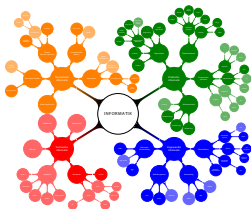


# Kapitel 36

## SQL – Modellierung

Wir bauen eine Datenbank

Vorlesung Einführung in die Informatik 1 vom 20. Mai 2014 von Till Tantau



## Lernziele von Kapitel 36

1. Das Konzept der Datenbank-Shell kennen
2. Die MySQL-Shell benutzen können
3. E/R-Diagramme in eine relationale Datenbank umwandeln können
4. SQL-Datentypen kennen

# Gliederung von Kapitel 36

## 36.1 Einführung zu SQL

36.1.1 SQL zur Kommunikation

36.1.2 Die MySQL-Shell

36.1.3 Erste Schritte

## 36.2 Erstellen einer Datenbank

36.2.1 Vom E/R-Modell zur Datenbank

36.2.2 Anlegen einer Datenbank

36.2.3 Anlegen von Tabellen

36.2.4 Löschen

## 36.1 Einführung zu SQL

- SQL zur Kommunikation  
Die MySQL-Shell  
Erste Schritte

## 36.2 Erstellen einer Datenbank

- Vom E/R-Modell zur Datenbank
- Anlegen einer Datenbank
- Anlegen von Tabellen
- Löschen

- SQL steht für *structured query language*, deren Syntax »angeblich menschenlesbar« ist.
- Sie erfüllt drei Hauptfunktionen:
  1. Sie stellt Befehle zur Verfügung, um Relationen und Datenbanken (Ansammlungen von Relationen) zu erstellen und zu verwalten.
  2. Sie stellt Befehle zur Verfügung, um Einträge in Tabellen *einzufragen* und *zu löschen*.
  3. Sie erlaubt es, *Anfragen* zu formulieren wie »Wer ist der Vater von Dolly?«
- Die Sprache ist *deklarativ*. Das bedeutet, dass man bei Anfragen angibt, was man gerne hätte, aber nicht, wie man das berechnen sollte.

## 36.1 Einführung zu SQL

- SQL zur Kommunikation  
Die MySQL-Shell  
Erste Schritte

## 36.2 Erstellen einer Datenbank

- Vom E/R-Modell zur Datenbank
- Anlegen einer Datenbank
- Anlegen von Tabellen
- Löschen

- Datenbanken werden von einem *Datenbanksystem* verwaltet.
- Das Programm, das die Verwaltung durchführt, heißt auch manchmal *Datenbankserver*.
- Um mit dem Datenbankserver zu reden, wird eine *Datenbank-Shell* benutzt. (Zur Erinnerung: Eine Shell ist ein Programm, mit dem man dialogbasiert mit einem anderen Programm spricht.)
- In der Datenbank-Shell gibt man SQL-Befehle ein, die an den Datenbankserver übermittelt werden. Die Antworten des Servers zeigt die Shell dann an.

- ▶ Die *Ensembl-Datenbank* ist eine große Datenbank, die vielfältige molekularbiologische Informationen speichert.
- ▶ Der Ensembl-Server ist ein Computer irgendwo auf diesem Globus, auf dem das *MySQL-Server-Programm* läuft und auf dessen Festplatten die Daten lagern.
- ▶ Auf einem beliebigen Computer (zum Beispiel einem im Rechnerpool) kann man nun die Datenbankshell `mysql` starten und die Adresse des Ensembl-Servers angeben.
- ▶ Daraufhin hat man (nur lesenden) Zugriff auf alle Tabellen der Ensembl-Datenbanken.
- ▶ Genauer verwaltet der Ensembl-Server viele Datenbanken und jede enthält wiederum viele Tabellen. Als Nutzer muss man am Anfang eine dieser Datenbanken auswählen, man arbeitet dann nur noch mit den Tabellen dieser einen Datenbank.

