

MySQL Grundkurs



www.mysql.com

© Florian Jurgeit, 2001/2002

Ziele/Inhalte des Kurses

- ◆ Datenbankgrundlagen (ERM, Normalformen, ...)
- ◆ Grundfunktionen von *MySQL*
(Datenbankkontrolle, Datendefinition, Datenmanipulation)
- ◆ (Grafische) Tools für MySQL
(Dezign, MySQL-Front)
- ◆ ODBC (MyODBC) – Zusammenarbeit mit Windows-Programmen
- ◆ Access → MySQL
- ◆ Webdatenbanken (Zugriff mit PHP oder Perl)

Was ist eine Datenbank ?

- ◆ Begriff DB als Bezeichnung für
 - Adresslisten ... Buchungssysteme für Banken
- ◆ Anwendung vom Handheld bis zum Mainframe
- ◆ Oft wird das Frontend als DB bezeichnet

- ◆ *Datenbank = Komponente eines Datenbanksystems für die physische Datenspeicherung.*



Was ist eine Datenbank ?

- ◆ Eine DB ist eine Ansammlung von Daten, die für einen spezifischen Zweck zusammengestellt werden. Eine DB ermöglicht den Zugriff auf Daten ohne die Kenntnis, in welcher Art und Weise die Daten in der Datenbank organisiert sind. Eine DB organisiert bestimmte Zugriffsberechtigungen für die Benutzer. Eine DB gewährleistet die Konsistenz der gespeicherten Daten und verhindert das Auftreten von redundanten Daten.

Einfache Datenbanken

- ◆ Die *csv*-Datei:
 - „Comma separated Values“ (auch *flat-file* DB genannt)
 - Trennzeichen der Felder: Beistrich
 - Jede Zeile ist ein Datensatz
- ◆ Beispiel:(Übung ?)
Vorname, Nachname, email
Sepp, Huber, Sepp.Huber@huber.at
..., ..., ...
- ◆ Vorteile/Nachteile (wichtiges Austauschformat)



Was ist MySQL ?

- ◆ MySQL = RDBMS (Relationales Datenbank Management System)
- ◆ Datenbanksystem: besteht aus einer Datenbank, dem Datenbankverwaltungssystem und einer Kommunikationsschnittstelle



Warum ein RDBMS ?

- ◆ Desktopsysteme (zB Access) haben Grenzen...
 - im Multiuser-Betrieb
 - Sicherheitsbedenken
 - bei großen Datenmengen



Wie funktioniert MySQL ?

- ◆ Client-Server Architektur
- ◆ Datenbank agiert als Server
- ◆ Client-Software stellt Verbindung her
- ◆ Kommunikation mit der DB durch eine einheitliche Sprache (SQL)

Das RDBMS MySQL

- ◆ Vollwertiges RDBMS (für bis ~1Mio Datensätze)
- ◆ SQL als Grundlage (ANSI-SQL)
- ◆ Unterliegt der GPL(open source)
 - Verwendung für nichtkommerzielle Nutzung frei
- ◆ Plattformvielfalt (Windows, Linux, Mac,...)
- ◆ Weite Verbreitung als Webdatenbank
- ◆ Beispiele für die Verwendung (im Web...)

Datenbankdesign

„Von der Realwelt zur Datenbank“

- ◆ Ziel: Realwelt (Objekte – Beziehungen) in eine digitale Form zu überführen
- ◆ Daher: ERM (Entity Relationship Model) und folgend das RDM (Relationship Data Model)
- ◆ *Designfehler* führen zu Redundanzen, Verlust von Information, Einfügen von Nullwerten, ...

Datenbankdesign – Anomalien

KURS						
<u>Kurs_Nr</u>	<u>Kurs_Bez</u>	Halbjahr	Doz_Kürzel	Doz_NN	Doz_VN	Doz_Tel
47	Windows-Einführung	95/2	SR	Raben	Susanne	7148554
103	Business-English I	95/2	WK	Kühnel	Winfried	3397631
142	Windows-Einführung	96/1	SR	Raben	Susanne	7148554
144	DOS-Einführung	96/2	AH	Huller	Anna	4788497
153	Englisch auf Reisen	96/2	WK	Kühnel	Winfried	3397631

Lösch-Anomalie:
Löschen von Kurs 144
löscht evtl. einzigen Eintrag
von A. Huller, und damit ungewollt
auch die Dozentin

Einfüge-Anomalie:
Neue Dozentin Tina Wolf
eintragen...

Änderungs-Anomalie:
Tel.-Nr. von Herrn
Kühnel ändern...

Knorz, G.: Datenbank-Entwurfsmethoden. Kapitel 4 in: Buder/Rehfeld/Seeger/Strauch: Grundlagen der praktischen Information und Dokumentation. Band 2, 4. Ausgabe. Saur Verlag, 1996, S. 664-687

Datenbankdesign - Beispiel

- ◆ Literaturdatenbank – erster Versuch:

Books					
Titel	Jahr	Hrsg	Autor1	Autor2	Autor3
MySQL	1999	Oreily	Huber		
Windows	2000	BHV	Muster	Test	
Office	1997	BHV	Huber	Maier	Test

- ◆ Verbesserungen ?!! (→ 1. Normalform)

Datenbankdesign - Beispiel

- ◆ 1. Normalform: Spalten mit gleichartigem Inhalt werden ausgeschieden und jeder Datensatz muss durch einen Primärschlüssel gekennzeichnet sein.

			Books			
ID	Titel	Jahr	Hrsg	Autor		
01	MySQL	1999	Oreily	Huber		
02	Windows	2000	BHV	Muster		
03	Office	1997	BHV	Huber		

Datenbankdesign - Beispiel

◆ Annäherung an die 2. NF:

Books				
ID	Titel	Jahr	Hrsg	
01	MySQL	1999	Oreily	
02	Windows	2000	BHV	
03	Office	1997	BHV	

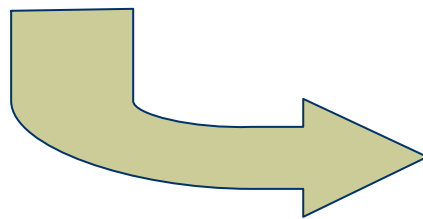
Autor		
Autor_ID	Titel_ID	Autor
1	01	Huber
2	03	Huber

Datenbankdesign - Beispiel

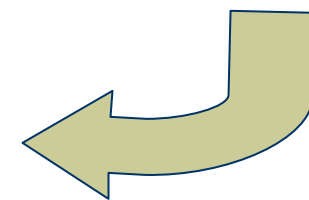
◆ Die 2. Normalform:

Books				
ID	Titel	Jahr	Hrsg	
01	MySQL	1999	Oreily	
02	Windows	2000	BHV	
03	Office	1997	BHV	

Autor	
Autor_ID	Autor
1	Huber
2	Muster
3	Maier
4	Test



Titel_ID	Autor_ID
01	1
02	2
02	4
03	1



Datenbankdesign - Beispiel

◆ Die 3. Normalform:

Books				
ID	Titel	Jahr	Hg_ID	
01	MySQL	1999	1	
02	Windows	2000	2	
03	Office	1997	2	

Hg_ID	Hrsg
1	oReily
2	BHV

Normalformen-Definitionen

- ◆ 1. Normalform: Eine Relation befindet sich in der ersten Normalform, wenn keines ihrer Attribute eine untergeordnete Relation darstellt und wenn alle Attribute nur atomare Werte beinhalten.
- ◆ 2. Normalform: Laut Definition muss die Datenbank immer zuerst in die erste Normalform versetzt werden, bevor man diese in die 2. Normalform versetzen kann. Hierbei müssen alle nicht zum Schlüssel gehörenden Attribute von diesem voll funktional abhängig sein. Besteht ein Schlüssel aus mehreren Teilschlüsseln, so ist das Element aus dem Datensatz herauszuziehen, welches nur von einem Teilschlüssel abhängt.
- ◆ 3. Normalform: Zusätzlich zur 2. Normalform gilt für jeden Schlüssel: Alle nicht zum Schlüssel gehörende Attribute sind nicht von diesem transitiv abhängig. Das bedeutet, dass alle Attribute nur vom Schlüsselattribut, nicht aber von anderen Attributen abhängig sein. Eine Abhängigkeit zwischen den Attributen muss aufgelöst werden.



