



Interaktive Webseiten mit PHP und MySQL

Vorlesung 3: Einführung in MySQL

Sommersemester 2003

Martin Ellermann

Heiko Holtkamp

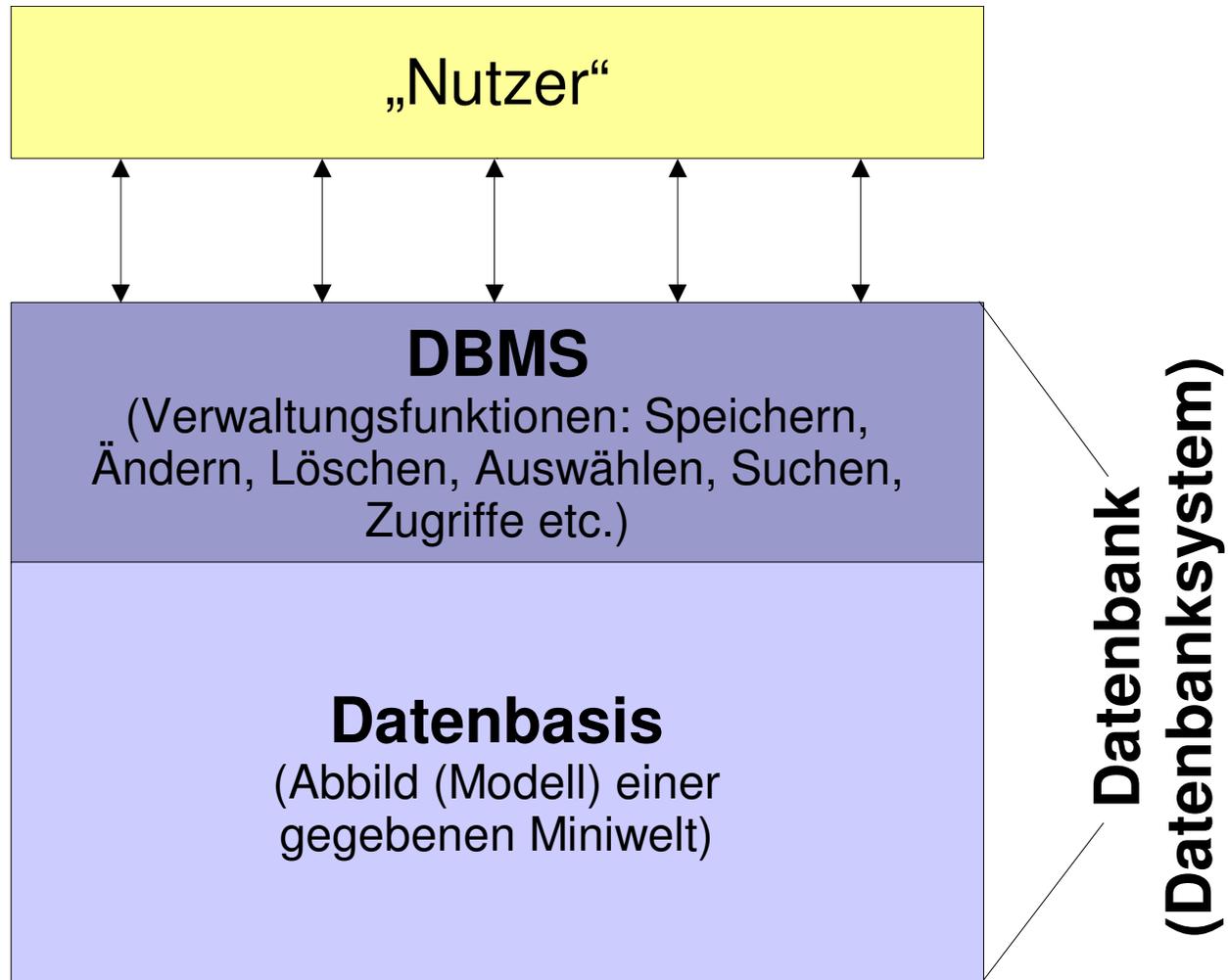
Quellenhinweise

- Offizielle MySQL-Webseite: <http://www.mysql.com>
- Krause, Jörg: PHP4 - Grundlagen und Profiwissen. Hanser, 2001, 2. Aufl.
- Greenspan, Jay; Bulger, Brad: MySQL/PHP-Datenbankanwendungen. MITP, 2001
- Reeg, Christoph; Hatlak, Jens: Datenbank, MySQL und PHP. 2002. (<http://www.reeg.net>).
- Raymans, Heinz-Gerd: MySQL im Einsatz. Addison-Wesley, 2001
- Lang S.M., Lockemann P.C.: Datenbankeinsatz. Springer, 1995

Was ist eine Datenbank?

