


# LAMP - Einführung


Stefan Maihack Dipl. Ing. (FH)  
Datum: 9. März 2015


# LAMP

## Begriffserläuterung

- L (X) – Linux
- A – Apache
- M – MySQL
- P – PHP

 XAMPP für Windows  
v1.8.3 (PHP 5.5.9)

 XAMPP für Linux  
v1.8.3 (PHP 5.5.9)

 XAMPP für Apple  
v1.8.3 (PHP 5.5.9)

# Warum eine Datenbankanbindung?

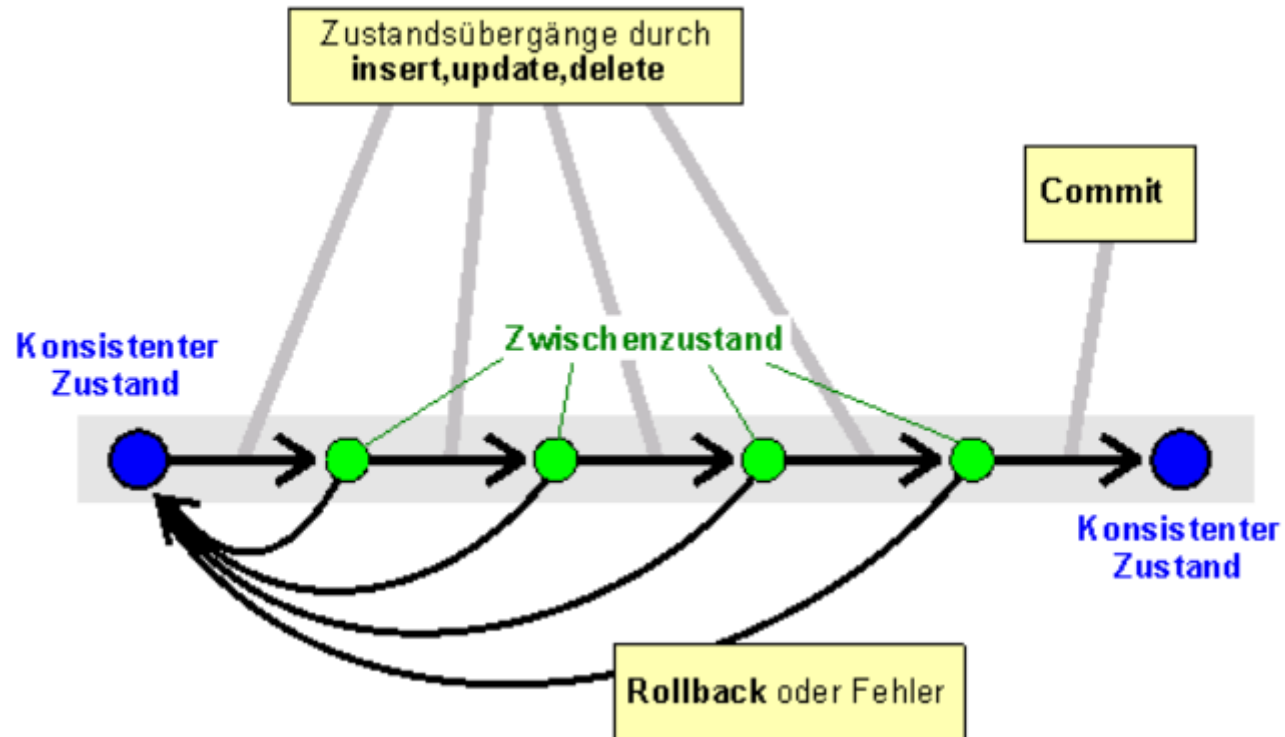
- Vermeidung von Datenredundanzen:
  - Daten nur an einer Stelle speichern: spart Speicherplatz
  - Änderungen nur an einer Stelle nötig: Daten bleiben konsistent
- Effiziente Datenverwaltung:
  - spezialisierte Algorithmen für große Datenbestände
  - z.B. Suchalgorithmen, Sortieralgorithmen, Backup-Methoden
- Konfliktfreier Mehrbenutzerbetrieb:
  - zentrale Zugriffssteuerung verhindert gegenseitige Störungen und unkontrolliertes Überschreiben (Datensicherheit)

Datenunabhängigkeit durch standardisierte Schnittstellen:

- interne Darstellung der Daten wird verborgen
- unabhängig von Speichermedien und Rechnerarchitektur

# Warum eine Datenbankankbindung?

Datenbank – LUW (Logical Unit of Work)



# Zugriffsmöglichkeiten auf MySQL

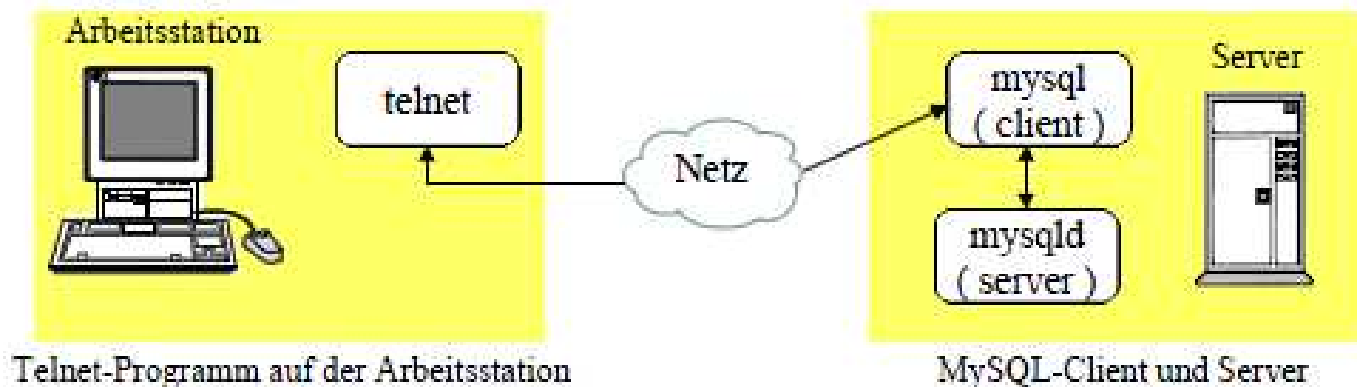
## Windows/UNIX - Client

Konfiguration: Telnet-Zugriff auf einen MySQL-Client auf einem Server

→ MySQL-Client mit Texteingabe und Ausgabe  
gut geeignet zum üben von SQL-Kommandos

Problem: SQL-Kenntnisse erforderlich

→ nicht für „normale Datenbank-Nutzer geeignet  
diese benötigen eine grafische Oberfläche.