Schlüsselfelder

- ◆ **Primärschlüssel: eindeutige Kennung** für jeden Datensatz
 - für jeden Datensatz anderer Wert
 - darf nicht leer sein
 - daher: **meist laufende Nummer**



Relationen

- ◆ Relationen machen eine DB der Tabellenkalkulation(Tabelle) überlegen
- ◆ Eine Relation zwischen zwei Tabellen entsteht dadurch, dass eine Tabelle ein Feld mit Primärschlüsselwerten einer anderen Tabelle enthält.

Relationen

- ◆ **1:1-Relation(Eineindeutig):** Einem Objekt vom Typ A ist genau ein Objekt vom Typ B zugeordnet
- ◆ **1:n(Funktional):** Jedes Objekt vom Typ A kann mit beliebig vielen vom Typ B in Beziehung stehen. Jedes Objekt vom Typ B kann jedoch nur mit genau einem Objekt vom Typ A in Beziehung stehen.
(Fremdschlüssel – Primärschlüssel)
- ◆ **M:n(Komplex):** Jedem Objekt vom Typ A können beliebig viele vom Typ B zugeordnet werden - und detto in die andere Richtung.(muss in 1:n aufgelöst werden)



DB-Design Übung

- ◆ Datenbank mit mindestens 3 Tabellen entwerfen – stelle die Beziehungen grafisch dar und gib ihren Typ an.

Installation von MySQL

- ◆ Download als **Binary** oder Source-Code für das entsprechende OS (www.mysql.com)
- ◆ Komponenten, die man erhält:
 - MySQL-Server: Datenbanksystem zur Verwaltung von Daten in einer DB
 - Client-Progs: Progs zur Administration und Abfrage. SQL-Kommunikationsschnittstelle
 - Zusatzsoftware: GUIs für MySQL, ...

Installation(win32)

- ◆ ZIP-Datei entpacken – SETUP ausführen
- ◆ Verzeichnisstruktur (..\mysql\):
 - \bench: Benchmarking Tools
 - \bin: ausführbare Programmdateien (u.a. der Server !!!)
 - \data: sämtliche Daten einer DB
 - \docs: Referenzen, ...(Referenz als HTML-File !!!)
 - \examples: Testprogramme für C/C++ API
 - \include: Dateien in C für API und ODBC
 - \lib: Bibliotheken f. C/C++ API
 - \scripts: Admin.-Progs in PERL(Compiler erforderlich)
 - \share: Fehlermeldungen

Server starten

- ◆ Win9x: `\mysql\bin\mysqld` in der Eingabeaufforderung
- ◆ WinNT/2000/XP: `\bin\mysqld-nt -install`
(`NET START mysql; NET STOP mysql`)
- ◆ Überprüfen: `\bin\mysqlshow`



```
C:\>cd mysql\bin
C:\mysql\bin>mysqlshow
+-----+
| Databases |
+-----+
| flo_test  |
| mysql    |
| test     |
+-----+
C:\mysql\bin>
```


Client starten

- ◆ ... \mysql\bin\mysql ausführen

```
C:\mysql\bin>mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3 to server version: 3.23.43-nt


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

mysql> status;
-----
mysql  Ver 11.15 Distrib 3.23.43, for Win95/Win98 (i32)

Connection id:          3
Current database:
Current user:           ODBC@localhost
Server version:        3.23.43-nt
Protocol version:      10
Connection:            . via named pipe
Client characterset:   latin1
Server characterset:   latin1
UNIX socket:           MySQL
Uptime:                9 min 39 sec

Threads: 1  Questions: 7  Slow queries: 0  Opens: 6  Flush ta
s: 0  Queries per second avg: 0.012
-----

mysql>
```



MySQL-Hinweise

- ◆ Im Client:
 - Befehle immer mit ; abschließen !!!
- ◆ Auf Groß-/Kleinschreibung achten !!!

Datenbankkontrolle in MySQL

- ◆ Vorhandene Datenbanken anzeigen:
`show databases ;`
- ◆ Datenbank erstellen:
`create database database_name ;`
- ◆ Datenbank löschen:
`drop database database_name ;`
- ◆ Datenbank auswählen:
`use database_name ;`
- ◆ Benutzerverwaltung/Rechte(s. Hilfe): GRANT, REVOKE



Datenbank anlegen - Übung

- ◆ Versuchen Sie eine Datenbank in MySQL anzulegen.
- ◆ Überprüfung auf vorhandene Datenbanken
- ◆ Löschen Sie die angelegte Datenbank
- ◆ Legen Sie die Datenbank erneut an

Datendefinition in MySQL

- ◆ Tabelle anlegen und Tabellenstruktur definieren:
`create table table_name(Felder,...);`
- ◆ **Wichtige Datentypen in MySQL:**
 - CHAR(M): Zeichenkette fester Länge mit max. 255 Zeichen
 - VARCHAR(M): Zeichenkette variabler Länge mit max. 255
 - TEXT: Text variabler Länge (max. 65535 Zeichen)
 - DATE: Datumsangaben – spezifisches Format
 - YEAR: Jahreszahlen aus 4 Ziffern
 - INTEGER: ganze Zahl mit oder ohne Vorzeichen
 - FLOAT: Fließkommazahl mit oder ohne Vorzeichen
 - BLOB: Binärdaten mit max. 65535 Bytes

Tabelle definieren - Übung

```
mysql> create table flo1 (ID int not null  
    auto_increment, NAME char(30), PLZ int  
    , primary key(ID));
```

Query OK, 0 rows affected (0.06 sec)

- ◆ Tabelle flo1 wird erstellt mit folgenden Feldern:
 - ID vom Typ Integer, darf nicht NULL sein, fortlaufende Nummerierung
 - NAME vom Typ Character mit der Länge 30
 - PLZ vom Typ Integer
 - Feld ID wird als Primärschlüssel definiert !!!

