

# Datenbanken

lukas.feiler@lukasfeiler.com

<http://teaching.lukasfeiler.com>

# Datenbanken

z.B. MySQL, Firebird, PostgreSQL, Oracle, DB2, MS SQL Server, ...

Ziel: jeder programmiert selbst  
ein Guestbook

# MySQL

Aktuelle Version: 4.1.14

Klein und schnell aber geringer Funktionsumfang

# Primärliteratur

<http://www.mysql.com/documentation/index.html>

# Sekundärliteratur

MySQL, Third Edition by Paul DuBois

# mysql

das mitgelieferte Command-Line-Tool

# phpMyAdmin

Zugriff über Browser

<http://www.phpmyadmin.net/>

# EMS MySQL Manager

kein Open Source sond. Shareware!

<http://ems-hitech.com/mymanager/>

## Grundbegriffe einer Datenbank:

- Tabelle (Table)
- Spalte (Column)
- Datensatz (Record)

Tabelle: employee

username	password	firstname	lastname
tom	jid7#-989s	Thomas	Mayer
joe	098778-ds87	Joe	Smith
susan	_skj989#as	Susan	Green

Angestellte mit Username, Passwort,  
Vorname, Nachname



# SQL – Structured Query Language

Die Abfragesprache für Datenbanken

```
CREATE TABLE employee (  
    f_employeeid INTEGER NOT NULL AUTO_INCREMENT,  
    f_username VARCHAR(15) NOT NULL,  
    f_password VARCHAR(15) NOT NULL,  
    f_firstname VARCHAR(255),  
    f_lastname VARCHAR(255),  
    PRIMARY KEY(f_employeeid),  
    UNIQUE (f_username)  
);
```

Records einfügen (INSERT)

Records lesen (SELECT)

Records ändern (UPDATE)

Records löschen (DELETE)

```
INSERT INTO  
    employee  
        (f_username, f_password, f_firstname, f_lastname)  
VALUES  
    ('admin', 'test', 'John', 'Doe');
```

#get all records (and all columns)

```
SELECT * FROM employee;
```

#get all records (but just the column f\_username)

```
SELECT f_username FROM employee;
```

#get only those records that have a value of 'admin' in  
#the column f\_username

```
SELECT * FROM employee WHERE f_username='admin';
```

```
UPDATE employee SET f_firstname='Moe',  
f_lastname='Green' WHERE f_username='admin';
```

#updates all (!) records

```
UPDATE employee SET f_firstname='Moe';
```

#delete the user guest

```
DELETE FROM employee WHERE f_username='guest';
```

#delete all (!) records

```
DELETE FROM employee;
```

# Spezifikation:

Eine Tabelle (mittels mysql od. phpMyAdmin), die alle für ein Gästebuch nötige Daten speichern kann (author, text, creation\_date, ...).

Einen Datensatz anlegen.

Zugang via phpMyAdmin:

<http://student.sae.at/phpmymad>

user: teaching-php

pass: \* \* \* \* \*



# Lösung:

```
CREATE TABLE guestbook (  
    f_messageid INTEGER NOT NULL AUTO_INCREMENT,  
    f_author VARCHAR(255) NOT NULL,  
    f_text TEXT,  
    f_creationdate DATETIME NOT NULL,  
    PRIMARY KEY (f_messageid)  
);
```

```
INSERT INTO guestbook (f_author, f_text, f_creationdate)  
VALUES('John Doe', 'How are ya?', NOW());
```

# Bugs finden und beheben:

newTable.sql

# PHP mit MySQL verbinden

Kommunikation mittels vordefinierten functions

→ PHP Manual ([www.php.net](http://www.php.net))

# Die wichtigsten Funktionen

```
$link = mysql_connect("localhost", "user", "pass");  
$link = mysql_connect("127.0.0.1", "user", "pass");  
  
mysql_select_db("teaching-php");  
  
$result = mysql_query("SELECT * FROM employee");  
  
mysql_fetch_assoc($result);  
  
mysql_close($link);
```

z.B. getEmployee.php

# Spezifikation:

Ein PHP-File, das alle Datensätze d. guestbook-Tabelle ausgibt; Name: getMessage.php