- Eine *Datenbank* ist eine strukturierte Sammlung von *Daten*.
- *Daten* sind physikalische Repräsentationen (*Symbole*), denen eine feste Bedeutung unterstellt werden kann.
- Bei der Verarbeitung dieser Symbole wird stets auf eine Bedeutung dieser Symbole abgehoben, deshalb lässt sich i.a. der Begriff „Daten“ rechtfertigen.
- Mit dem Begriff „Informationen“ wird i.a. noch der Zweck der Benutzung von Daten verbunden.
- Um mit den Daten umzugehen wird ein System benötigt, um mit diesen arbeiten zu können, das *Datenbank-Management-System (DBMS – Data Base Management System)*.

Was ist eine Datenbank?



Ein Sprung... Datenmodelle

- Eine Datenbasis muss zu jedem Zeitpunkt ein Abbild (Modell) einer gegebenen Miniwelt sein.
- Eine Datenbasis muss zu jedem Zeitpunkt den Konsistenzregeln für ein Modell einer gegebenen Miniwelt folgen.
- Ein Datenbanksystem gewährleistet Konsistenz, wenn seine Dienstfunktionen stets einen konsistenten Zustand seiner Datenbasis wieder in einen konsistenten Zustand überführen.
- Ein Datenbasisverwaltungssystem (DBMS) realisiert ein (einziges) Datenmodell.
- Datenmodelle lassen sich anhand verschiedener Bewertungskriterien beurteilen (Strukturelle Mächtigkeit, Strukturelle Orthogonalität, Operationelle Verfügbarkeit, Operationelle Generizität)

Datenmodelle

- NF²-Modell (Non First Normal Form) / ENF²-Modell
- Netzwerkmodell (CODASYL-Modell)
- Deduktive Modelle (XPS)
- Objektorientierte Modelle
- Semantische Datenmodelle
- **Relationales Modell**

Das relationale Modell

- 1970 von E.F. Codd eingeführt durch eine Veröffentlichung in Communications of the ACM (Band 13, Nr. 6, S. 377-387, Juni 1970): "A relational model for large shared data banks" ("Ein relationales Modell für Daten in großen Datenbanken").
- Larry Ellison war einer der ersten, der die Arbeit von Codd von der Theorie in die Praxis umsetzte: ORACLE.

Relationale Datenbanken / Relationenmodell

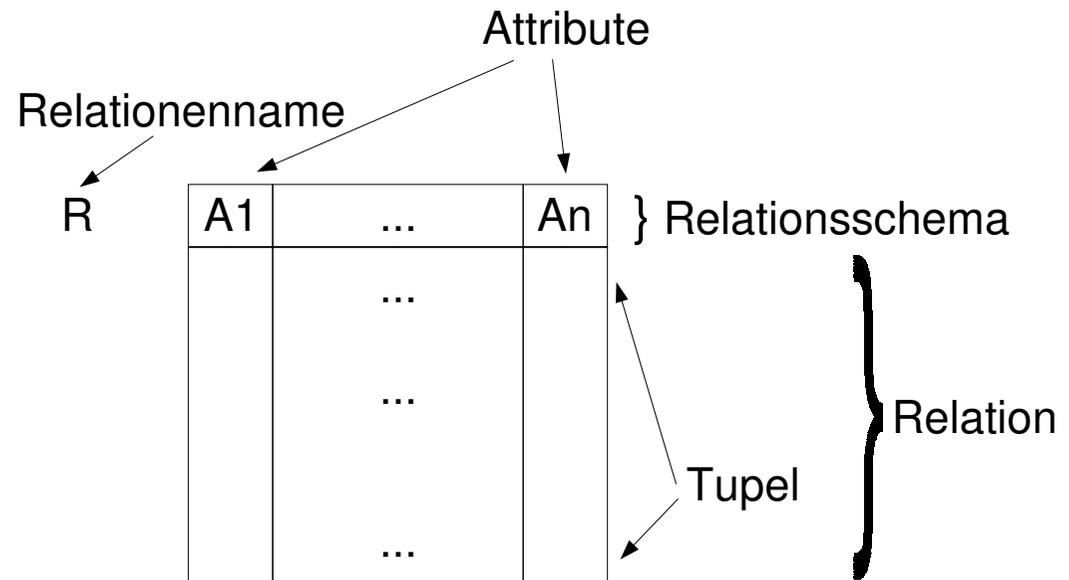
- Im *relationalen Modell* werden Datenbestände durch *Relationen* repräsentiert.
- Relationen sind *Mengen* von gleichartig strukturierten *Tupeln*, wobei jedes Tupel ein Objekt oder eine Beziehung in einer Miniwelt beschreibt.
- Jede Komponente eines Tupels eines Tupels enthält dabei eine Merkmalsausprägung des entsprechenden Objekts (oder der Beziehung) bezüglich eines ganz bestimmten Merkmals.
- Diese Merkmale nennt man *Attribute*.
- Die Menge aller möglichen Ausprägungen eines Attributs heißt *Domäne* des Attributs.
- Oder formaler...