Datendefinition

- ◆ Tabellen anzeigen: `show tables;`
- ◆ Tabelle löschen: `drop table table_name`
- Bsp.: `drop table flo1;`
- ◆ Datenstruktur einer Tabelle anzeigen: `explain`
- Bsp.: `explain flo1;`
- ◆ Index erstellen/löschen: `create/drop`
- Bsp.: `create index flo1_index on flo1(name);`



Datendefinition – Übung(2)



- ◆ Wie kann man sich die Datenstruktur einer Tabelle anzeigen lassen – Was bedeuten die Angaben ?
- ◆ Wie erhält man eine Übersicht über die Tabellen in einer Datenbank ?
- ◆ Löschen Sie eine zuvor erstellte Tabelle

Datendefinition

- ◆ Datenstruktur einer Tabelle ändern:
`alter table table_name aktion;`
- ◆ Mögliche Aktionen:
 - `rename Tabelle;`
 - `add Spaltendefinition;`
 - `drop Spalte;`
 - `add primary key (Spalte);`
 - `drop primary key (Spalte);`
 - `change Spalte_alt Sp_neu Definition;`
 - `modify Spalte Definition;`
 - `alter Spalte {set/drop default wert};`

Datendefinition – Übung(3)

- ◆ Erstellen Sie eine Tabelle „Buch“ mit folgenden Feldern:
 - id: integer, fortlaufend, PK
 - autor: char
 - titel: char
 - preis: float
 - jahr: year
- ◆ Überprüfen Sie die Tabellenstruktur
- ◆ Benennen Sie diese Tabelle in „buchliste“ um

Datendefinition – Übung(3)

- ◆ Fügen Sie der Tabelle eine Spalte(Feld) des Typ CHAR hinzu mit der Bezeichnung „verlag“
- ◆ Löschen Sie eine Spalte(verlag) der Tabelle
- ◆ Entfernen Sie den Primary-Key (Beachte: die Spalte bleibt erhalten ! Probleme !??)
- ◆ Fügen Sie den gelöschten PK wieder ein

Datendefinition – Übung(3)

- ◆ Modifizieren Sie die Spalte „buch“ – benennen diese um („buchtitel“) und verändern die Länge des Typs CHAR
- ◆ Legen Sie für den Preis einen Standard-Wert von 10 € fest
- ◆ Überprüfen Sie die getätigten Änderungen in der Tabellestruktur



Datenmanipulation

- ◆ Natürlich soll eine erstellte Tabellenstruktur auch mit Daten „gefüttert“ werden und abgefragt werden.
- ◆ Grundlegende Operationen:
 - INSERT
 - UPDATE
 - DELETE
 - LOAD DATA

Datenmanipulation

- ◆ Datensatz einfügen: **INSERT**
 - Syntax: `INSERT [INTO] Tabelle (Spalten) VALUES (Werte);`
 - Bsp.: `INSERT INTO buch (id, autor, titel, ...) VALUES (1, "Huber", „MySQL“, ...);`
 - Beachte: **ID** → **Auto_increment**; daher:
`INSERT INTO buch (Autor, titel, ...) VALUES („Huber“, „MySQL“, ...);`
 - Beachte: **Default-Werte**

Datenmanipulation-Übung

- ◆ Füllen Sie die Tabelle „buchliste“ mit Werten (Beachte die Felder ID und Preis)
- ◆ Teste die Insert-Anweisung ohne Attributliste
- ◆ Teste die Default-Wert Funktion beim Attribut „preis“ – daher: Datensatz ohne Preisangabe und einen mit einem vom Default-Preis abweichenden Wert.
- ◆ Versuche mehrere Datensätze in einem ‚Wisch‘ einzufügen `... VALUES (Werte) , ... , (Werte) ;`
- ◆ Ziel: 5 Datensätze sollten in der Tabelle sein

Datenmanipulation

◆ Datenbank abfragen: SELECT

- **Syntax:** SELECT [DISTINCT]
{*/Spalte/Tabelle.Spalte} FROM Tabelle
WHERE Bedingung [GROUP BY Spalte]
[HAVING Bedingung] [ORDER BY Spalte]
[ASC/DEC]

- ein einfaches Beispiel:

```
select * from Tabelle;
```

Zeigt alle Datensätze der Tabelle an !



Datenmanipulation - Übung

- ◆ Frage alle Datensätze deiner Tabelle „buchliste“ (buch) ab !

Datenmanipulation

- ◆ **SELECT** – die Argumente(1):
 - **DISTINCT**: im Ergebnis werden mehrfach auftretende, identische Datensätze nur 1x angezeigt
 - **FROM**: definiert eine Tabelle oder mehrere Tabellen (Verknüpfung)
 - **WHERE**: Abfragebedingungen
 - **GROUP BY**: Datensätze werden gruppiert



Datenmanipulation

- ◆ **SELECT** – die Argumente(2):
 - **HAVING**: Bedingungen für die Gruppe
 - **ORDER BY**: Legt die Sortierung der Datensätze fest
 - **ASC/DESC**: Sortierung aufsteigend/absteigend

Datenmanipulation-Übung

- ◆ WH: Frage alle Datensätze ab
- ◆ Es sollen nur die Spalten Autor und Titel angezeigt werden
- ◆ Alle Datensätze – jedoch sortiert nach dem Autor (Aufsteigend und Absteigend)

Lösungen folgen...😊

Datenmanipulation-Übung

◆ Lösungen:

1) `select * from Tabelle;`

2) `select autor, titel from Tabelle;`

3) `select * from Tabelle order by autor
asc/desc;`

Datenmanipulation

- ◆ **SELECT – wichtige Ausdrücke:**

- Wozu Ausdrücke ?

Diese dienen der Durchführung von Berechnungen oder Vergleichsoperationen zB mit der SELECT-Anweisung.

Diese Aufgabe übernehmen später die Programmapplikationen.

Datenmanipulation

- ◆ SELECT – wichtige Ausdrücke:
 - Rechenoperationen: +, -, *, /
 - Statistik: AVG, MAX, MIN, SUM, COUNT
 - Logik-Vergleichsop.: =, <>, <, >, <=, >=
 - Verknüpfungsop.: AND, OR, NOT
 - Sonstige: IN, LIKE, BETWEEN

Datenmanipulation-Übung

- ◆ Angenommen die Preise sind in €- wir wollen aber die ATS-Preise wissen und zwar in Form einer Ausgabe mit Titel und Preis(ATS)
- ◆ Wie oben – nur die Preise sind Nettopreise, uns interessieren diese aber mit 20% Steuer
- ◆ Wir wollen den kleinsten Preis, den Größten und den Durchschnittspreis der Bücher in unserer Tabelle wissen.



Datenmanipulation-Übung

- ◆ Die Zahl der Datensätze soll ermittelt werden
- ◆ Wir wollen wissen welche Bücher ein bestimmter Autor verfasst hat
- ◆ Wie viele Bücher hat ein Autor verfasst ?
- ◆ Welche Bücher sind teurer/billiger als der Default-Preis(oder irgendein Wert) ?
- ◆ Welche Bücher sind teurer als der Durchschnittspreis ?



Datenmanipulation-Übung

- ◆ Welche Bücher hat ein bestimmter Autor in einem bestimmten Jahr verfasst ?
- ◆ Alle Bücher ohne die eines bestimmten Autors
- ◆ Bücher von bestimmten Autoren abfragen (mehrere in einer Abfrage) – 2 Möglichkeiten



Datenmanipulation-Übung

- ◆ Eine String-Suche (Zeichenkettenvergleich):
Suche Sie nach Titeln die mit einem bestimmten Buchstaben beginnen !!
- ◆ Suche Sie Bücher die in einem Zeitraum (untere/obere Jahresschranke) verlegt worden sind (2 Möglichkeiten)
- ◆ Durchschnittspreis der Bücher der Autoren ?