# Bugs finden und beheben:

getEmployeeBug.php

# Neue Datensätze erstellen

```
$link = mysql_connect("localhost", "user", "pass");  
$link = mysql_connect("127.0.0.1", "user", "pass");  
  
mysql_select_db("teaching-php");  
  
mysql_query("INSERT INTO employee (...) VALUES (...)");  
  
mysql_close($link);
```

z.B. createEmployee.php

# Spezifikation:

Ein File bestehend aus PHP und HTML; über das Submiten einer HTML-Form per POST soll ein neuer Eintrag im Gästebuch erstellt werden; Name: createMessage.php

## Tips:

- die primary key column sollte nicht gesetzt sein (AUTO\_INCREMENT)
- Das aktuelle Datum gibt die MySQL-Funktion NOW() zurück.

## Advanced

- Validierung der eingegebenen Daten
- SQL Injection Protection
- Letzte Aufgabe mit dieser verbinden

## Aufgabe



# Bugs finden und beheben:

`createEmployeeBug.php`

# Datensätze löschen

```
$link = mysql_connect("localhost", "user", "pass");  
$link = mysql_connect("127.0.0.1", "user", "pass");  
  
mysql_select_db("teaching-php");  
  
mysql_query("DELETE FROM employee WHERE ...");  
  
mysql_close($link);
```

z.B. deleteEmployee.php

# SQL Injection

`$id = "0 OR 1=1"`

```
mysql_query("DELETE FROM employee WHERE f_employeeid = $id");
```

→ "DELETE FROM employee WHERE f\_employeeid = 0 OR 1=1"

vgl. deleteEmployee.php

# Lösung für das Problem SQL Injection:

- 1) Variablen in Queries immer unter (einfachen) Anführungszeichen: "... WHERE f\_id='\$id' "
- 2) „escapen“ mit: \$id = **mysql\_escape\_string**(\$id)