Relationstypen, Relationen und Tupel

- Definition:
Ein *Relationstyp* T_R ist ein *Tupel* (A_{R1}, \dots, A_{Rn}) . Dabei sind A_{R1}, \dots, A_{Rn} paarweise verschiedene Attributnamen, und zu jedem A_{Ri} ist eine Menge D_i , die Domäne von A_{Ri} , gegeben. Zu jedem Typ T_R existiert eine n -stellige *Relation* R als eine zeitlich veränderliche Untermenge des kartesischen Produkts $D_1 \times \dots \times D_n$, also eine Menge von *Tupeln* (d_1, \dots, d_n) mit $d_i \in D_i$ für $1 \leq i \leq n$.
- Oder einfacher...

Tabellen, Spalten, Zeilen

- Relationen lassen sich sehr übersichtlich als Tabellen darstellen, indem man ihre Tupel untereinander anordnet.
- Jeder der dabei entstehenden Spalten kann dann ein Attribut zugeordnet werden.
- Die Zeilen der Tabelle, die jeweils ein Objekt darstellen, enthalten in jeder Spalte die Ausprägung dieses Objekts unter dem der Spalte zugeordneten Attribut.



Mitarbeiter			
MNr	Name	Vorname	Telefon
1	Holtkamp	Heiko	1063567
2	Ellermann	Martin	1063567
3	Holtmann	Marcel	1063566
4	Hilbert	Mirco	1063568

Das Relationenmodell

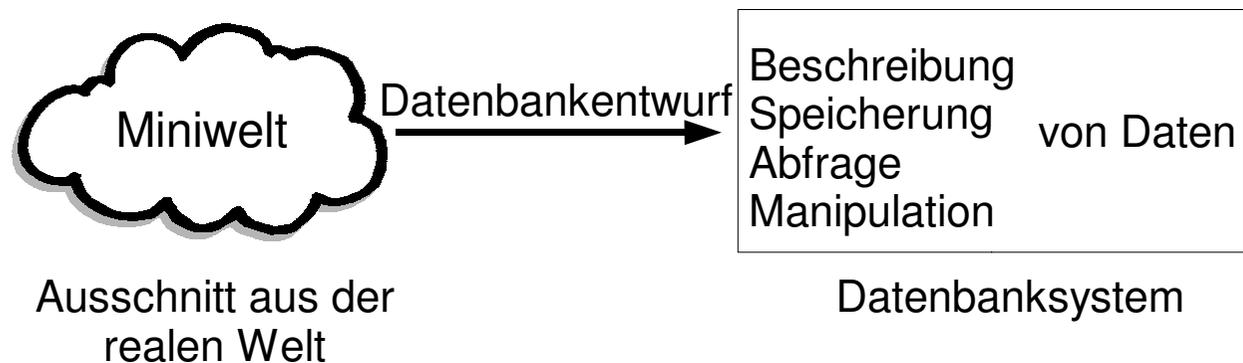
- Neben diesem sog. *Strukturteil* besteht das Relationenmodell auch aus einem *Operationenteil*.
- Mit den Konzepte des Strukturteils können die Objekttypen der Anwendungswelt beschrieben werden.
- Der Operationenteil stellt eine Menge von (implizit) im Modell vorhandenen Operationen auf den zu diesen Objekttypen gehörenden Instanzen (Daten) zur Verfügung.