Datenmanipulation

- ◆ Inhalt eines Datenfeldes ändern: UPDATE
 - Syntax: UPDATE Tabelle SET Spalte1=Wert1, ... WHERE Bedingung
- ◆ UPDATE und die WHERE-Bedingung liefern ähnliche Möglichkeiten wie bei der SELECT-Anweisung.
- ◆ Beispiel:

```
update buchliste set jahr=„2002“  
where titel = „MySQL“;
```



Datenmanipulation-Übung

- ◆ Ändern Sie das Erscheinungsjahr eines Buches
- ◆ Ändern Sie alle Preise eines gewissen Wertes (evtl. die mit Default-Wert 10) auf 12€
- ◆ Ändern Sie den Buchtitel eines bestimmten Autors
- ◆ Fügen Sie eine Spalte mit einem Preis in einer anderen Währung ein – füllen Sie diese mit dem Umrechnungspreis.

Datenmanipulation

- ◆ Datensätze löschen: DELETE
 - Syntax: DELETE FROM table [WHERE Bedingung];
- ◆ Beispiel:

```
delete from buchliste where  
titel=„mysql“;  
delete from buchliste; (Löscht alle  
Datensätze)
```



Datenmanipulation-Übung

- ◆ Wie kann man alle Datensätze einer Tabelle löschen ? (nicht ausführen !)
- ◆ Lösche die Bücher eines bestimmten Autors
- ◆ Lösche ein Buch eines Autors, der mehrere Bücher verfasst hat.

Datenmanipulation

- ◆ Datensätze aus einer Datei einlesen: **LOAD DATA**
 - Wichtig: Welche Trennzeichen trennen die Felder und die Datensätze (zB.: , zwischen den Feldern, \n trennt die Datensätze)
 - **Syntax:** `LOAD DATA [LOCAL] INFILE 'Datei' INTO TABLE Tabelle [Options]`

Datenmanipulation

- ◆ LOCAL: lokale Datei wird an Server übermittelt.
- ◆ LOAD DATA – Options:
Definieren das Datenformat; keine Angaben führen zur Verwendung der Standard-Trennzeichen(Tab und Zeilenumbruch)
 - FIELDS TERMINATED BY „Trennz.“
 - LINES TERMINATED BY „Trennz.“

Datenmanipulation-Übung

- ◆ Erstelle in einem Editor eine Textdatei, die Datensätze für die Tabelle buchliste enthält – als Trennzeichen werden Tabulator und Zeilenumbruch verwendet
- ◆ Importiere diese Datensätze in die Tabelle buchliste
- ◆ (Wie oben – nur mit einem anderen Spaltentrennzeichen)

Daten sichern

- ◆ Natürlich muss auch bei MySQL auf Datensicherheit geachtet werden – daher ist eine Datensicherung ratsam !
- ◆ Datensicherung mit MySQL:MYSQLDUMP
 - mysqldump ist ein Dienstprogramm und befindet sich im Verzeichnis `\bin` !
 - Die Sicherung erfolgt in eine Textdatei, die somit mit jedem Editor betrachtet werden kann !!!

Daten sichern

- ◆ **Syntax:** `mysqldump [Options]`
Datenbank > Textdatei
- Options: `-A` speichert alle DBs des Servers
- ◆ **Beispiel:**
`mysqldump DB_name > File.txt`



Daten sichern - Übung

- ◆ Sichern Sie ihre Datenbank mit der Tabelle buchliste
- ◆ Betrachten Sie das Sicherungsfile mit einem Editor – was befindet sich im File, versuchen Sie dies ein wenig zu interpretieren

Gesicherte Daten einlesen

- ◆ Dies kann direkt über den mysql-client erfolgen:

```
mysql database < File.txt
```



Datensicherung - Übung

- ◆ Legen Sie eine neue Datenbank an.
- ◆ Lesen Sie die gesicherten Buchdaten in diese ein
- ◆ Überprüfe die erfolgreiche Durchführung

Tabellenverknüpfung

- ◆ Tabellenverknüpfungen ermöglichen uns erst das relationale Datenmodell zu definieren !
- ◆ Wiederholung: die 1:n und m:n Relation
- ◆ Durchführung über eine Abfrage (select)
`select * from Tabellen where ...`

Tabellenverknüpfung

- ◆ **Beispiel: Buch – Herausgeber (1:n)**

```
select * from buch, hrsg where buch.b_id  
= hrsg.hg_id;
```

- ◆ **Left/Right JOIN:**

```
select * from buch left join hrsg on  
buch.b_id = hrsg.hg_id;
```

Tabellenverknüpfung-Übung

- ◆ Erstelle jeweils eine Tabelle ‚buch‘ und ‚verlag‘
(Achtung: 1:n ...)
- ◆ Gib jeweils Werte ein
- ◆ Führe eine Verknüpfungsabfrage durch
- ◆ Führe einen JOIN durch (left und right) –
Finde heraus was ein JOIN ist ! Notiere deine Erkenntnisse !

Tabellenverknüpfung-Übung(2)

- ◆ Führe eine Verknüpfungsabfrage durch, die alle Bücher eines bestimmten Verlages ausgibt (Buchdaten und Verlagsdaten)
- ◆ Erstelle eine Tabelle ‚Autor‘, gib Werte ein
- ◆ Erstelle eine Relationstabelle ‚buch_autor‘ und gib Werte ein
- ◆ Führe eine Verknüpfungsabfrage durch (m:n)



ODBC

- ◆ ODBC (=Open DataBase Connectivity)
- ◆ Was ist ODBC ?
 - eine Art Treiber, von MS entwickelt
 - standardisierte Programmschnittstelle (Kommunikationsschnittstelle) für Datenbanksysteme

