

Einführung in SQL und MySQL

Kursleiter: David Niegisch

Inhalte:

- Einführung in relationale Datenbanken
- SQL Grundlagen
- Erstellung von Beispiel-Datenbanken
- Hinweise zur Administration von Datenbanken



Einführung in SQL und MySQL

Sämtliche Kursmaterialien befinden sich online, unter:

<http://yahg.net/kurse/>

Kontakt zum Kursleiter:

niegisch@yahg.net



Einführung in SQL und MySQL

Was ist Information? Wie entsteht Wissen? Und was haben Daten damit zu tun?

Daten sind isolierte und uninterpretierte Fakten und Kennwerte der Realitätsbeschreibung.

Bspw: Name = Meier, Straße = Hauptstraße 12, Ort = 72076 Tübingen

Informationen sind verknüpfte und mit Bedeutung versehene Daten.

Bspw: Herr Meier wohnt in der Hauptstraße 12 in 72076 Tübingen.

Von **Wissen** spricht man erst, wenn diese Informationen individuell verarbeitet sind und einen mehr oder weniger starken Bezug zur eigenen Erfahrungswelt erhalten haben.

Bspw: Mein ehemaliger Schulfreund Peter Meier wohnt auch hier in Tübingen!



Einführung in SQL und MySQL

Daten sind der Ursprung allen Wissens

Daten müssen gespeichert und Zugriffsmöglichkeiten müssen geschaffen werden.

Dies geschieht mit Hilfe von **Datenbanken** und **Datenbank-Management-Systemen (DBMS)**.



Einführung in SQL und MySQL

Unterscheidung Datenbank und DBMS

Eine **Datenbank** ist eine logische Einheit zusammengehörender Daten und sämtlicher Meta-Informationen dazu.

Bspw: Eine Access-MDB/ACCDB-Datei oder eine MySQL-Datenbank.

Ein **Datenbank-Management-System (DBMS)** stellt unterschiedliche Werkzeuge bereit, mit welchen eine oder mehrere Datenbanken erstellt, mit Daten gefüllt und verwaltet werden können.

Bspw: Die Access-Installation auf einem Arbeitsplatzrechner, oder die MySQL-Installation auf einem Server.



Einführung in SQL und MySQL

Unterscheidung Datenbank und DBMS

Ohne das zugehörige Datenbank-Management-System sind die Daten einer Datenbank unverständlich und nicht verwendbar.

Das Datenbank-Management-System stellt die eigentliche Rahmenarchitektur zur Erstellung und Pflege von Datenbanken dieses Typs mit ihren speziellen Inhalten dar und bietet datenbankübergreifend zusätzliche Serviceleistungen an.

In diesem Kurs werden wir MySQL als DBMS verwenden um eigene Datenbanken zu erstellen und zu pflegen.

Einführung in SQL und MySQL

Desktop- und serverbasierte DBMS

Desktop-basierte DBMS kennzeichnen sich dadurch aus, dass der Betriebssystem-Prozess für die Datenbank nur dann aktiv sein muss, wenn ein Nutzer auf die Datenbank zugreift.



Jeder Nutzer erhält eine Kopie der gesamten Daten für die Zeit seiner Sitzung und kann nach Belieben Inhalte oder sogar Strukturen ändern und löschen.

Bspw: Eine Access-Datenbank auf einem freigegebenen Netzlaufwerk.

Einführung in SQL und MySQL

Beispiel für ein desktopbasiertes DBMS:

Auf dem Arbeitsplatzrechner ist MS Access als DBMS installiert.

Mit Hilfe von MS Access hat der Mitarbeiter Zugriff auf die Datenbank „Kunden.accdb“.

Während einer Sitzung hat der Mitarbeiter somit uneingeschränkten Zugriff auf alle Daten, die sich in dieser Datenbank befinden.



Einführung in SQL und MySQL

Desktop- und serverbasierte DBMS

Im Gegensatz dazu läuft bei serverbasierten DBMS kontinuierlich mindestens ein Serverprozess für die Datenbank. Dieser Prozess stellt die einzige Schnittstelle zur Verfügung, über die die Nutzer auf die Daten der Datenbank(en) zugreifen können. Nur wenn die Nutzer ihre Befehle syntaktisch korrekt formulieren und die Berechtigung für einen solchen Zugriff vorliegt, wird die Befehlsfolge ausgeführt und gegebenenfalls Daten an den Nutzer zurückgesandt.

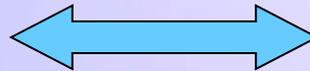
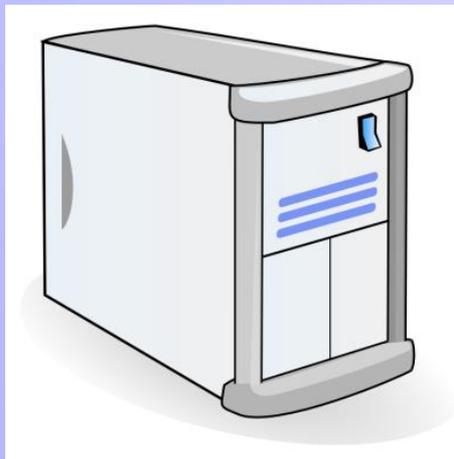
Bspw: Ein MySQL-Server.



Einführung in SQL und MySQL

Beispiel für ein serverbasiertes DBMS:

Auf dem Server ist MySQL als DBMS installiert.



