

# Multimedia im Netz

Wintersemester 2013/14

Übung 03 (Nebenfach)

# Datenbanken und SQL

- Mit Hilfe von Datenbanken kann man Daten permanent speichern
- Es gibt verschiedene Datenbankverwaltungssysteme. Für MMN verwenden wir MySQL.
- SQL (= Structured Query Language) ist eine Sprache um auf Datenbanken zuzugreifen, Daten anzufragen und zu manipulieren.
- Mit SQL kann man:
  - Datenbank anlegen
  - Tabellen anlegen
  - Daten aus der Datenbank holen
  - Daten in der Datenbank ablegen
  - ...

# Datenbanktabellen

- Eine (relationale) Datenbank besteht meistens aus ein oder mehreren Tabellen.
- Jede Tabelle hat einen Namen mit ein oder mehreren Spalten
- Eine Tabelle enthält (k)eine oder mehrere Einträge (= Zeilen der Tabelle)

*Tabelle: Kontakte*

PersonenID	Vorname	Nachname	Telefonnummer
1	Max	Mustermann	089455544431
2	Laura	Stern	070815643593
3	Tanja	Baumann	0895673138
4	Felix	Maurer	0894562897

# MySQL im CIP-Pool (I)

- Unter <https://tools.rz.ifi.lmu.de/> “Datenbank Management” aufrufen
- Neuen Datenbank Account anlegen (MUST DO)
- Neue Datenbank anlegen (MUST DO)
- Verbindung via db2.cip.ifi.lmu.de

# MySQL im CIP-Pool (II)

- Um mit der Datenbank zu arbeiten, muss man sich vorher mit dem Datenbankserver verbinden:

1. SHELL aufrufen

2. Folgenden Befehl eingeben:

```
mysql -h db2.cip.ifi.lmu.de -u [username] -p
```

3. Passwort eingeben

4. Es sollte folgender Text erscheinen:

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 22399
Server version: 5.1.72-2 (Debian)

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> █
```

# MySQL mit einer lokalen Datenbank (I)

- XAMPP bietet auch die Möglichkeit, mit MySQL lokal zu arbeiten
- Dazu muss man im Kontrollzentrum das Modul MySQL starten



The screenshot shows the XAMPP Control Panel v3.0.12 interface. It features a table of services and a set of control buttons on the right. The MySQL service is highlighted in green, and its 'Stoppen' button is selected with a blue dashed border.

Module Dienst	Modul	PID(s)	Port(s)	Aktionen
<input checked="" type="checkbox"/>	Apache	6404 6596	443, 8080	Stoppen Admin Konfig Logs
<input checked="" type="checkbox"/>	MySQL	5544	3306	Stoppen Admin Konfig Logs
<input checked="" type="checkbox"/>	FileZilla			Starten Admin Konfig Logs
<input type="checkbox"/>	Mercury			Starten Admin Konfig Logs
<input checked="" type="checkbox"/>	Tomcat			Starten Admin Konfig Logs

Control Panel Buttons (Right Side):  
Konfig  
Netstat  
XAMPP-Shell  
Explorer  
Win-Dienste  
Hilfe  
Beenden

# MySQL mit einer lokalen Datenbank (II)

- Verbindung mit einem lokalen Datenbankserver:
  1. In der Kommandozeile zum Pfad „.../xampp/mysql/bin“ wechseln
  2. Folgenden Befehl eingeben:  
`mysql -h localhost -u [username] -p`
  3. Passwort eingeben
  4. Es sollte folgender Text erscheinen:

```
D:\PHP\xampp\mysql\bin>mysql -h localhost -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.5.25a MySQL Community Server (GPL)