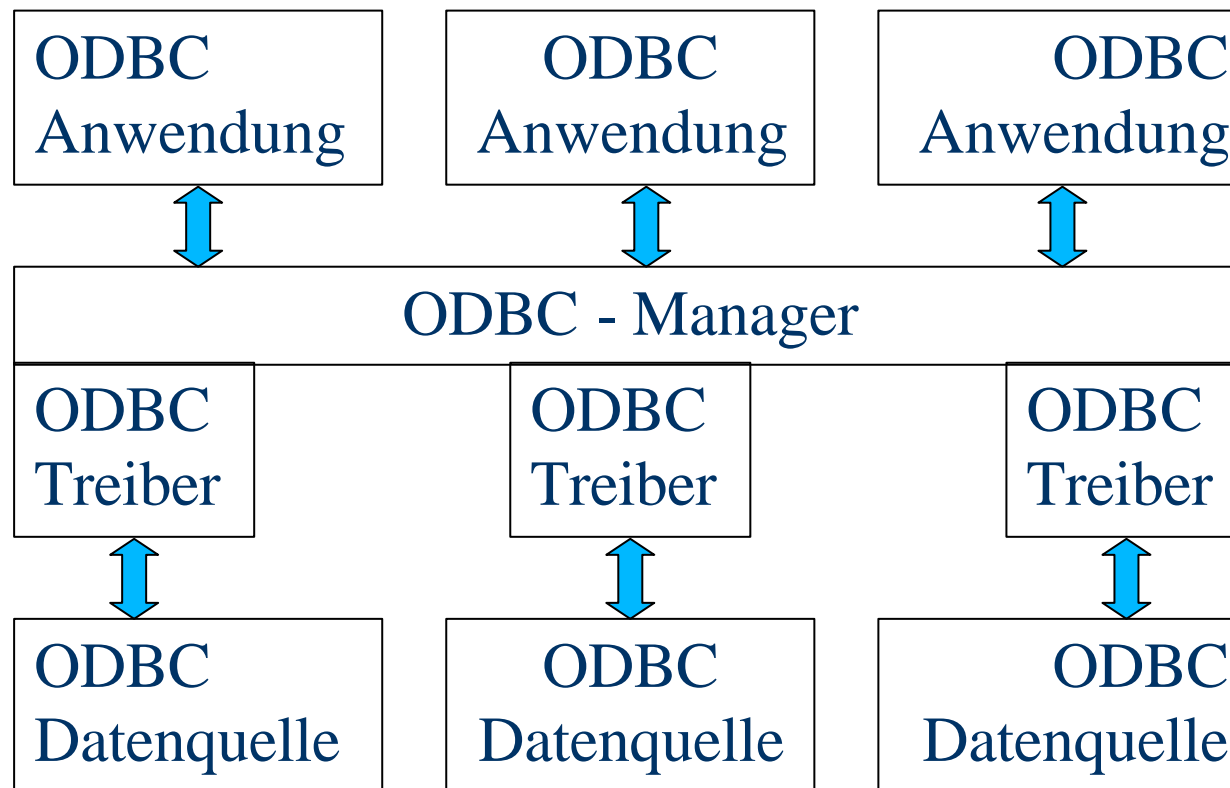


ODBC

- ◆ Vorteile:
 - Plattformübergreifend
 - Netzwerkfähig
 - Standardisierte Kommunikation (SQL)
 - Zugriff auf MySQL-DB über grafisches Front-End möglich

ODBC

◆ Architektur:



ODBC f . MySQL

- ◆ Treiber wird MyODBC genannt
- ◆ Unter www.mysql.com erhältlich
- ◆ Installation/Einrichten
 - nach dem Setup: in der Systemsteuerung unter Datenquellen (bei NT Verwaltung – Datenquellen)

MyODBC und Access - Übung

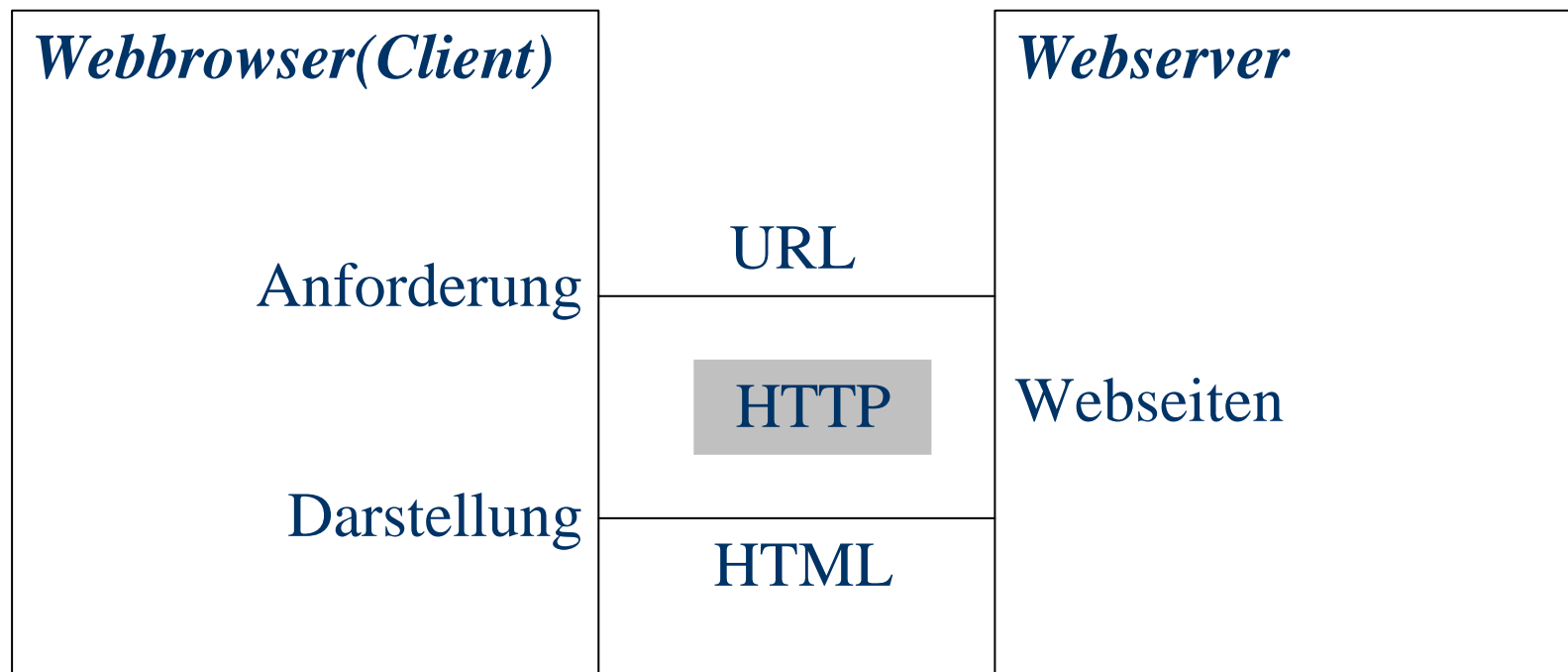
- ◆ Eine ODBC Datenquelle einrichten
- ◆ Versuche von Access aus eine Verbindung zur MySQL-DB herzustellen
 - Hinweis: Externe Daten – Tabellen verknüpfen
- ◆ Füge Datensätze hinzu
- ◆ Führe eine Abfrage durch – auch mit SQL-Pass-Through

Webdatenbanken

- ◆ Was sind Webdatenbanken ?
 - DB-Anbindung an WWW
 - GUI für die DB wird über den Webbrowser dargestellt (kein Front-End Programm notwendig; einfache Bedienung)
 - Verwendung: meist dynamische Webseiten (vom Gästebuch bis zum Online-Shop)
 - Intranet-Anwendungen