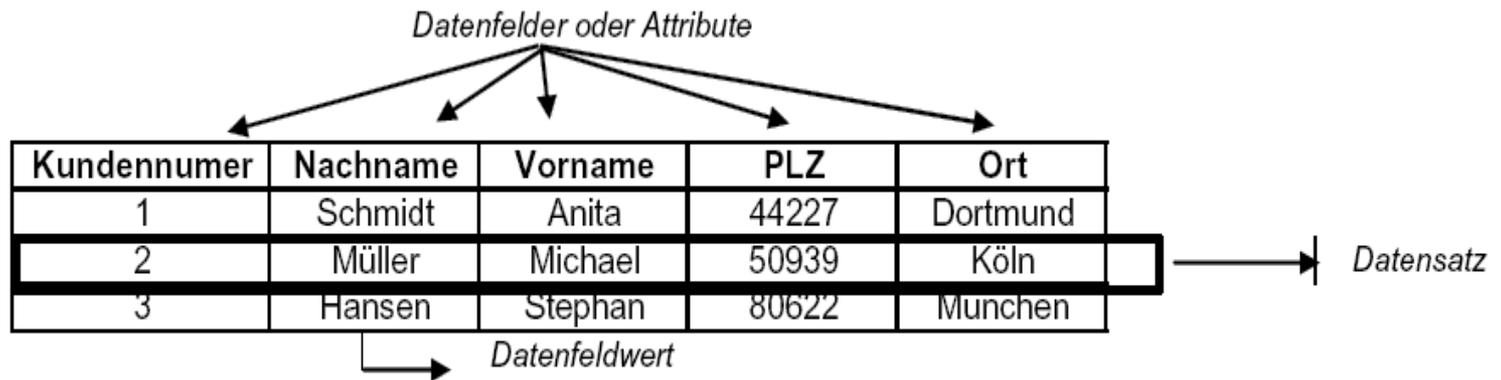
Die Arbeitsplatzrechner kommunizieren mit dem MySQL-Server und der Server entscheidet, ob und welche Daten zurückgesendet werden.



Einführung in SQL und MySQL

Aufbau einer Datenbank

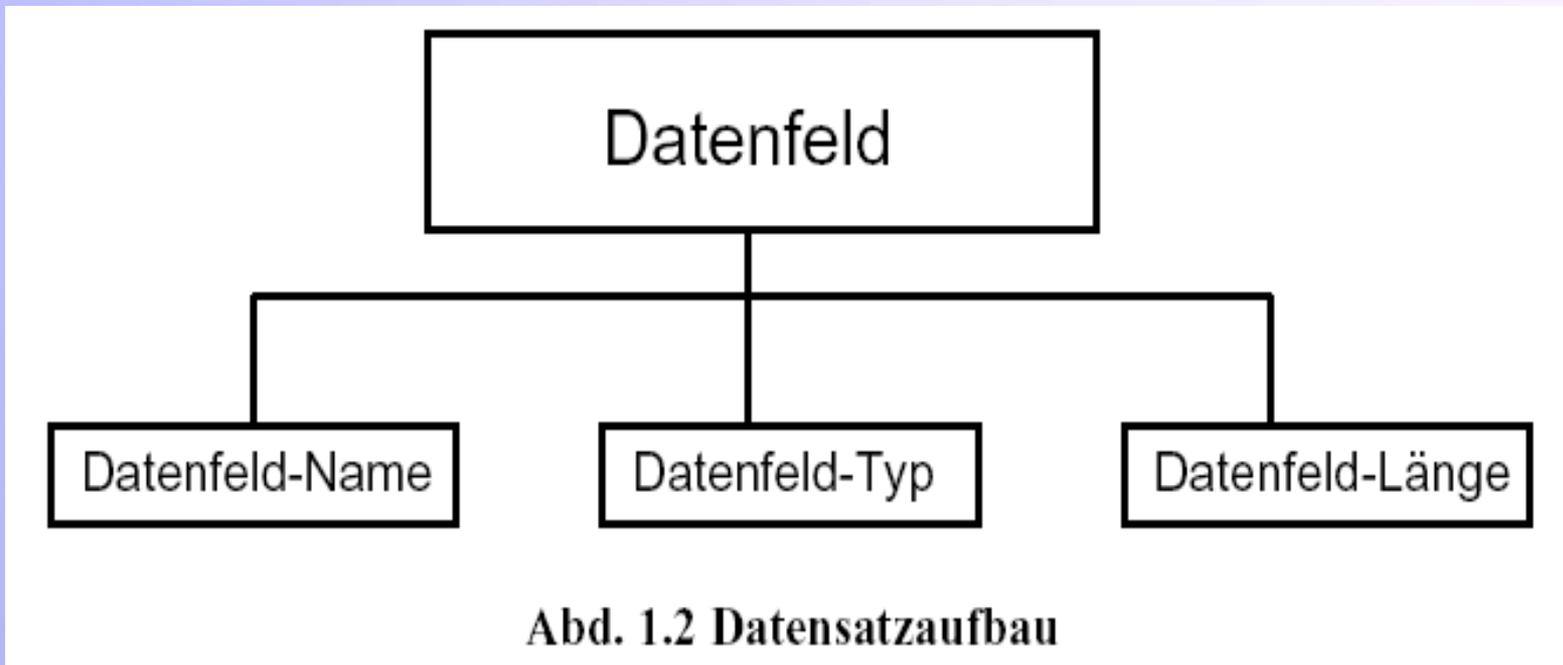
Tabelle Kunden



Abd. 1.1 Datenbankinformationen in Tabellenform

Einführung in SQL und MySQL

Datensatzaufbau



Einführung in SQL und MySQL

Datenbankmodelle - Einführung

Aufgabe: Erstellen Sie eine Adress-Datenbank (auf Papier).
Die folgenden Daten liegen bereits vor:



en-mix
energie • natürlich • für immer

Dipl.-Ing. Max Mustermann

| | | |
|--|--|--|
| en-mix Energie GmbH Hauptstraße 1 12345 Hauptstadt | Telefon 0 67 89 / 123 456 Telefax 0 67 89 / 123 45 67 | Web: www.en-mix.de eMail: info@en-mix.de |
|--|--|--|

Dr. med. Alexander Medicus
Arzt für Allgemeinmedizin
Naturheilverfahren
Physikalische Therapie

Dieselstraße 2
50859 Köln (Lövenich)
Telefon (02234) 70 11-0
Telefax (02234) 70 11 470

Sprechstunden:
Mo.–Fr. 9–12 und 15–18 Uhr
außer Mittwochnachmittag
und nach Vereinbarung



Musterfirma
We make Business TRUE

Hans Mustermann
Dpl. Ing.