## 36.1 Einführung zu SQL

- ▶ SQL zur Kommunikation  
Die MySQL-Shell  
Erste Schritte

## 36.2 Erstellen einer Datenbank

- Vom E/R-Modell zur Datenbank
- Anlegen einer Datenbank
- Anlegen von Tabellen
- Löschen

- ▶ Das Programm `mysql` dient dazu, eine Verbindung zu einem Datenbankserver aufzubauen.
- ▶ Es nimmt viele optionale Parameter, die man sich mittels `mysql --help` alle anzeigen lassen kann.
- ▶ Auf den *Prompt* hin kann man nun SQL-Befehle eingeben.
  - ▶ Nützlich sind dort die Hoch- und Runterpfeiltasten, um die letzten eingegebenen Befehle nochmal zu betrachten.
  - ▶ Befehle sollte man mit einem Semikolon (wie in Java) abschließen.

Die wichtigsten Optionen sind:

- `-host=` Dieser Option sollte der Name des Servers folgen (bei Ensembl beispielsweise `ensemldb.ensembl.org`).
- `-user=` Dieser Option sollte der Name des Benutzers folgen, auf dessen Account man sich einloggen möchte (bei Ensembl beispielsweise `anonymous`).

## 36.1 Einführung zu SQL

SQL zur Kommunikation

- ▶ Die MySQL-Shell  
Erste Schritte

## 36.2 Erstellen einer Datenbank

Vom E/R-Modell zur Datenbank

Anlegen einer Datenbank  
Anlegen von Tabellen  
Löschen

# Beispiele für den Verbindungsaufbau mit zwei Datenbanken.

## Verbindung mit Ensembl aufbauen

```
murmel:~ tantau$ mysql \  
--host=ensemldb.ensembl.org --user=anonymous  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
mysql>
```

## Verbindung mit einem Datenbankserver der Uni

```
murmel:~ tantau$ mysql \  
--host=theogate.tcs.uni-luebeck.de \  
--user=tantau --password  
Enter password: *****  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
mysql>
```

Die Backslashes an den Zeilenenden bedeuten, dass es dort eigentlich *keinen* Zeilenumbruch geben darf (der Text hat aber einfach nicht auf eine Zeile gepasst).

### 36.1 Einführung zu SQL

SQL zur Kommunikation

- Die MySQL-Shell
- Erste Schritte

### 36.2 Erstellen einer Datenbank

Vom E/R-Modell zur Datenbank

- Anlegen einer Datenbank
- Anlegen von Tabellen
- Löschen



# Erste nützliche Befehle.

## Auswahl der Datenbank.

Wenn man zum ersten Mal mit einem Datenbankserver verbunden ist, dann sind folgende Befehle nützlich:

- ▶ Der `status`; Befehl gibt jede Menge (meist unverständliche) Informationen über den aktuellen Zustand der Verbindung mit dem Server.
- ▶ Der `show databases`; Befehl liefert eine Liste aller Datenbanken, die der Server verwaltet. Ein Server kann also mehrere Datenbanken verwalten, die ihrerseits viele Tabellen enthalten.
- ▶ Der `use database`; macht eine Datenbank zur aktiven Datenbank. Alle weiteren Befehle beziehen sich auf diese Datenbank.  
Dabei ist `database` der Name einer der Datenbanken, die bei `show databases` aufgelistet werden.

### 36.1 Einführung zu SQL

SQL zur Kommunikation

Die MySQL-Shell

▶ Erste Schritte

### 36.2 Erstellen einer Datenbank

Vom E/R-Modell zur Datenbank

Anlegen einer Datenbank

Anlegen von Tabellen

Löschen

# Erste nützliche Befehle.

## Beschreibung der Tabellen.

### Kapitel 36 SQL – Modellierung

#### 36.1 Einführung zu SQL

SQL zur Kommunikation

Die MySQL-Shell

► Erste Schritte

#### 36.2 Erstellen einer Datenbank

Vom E/R-Modell zur  
Datenbank

Anlegen einer Datenbank

Anlegen von Tabellen

Löschen

Hat man mittels `use database;` eine Datenbank auf dem Server ausgewählt, dann sind folgende Befehle weiter nützlich:

- Der `show tables;` Befehl liefert eine Liste aller Tabelle, die in der Datenbank vorhanden sind.
- Der `describe table;` liefert eine Beschreibung der Spalten (Attribute) der Tabelle `table`.

