Multimedia im Netz (Online Multimedia)

Wintersemester 2014/15

Übung 03 (Nebenfach)

Databases and SQL

- Data can be stored permanently in databases
- There are a number of database management systems (DBMS). In this lecture & tutorial we use **MySQL**
- SQL (= Structured Query Language) is a language that allows us to access databases. We can retrieve and manipulate data with it.
- With SQL you can:
  - Create databases
  - Create tables
  - Retrieve data from a database
  - Store data in a database
  - ...

Tables in relational databases

- A relational database usually consists of one or more **tables**
- Each table has a unique name with one or more **columns**
- Each table can have multiple entries (or none).
- A table **row** represents an entry

Table: Contacts

PersonID	FirstName	LastName	PhoneNumber
1	Max	Mustermann	089455544431
2	Laura	Stern	070815643593
3	Tanja	Baumann	0895673138
4	Felix	Maurer	0894562897

MySQL at the CIP-Pool

- Access “Datenbank Management” here: https://tools.rz.ifi.lmu.de/
- Create a new account (required)
- Create a new database (required)
- Connect to db2.cip.ifi.lmu.de

MySQL at the CIP-Pool (II)

- To work with the database, you have to connect to the database server:
  1. Start a SHELL (Ctrl+Alt+T)
  2. Enter the following command:
     `mysql -h db2.cip.ifi.lmu.de -u [username] -p`
  3. Provide your password
  4. If successfull you should see something like this:

```
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 with your local database(I)

- XAMPP lets you work with your own, local MySQL database
- Make sure you start the MySQL Service in the control center

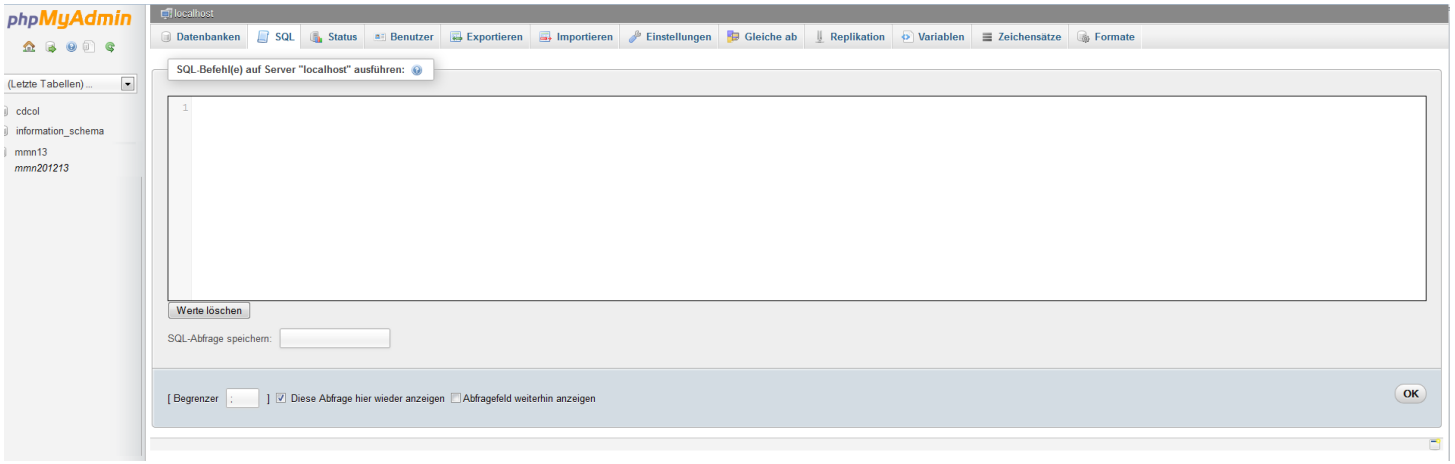
MySQL with your local database (II)

- Connect to a local database server:
  1. Change to the „…/xampp/mysql/bin“ directory
  2. Enter the following command:
     `mysql -h localhost -u [username] -p`
  3. Enter the password (usually “root”, “admin”, “password” or none)
  4. You should see something like the following:

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.5.34 Source distribution

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 with your local database (III)

- You can perform work with MySQL through a very common web interface: phpMyAdmin
- Once you've started the Apache & MySQL Servers in XAMPP, enter the following URL in a web browser:
  - http://localhost/phpmyadmin



SQL: Creating a database

- Get an overview on all existing databases:
  ```
  SHOW DATABASES;
  ```

- Create a new database:
  ```
  CREATE DATABASE mydb;
  ```

- Select a database for further usage:
  ```
  USE mydb;
  ```

- Delete a database (be careful!):
  ```
  DROP DATABASE mydb;
  ```

SQL: Creating a table (I)

- Get an overview on all exisiting tables (of a database):

```
SHOW TABLES;
```

- Create a new table

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

Table: myTable

column_name1	column_name2	column_name3	…

SQL: Creating a table(II)

- Problems with the statement from previous slide:
  - You can add empty entries to the table
  - Entries could be duplicates

- Solution: Create a table with certain constraints. Thus, you define certain rules for columns

- Most important constraints (among many others):
  - NOT NULL
  - PRIMARY KEY [often in conjunction with] AUTO INCREMENT

SQL: Creating a table (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),
…
);
```

Example: Creating a table

```
CREATE TABLE Contacts
(
PersonID int NOT NULL PRIMARY KEY AUTO_INCREMENT,
FirstName varchar(255) NOT NULL,
LastName varchar(255) NOT NULL,
PhoneNumber int NOT NULL,
);
```

Table: Contacts

PersonID	FirstName	LastName	PhoneNumber

SQL: Adding & Retrieving data

- Add entries:

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

- Retrieve all entries from a table:

```
SELECT * FROM myTable;
```

- Retrieve only a subset of entries
  - Entries that fulfill certain conditions with the **`WHERE`** keyword

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

  - Entries from specific columns:

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

Example: Add an entry

```
INSERT INTO Contacts
   (FirstName, LastName, PhoneNumber)
VALUES
   ("Max", "Mustermann", 089455544431);
```

Table: Contacts

PersonID	FirstName	LastName	PhoneNumber
1	Max	Mustermann	089455544431

Example: Read data

- Retrieve all data from a table

```
SELECT * FROM Contacts
```

- Retrieve entries that fulfill a certain condition:

```
SELECT * FROM Contacts WHERE FirstName="Laura";
```

Table: Contacts

PersonID	FirstName	LastName	PhoneNumber
1	Max	Mustermann	089455544431
2	Laura	Stern	070815643593
3	Tanja	Baumann	0895673138
4	Felix	Maurer	0894562897

Assignment 3

- **Topic: Address-book with SQL Queries**
- Due in: 1 Week
- Due date: 03.11.2014 14:00h

Thanks!

What are your questions?