Musterstr. 12
Am Medienhafen

0123/456789
0123/456789-5
hans@mustermail.de
www.mustersite.de



AvataR design & computer gbr
grafikdesign • internet • hardware • service

Garvin Hicking

Pintgasse 13
D-53424 Remagen

mail garvin@avatar-dc.de
phone 02642-90 12 81
fax 02642-90 12 82
mobile 0177-5 88 14 45

<http://www.avatar-dc.de/>

Einführung in SQL und MySQL

3 Typen von Datenbankmodellen

1. Hierarchisches Datenbankmodell (überholt)
Bspw.: IMS von IBM, existiert seit ca. 50 Jahren
<http://www-306.ibm.com/software/data/ims/>
2. Relationales Datenbankmodell (aktuell)
Bspw.: MS Access und MS SQL Server, Oracle, MySQL
3. Objektorientiertes Datenbankmodell (die Zukunft?)
Bspw.: Lotus Notes mit Domino Server oder PostgreSQL
<http://www.postgres.de/>



Einführung in SQL und MySQL

Relationales Datenbankmodell – Beschreibung

Zusammenhängende Informationen werden auf mehrere Tabellen aufgeteilt, die untereinander in Beziehung stehen.

Eine große Tabelle für alle Informationen

| Matrikelnummer | Vorname | Nachname | Adresse | Fachbereich |
|----------------|-----------|----------|----------|-------------|
| 12345678 | Alexandra | von Cube | Dortmund | Mathematik |
| 12345678 | Alexandra | von Cube | Dortmund | Kunst |
| 12345678 | Alexandra | von Cube | Dortmund | Pädagogik |

Einführung in SQL und MySQL

Relationales Datenbankmodell – Beschreibung

Tabelle 1

| Matrikelnummer | Vorname | Nachname | Adresse |
|----------------|-----------|----------|----------|
| 12345678 | Alexandra | von Cube | Dortmund |
| 91011121 | Christine | Domberg | Bochum |
| 97654377 | Marcus | Meier | Minden |

Tabelle 2, die mit der ersten verknüpft ist

| Matrikelnummer | Fachbereich |
|----------------|----------------|
| 12345678 | Mathematik |
| 12345678 | Kunst |
| 12345678 | Pädagogik |
| 91011121 | Hauswirtschaft |
| 91011121 | Pädagogik |



Einführung in SQL und MySQL

Relationales Datenbankmodell – Beschreibung

Primärschlüsselfeld:

- Keine Duplikate
- Eindeutige Identifikation der Datensätze
- Wird mit Fremdschlüsselfeld einer anderen Tabelle verknüpft und dient der eindeutigen Zuordnung verschiedener Daten aus mehreren Tabellen

Fremdschlüsselfeld:

- Feld in einer Tabelle, das mit dem Primärschlüsselfeld einer anderen Tabelle verknüpft ist



Einführung in SQL und MySQL

Relationales Datenbankmodell – Beschreibung

Indizes:

- Indizes beschleunigen Such- und Sortierfunktionen.
- Ein Primärschlüssel einer Tabelle ist immer ein Index (macht MySQL automatisch).
- Umgekehrt muss ein Index nicht unbedingt Primärschlüssel sein.



Einführung in SQL und MySQL

Beziehungstypen in relationalen Datenbanken:

- 1:n-Beziehung
- n:m-Beziehung
- 1:1-Beziehung



Einführung in SQL und MySQL

1:n-Beziehung:

- 1-Seite: Tabelle ohne redundante Informationen (Mastertabelle)
- n-Seite: Tabelle in der der Wert einmal, keinmal oder mehrmals vorkommen kann (Detailtabelle)



Einführung in SQL und MySQL

1:n-Beziehung:

Tabelle 1

| Matrikelnummer | Vorname | Nachname | Adresse |
|----------------|-----------|----------|----------|
| 12345678 | Alexandra | von Cube | Dortmund |
| 91011121 | Christine | Domberg | Bochum |
| 97654377 | Marcus | Meier | Minden |

Tabelle 2, die mit der ersten verknüpft ist

| Matrikelnummer | Fachbereich |
|----------------|----------------|
| 12345678 | Mathematik |
| 12345678 | Kunst |
| 12345678 | Pädagogik |
| 91011121 | Hauswirtschaft |
| 91011121 | Pädagogik |



Einführung in SQL und MySQL

n:m-Beziehung:

- n-Seite: Tabelle in der der Wert einmal, keinmal oder mehrmals vorkommen kann
- m-Seite: Tabelle in der der Wert einmal, keinmal oder mehrmals vorkommen kann

Wichtig: Die n:m-Beziehung benötigt eine eigene Tabelle!



Einführung in SQL und MySQL

n:m-Beziehung:

Tabelle 1 (Mastertabelle)

| Matrikelnummer | Nachname |
|----------------|----------|
| 12345678 | von Cube |
| 10111213 | Müller |
| 14151617 | Schmidt |
| 18192021 | Damberg |
| usw. | |

Tabelle 2 (Mastertabelle)

| ID_Seminar | Seminarname | Dozent |
|------------|-----------------------------|----------|
| 0001 | Sozioökonomie des Haushalts | Schmetz |
| 0002 | Ernährungslehre | Bönnhoff |
| 0003 | Wirtschaftslehre | Kuklik |
| 0004 | Didaktik der Haushaltslehre | Stiller |

In Tabelle 2 ist die ID der Primärschlüssel.