Copyright (c) 2000, 2011, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql>
```

# MySQL mit einer lokalen Datenbank (III)

- Man kann auch über ein Web-Interface mit MySQL Datenbanken arbeiten: phpMyAdmin
- Dazu muss man im Browser einfach folgende URL eingeben
  - <http://localhost/phpmyadmin>
  - Darauf achten, dass der PHP Server ebenfalls gestartet ist



# SQL: Datenbank anlegen

- Übersicht der angelegten Datenbanken  
`SHOW DATABASES ;`
- Neu Datenbank anlegen  
`CREATE DATABASE mydb ;`
- Datenbank für die Verwendung auswählen  
`USE mydb ;`
- Datenbank löschen  
`DROP DATABASE mydb ;`

# SQL: Tabelle anlegen (I)

- Tabellenübersicht anzeigen:

```
SHOW TABLES;
```

- Tabelle anlegen

```
CREATE TABLE myTable  
(  
column_name1 data_type(size) ,  
column_name2 data_type(size) ,  
column_name3 data_type(size) ,  
...  
);
```

*Tabelle: myTable*

column_name1	column_name2	column_name3	...
--------------	--------------	--------------	-----

# SQL: Tabelle anlegen (II)

- Probleme der Tabelle auf der vorherigen Folie:
  - Man kann leere Einträge in die Tabelle einfügen
  - Einträge sind nicht unbedingt eindeutig
- Lösung: Tabelle anlegen und Constraints angeben: Damit kann man bestimmte Regeln für die Daten innerhalb einer Spalte anlegen
- Wichtige Constraints (nur Auszug):
  - NOT NULL
  - PRIMARY KEY, oft verwendet mit AUTO INCREMENT

# SQL: Tabelle anlegen (III)

- Tabelle anlegen mit Constraints

```
CREATE TABLE myTable
(
column_name1 data_type(size) NOT NULL
                                PRIMARY KEY
                                AUTO_INCREMENT,
column_name2 data_type(size) NOT NULL,
column_name3 data_type(size),
...
);
```

# Beispiel: Tabelle Anlegen

```
CREATE TABLE Kontakte
(
  PersonenID int NOT NULL PRIMARY KEY AUTO_INCREMENT,
  Vorname varchar(255) NOT NULL,
  Nachname varchar(255) NOT NULL,
  Telefonnummer int NOT NULL,
);
```

*Tabelle: Kontakte*

PersonenID	Vorname	Nachname	Telefonnummer
------------	---------	----------	---------------

# SQL: Daten einfügen / auslesen

- Einträge hinzufügen

```
INSERT INTO myTable
    (column_name1, column_name2, ...)
VALUES
    (value1, value2, ...)
);
```

- Alle Einträge auslesen

```
SELECT * FROM myTable;
```

- Bestimmte Einträge auslesen

- Einträge auslesen mit Bedingung

```
SELECT * FROM myTable WHERE column_name=value;
```

- Einträge einer Spalte auslesen

```
SELECT column_name1 FROM myTable;
```

```
SELECT column_name1, column_name2 FROM myTable;
```

# Beispiel: Daten einfügen

```
INSERT INTO Kontakte  
  (Vorname, Nachname, Telefonnummer)  
VALUES  
  ( "Max" , "Mustermann" , 089455544431 ) ;
```

*Tabelle: Kontakte*

PersonenID	Vorname	Nachname	Telefonnummer
1	Max	Mustermann	089455544431

# Beispiel: Daten auslesen

- Alle Daten einer Tabelle auslesen  
`SELECT * FROM Kontakte`
- Daten, die eine bestimmte Bedingung erfüllen auslesen  
`SELECT * FROM Kontakte WHERE Vorname="Laura" ;`

*Tabelle: Kontakte*

PersonenID	Vorname	Nachname	Telefonnummer
1	Max	Mustermann	089455544431
2	Laura	Stern	070815643593
3	Tanja	Baumann	0895673138
4	Felix	Maurer	0894562897

# Übungsblatt 3

- **Thema: SQL Anfragen**
- Bearbeitungszeit: 1 Woche
- Abgabe: 20.11.2013 23:00 Uhr

**Danke! Fragen?**