Relationenalgebra

- **Selektion**
Wählt bestimmte Tupel aus einer Relation
- **Projektion**
Wählt bestimmte Spalten aus einer Relation
- **Vereinigung**
Fasst Tupel zweier Relationen in einer neuen Relation zusammen
- **Differenz**
Die Tupel zweier Relationen werden verglichen; die in der ersten, nicht aber in der zweiten Relation befindlichen Tupel finden Eingang in die Ergebnisrelation
- **Durchschnitt**
Wählt die Tupel, die gleichzeitig in zwei verschiedenen Relationen enthalten sind.
- **Kartesisches Produkt**
Verbindet alle Tupel zweier Relationen kombinatorisch miteinander.
- **Verbindung (Join)**
Ein Join ist die Kombination von kartesischem Produkt und nachfolgender Selektion.
- **Umbenennung**
Die Relation bleibt unverändert, nur ihr Schema ändert sich.

Datenbanken entwickeln

- Die Entwicklung einer Datenbank vollzieht sich in mehreren Schritten (stark vereinfacht!)
 - Welche Informationen erwarten die Anwender vom DBS?
 - Welche Tabellen werden benötigt?
 - Welche Datentypen werden für die Tabellenspalten benötigt?
- Dieser Prozess wird als Datenmodellierung bezeichnet.
- Grundsätze für die Datenmodellierung
 - Keine Redundanz
 - Eindeutigkeit
 - Keine Prozessdaten



Ein blödes Beispiel

Redundanz			
Name	Vorname	GebDatum	Alter
Heiko Holtkamp	Heiko	1.1.1972	31
Martin Ellermann	Martin	1.1.1970	33
Marcel Holtmann	Marcel	1.1.1976	27
Mirco Hilbert	Mirco	1.1.1975	28

Eindeutigkeit			
Name	Vorname	GebDatum	Alter
Holtkamp	Heiko	1.1.1972	31
Ellermann	Martin	1.1.1970	33
Holtmann	Marcel	1.1.1976	27
Hilbert	Mirco	1.1.1975	28
Holtkamp	Andreas	1.1.1962	41

Prozessdaten			
Name	Vorname	GebDatum	Alter
Heiko Holtkamp	Heiko	1.1.1972	31
Martin Ellermann	Martin	1.1.1970	33
Marcel Holtmann	Marcel	1.1.1976	27
Mirco Hilbert	Mirco	1.1.1975	28

Datenmodellierung / Tabellen erstellen

- **Normalisierung**
Der Normalisierungsprozess besteht aus der Erzeugung verschiedener Normalformen. Normalisierung ist wichtig, aber nicht alleine das A und O guten Datenbankdesigns. Erfahrung und Instinkt spielen ebenso eine große Rolle
- **Entity-Relationship-Modell (ER-Modell)**
Ein *Gegenstand* (engl. *Entity*) repräsentiert ein abstraktes oder physisches Objekt der realen Welt. Zur Beschreibung von Gegenständen sowie deren Identifizierung bedient man sich Attributen, die geeignet belegt werden. Eine *Beziehung* (engl. *Relationship*) beschreibt einen Zusammenhang zwischen mehreren Gegenständen und reichert diesen ggf. durch Informationen an.
- **Ansätze: Normalisierung und/oder Transformation eines ER-Modells**

Der Normalisierungsprozess (nicht formal, praxisorientiert)

- 0. Normalform
 - Alle Datenelemente auflisten
 - Für jede Tabelle einen *Primärschlüssel* festsetzen (ein Schlüssel dient zur Identifikation eines Tupels)
- 1. Normalform: *Wiederholungsgruppen auflösen*
 - Eliminieren sich wiederholender Gruppen in einzelnen Relationen
 - Alle Wertebereiche der Attribute einer Relation müssen atomar sein