Tabelle 3 (Detailtabelle)

| Matrikelnummer | ID_Seminar |
|----------------|------------|
| 12345678 | 0001 |
| 12345678 | 0002 |
| 14151617 | 0003 |
| 12345678 | 0004 |
| 18192021 | 0003 |
| 18192021 | 0001 |



Einführung in SQL und MySQL

1:1-Beziehung:

- Sonderfall, tritt relativ selten auf
- Wert existiert in beiden Fällen nur genau einmal und ist jeweils Primärschlüssel



Wird bspw. verwendet um eine sehr große Tabelle in mehrere kleine Tabellen aufzuteilen oder um bestimmte Informationen von den restlichen Daten eindeutig abzugrenzen.

Einführung in SQL und MySQL

1:1-Beziehung:

Tabelle Adressen

| ID_Mitarbeiter | Vorname | Name |
|----------------|---------|-------|
| 00001 | Bill | Gates |
| 00002 | Steve | Jobs |
| usw. | | |

Tabelle Bezüge (vertraulich)

| ID_Mitarbeiter | Jahresgehalt 2006 (in US-Dollar) |
|----------------|----------------------------------|
| 00001 | 660.000,- |
| 00002 | 1,- |
| usw. | |



Einführung in SQL und MySQL

Referentielle Integrität:

1. In Detailtabellen dürfen nur Fremdschlüssel verwendet werden, die auch in den entsprechenden Mastertabellen als Primärschlüssel vorkommen.
2. Datensätze aus Mastertabellen dürfen nicht einfach gelöscht werden, wenn diese mit anderen Detailtabellen verknüpft sind.

Nur die InnoDB-Speicher-Engine von MySQL unterstützt die automatische Überprüfung auf Einhaltung der Regeln der referentiellen Integrität (CONSTRAINT).



Einführung in SQL und MySQL

Normalisierung (Normalformen):

1. Normalform (1NF)
2. Normalform (2NF)
3. Normalform (3NF)

In der Praxis werden meist nur die ersten drei Normalformen angewendet.



Boyce-Codd (BCNF)



4. Normalform (4NF)



5. Normalform (5NF)

Einführung in SQL und MySQL

1. Normalform:

Die Werte in jeder Zeile und jeder Spalte der Tabelle befinden sich im atomaren Zustand, d.h. in den kleinsten möglichen Einheiten.



| ID | Name | Adresse |
|------|------------|------------------------------------|
| 001 | Bill Gates | 1835 73rd Ave NE, Medina, WA 98039 |
| usw. | | |

Die Informationen in den Feldern Name und Adresse lassen sich in noch kleinere Einheiten zerlegen.

Einführung in SQL und MySQL

1. Normalform erfüllt:

| ID | Vorname | Nachname | Straße | Hausnr. | PLZ | Ort |
|------|---------|----------|-------------|---------|----------|--------|
| 001 | Bill | Gates | 73rd Ave NE | 1835 | WA 98039 | Medina |
| usw. | | | | | | |



Einführung in SQL und MySQL

2. Normalform:

Die 1. Normalform muss erfüllt sein.

Jede Tabellenzeile enthält nur Informationen die sich auf das Subjekt beziehen, das durch den Primärschlüssel definiert ist.



Einführung in SQL und MySQL

Beispiel für 2. Normalform:

Tabelle Fuhrpark

| ID_Fahrzeug | Marke | Typ | Baujahr | Fahrer |
|-------------|------------|------|---------|--------------|
| 00001 | Audi | A6 | 2005 | Dieter Meier |
| 00002 | Volkswagen | Golf | 2006 | Heinz Schmid |
| usw. | | | | |

Der Fahrer des Fahrzeugs bezieht sich nicht auf den Primärschlüssel (ID_Fahrzeug).



Einführung in SQL und MySQL

2. Normalform erfüllt:

Tabelle Fuhrpark

| ID_Fahrzeug | Marke | Typ | Baujahr | |
|-------------|------------|------|---------|--|
| 00001 | Audi | A6 | 2005 | |
| 00002 | Volkswagen | Golf | 2006 | |
| usw. | | | | |

Tabelle Mitarbeiter

| ID_Mitarbeiter | Vorname | Nachname |
|----------------|---------|----------|
| 00001 | Dieter | Meier |
| 00002 | Heinz | Schmid |
| usw. | | |

Tabelle FahrerFahrzeuge

| ID_Mitarbeiter | ID_Fahrzeug |
|----------------|-------------|
| 00001 | 00001 |
| 00002 | 00002 |
| usw. | |

Einführung in SQL und MySQL

3. Normalform:

Die 2. Normalform muss erfüllt sein.

Jede Spalte der Tabelle darf nur einmal vorkommen und alle Spalten, die nicht den Primärschlüssel bilden, müssen voneinander unabhängig sein.



Einführung in SQL und MySQL

Beispiel für 3. Normalform:

| Matrikelnummer | Nachname | Strasse | Hausnr | Postleitzahl | Ort | Geburtsdatum |
|----------------|----------|-------------|--------|--------------|----------|--------------|
| 12345678 | von Cube | Schmitzweg | 9 | 44225 | Dortmund | 20.02. 1980 |
| 10111213 | Schmidt | Tannenstr. | 18 | 45632 | Bochum | 23.04. 1970 |
| 14151617 | Schmidt | Harkortstr. | 4 | 87564 | Lünen | 15.06. 1982 |
| 18192021 | Damberg | Colastr. | 8 | 44123 | Dortmund | 30.12. 1977 |
| usw. | | | | | | |

Die Felder „Postleitzahl“ und „Ort“
sind direkt voneinander abhängig.

