein Werk mit Vorkommen von epistemischen Modalverben geschrieben haben. Suchen Sie hierzu die Signatur der jeweiligen Werke in der UB.

Suchen Sie alle Autoren, deren Modalverben beinhaltenden Werke von Helgard Ulmschneider herausgegeben wurden. Suchen Sie zu diesen Autoren die Werke und Jahreszahlen.