# Erste nützliche Befehle.

## Anzeigen einer Tabellen.

Ist man an einer bestimmten Tabelle interessiert, dann kann man sich ihren Inhalt mit folgendem Befehl anschauen:

- ▶ `select count(*) from table;` zeigt an, wie viele Zeilen die Tabelle `table` hat.
- ▶ `select * from table;` zeigt den gesamten Inhalt der Tabelle an.
- ▶ `select * from table limit 10;` zeigt nur die ersten 10 Zeilen der Tabelle an.

Der `select`-Befehl kann noch *viel mehr*, ihm sind noch zwei Kapitel gewidmet.

### 36.1 Einführung zu SQL

SQL zur Kommunikation

Die MySQL-Shell

▶ Erste Schritte

### 36.2 Erstellen einer Datenbank

Vom E/R-Modell zur Datenbank

Anlegen einer Datenbank

Anlegen von Tabellen

Löschen

# Wiederholung: Entitätstabellen enthalten Entitäten mit ihren Attributen.

- ▶ Für jeden Entitätstyp gibt es in der Datenbank eine *Tabelle*.
- ▶ Diese enthält für jede Entität eine *Zeile*.
- ▶ Die *Spalten* sind die Attribute der Entität.
- ▶ Ein Spalte, anhand derer man die Entität eindeutig identifizieren kann, heißt *Schlüsselattribut* oder einfach nur *Schlüssel*.

## Beispiel

Name	Farbe	Fell
Dolly	weiß	lockig
Max	schwarz	wuschelig
Peter	schwarz	wuschelig
Flauschi	beige	glatt

### 36.1 Einführung zu SQL

SQL zur Kommunikation  
Die MySQL-Shell  
Erste Schritte

### 36.2 Erstellen einer Datenbank

- ▶ Vom E/R-Modell zur Datenbank
- Anlegen einer Datenbank
- Anlegen von Tabellen
- Löschen

# Wiederholung: Relationstabelle enthalten Tupel von Entitätsschlüssel.

## 36.1 Einführung zu SQL

SQL zur Kommunikation  
Die MySQL-Shell  
Erste Schritte

## 36.2 Erstellen einer Datenbank

► Vom E/R-Modell zur Datenbank  
Anlegen einer Datenbank  
Anlegen von Tabellen  
Löschen

- Für jeden Relationstyp gibt es in der Datenbank wieder eine *Tabelle*.
- Diese enthält für jede Relationship eine *Zeile*.
- Die *Spalten* sind die Schlüssel der beteiligten Entitätstypen.

## Beispiel

Die Tabelle des Relationstyps »haben«.

Schafs-Name	Gen-Name
Dolly	Asthma-Gen
Dolly	Intelligenz-Gen
Max	Asthma-Gen
Peter	Intelligenz-Gen

# Wie legt man eine Datenbank in SQL an?

## 36.1 Einführung zu SQL

SQL zur Kommunikation  
Die MySQL-Shell  
Erste Schritte

## 36.2 Erstellen einer Datenbank

Vom E/R-Modell zur  
Datenbank

- Anlegen einer Datenbank
- Anlegen von Tabellen
- Löschen

- Zunächst baut man eine Verbindung zum Datenbankserver auf.
- Nun benutzt man aber nicht den `use`-Befehl, um eine Datenbank auszuwählen, sondern den `create database`-Befehl.
- Dann kann man beginnen, für jeden Entitätstyp und für jeden Relationshiptyp eine Tabelle anzulegen.

```
murmel:~ tantau$ mysql \
    --host=theogate.tcs.uni-luebeck.de --user=tantau
Welcome to the MySQL monitor.  Commands end with ; or \g.
mysql> create database molecular_sheep;
mysql> use molecular_sheep;
Database changed
mysql>
```

# Wie legt man Tabellen an?

## 36.1 Einführung zu SQL

SQL zur Kommunikation  
Die MySQL-Shell  
Erste Schritte

## 36.2 Erstellen einer Datenbank

Vom E/R-Modell zur  
Datenbank  
Anlegen einer Datenbank  
► Anlegen von Tabellen  
Löschen

In der aktuellen Datenbank kann man eine Tabelle wie folgt anlegen:

```
mysql> create table schaf (  
    name char(20) primary key,  
    farbe char(20),  
    fell char(20)  
);  
  
mysql> create table stall (  
    nummer integer primary key,  
    groesse float  
);  
  
mysql> create table liegt_in (  
    name char(20),  
    nummer integer  
);
```