Der Normalisierungsprozess: Beispiel

„0. Normalform“

PersNr	Name	Vorname	AbtNr	Abteilung	ProjektNr	Beschreibung
1	Lorenz	Sophia	1	Personal	2	Verkaufspromotion
2	Hohl	Tatjana	2	Einkauf	3	Konkurrenzanalyse
3	Wilschrein	Theo	1	Personal	1,2,3	Kundenumfrage, Verkaufspromotion, Konkurrenzanalyse
4	Richter	Hans	3	Verkauf	2	Verkaufspromotion
5	Wiesenland	Werner	2	Einkauf	1	Kundenumfrage

1. Normalform

PersNr	Name	Vorname	AbtNr	Abteilung	ProjektNr	Beschreibung
1	Lorenz	Sophia	1	Personal	2	Verkaufspromotion
2	Hohl	Tatjana	2	Einkauf	3	Konkurrenzanalyse
3	Wilschrein	Theo	1	Personal	1	Kundenumfrage
3	Wilschrein	Theo	1	Personal	2	Verkaufspromotion
3	Wilschrein	Theo	1	Personal	3	Konkurrenzanalyse
4	Richter	Hans	3	Verkauf	2	Verkaufspromotion
5	Wiesenland	Werner	2	Einkauf	1	Kundenumfrage

Der Normalisierungsprozess (nicht formal, praxisorientiert)

- 2. Normalform: *Abhängigkeit vom Primärschlüssel*
 - Eine Relation befindet sich in der zweiten Normalform, wenn
 - sie in der 1. Normalform ist und
 - jedes Nicht-Schlüssel-Attribut vom Primärschlüssel voll funktional abhängig ist.
 - Ein Attribut B ist vom Attribut A funktional genau dann abhängig, wenn gilt: aus A folgt direkt B ($A \rightarrow B$)
 - Relationsstypen, die in der 1. NF sind, sind automatisch in der 2. NF, wenn ihr Primärschlüssel nicht zusammengesetzt ist (nur aus einem Attribut besteht)
 - Es muss also geprüft werden, ob man aus einem Teilschlüssel auf Nichtschlüsselattribute folgern kann
 - Bei der „Erzeugung“ der 2. NF entstehen i.d.R. mehrere Relationen

Der Normalisierungsprozess: Beispiel

1. Normalform

PersNr	Name	Vorname	AbtNr	Abteilung	ProjektNr	Beschreibung
1	Lorenz	Sophia	1	Personal	2	Verkaufspromotion
2	Hohl	Tatjana	2	Einkauf	3	Konkurrenzanalyse
3	Wilschrein	Theo	1	Personal	1	Kundenumfrage
3	Wilschrein	Theo	1	Personal	2	Verkaufspromotion
3	Wilschrein	Theo	1	Personal	3	Konkurrenzanalyse
4	Richter	Hans	3	Verkauf	2	Verkaufspromotion
5	Wiesenland	Werner	2	Einkauf	1	Kundenumfrage

2. Normalform

PersNr	Name	Vorname	AbtNr	Abteilung
1	Lorenz	Sophia	1	Personal
2	Hohl	Tatjana	2	Einkauf
3	Wilschrein	Theo	1	Personal
4	Richter	Hans	3	Verkauf
5	Wiesenland	Werner	2	Einkauf

ProjektNr	Beschreibung
2	Verkaufspromotion
3	Konkurrenzanalyse
1	Kundenumfrage

PersNr	ProjektNr
1	2
2	3
3	1
3	2
3	3
4	2
5	1

Der Normalisierungsprozess (nicht formal, praxisorientiert)

- 3. Normalform: Transitive Abhängigkeit
 - Eine Relation befindet sich in der 3. Normalform, wenn
 - Sie in der 2. NF ist und
 - Die Nichtschlüsselattribute voneinander nicht funktional abhängig sind, d.h. es existieren keine transitiven Abhängigkeiten.
 - Transitivität: $X \rightarrow Y$ und $Y \rightarrow Z$, dann $X \rightarrow Z$
 - Ein Attribut Z ist vom Attribut X über das Attribut Y genau dann transitiv abhängig, wenn gilt: $X \rightarrow Y$ und $\neg(Y \rightarrow X)$ und $Y \rightarrow Z$
 - Es muss also geprüft werden, ob aus einem Nichtschlüsselattribut auf ein weiteres Nichtschlüsselattribut gefolgert werden kann
 - Bei der „Erzeugung“ der 3. NF entstehen weitere Relationen