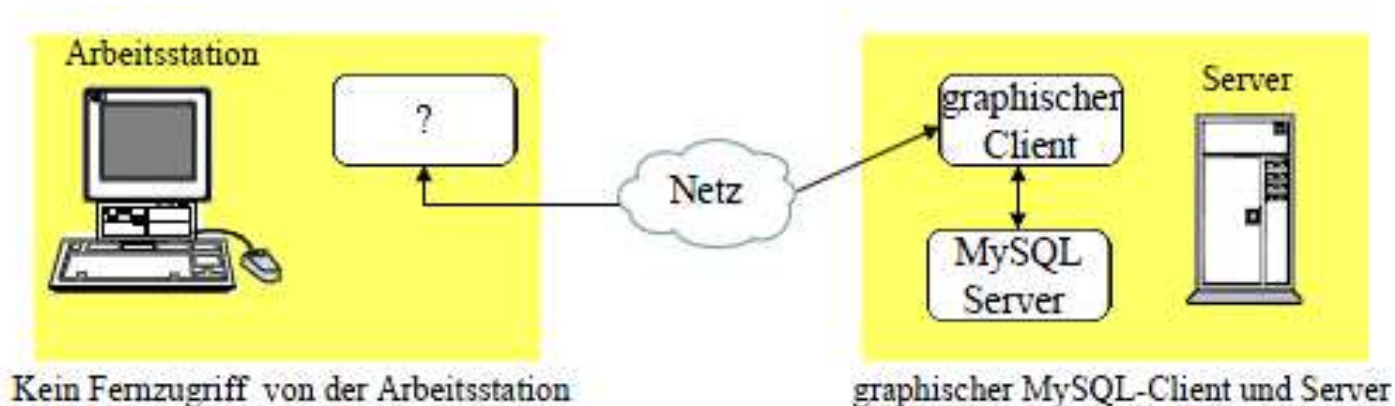
# Mögliche Lösungen um auf Datenbanken zu zugreifen

„grafischer“ Client = in C++ selbst geschrieben, mit grafischer Oberfläche

→ MySQL-Client bietet ein C-API (application programming interface)  
= eine Sammlung von Funktionen, in das C-Programm einzubinden.

Aber:

→ Client ist betriebssystemabhängig (Linux, Mac OS, Windows?)  
→ nur optimal, wenn Client und Server auf einem Rechner (lokal)  
→ Datenbanken benötigen meist Mehrbenutzerbetrieb-Netzzugriff...



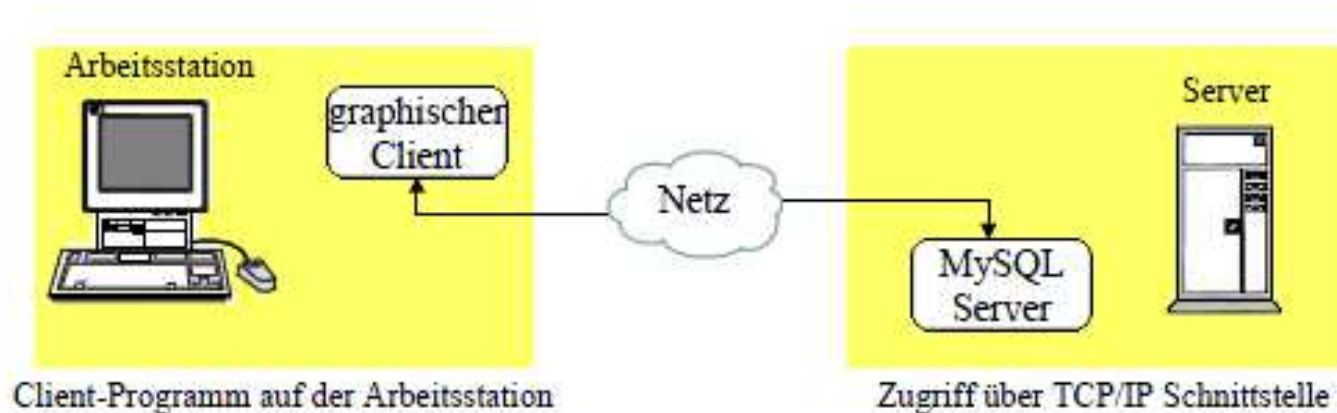
# Konfiguration mit grafischen Client (1)

Variante a) Ein Client-Programm auf jeder Arbeitsstation  
→ Client/Server Kommunikation über Netzwerkschnittstelle

Vorteil: Server wird nicht mit Grafik-Ausgabe belastet

Probleme:

- hoher Administrationsaufwand  
(bei Softwareänderungen sind alle Arbeitsstationen betroffen.)
- Client für unterschiedliche Betriebssysteme notwendig