# Spezifikation:

Gästebucheinträge löschen; von getMessage.php aus;  
Name: deleteMessage.php

## Tips:

- Mit der primary key column arbeiten

## Advanced

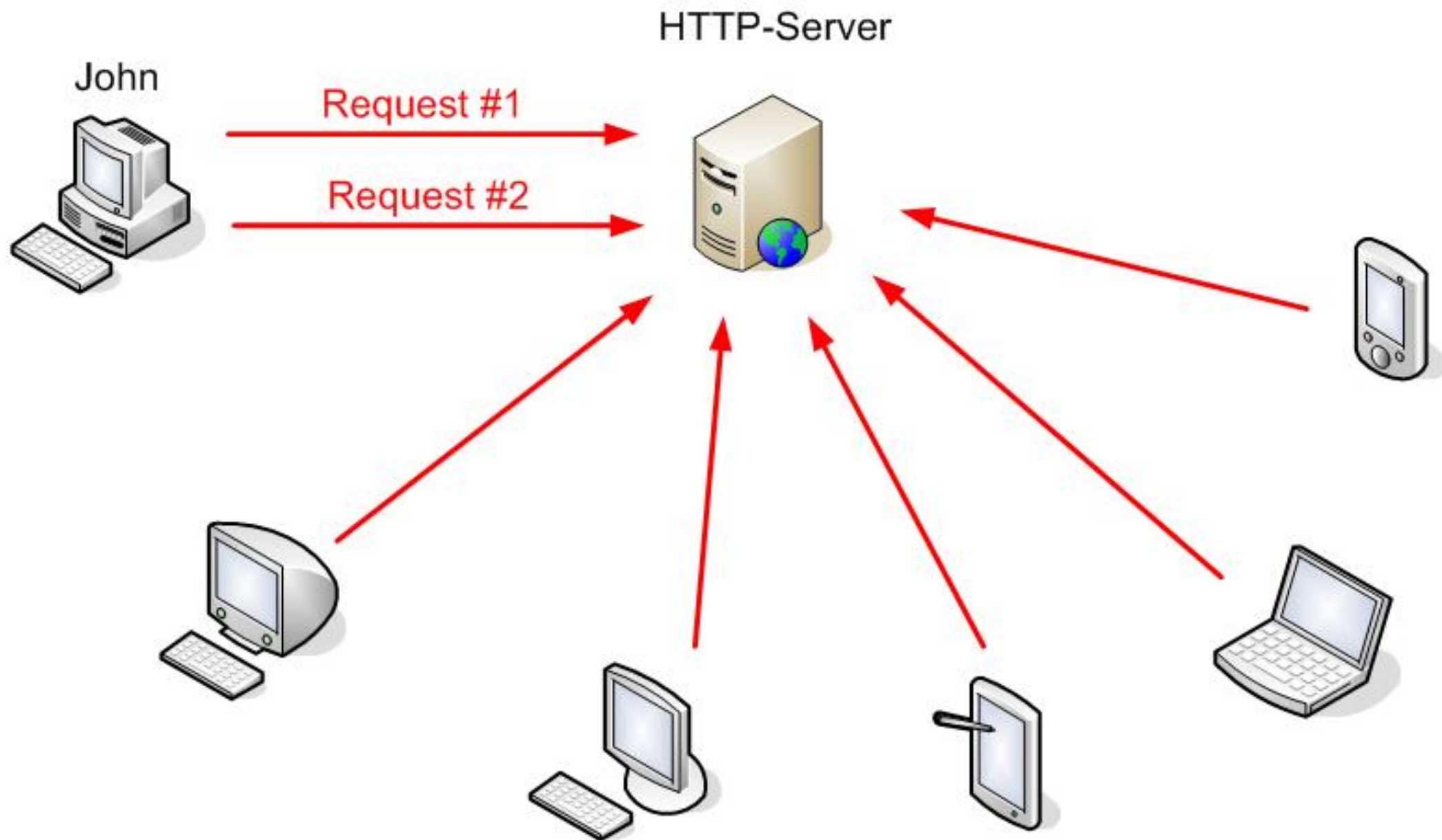
- Validierung der eingegebenen Daten
- SQL Injection Protection
- Letzte Aufgaben mit dieser verbinden

# Bugs finden und beheben:

`deleteEmployeeBug.php`

# Session Management

Ermöglicht die Implementierung einer „Login“-Session



**Das Problem**



John



Request #1



SID=h28hau98a...

Request #2 (inkl. SID)



HTTP-Server



Session  
Data for SID  
h28hau98a...

# Session Handling in PHP

```
//starts the session  
session_start();
```

```
//the array $_SESSION contains all session variables  
$_SESSION['loggedIn'] = true;
```

z.B: testSession.php

# Eine Login Session implementieren

- Mit `session_start()` wird das Session Management aktiviert
- Benutzer muss sich mit Username & Passwort anmelden
  - wenn erfolgreich: eine Session-Variable wird gesetzt um Session als logged-in zu kennzeichnen:  
*`$_SESSION['loggedIn'] = true;`*

z.B: deleteEmployee2.php

# Spezifikation:

Nur authentifizierte Benutzer dürfen Gästebucheinträge löschen.

Advanced

- Logout soll auch möglich sein.

# Bugs finden und beheben:

`deleteEmployeeBug2.php`

# Macromedia Flash & PHP verbinden

PHP: `echo "var1=value1&var2=value2";`

Flash: `var receiver = new LoadVars();`  
`var sender = new LoadVars();`

# PEAR: PHP Extension and Application Repository

<http://pear.php.net>

Packages: DB, Mail, ...