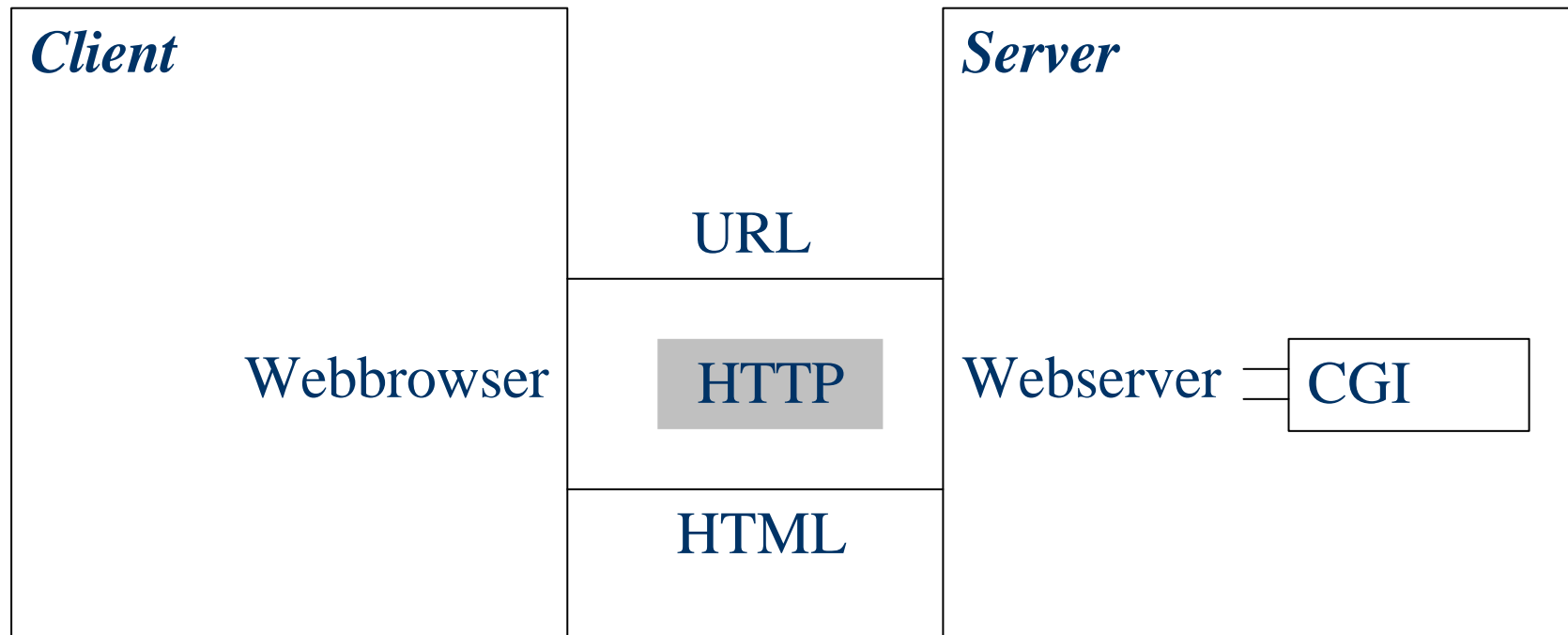
Webdatenbanken - Technik

- ◆ Client-Server Prinzip im WWW:



Webdatenbanken - Technik

- ◆ Client-Server bei Webdatenbanken:
 - CGI: Common Gateway Interface





Webdatenbanken - Technik

- ◆ Systemvoraussetzungen-Webserver:
 - Webserver (zB Apache)
 - CGI: zB PHP oder Perl
 - Datenbankserver (zB MySQL)

Webdatenbanken - Übung

- ◆ Installation des
 - Webservers (Apache)
 - der Script-Sprache PHP
 - Konfiguration von Apache
- ◆ Verzeichnisstruktur von Apache (htdocs !)
Hinweis Linux: /usr/local/httpd/htdocs
- ◆ Apache testen: Localhost (127.0.0.1)

HTML-Grundlagen

- ◆ Ist bekannt ?!!
- ◆ Wichtig f. Dynamische Webseiten:
 - Formular-Tags in HTML (siehe SelfHTML: www.teamone.de/selfhtml)
- ◆ Übung: Eigene Startseite (index.html) erstellen und ins /htdocs stellen (+testen)
Diese wird den Ausgangspunkt für die weiteren Übungen bilden !!!

PHP-Grundlagen

- ◆ Was ist PHP ?
 - PHP: *Hypertext Preprocessor*
 - Scriptsprache in HTML eingebettet
 - OpenSource !!! (www.php.net)
 - PHP-Code wird vom Server interpretiert
(Client sieht Code nicht !)
 - gute DB-Anbindung (MySQL, Oracle, ...)
 - Endung: *.php (Server sucht nach Code)

PHP-Grundlagen

- ◆ Einbindung in HTML erfolgt über den „Tag“
`<?php`
...
...
`?>`
- ◆ Textausgabe: `print "Text";` oder
`echo "Text";`
- ◆ Beachte: Befehlszeilen enden mit `;`

PHP-Grundlagen Übung

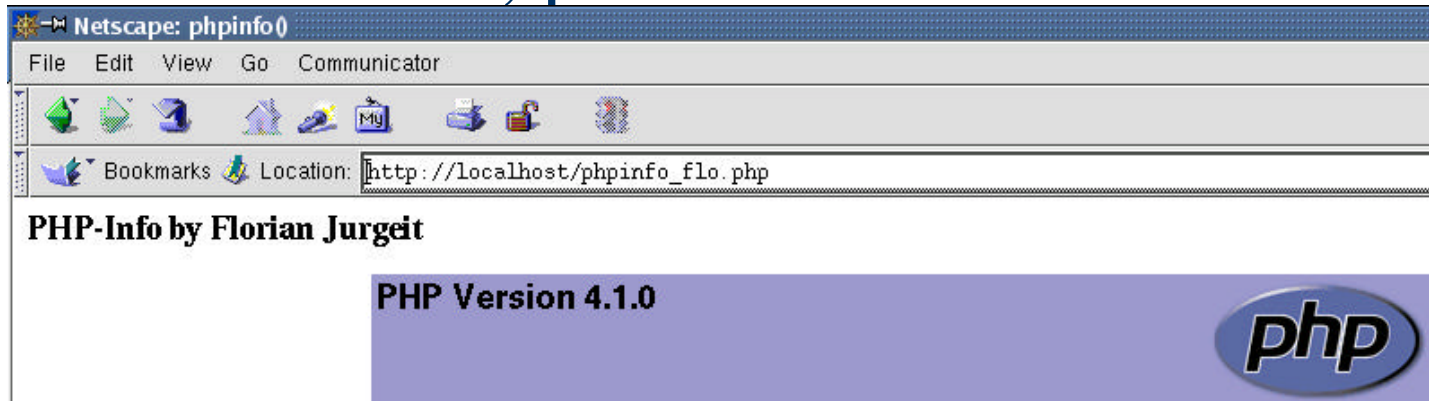
- ◆ Wir wollen nun den installierten PHP-Parser testen:

```
<?php  
phpinfo( ) ;  
?>
```
- ◆ Als Datei info.php speichern (/htdocs) und im Browser aufrufen

PHP-Grundlagen Übung

- ◆ Erweitere das info-Script mit einer Textausgabe, die vor den Infos folgendes ausgibt: PHPinfo von Name ...

Hinweis: echo, print



PHP-Grundlagen

- ◆ Formatierung der Textausgabe...
 - erfolgt mit HTML-Tags
 - kann direkt in die Ausgabeanweisungen integriert werden !
 - Bsp.: `print "Hello world !";`
 - Achtung: Tags die " beinhalten – durch ' ersetzen: zB: `print "";`

PHP-Grundlagen Übung

- ◆ Formatiere beim Script info.php deine individuelle Ausgabe (Name fett)
- ◆ Erweitere die Ausgabe um einen Link auf die Homepage von PHP (www.php.net) - so soll's aussehen:
PHPinfo von **Name** ... (www.php.net)
- ◆ Erweitere den Link so, dass die Seite php.net in einem neuen Fenster geöffnet wird

PHP-Grundlagen

- ◆ **Php-Code in einer HTML-Datei:**

```
<html>
<head>
<title>PHP-Test</title>
</head>
<body>
<?php
...
?>
</body>
</html>
```



PHP-Grundlagen Übung

- ◆ Erstelle eine Datei info2 mit „schönem“ HTML als Grundlage
- ◆ Gib den individuellen Text noch im HTML-Teil aus (nicht im php-Bereich)

PHP-Grundlagen

- ◆ Kommentare in PHP:
 - über eine Zeile: `// Kommentar`
 - über mehrere Zeilen: `/* Kommentar */`
- ◆ Namen: wie üblich...
 - keine reservierten Wörter
 - keine Leerzeichen
 - keine Sonderzeichen

PHP-Grundlagen

- ◆ Variablen: \$-Zeichen vorangestellt !
 - Zuweisungsoperator: =
 - Datentyp wird anhand des Wertes erkannt
- ◆ Beispiele:

```
$x = 10; $name = "Huber"; $phi = 3.14;
```


PHP-Grundlagen

- ◆ Arithmetische Operatoren: +, -, *, /, ++, --

- ◆ Beispiele:

```
$x = 2; $y = 3; $z = $x+$y;
```

```
$i = 1; $i++;
```

PHP-Grundlagen

- ◆ Zeichenkettenoperatoren:

- Verkettung: .