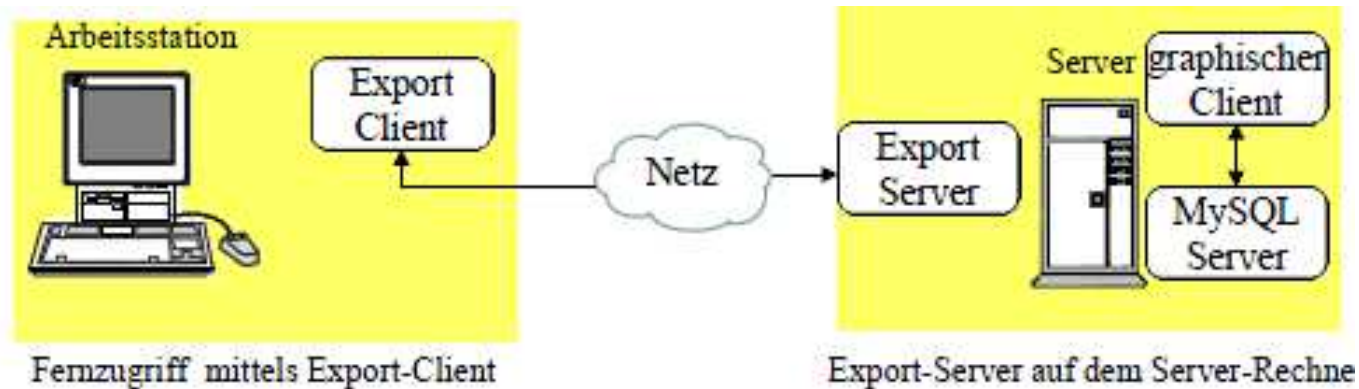
# Konfiguration mit grafischen Client (2)

- B) Export der gesamten Grafik-Oberfläche („grafisches Telnet“)

Vorteil: weniger Konfigurationsaufwand, nur einmal Export-Client installieren

Problem:

- hohes Datenvolumen
- Sicherheit (Zugriff auf alle Anwendungen des Servers?)
- Export-Software evtl. kostenpflichtig  
z.B. Citrix Metaframe Application Serving,  
Windows Remote Desktop (Terminal Server) etc.





# Graphische Clients für MySQL

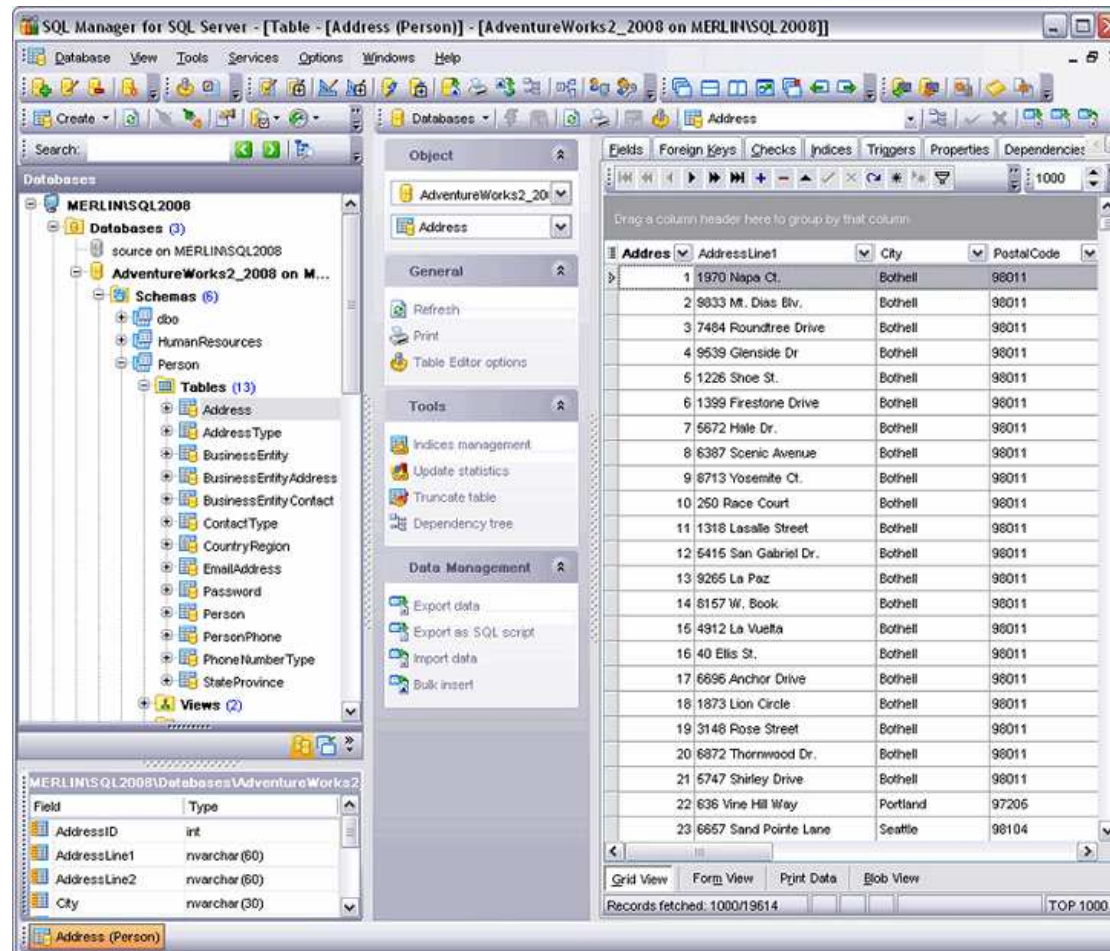
## EMS SQL Manager 2005 for MySQL

EMS SQL Manager ist eine optimale Anwendung zur Entwicklung und Verwaltung von MySQL-basierten Datenbanken. Die Anwendung arbeitet mit verschiedenen MySQL-Versionen von 3.23 bis 5.1 und unterstützt sämtliche MySQL-Eigenschaften, einschließlich Views, Stored Procedures und Funktionen, InnoDB-Fremdschlüssel usw. Sie bietet viele nützliche Werkzeuge, mit denen die Entwicklung und Verwaltung optimal unterstützt wird.