## PEAR::DB

ist "Database abstraction layer"

DSN - Data Source Name:

phptype://username:password@hostspec/database

z.B: PEARDB.php



## PEAR::Mail

zum Versenden von E-Mails über

- lokalen Mail Server (sofern vorhanden)
- entfernten SMTP Server

z.B: PEARMail.php

# Relationales Datenbankdesign

Tabelle: employee (schlechtes Design)

f_employeeid	f_username	f_password	f_firstname	f_lastname	f_department	f_departmentphone
1	tom	Uai908#as	Thomas	Mayer	Science	391 122
2	john	Sa7as9as	John	Doe	Development	231 399
3	susan	#asdf+as8	Susan	Green	Science	391 122
4	lukas	989asadf9	Lukas	Feiler	Marketing	198 321
5	moe	9as9adsf0	Moe	Green	Design	882 233

Angestellte mit Username, Passwort,  
Vorname, Nachname, Abteilung u.  
Telefonnummer d. Abteilung

## Datenbankdesign Richtlinien:

- Datenelemente in verwandte Gruppen teilen
- Definition der richtigen Beziehung
- Tabellen und Spaltennamen selbsterklärend
- Daten werden niemals mehrfach abgespeichert

## Tabelle: employee

f_employeeid	f_username	f_password	f_firstname	f_lastname	f_departement_id
1	tom	Uai908#as	Thomas	Mayer	1
2	john	Sa7as9as	John	Doe	2
3	susan	#asdf+as8	Susan	Green	1
4	lukas	989asadf9	Lukas	Feiler	3
5	moe	9as9adsf0	Moe	Green	4

## Tabelle: department

f_departmentid	f_departmentname	f_departmentphone
1	Science	391 122
2	Development	231 399
3	Marketing	198 321
4	Design	882 233

z.B: relational.sql

# Zwei in Relation stehende Tabellen

# Objekt-Orientiertes Programmieren (OOP)

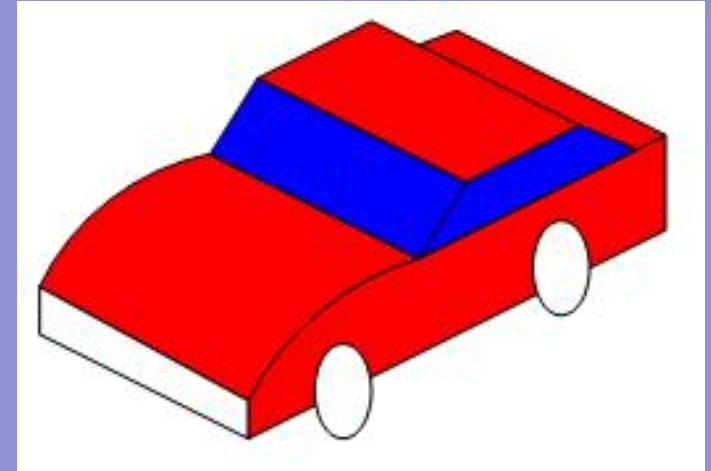
- Everything is a object
- Every object has a type

Type/Class Name:

Car

Interface:

start()  
stop()  
accelerate()  
decelerate()



Defining the class Car

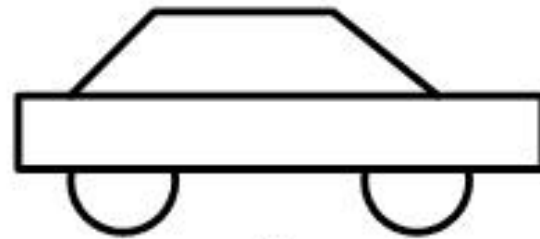
```
class Car {  
    ...  
}
```

Creating a new instance of the class Car:

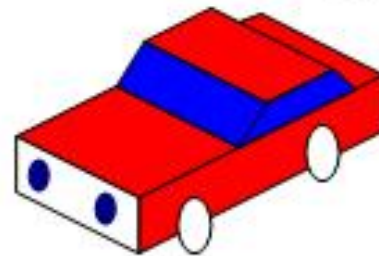
```
$car = new Car();  
$car->start();
```

**The class Car**

Car
start() stop() accelerate() decelerate()