- Zeichenketten erweitern: .=

- ◆ Beispiele:

```
$vname = "Sepp" ; $nname = "Huber" ;
```

```
$name = $vname . "_" . $nname ;
```

```
$name2 = "sepp" . "_" ; $name2 .= $nname ;
```

PHP-Grundlagen Übung

- ◆ Erstelle ein php-Script: es sollen 2 Variablen definiert werden (vname, nname) und Werte zugewiesen werden (dein Name). Diese sollen dann ausgegeben werden:

Meine Name ist vname nname.

PHP-Grundlagen

- ◆ Programmablaufstrukturen:
 - Fallunterscheidung: IF
`If (Bedingung) {Anweisungsfolge}`
`else {Anweisungsfolge};`
- ◆ Mögliche Vergleichsoperatoren:
`==, !=, <, >, >=, <=`
- ◆ Logische Operatoren:
`!, &&, ||` (nicht, und, oder)

PHP-Grundlagen

- ◆ Programmablaufstrukturen:
 - Zählschleife: FOR
`for (Startwert; Bedingung; Änderung)
{Anweisungen};`
- ◆ Übung: php-Script, das die Variable `i` bis 100 hochzählt und jeden Wert ausgibt.

PHP-Grundlagen

- ◆ Programmablaufstrukturen:
 - Kopfschleife: WHILE
- ```
while (Bedingung)
{Anweisungen};
```

# PHP-Grundlagen

- ◆ Wertübergabe aus HTML-Dateien:

- Formulare: name = Variable !
- method: post
- action: scriptname

Das PHP-Script erkennt die

Formularfeldnamen als Variablen ! (zB:

Feldname: name=a → im PHP-Script: \$a)

# PHP-Grundlagen Übung

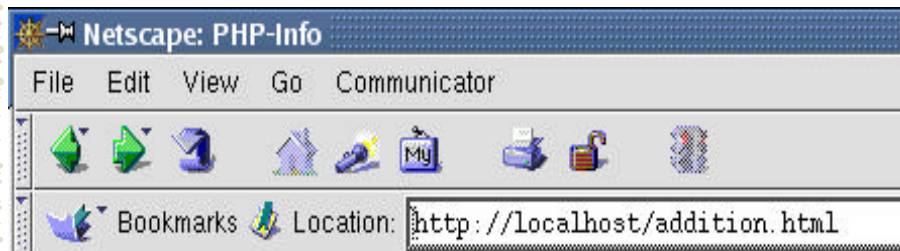
- ◆ Erstelle eine einfache HTML-Datei mit einem Formular: 2 Eingabefelder für die Variablen a und b !
- ◆ Erstelle ein Script summe, das aus der HTML-Datei aufgerufen wird (action) und die Variablen a und b addiert und das Ergebnis ausgibt.



# PHP-Grundlagen Übung

- ◆ Erweitere das Script und die HTML-Datei so, dass der Benutzer 2 Zahlenwerte eingeben kann und zwischen 2 Rechnungsarten wählen kann (Addition/Multiplikation) – entsprechende Berechnung soll durchgeführt werden und das Ergebnis angezeigt werden (+ evtl. Angabe der gewählten Rechnungsart)

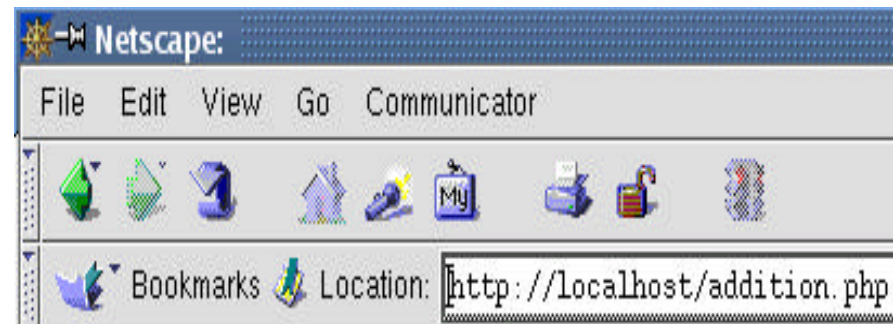
# PHP-Grundlagen Übung



## 2 Zahlen eingeben und einen Operator auswählen

- ◆ Addition
- ◆ Multiplikation



Die Multiplikation von 2 und 3 ergibt 6



---

# PHP-Grundlagen

---

- ◆ Die MySQL-PHP API:
  - stellt Funktionen zum DB-Zugriff zur Verfügung
  - Auf- und Abbau der Verbindung zum DB-Server
  - Datenbankabfrage
  - ...

# PHP-Grundlagen

- ◆ MySQL-PHP API: Verbindung zum DB-Server
  - `mysql_connect()`: Verbindung herstellen  
`mysql_connect( "Host", "User", "Passw." );`
  - `mysql_close()`: Verbindung beenden

# PHP-Grundlagen

- ◆ MySQL-PHP API: Administration der DB
  - `mysql_create_db()`;
  - `mysql_drop_db()`;
  
  - `mysql_select_db()`: Wählt eine DB aus (vgl. USE-Befehl in MySQL)  
`mysql_select_db( "datenbankname" );`

# PHP-Grundlagen

- ◆ MySQL-PHP API: Datenbankabfrage
  - `mysql_query()`: Datenbankabfrage stellen

```
mysql_query("SQL-Befehl");
```

Beispiel:

```
mysql_query("select * from Tabelle");
```

# PHP-Grundlagen

- ◆ MySQL-PHP API: Suchergebnisse
  - `mysql_fetch_row()`;
  - `mysql_num_fields()`;
  - `mysql_num_rows()`;
  - `mysql_field_name()`;
  
  - **`mysql_fetch_object()`;**

# PHP-Grundlagen

- ◆ MySQL-PHP API: Fehlermeldungen
  - `mysql_error()`: Fehlermeldung
  - `mysql_errno()`: Fehlernummer
- ◆ Fehlerbehandlung im Script:

```
mysql_connect(...)
or die("Keine Verbindung");
```





---

# PHP-Grundlagen Übung

---

- ◆ Erstelle ein Script, das eine Verbindung zu deiner DB herstellt

# PHP-Grundlagen

- ◆ Vorgangsweise beim DB-Zugriff:
  - 1) Verbindung zur DB herstellen und auswählen
  - 2) SQL-Befehl definieren (als Variable)
  - 3) Anfrage durchführen (mysql\_query)
  - 4) Datenausgabe bzw. Statusmeldung



---

# PHP-Grundlagen Übung

---

- ◆ Ein Gästebuch für die Homepage:
  - Was soll ein Gästebuch können ?
  - Was benötigt man dazu ?
  - Welche Felder soll die DB enthalten ?