- Volle Unterstützung von MySQL-Versionen von 3.23 bis 5.1
- Schnelle Navigation und Datenbankmanagement
- Einfaches Management aller MySQL-Objekte
- Fortgeschrittene Datenmanipulations-Werkzeuge
- Zugang zum MySQL-Server durch HTTP-Protokoll
- Ausgezeichnete visuelle Textwerkzeuge für Query-Bildung
- Das Verbinden über den lokalen Port durch den SSH-Tunnel
- Optimierte Datenexport und -import-Eigenschaften
- Erweitertes Sicherheitsmanagement
- Visual Database Designer
- Kosten: 100.- – 350.- Euro

# Graphische Clients für MySQL

EMS SQL Manager 2005 for MySQL



# Graphische Clients für MySQL

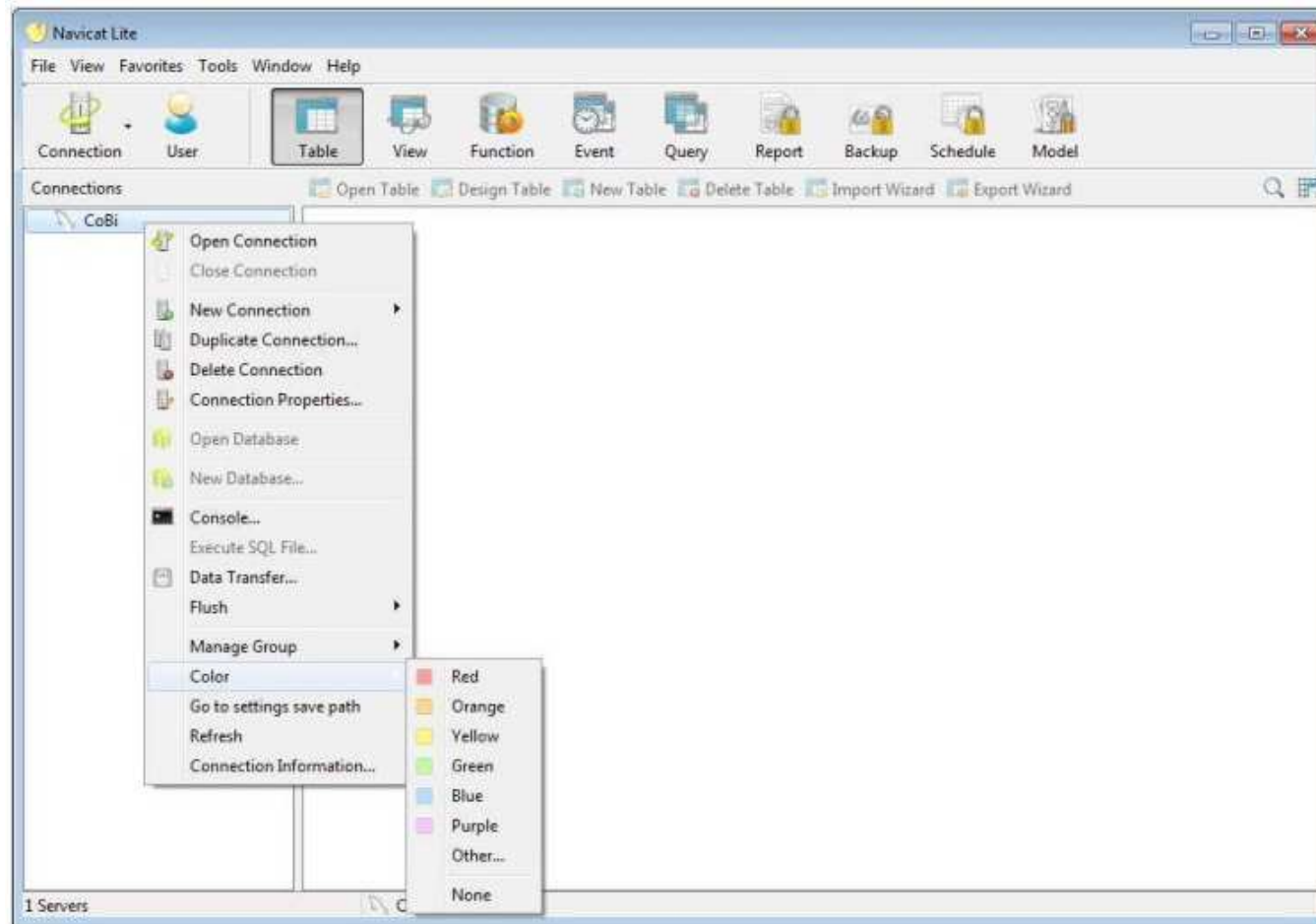
## Navicat

Navicat ist ein leistungsfähiges Datenbankadministrations- und Entwicklungswerkzeug, das nicht nur optimal für professionelle Entwickler, sondern auch für Neueinsteiger leicht einzusetzen ist. Die Benutzeroberfläche von Navicat hilft schnell Informationen zu erzeugen, zu organisieren, abzurufen und gemeinsam zu nutzen.