- ▶ Dem `create table`-Befehl folgt der Name der Tabelle.
- ▶ Danach kommen in Klammern die Attribute der Tabelle.
- ▶ Jedem Attribut *folgt* der Typ des Attributs (anders herum als in Java).
- ▶ Die wichtigsten erlaubten Typen sind:
  - ▶ `tinyint` ist ein 8-Bit Integer
  - ▶ `int` ist ein 32-Bit Integer
  - ▶ `bigint` ist ein 64-Bit Integer
  - ▶ `double` ist eine 64-Bit Gleitkommzahl
  - ▶ `time`, `date` Zeit und Datum
  - ▶ `char` ( $n$ ) ist ein String der Länge  $n \leq 255$
  - ▶ `varchar` ( $n$ ) ist ein String der Länge maximal  $n \leq 255$
  - ▶ `text` ist ein String der Länge maximal 65535
  - ▶ `longtext` ist ein String der Länge maximal 4GB.
  - ▶ `blob` (binary large object) ist ein (rotes blubberndes) Ding der Größe maximal 65536 Byte.
  - ▶ `longblob` ist ein Ding der Größe maximal 4GB.

- 36.1 Einführung zu SQL
  - SQL zur Kommunikation
  - Die MySQL-Shell
  - Erste Schritte
- 36.2 Erstellen einer Datenbank
  - Vom E/R-Modell zur Datenbank
  - Anlegen einer Datenbank
  - ▶ Anlegen von Tabellen
  - Löschen



# Wie fügt man etwas in eine Tabelle ein?

## 36.1 Einführung zu SQL

SQL zur Kommunikation  
Die MySQL-Shell  
Erste Schritte

## 36.2 Erstellen einer Datenbank

Vom E/R-Modell zur  
Datenbank  
Anlegen einer Datenbank  
► Anlegen von Tabellen  
Löschen

Man kann Werte in eine Tabelle wie folgt einfügen:

```
mysql> insert into schaf
      values ("Dolly","weiss", "lockig"),
             ("Flauschi", "schwarz","wuschelig"),
             ("Peter",  "schwarz","wuschelig");

mysql> insert into stall
      values (1,50.0),
             (5,25.0);

mysql> insert into liegt_in
      values ("Dolly", 1),
             ("Flauschi", 5);
```

# Wie löscht man Dinge aus einer Datenbank?

## 36.1 Einführung zu SQL

SQL zur Kommunikation  
Die MySQL-Shell  
Erste Schritte

## 36.2 Erstellen einer Datenbank

Vom E/R-Modell zur  
Datenbank  
Anlegen einer Datenbank  
Anlegen von Tabellen  
► Löschen

- Um die ganze Datenbank zu löschen, schreibt man

```
mysql> drop database molecular_sheep;
```

- Um eine Tabelle zu löschen, schreibt man

```
mysql> drop table liegt_in;
```

- Um einen Eintrag aus einer Tabelle zu löschen, schreibt man:

```
mysql> delete from schaf  
        where name = "Dolly" and farbe =  
        "weiss";
```

Wir werden später noch sehen, dass dies hier nur ein Spezialfall ist. Man kann allgemein Anfragen benutzen zum Löschen.

1. Eine *Datenbank-Shell* wie *mysql* dient dazu, eine Kommunikation mit einem Datenbank-Server herzustellen.
2. Die Sprache *SQL* dient dazu, Datenbank zu verwalten und Anfragen zu formulieren.
3. Datenbanken und Tabellen kann man mit den Befehlen *create*, *insert*, *drop* und *delete* verwalten.

## 36.1 Einführung zu SQL

SQL zur Kommunikation  
Die MySQL-Shell  
Erste Schritte

## 36.2 Erstellen einer Datenbank

Vom E/R-Modell zur  
Datenbank  
Anlegen einer Datenbank  
Anlegen von Tabellen  
► Löschen

36-19



H. P. Gumm, M. Sommer.

*Einführung in die Informatik*, Oldenbourg Verlag, 2004.

Kapitel 10.3