Der Normalisierungsprozess: Beispiel

2. Normalform

PersNr	Name	Vorname	AbtNr	Abteilung
1	Lorenz	Sophia	1	Personal
2	Hohl	Tatjana	2	Einkauf
3	Wilschrein	Theo	1	Personal
4	Richter	Hans	3	Verkauf
5	Wiesenland	Werner	2	Einkauf

ProjektNr	Beschreibung
2	Verkaufspromotion
3	Konkurrenzanalyse
1	Kundenumfrage

PersNr	ProjektNr
1	2
2	3
3	1
3	2
3	3
4	2
5	1

3. Normalform

PersNr	Name	Vorname	AbtNr
1	Lorenz	Sophia	1
2	Hohl	Tatjana	2
3	Wilschrein	Theo	1
4	Richter	Hans	3
5	Wiesenland	Werner	2

ProjektNr	Beschreibung
2	Verkaufspromotion
3	Konkurrenzanalyse
1	Kundenumfrage

PersNr	ProjektNr
1	2
2	3
3	1
3	2
3	3
4	2
5	1

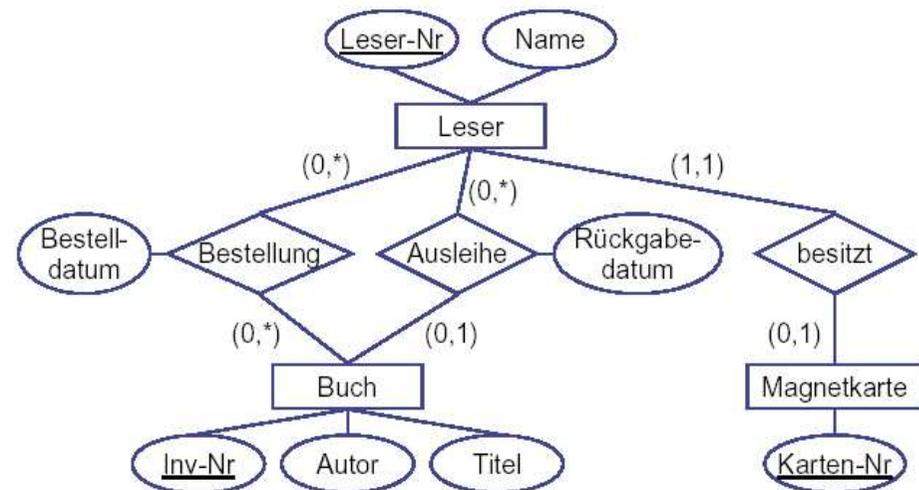
AbtNr	Abteilung
1	Personal
2	Einkauf
3	Verkauf

Der Normalisierungsprozess (nicht formal, praxisorientiert)

- Weitere Normalisierungsformen:
 - In der Regel werden in der Praxis nur die ersten drei Normalformen angewendet, es gibt aber eine Reihe weiterer Normalformen;
Stichworte:
 - Vierte Normalform
 - Fünfte Normalform
 - Boyce-Codd-Normalform (BCNF)

Das Entity-Relationship-Modell – Kurz & knapp...

- Vorgehen
 - Jeder Entitytyp wird in eine Relation transformiert.
 - Ein 1:1-Beziehungstyp wird durch Aufnahme eines Schlüssels in eine der beiden Tabellen abgebildet.
 - Ein 1:n-Beziehungstyp wird durch Aufnahme des Schlüssels des Entitytyps, der in der Beziehung n-mal auftreten kann, in die dem anderen Entitytyp entsprechende Tabelle abgebildet.
 - Ein m:n-Beziehung wird durch die Definition einer zusätzlichen Tabelle, in der die Schlüssel aller beteiligten Entitytypen enthalten sind, abgebildet.
 - Bei attributierten Beziehungen werden die Attribute des Beziehungstyps in die Tabelle, in der die Beziehung abgebildet ist, aufgenommen.