- Unterstützung mehrfacher Verbindungen zu MySQL-Servern
- Erzeugen und Löschen von Datenbanken, Tabellen, Indizes und Benutzern
- Erstellen und Ausführen von SQL-Abfragen
- Bearbeiten von Benutzerrechten (Sicherheitsrichtlinien)
- Sichern/Wiederherstellen von Datenbanken
- Datenübertragung – Replikation von Datenbanken
- Foreign Keys – unterstützt die Definition von Fremdschlüsseln
- Stapelverarbeitung und Zeitplanung
- Datenimport/-export der populärsten Datenformate, einschließlich Access, Excel, XML, PDF und TXT
- Datenimport von ODBC-Datenquellen
- Datensynchronisation und Struktursynchronisation
- Unicode-Unterstützung
- Stored Procedure
- Kosten für Navicat Light: 0.- Euro

# Graphische Clients für MySQL

## Navicat



# Graphische Clients für MySQL

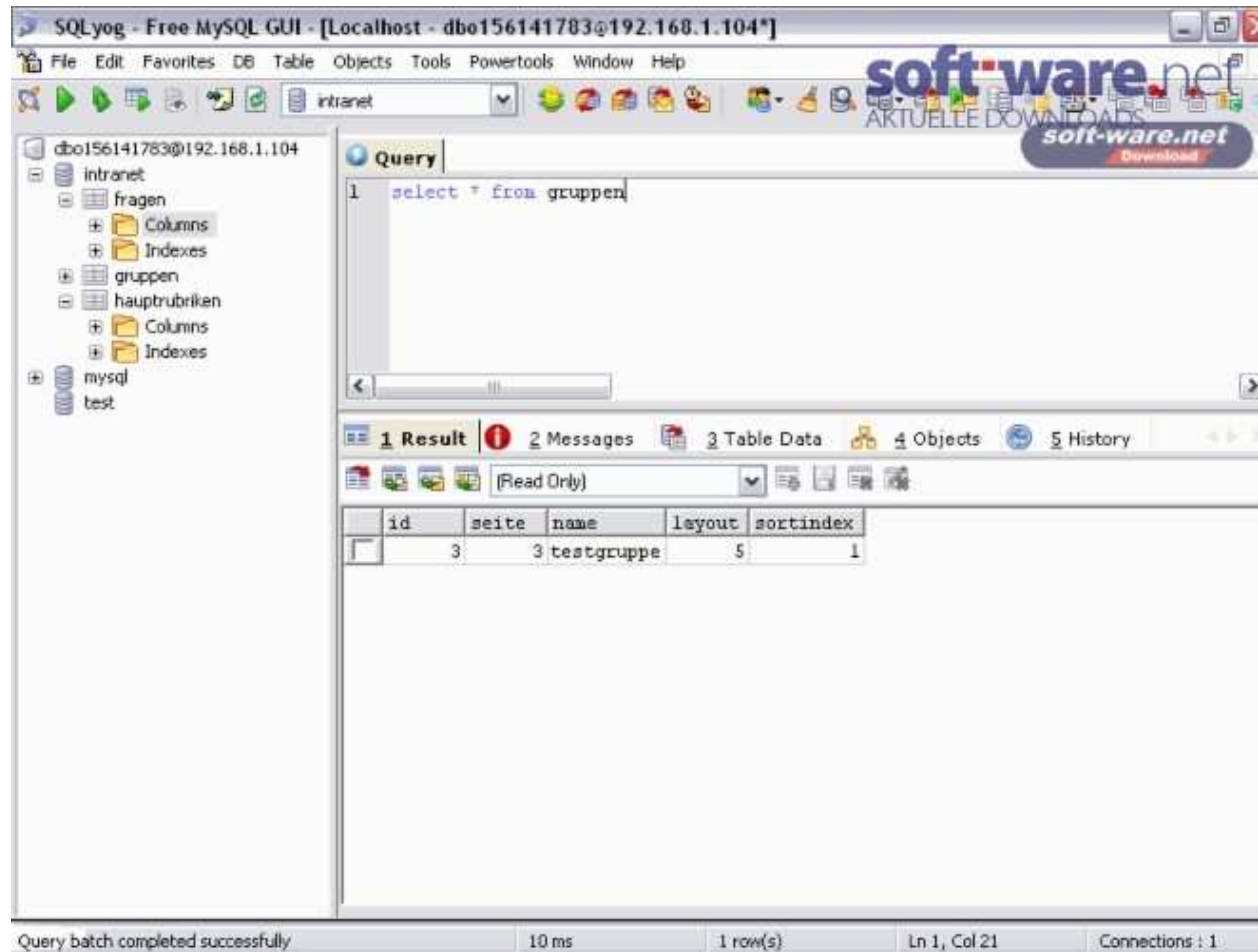
## SQLyog

SQLyog steht dem EMS SQL Manager und Navicat in nichts nach. Webyog Softworks stellt mit SQLyog ein leistungsfähiges Datenbankadministrations- und Entwicklungswerkzeug zur Verfügung.