Einführung in SQL und MySQL

Alle drei Normalformen erfüllt:

| Matrikelnummer | Nachname | Strasse | Hausnr | Postleitzahl | Geburtsdatum |
|----------------|----------|-------------|--------|--------------|--------------|
| 12345678 | von Cube | Schmitzweg | 9 | 44225 | 20.02. 1980 |
| 10111213 | Schmidt | Tannenstr. | 18 | 45632 | 23.04. 1970 |
| 14151617 | Schmidt | Harkortstr. | 4 | 87564 | 15.06. 1982 |
| 18192021 | Damberg | Colastr. | 8 | 44123 | 30.12. 1977 |
| usw. | | | | | |



Einführung in SQL und MySQL

Entity Relationship Model (ERM):

Übliche Vorgehensweise beim Entwurf einer relationalen Datenbank:

1. Erstellen des Entity Relationship Model (ERM)
2. Ableiten der Tabellen aus der ERM-Skizze
3. Normalisierung (Anwendung der Normalformen auf die Tabellen)



Einführung in SQL und MySQL

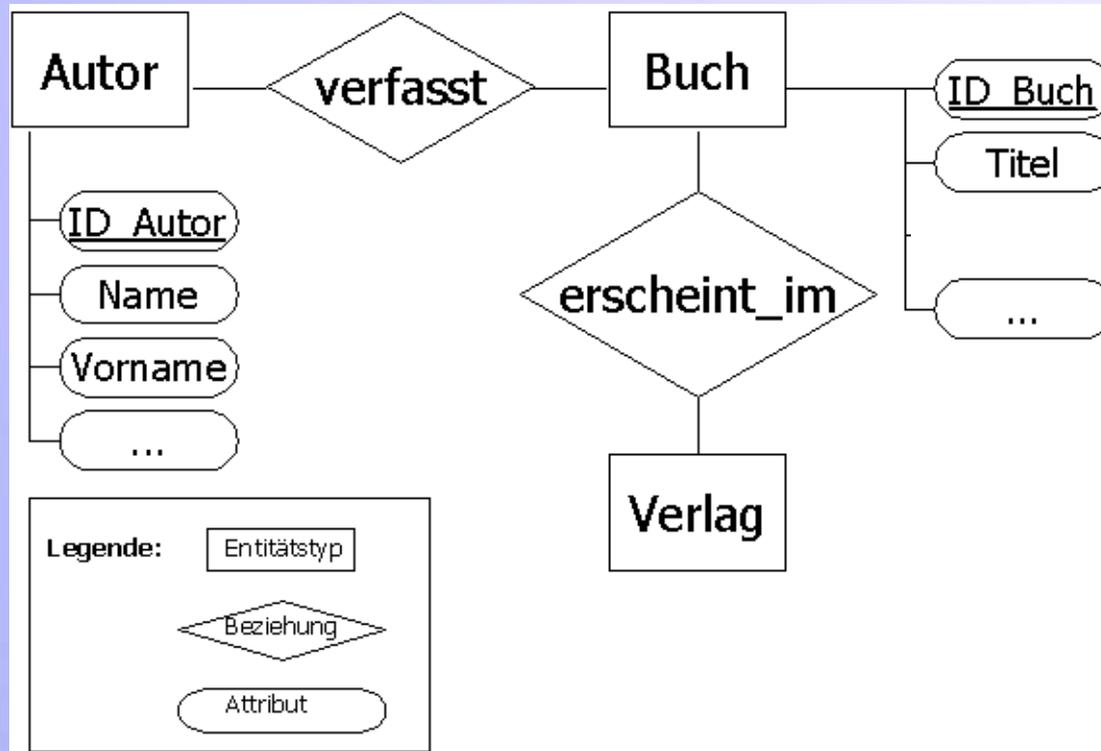
Entity Relationship Model (ERM):

- Beschreibungsmittel zur Erstellung von konzeptionellen Schemata
- ERM beschreibt Daten und ihre Beziehungen untereinander
- Grundlage ist die Typisierung von Objekten und deren Beziehungen untereinander



Einführung in SQL und MySQL

Entity Relationship Model (ERM):



Einführung in SQL und MySQL

Vorübung für unsere erste MySQL-Datenbank:

Aufgabe: Erstellen Sie den Datenbankentwurf (auf Papier) für ein News-System auf einer Internetseite (bspw. Blog) mit folgenden Bedingungen:

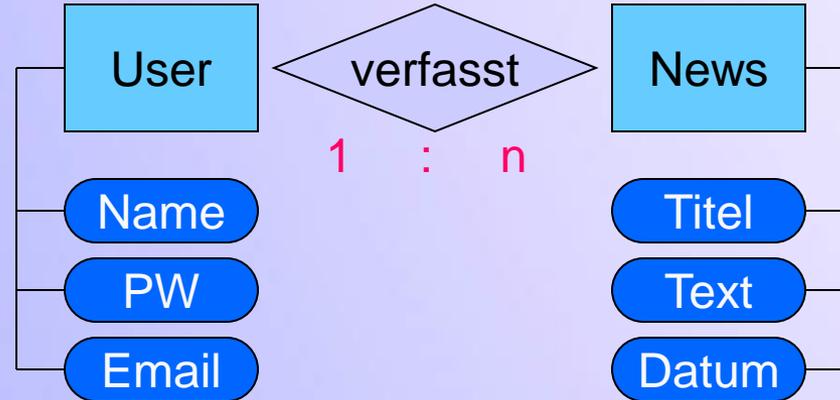
- Nur registrierte Benutzer dürfen News verfassen.
- Zu jedem Benutzer werden Login-Name, Passwort und E-Mailadresse gespeichert.
- Jeder Newsbeitrag besteht aus Titel, Text, Datum des Beitrags und Name des Verfassers.