Inheritance



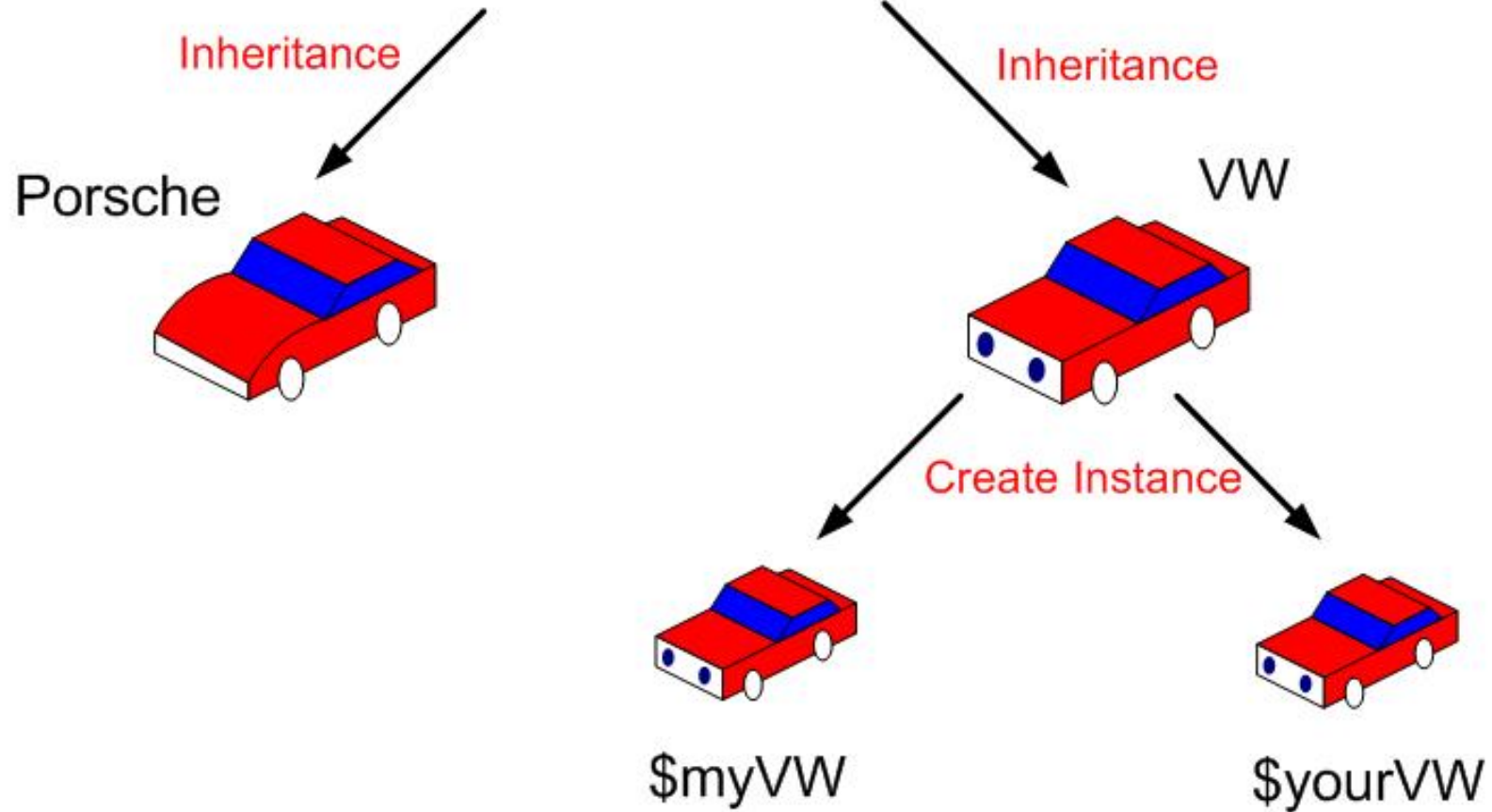
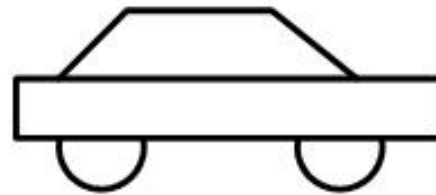
VW

Die Klasse VW erbt von der Klasse Car:

```
class VW extends Car {  
    ...  
}
```



Car
start() stop() accelerate() decelerate()



The class tree: siehe Car.php