Genug der Theorie...
MySQL

MySQL

- Das *MySQL RDBMS* wird von einem kommerziell orientierten Unternehmen, der *MySQL AB*, entwickelt.
- MySQL AB ist ein schwedisches Unternehmen, das von Michael >Monty< Widenius, Allan Larson und David Axmark gegründet wurde.
- Das Produkt MySQL wird in zwei getrennten Lizenzmodellen angeboten:
 - Lizenzmodell der GPL (OpenSource)
 - Kommerzielle Fassung
 - Die Lizenzpolitik von MySQL AB ist allerdings nicht so ganz klar verständlich und sollte auf jeden Fall aufmerksam studiert werden, wenn MySQL im kommerziellen Umfeld zum Einsatz kommt.

MySQL

- MySQL hält sich an den Standard Entry-Level ANSI SQL92 (Angabe aus der Dokumentation zu Version 3.23.30-gamma).
- Es ist beabsichtigt in Zukunft ANSI SQL99 vollständig zu unterstützen.
- MySQL enthält einige Erweiterungen zu ANSI SQL92 (Auf diese wird hier nicht näher eingegangen. Es sei allerdings bemerkt, dass bei Verwendung dieser Erweiterungen der Code nicht mehr kompatibel mit anderen SQL-Servern ist).

SQL

- SQL steht für *Structured Query Language* (strukturierte Abfrage Sprache).
- Damit ist der Funktionsumfang allerdings nur unzureichend umrissen:
 - Sie enthält nicht nur Funktionen für Datenbankabfragen,
 - sondern deckt das volle Aufgabenspektrum vom Anlegen und Pflegen von Datenstrukturen über das Einfügen, Ändern und Löschen von Daten bis hin zur Benutzerverwaltung ab.
- Wichtige Funktionsbereiche von SQL
 - Datendefinition (DDL – Data Definition Language)
 - Datenmanipulation (DML – Data Manipulation Language)
 - Datenabfrage (QL – Query Language)
 - Zugriffskontrolle
 - Ggf. spezielle Befehle zur internen Datenbankadministration
- Eine richtige "Programmiersprache" ist SQL trotz ihrer Mächtigkeit aber nicht.
 - Sie enthält beispielsweise keine Schleifen- oder Verzweigungsbefehle
- SQL ist eher eine "Schnittstellensprache", in der ein Datenbankserver die Anforderungen einer Client-Anwendung entgegen nimmt.

MySQL starten und anhalten (Systemspezifisch)

- Das hier beschriebene Vorgehen zum starten und anhalten von MySQL (mysqld) ist systemspezifisch, hier SuSE Linux 8.1 mit installierter MySQL Datenbank in der Version 3.23.52

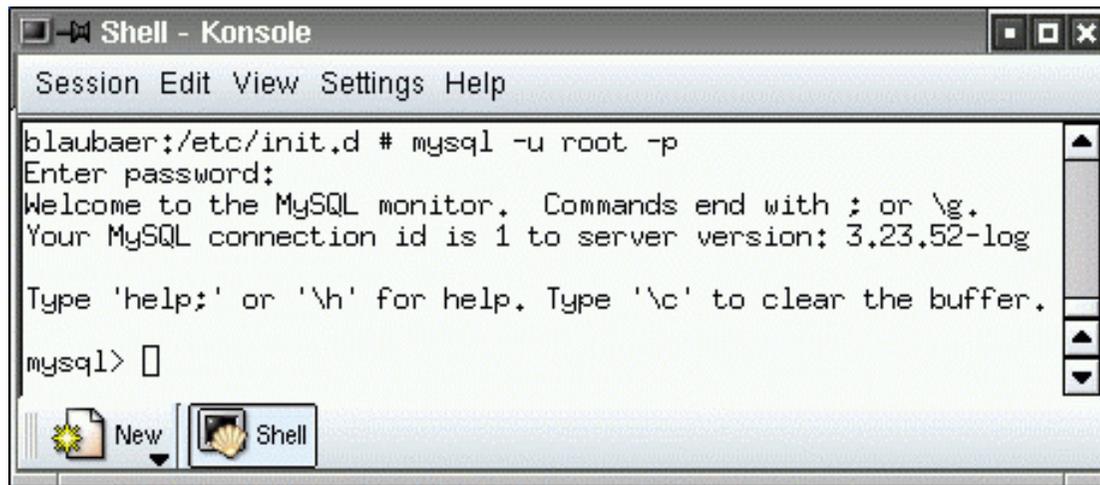
```
blabaer> /etc/init.d/mysql start
```

```
blabaer> /etc/init.d/mysql stop
```

```
blabaer> /etc/init.d/mysql restart
```