Einführung in SQL und MySQL

Vorübung für unsere erste MySQL-Datenbank:

ERM:



+ 1 Feld für Primärschlüssel

+ 1 Feld für Fremdschlüssel

+ 1 Feld für Primärschlüssel

Einführung in SQL und MySQL

Vorübung für unsere erste MySQL-Datenbank:

Tabellen, abgeleitet von ERM:

Tabelle User

| | | | |
|---------|----------|----------|-------|
| ID_user | Username | Passwort | Email |
|---------|----------|----------|-------|

Tabelle News

| | | | | |
|---------|-----------|----------|-----------|---------|
| ID_news | Newstitel | Newstext | Newsdatum | ID_user |
|---------|-----------|----------|-----------|---------|



Einführung in SQL und MySQL

SQL-Grundlagen: Historie

- **1970:** Edgar Frank Codd veröffentlicht *A Relational Model of Data for Large Shared Data Banks*. Darin wird das relationale Datenmodell mit seinen fünf Normalformen vorgestellt.
- **1974:** Basierend auf der Arbeit von Codd entwickelten D. D. Chamberlin und R. F. Boyce bei IBM die Sprache SEQUEL (Structured English Query Language).
- **1979:** Auslieferung des ersten kommerziellen relationalen DBMS. Das Produkt hieß Oracle, die Firma wurde später in Oracle Corporation umbenannt.



Einführung in SQL und MySQL

SQL-Grundlagen: ANSI Standard

- **1986:** SQL1, erster ANSI Standard
- **1987:** SQL1 wird auch als ISO Standard anerkannt
- **1989:** Überarbeitung von SQL1, ist seitdem von RDBMS Herstellern weithin übernommen worden
- **1992:** SQL92 oder SQL2
- **1999:** SQL99 oder SQL3
- **2003:** SQL2003
- **2006:** SQL/XML2006, Verwendung von SQL mit XML
- **SQL:2008** und **SQL:2011**
- **2016:** SQL:2016, aktueller Standard



Einführung in SQL und MySQL

SQL-Grundlagen: Dialekte

Durch die ständige Weiterentwicklung der SQL Standards haben sich bei den RDBMS Herstellern unterschiedliche Dialekte von SQL gebildet. Entweder um von den Usern gewünschte Features anzubieten, zu denen noch kein Standard existiert oder um einen Wettbewerbsvorteil zu erlangen.

Manche Dialekte haben prozedurale Befehle eingeführt, um typische Funktionen einer Programmiersprache anzubieten, bspw. Bedingungen und Schleifen.



Einführung in SQL und MySQL

SQL-Grundlagen: Befehlseingabe

SQL-Befehle können auf zwei verschiedene Arten eingegeben und an die Datenbank übermittelt werden:

- **SQL-Ausführung durch einen Interpreter**
Das DBMS oder ein besonderer Client bieten direkten Zugriff auf die Datenbank, bspw. phpMyAdmin oder MySQL-Monitor.
- **SQL eingebettet in eine Wirtssprache**
Innerhalb einer Programmiersprache wird ein Objekt verwendet, welches Kontakt zum DBMS aufnehmen kann.



Einführung in SQL und MySQL

SQL-Grundlagen: Mengenoperationen

SQL ist eine **mengenorientierte Sprache** im Gegensatz zum datensatzorientierten Zugriff.

Die Stärke von SQL liegt darin, große Datenmengen auf einmal bearbeiten zu können.

Mit Hilfe von JOINS werden zusammenhängende Datensätze aus mehreren Tabellen verknüpft. Durch Anwendung von Filterregeln (WHERE) können aus dieser Menge bestimmte Datensätze ausgewählt und bearbeitet werden.



Einführung in SQL und MySQL

SQL-Grundlagen: 4th Generation Language

SQL gilt als Sprache der vierten Generation (4GL).

SQL-Befehle beschreiben nur, was gemacht werden soll, aber nicht wie es gemacht werden soll.



Eine Sprache der dritten Generation müsste ein eigenes Tabellenobjekt definieren und eigenständig Verfahren entwickeln, um bspw. eine Zeile möglichst effizient zu suchen. Damit wäre jedoch der Datenzugriff nur an diese eine Programmiersprache gebunden, was bei SQL nicht der Fall ist.

Einführung in SQL und MySQL

Exkurs: VIEW (Sicht)

SELECT-Abfragen können in einer VIEW auf dem Server gespeichert werden. Die VIEW zeigt dabei stets das Aktuelle Ergebnis der zugrunde liegenden SELECT-Abfrage und verhält sich fast wie eine reale Tabelle, d.h. es können bspw. SELECT-Abfragen erstellt werden, die auf die VIEW zugreifen als wäre sie eine echte Tabelle.

Als VIEW gespeicherte SELECT-Abfragen sind in der Regel komplizierte oder besonders umfangreiche Abfragen, deren Ergebnisse häufig benötigt werden.



Einführung in SQL und MySQL

Transaktion, Commit und Rollback

Transaktion bezeichnet die Bündelung mehrerer Datenbankoperationen zu einer Einheit.

Die Folge der Operationen (lesen, ändern, einfügen, löschen) soll die Datenbank von einem konsistenten Zustand in einen anderen konsistenten Zustand überführen.