# PHP-Grundlagen Übung

- ◆ Ein Gästebuch für die Homepage:
  - 1) Definiere die Tabelle für das Gästebuch in MySQL
  - 2) Dateneingabe: 2 Arbeitsschritte
    - a) HTML-Formular
    - b) PHP-Script
  - 3) Teste dein Formular (Überprüfe die Werte im MySQL-Client)

# PHP-Grundlagen Übung

- ◆ Ein Gästebuch für die Homepage:  
Natürlich sollen Benutzer die Einträge auf der Homepage lesen können :-)
  - 1) Erstelle ein Script zur Datenausgabe  
(Grundlage wird gemeinsam mit dem Referenten erarbeitet)  
(Lösungen siehe Lösungszettel)
  - 2) Versuche die Ausgabe zu formatieren und mit „Funktionen“ (email, Links) zu versehen

# PHP-Grundlagen Übung

- ◆ Ein Gästebuch für die Homepage:
  - Verbesserungen: Ändere das Script so, dass die letzten Einträge als erste erscheinen !
  - Gib vor den Einträgen an, wie viele überhaupt vorhanden sind !  
(Es sind x Einträge im Gästebuch)

# PHP-Grundlagen

- ◆ Weitere wichtige Befehle in PHP:  
Fallunterscheidung mit **switch**
  - Vorteil: übersichtlicher als geschachtelte if-elseif-else Anweisungen !
  - Syntax:

```
switch (ausdruck)
{ case ausdruck1: anweisung1;
 ...
 Default: anweisung;
}
```

# PHP-Grundlagen

- ◆ Weitere wichtige Befehle in PHP:  
Fallunterscheidung mit **switch**

→ Beispiel:

```
/* Beispiel 1 */
if ($i == 0) {
 print "i ist gleich 0"; }
if ($i == 1) {
 print "i ist gleich 1"; }
```

```
/* Beispiel 2 */
switch ($i) {
 case 0: print "i ist
gleich 0"; break;
 case 1: print "i ist
gleich 1"; break;
}
```



# PHP-Grundlagen

- ◆ Weitere wichtige Befehle in PHP:  
Einfügen externer Dateien: `include`, `require`  
→ Quelltext aus einer anderen Datei – wird behandelt als stünde er an der Stelle !!  
→ Unterschiede: `require` wird auf jeden Fall eingelesen – nur einmal !

# PHP-Grundlagen

- ◆ Weitere wichtige Befehle in PHP:  
Einfügen externer Dateien: `include`  
→ Beispiel:  
`include(Dateiname.ext);`

# Tools zum Thema Web-DB

- ◆ Editoren mit Syntaxhighlighting für PHP-Code:  
Proton (Win32, kostenlos, [www.meybohm.de](http://www.meybohm.de))  
TextPad (Win32, [www.textpad.com](http://www.textpad.com))  
Kate (Linux, Teil von KDE)
- ◆ Dezign (Win32)
- ◆ MySQL-Front (Win32)
- ◆ PHPMyAdmin
- ◆ MySQLnavigator (Linux)
- ◆ MySQL-Manager (Win32)



---

# Tools zum Thema Web-DB

---

- ◆ Dezn: Datenbankentwicklungstool
  - kostenlos (eingeschränkte Funktionalität)
  - DB-Design unter einer GUI erstellen und SQL-Code generieren



---

# Tools zum Thema Web-DB

---

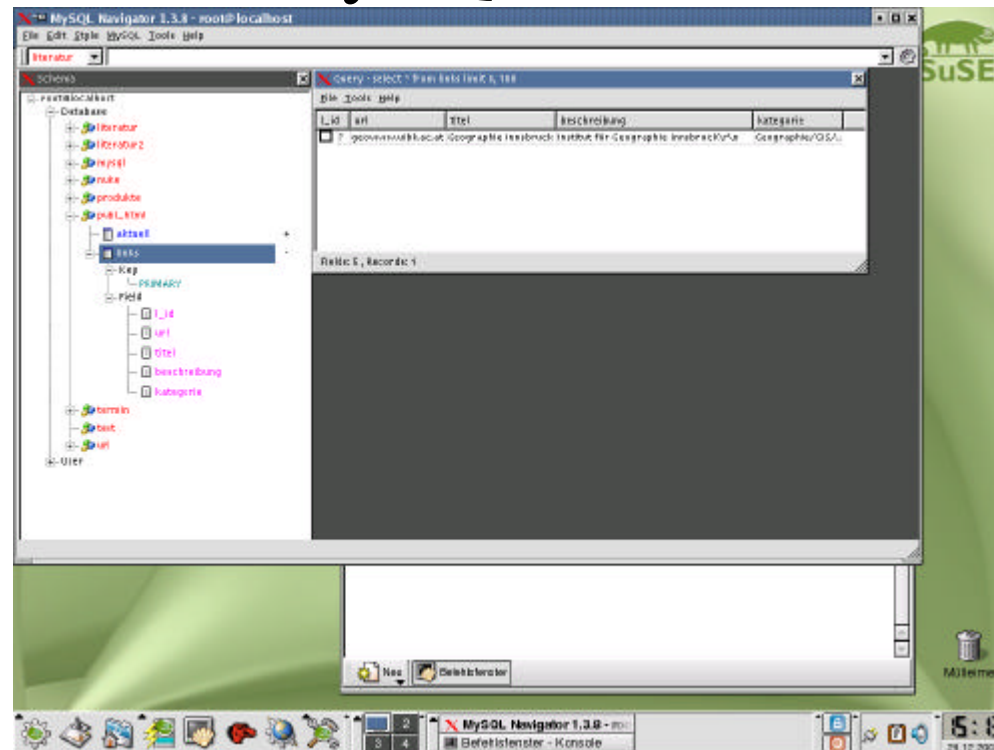
- ◆ MySQL-Front: [www.mysqlfront.de](http://www.mysqlfront.de)  
→ Grafisches Front-End um mit der DB zu arbeiten

# Tools zum Thema Web-DB

- ◆ PHPMyAdmin:
  - Admin-Programm auf Basis einer Weboberfläche (mit PHP programmiert-Serverapplikation)
  - Datendefinition, Datenmanipulation etc. möglich
  - Vorteil: kann von überall mit einem Browser aufgerufen werden

# Tools zum Thema Web-DB

- MySQLnavigator (Linux):
  - Pendant zu MySQL-Front unter Linux



# Tools zum Thema Web-DB

- MySQL-Manager (Win32)
  - Teil der MySQL Distribution
  - Kommunikationsprogramm mit GUI
  - nur lokal nutzbar ;-)
  - `..\mysql\bin\mysqlmanager`





---

# Literatur

---

- ◆ Däßler, R.: MySQL. – bhv, Kaarst 2001.
- ◆ Kofler, M.: MySQL. – Addison-Wesley.
- ◆ Wigard, S.: PHP4. – bhv, Bonn 2001.
- ◆ Dehnhart, W.: Scriptsprachen für dynamische Webauftritte. – Hanser, München 2001.