Der MySQL-Monitor

- Verbindung zum Server herstellen und trennen
 - Um sich mit dem MySQL-Server zu verbinden, muss beim Aufruf von `mysql` in der Regel ein MySQL-Benutzername und üblicherweise auch ein Passwort angegeben werden.
 - Wenn der Server auf einer anderen Maschine läuft als der, von der der MySQL Monitor gestartet wird, muss auch ein Hostname angegeben werden.



```
Shell - Konsole
Session Edit View Settings Help
blaubauer:/etc/init.d # mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1 to server version: 3.23.52-log

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> 
```

- Die Verbindung wird mit `quit` getrennt.

MySQL-Befehle

- **Anlegen einer Datenbank** CREATE DATABASE:

```
CREATE DATABASE [IF NOT EXISTS] <NameDerDB>;
```

- Existiert die Datenbank bereits, liefert MySQL eine Fehlermeldung zurück, es sei den die Klausel "if not exists" wurde angegeben.
- Bei einigen Betriebssystemen (z.B. Unix) ist der Name der Datenbank case-sensitiv (DB3567 ist nicht gleich db3567!)
- Der Name der Datenbank darf maximal 64 Zeichen haben

- **Löschen einer Datenbank** DROP DATABASE:

```
DROP DATABASE [IF EXISTS] <NameDerDB>;
```

- **Verwenden einer Datenbank** USE:

```
USE <NameDerDB>;
```

- **Anzeigen der vorhandenen Datenbanken:**

```
SHOW DATABASES;
```

MySQL-Befehle

- Erstellen einer Tabelle CREATE TABLE:

```
CREATE TABLE [IF NOT EXISTS] <TabellenName> (  
    spaltenname1 datentyp [einschränkung],  
    spaltenname2 datentyp [einschränkung],  
    ...  
)
```

- Als [einschränkung] können dabei bestimmte Eigenschaften der Spalte von vornherein festgelegt werden:
 - NULL oder NOT NULL
 - DEFAULT
 - AUTO_INCREMENT
 - UNIQUE
 - PRIMARY KEY
 - etc.

MySQL-Befehle

- Verfügbare Datentypen in MySQL (Kleine Auswahl):

Typ	Beschreibung
TINYINT [UNSIGNED]	-128 .. 127 bzw. 0 .. 255
INT	-2.147.483.648 .. 2.147.483.647 bzw. 0 .. 4.294.967.295
FLOAT	-3.402823466E+38 .. -1.75494351E-38, 0 und 1.75494351E-38 .. 3.402823466E+38
DATE	Datumstyp, YYYY-MM-DD 1000-01-01 bis 9999-12-31
TIME	Zeittyp, HH:MM:SS -836:59:59 bis 838:59:59
TIMESTAMP	Zeitstempel, YYYYMMDDHHMMSS, 1970-01-01 00:00:00 bis irgendwann im Jahr 2037
CHAR(NUM)	Zeichenkette fester Länge mit max. NUM-Stellen, 1 <= NUM <=255
VARCHAR(NUM)	Zeichenkette variabler Länge mit max. NUM-Stellen, 1 <= NUM <=255
TEXT	Eine Textspalte mit max. 65.535 (2 ¹⁶ -1) Zeichen
BLOB	Binary Large Object; großes Binärobjekt (2 ¹⁶ -1)

MySQL-Befehle → phpMyAdmin

- Bevor wir aber mit dem mysql-Monitor weiter machen, ein Hinweis auf ein sehr nützliches Tool: **phpMyAdmin**
- phpMyAdmin erleichtert die Administration und Manipulation von MySQL-Datenbanken
 - Es ist zu bekommen unter der URL <http://www.phpmyadmin.net>
 - phpMyAdmin ist eine browserbasierte Oberfläche zur Bedienung von MySQL über das WWW