Einführung in SQL und MySQL

Transaktion, Commit und Rollback

Beispiel für Transaktion: Überweisung von 50,- Euro von Konto A nach Konto B

Pseudocode für die Überweisung:

```
read(KontoA, a); -- Kontostand für Konto A ermitteln  
a := a - 50; -- neuen Kontostand berechnen  
write(KontoA, a); -- neuen Kontostand speichern  
read(B, b); -- Kontostand für Konto B ermitteln  
b := b + 50; -- neuen Kontostand berechnen  
write(B, b); -- neuen Kontostand speichern
```



Einführung in SQL und MySQL

Transaktion, Commit und Rollback

Transaktionen in SQL beginnen mit Ausführung der ersten Anweisung. Eine Transaktion wird abgeschlossen durch

- **commit work:** Alle Änderungen sollen festgeschrieben werden (ggf. nicht möglich wegen Konsistenzverletzungen).
- **rollback work:** Alle Änderungen sollen zurückgesetzt werden (ist immer möglich).



Einführung in SQL und MySQL

Transaktion, Commit und Rollback

Innerhalb einer Transaktion sind auch Inkonsistenzen erlaubt. Im folgenden Beispiel fehlt vorübergehend der Eintrag in die Tabelle Professoren (MySQL Syntax):



```
start transaction;  
insert into Vorlesungen  
values (5275, `Kernphysik`, 3, 2141);  
insert into Professoren  
values (2141, `Meitner`, `C4`, 205);  
commit;
```

Einführung in SQL und MySQL

Stored Routines (Prozeduren und Funktionen)

In Prozeduren und Funktionen können SQL-Befehle direkt auf dem Server gespeichert werden. Diese Prozeduren und Funktionen führen dann die entsprechenden SQL Befehle bei ihrem Aufruf aus.



Einführung in SQL und MySQL

Beispiel für eine Funktion:

```
mysql> CREATE FUNCTION hello (s CHAR(20))  
mysql> RETURNS CHAR(50) DETERMINISTIC  
-> RETURN CONCAT('Hello, ',s, '!');
```

```
mysql> SELECT hello('world');
```

```
+-----+
```

```
| hello('world') |
```

```
+-----+
```

```
| Hello, world! |
```

```
+-----+
```

Einführung in SQL und MySQL

Beispiel für eine Prozedur:

```
mysql> delimiter //  
mysql> CREATE PROCEDURE simpleproc (OUT param1 INT)  
-> BEGIN  
-> SELECT COUNT(*) INTO param1 FROM t;  
-> END//  
mysql> delimiter ;
```

```
mysql> CALL simpleproc(@a);
```

```
mysql> SELECT @a;
```

```
+-----+
```

```
| @a |
```

```
+-----+
```

```
| 3 |
```

```
+-----+
```

Einführung in SQL und MySQL

Trigger

Trigger sind mit einer Tabelle verbunden und werden aktiviert, sobald ein bestimmtes Ereignis eintritt, bspw. beim Einfügen einer neuen Zeile.

Beispiel eines Triggers der die aktuelle Summe einer Spalte berechnet:



```
mysql> CREATE TABLE account (acct_num INT, amount DECIMAL(10,2));
```



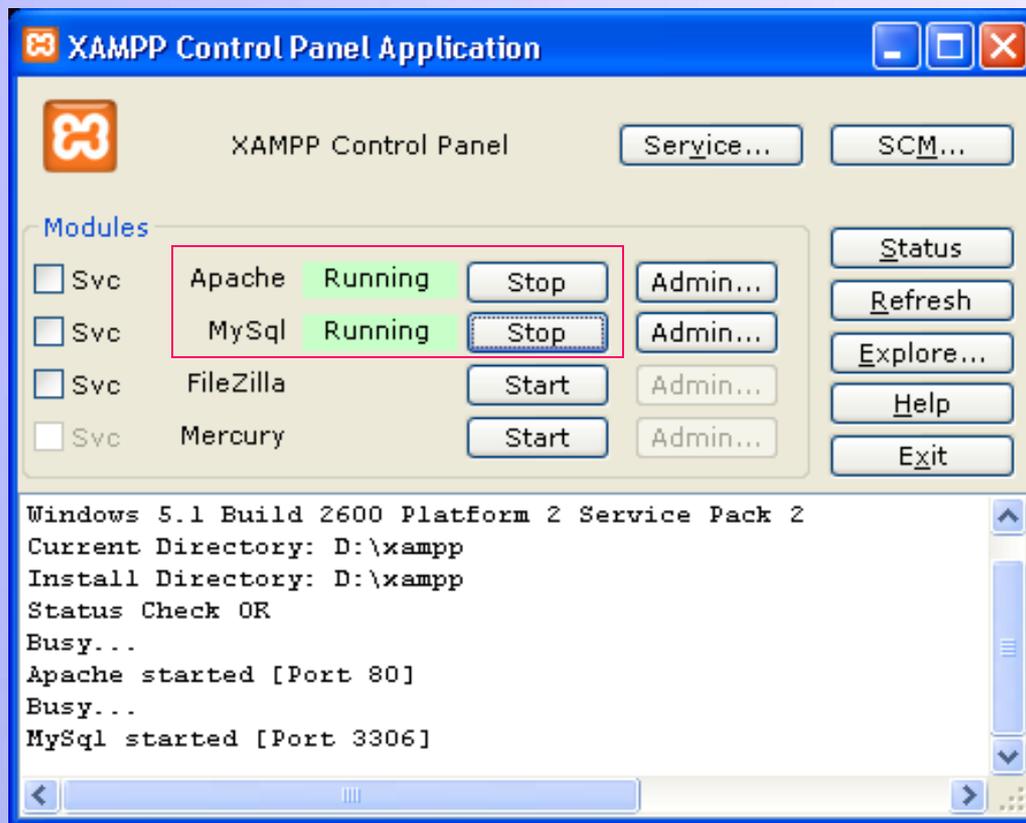
```
mysql> CREATE TRIGGER ins_sum BEFORE INSERT ON account  
-> FOR EACH ROW SET @sum = @sum + NEW.amount;
```



```
mysql> SELECT @sum;
```