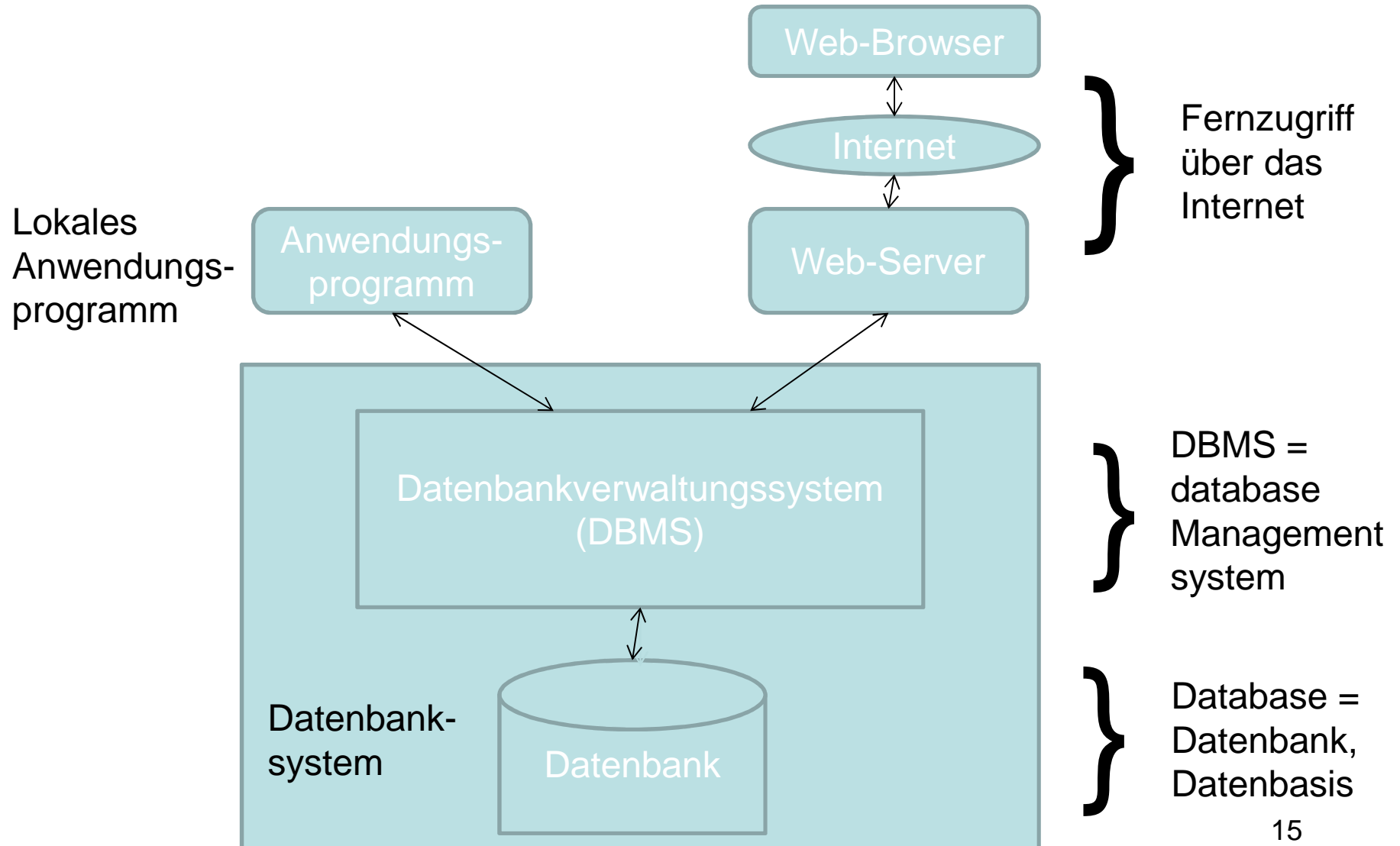
- Erzeugen und Löschen von Datenbanken, Tabellen, Indizes und Benutzern
- Erstellen und Ausführen von SQL-Abfragen
- Bearbeiten von Benutzerrechten (Sicherheitsrichtlinien)
- Sichern/Wiederherstellen von Datenbanken
- Datenübertragung – Replikation von Datenbanken
- Datensynchronisation und Struktursynchronisation
- Ausgezeichnete visuelle Textwerkzeuge für Query-Bildung
- Das Verbinden über den lokalen Port durch den SSH-Tunnel
- Optimierte Datenexport und -import-Eigenschaften
- Stored Procedure

# Graphische Clients für MySQL

## SQLyog



# Zugriff auf eine MySQL-Datenbank

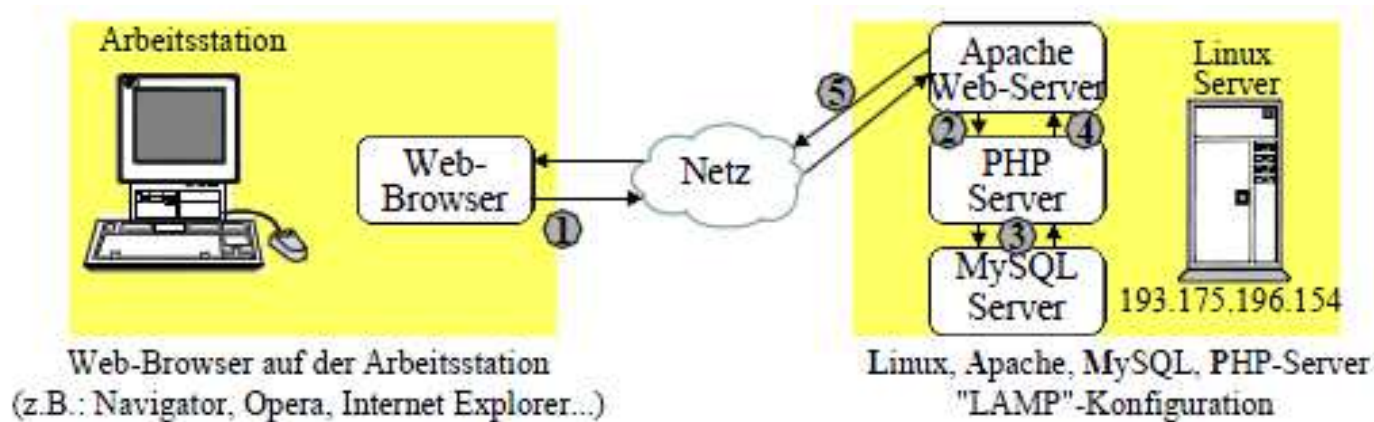


# „LAMP“ - Konfiguration

Alternativ zum grafischen Client: Technik des WWW einsetzen!