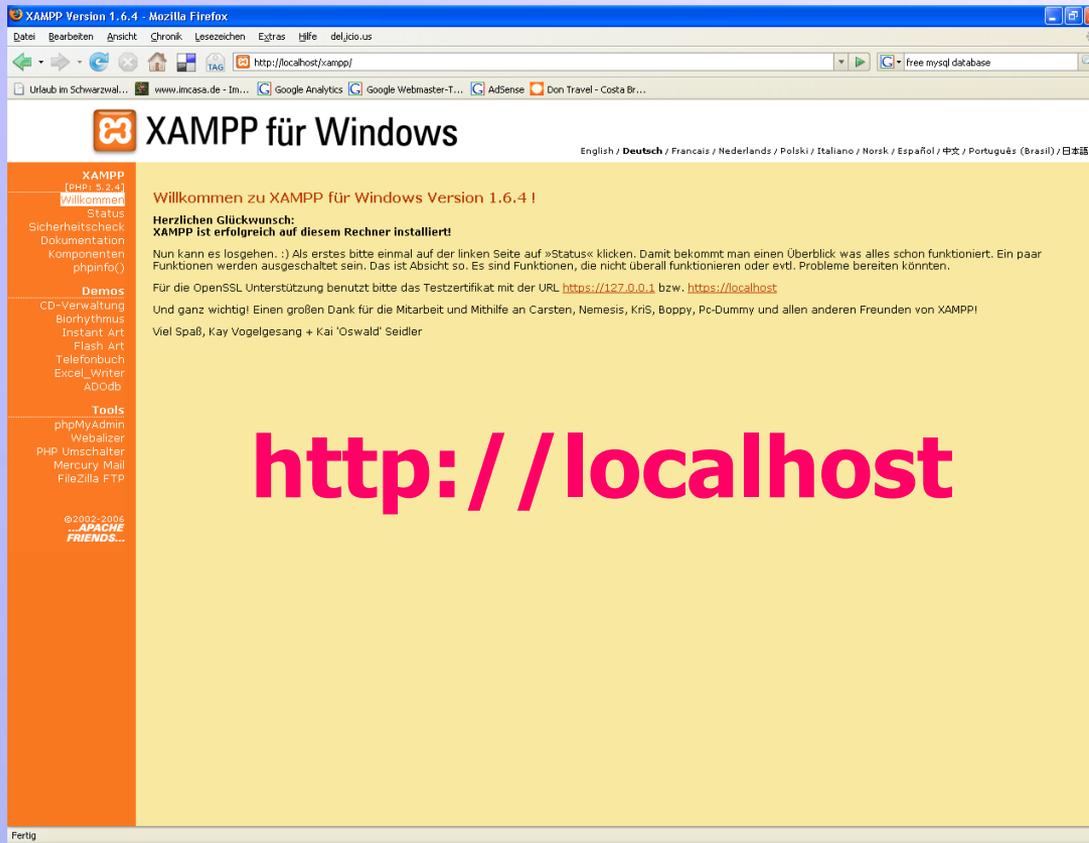
Einführung in SQL und MySQL

XAMPP als Entwicklungsumgebung:



Einführung in SQL und MySQL

XAMPP als Entwicklungsumgebung:



XAMPP Version 1.6.4 - Mozilla Firefox

http://localhost/xampp/

XAMPP für Windows

English / **Deutsch** / Francais / Nederlands / Polski / Italiano / Norsk / Español / 中文 / Português (Brasil) / 日本語

XAMPP (PHP: 5.2.4) Willkommen

Status
Sicherheitscheck
Dokumentation
Komponenten
phpInfo()

Demos
CD-Verwaltung
Biorhythmus
Instant Art
Flash Art
Telefonbuch
Excel_Writer
ADODB

Tools
phpMyAdmin
Webalizer
PHP Umschalter
Mercury Mail
FileZilla FTP

© 2002-2008
APACHE FRIENDS...

Willkommen zu XAMPP für Windows Version 1.6.4 !

Herzlichen Glückwunsch: XAMPP ist erfolgreich auf diesem Rechner installiert!

Nun kann es losgehen. :) Als erstes bitte einmal auf der linken Seite auf »Status« klicken. Damit bekommt man einen Überblick was alles schon funktioniert. Ein paar Funktionen werden ausgeschaltet sein. Das ist Absicht so. Es sind Funktionen, die nicht überall funktionieren oder evtl. Probleme bereiten könnten.

Für die OpenSSL Unterstützung benutzt bitte das Testzertifikat mit der URL <https://127.0.0.1> bzw. <https://localhost>

Und ganz wichtig! Einen großen Dank für die Mitarbeit und Mithilfe an Carsten, Nemesis, Kris, Boppy, Pc-Dummy und allen anderen Freunden von XAMPP!

Viel Spaß, Kay Vogelgesang + Kai 'Oswald' Seidler

http://localhost

Fertig

Einführung in SQL und MySQL

Übung: Ein Website News-System mit MySQL

Alle Anweisungen und Informationen zu dieser Übung finden Sie im Handout.



Einführung in SQL und MySQL

Administration von MySQL-Datenbanken

Bei den Kursmaterialien finden Sie einen Link zu dem Artikel „MySQL-Anwendungen sichern“ sowie die PDF-Version des Artikels.

Ebenfalls als Link finden Sie bei den Kursmaterialien auch eine Anweisung für verschiedene Möglichkeiten um Backups (Datensicherungen) und Restores (Datenwiederherstellungen) bei MySQL-Datenbanken durchzuführen.



Einführung in SQL und MySQL

Beispiele für ungesicherte MySQL-Datenbanken

http://www.google.com/search?q=intitle%3AphpMyAdmin+%22Welcome+to+phpMyAdmin+***%22+%22running+on+*+as+root%40*%22&start=0

<https://www.exploit-db.com/google-hacking-database>