Prinzipieller Ablauf:

- 1) Nutzer ruft mit Browser eine Web-Seite auf dem Apache-Server auf
- 2) Apache Webserver übergibt Webseite an PHP-Server
- 3) PHP-Server liest Daten aus MySQL-Datenbank
- 4) PHP-Server verpackt die Daten in HTML-Format (für den Browser)
- 5) Apache Server übergibt die Web-Seite an den Browser





# Inhaltsverzeichnis

- Grundlagen von PHP
  - [Dokumentation](#)
  - [Einbettung in HTML](#)
  - [Variablen](#)
  - [Parameterübergabe](#)
  - [PHP-Beispiele](#)
  - [Arrays](#)
- Grundlagen von MySQL
  - [Dokumentation](#)
  - [MySQL-Server](#)
  - [MySQL-Workbench](#)
  - [ODBC-Client](#)
- PHP-Zugriff auf MySQL-Datenbanken
  - [PHP-Client](#)
  - [MySQL-Funktionen in PHP](#)
  - [MySQL-Beispiele](#)
  - [phpMyAdmin](#)
- [Übungsaufgaben](#)

# Informationen - Dokumentationen

- [HRZ-Dokumentation PHP-Handbuch](#) Sehr umfangreiches Referenz-Manual in 10 Sprachen
- [SELFPHP](#) Sehr umfangreiche Funktionsreferenz mit vielen Beispielen
- [Weaverslave](#) PHP-Editor
- [Zend Technologies](#) "where PHP meets eBusiness,,
- [phpWizard](#) Web Application Development with PHP (Tobias Ratschiller)
- [phpCenter.de](#) Deutschsprachiges PHP-Portal [phpWelt.de](#) Deutschsprachiges PHP-Portal
- [PHP-Homepage.de](#) Deutschsprachiges PHP-Portal
- [Bücher RRZN-Handbuch](#) im HRZ erhältlich
- [Dynamische Webseiten-Generierung](#) Vorlesung mit Übungen von (A. Wassermann)
- [PHP Schulung](#) Tutorium (Ulf Wendel, Johann-Peter Hartmann)
- [PHP-Kochrezepte für typische Anwendungen](#) Beispielsammlung (Bernd Cappel, Universität Düsseldorf)
- [PHP-Usergroups PEAR](#), [Vortrag dazu](#) PHP Extension and Application Repository

# Tools

- XAMPP
- Notepad++
- MySQL-Workbench
- Phase 5 HTML-Editor

# HTML (Hypertext Markup Language)

- PHP-Code kann mit HTML-Code gemischt werden  
Prinzipieller Aufbau einer Webseite in HTML:

```
<html>  
  <head>  
    <title> Seitenüberschrift </title>  
  </head>  
  <body>  
    Hier steht die Beschreibung des Seiteninhalts  
  </body>  
</html>
```

- Kennzeichnend sind die Start- und Ende-Marken (→ „Markup-Language“)

Weitere Marken:

- <center> = zentrieren </center>
- <h1> = Überschrift (header) </h1>
- <font ...>=Schriftart </font>
- <a href=...> = Hyperlink </a>
- <br> = neue Zeile (break) (ohne Ende-Marke) </br> usw...



Xhtml  
verwenden

# PHP in HTML einbetten

- PHP ist eine Scriptsprache  
→ Datei ist eine Text-Datei (ASCII-Code, mit Texteditor lesbar)
- PHP-Kommandos sind in HTML-Code eingebettet  
→ als Kommentar, daher im Browser-Darstellung nicht sichtbar
- PHP-Kommandos haben Start-Marke und End-Marke

```
<?PHP
    echo "Guten Abend!"
    mysql_connect(...);
    mysql_close(...);
?>
<!-- empfohlen! -->
```

Regel: Die HTML-Bestandteile bleiben unangetastet.  
Die PHP-Teile werden verarbeitet und müssen zu  
korrektem HTML-Code führen.

- PHP-Sprache ist von Programmiersprache C abgeleitet.
- PHP-Kommandos werden vom PHP-Server interpretiert.
- PHP-Script kann Daten einlesen und ausgeben.
- PHP-Dateien haben die Endung \*.PHP statt \*.HTM

# Historie von PHP

- 1994: Rasmus Lerdorf entwickelt eine kleine Skriptsprache
  - Ziel: Programmierung eines Web-Servers
  - Name: Personal Home Page Tools = PHPQuellcode wird im Internet frei verbreitet = PHP Version 1
- 1995: Open-Source-Szene entwickelt PHP weiter zu PHP/FI (Personal Homepage Tools/Form Interface: PHP kann mit HTML-Formularen Daten austauschen.
- 1997: PHP3 wird veröffentlicht
- 2000: PHP4 wird veröffentlicht
  - neuer Kernel: ZEND (von Zeev Suraski und Andi Gutman)
- 2004: PHP 5 wird veröffentlicht (Zend II Engine; verbessertes Objektmodell)
- 2011: PHP 5.3 Objektorientierung wurde erweitert

Alternativen zu PHP:

- Perl: bereits seit 1986 verfügbar, sehr gute Scriptverarbeitung
- ASP (Active Server Pages), basiert auf VB-Script (nur für Microsoftsysteme)
- Python, Java, TCL ...

# Variablen in PHP

- **Variable in PHP**
- Variablen-Namen beginnen immer mit einem Dollarzeichen:

```
$pi = 3.1415982;  
$Meldung = "Bitte geben Sie Ihre Geheimzahl ein!";  
$auch_eine_Textkonstante='Die Geheimzahl ist falsch,;
```

In Variablen-Namen wird Groß-Klein-Schreibung unterschieden.

- Bei der Ausgabe von Texten mit eingebetteten Variablen-Namen werden diese durch ihre Werte ersetzt:

```
echo "Die Konstante pi hat den Wert $pi";
```

- Das gilt nur, wenn die Texte in " eingefasst sind, nicht aber wenn sie in ' eingefasst sind.

# Parameterübergabe

- **Parameterübergabe**

- Methode GET

```
HTTP://meinServer.de/beispiel.php?name  
=Weiss&vorname=Dieter
```

- Methode POST

Damit stehen dem Programm beispiel.php zwei Variable zur Verfügung, so als ob sie wie folgt vereinbart worden wären:

```
$name = "Weiss";  
$vorname = "Dieter";  
Methode POST Browser-DarstellungHTML-  
Source dazuName: Vorname:  
<FORM METHOD="POST"  
ACTION="beispiel.php">  
Name: <INPUT TYPE="TEXT" NAME="name">  
Vorname: <INPUT TYPE="TEXT"  
NAME="vorname">  
<INPUT TYPE="SUBMIT" NAME="Absenden"  
VALUE="Absenden"></FORM>
```

Auch hier stehen dem Programm beispiel.php die entsprechenden Variablen zur Verfügung.

Browserdarstellung:

Name:

Vorname:



# Grundlagen von MySQL

## Informationen

- MySQL-Homepage: <http://www.mysql.com>
- Referenz Manual: [http://www.uni-giessen.de/hrz/htmldocs/mysql/manual\\_toc.html](http://www.uni-giessen.de/hrz/htmldocs/mysql/manual_toc.html)
- Deutsches Handbuch: <http://www2.little-idiot.de/mysql>
- phpMyAdmin: <http://www.phpwizard.net/projects/phpMyAdmin/>
- MySQL-Buch mit Online Ergänzungen: <http://www.kofler.cc/mysql/mysqlbuch.html#updates>

# MySQL - Beispiel

The screenshot shows a Microsoft Internet Explorer browser window displaying a web page. The address bar shows the URL: `http://www.uni-giessen.de/~g004/php/demo.php?datei=datenbankabfrage.php`. The page content is split into two columns.

The left column displays the output of a PHP script, which is a list of resource IDs and names with dates. The output starts with `$a=Resource id #3` and `$b=1`, followed by a list of entries like `TRallala 12334155 2004-06-13`, `Mattes 78624984 2004-06-14`, and so on, ending with `shsfhfh 323525 2004-08-14`.

The right column is titled "Source-Code" and shows the PHP code used to generate the output. The code includes a header, a title "Datenbankabfrage", and a PHP script that connects to a MySQL database, selects data from a table named "teilnehmer", and echoes the results in a formatted way.

```
Source-Code
<!DOCTYPE HTML PUBLIC "-//SoftQuad Software//DTD HoTMetal PRO
6.0::19990601:extensions to HTML 4.0//EN" "hmpro6.dtd">
<HTML>
  <HEAD>
    <TITLE>Datenbankabfrage</TITLE>
  </HEAD>
  <BODY> <SCRIPT LANGUAGE="php">

require ".htgeheim";
$a=mysql_connect ($dbhost,$dbuser, $dbpasswd) or die ("Keine Verbindung
zum Server");
echo "\$a=$a<br>": $b=mysql_select_db ("workshop_w6") or die ("Keine
Verbindung zur Datenbank");
echo "\$b=$b<br>":$ergebnis = mysql_query ("SELECT * FROM teilnehmer") or
die ("SQL-Fehler");
echo "\$ergebnis=$ergebnis<br>";
while ($zeile = mysql_fetch_array($ergebnis))
{ //echo "\$zeile=$zeile<br>";
$name=$zeile["name"];
$tel=$zeile['tel'];
$dat=$zeile['Einfuegedatum'];
echo "\$name $tel $dat<BR> ";
}
mysql_close ();
</SCRIPT> </BODY>
</HTML>
1
```

# MySQL – Funktionen in PHP

- Sequenz einer Datenbank-Abfrage

```
mysql_connect ("host", "user", "password") or die  
("Keine Verbindung zum Server");
```

```
mysql_select_db ("datenbank") or die  
("Keine Verbindung zur Datenbank");
```

```
$ergebnis = mysql_query ("SELECT spalten FROM  
tabelle") or die  
("SQL-Fehler");
```

```
while ($zeile = mysql_fetch_array($ergebnis))  
{  
    $var=$zeile['feldname1'] ;  
    echo "$var <BR>";  
}
```

```
mysql_close ();  
Verbindung zum Server schließen
```

- Verbindung zum Server
- Datenbank auswählen
- SQL – Abfrage
- Abfrage – Ergebnis zeilenweise ausgeben
- Verbindung zum Server schließen

# MySQL - Beispiel

The screenshot shows a Microsoft Internet Explorer browser window displaying a web page titled "Testdatenbank workshop\_w6". The page contains a form for entering data for a new participant, with fields for "Name:" and "Telefon:" and a "absenden" button. The source code of the page is displayed on the right side, showing PHP code that connects to a MySQL database, inserts data into a table named "teilnehmer", and outputs the entered data.

**Testdatenbank workshop\_w6**

**Dateneingabe für neuen Teilnehmer:**

Name:

Telefon:

**Source-Code**

```
<!DOCTYPE HTML PUBLIC "-//SoftQuad Software//DTD HoTMetaL PRO
6.0:19990601::extensions to HTML 4.0/EN" "hmpro6.dtd">
<HTML>
<HEAD>
<TITLE>Datenbankeingabe</TITLE>
</HEAD>
<BODY>
<H1>Testdatenbank workshop_w6</H1> <?

require ".htgeheim";

if (isset($name))
{
    $a=mysql_connect ($dbhost, $dbuser, $dbpasswd) or die ("Keine
Verbindung zum Server");
    $b=mysql_select_db ("workshop_w6") or die ("Keine Verbindung zur
Datenbank");
    $querystring=
    "INSERT INTO teilnehmer (name, tel, Einfuegedatum) VALUES
(' $name', '$tel',NOW())" ;
    $ergebnis = mysql_query ($querystring) or die ("SQL-Fehler");
    mysql_close ();
    ?>
<P><B>Daten eingegeben:</B><BR> <? echo "$name<br> $tel <br>"; ) else
( ?>

</P>
<H2>Dateneingabe fsuml;r neuen Teilnehmer:</H2>
<FORM METHOD="POST" ACTION="<?echo $PHP_SELF;?>">
<TABLE>
<TR>
<TH>Name:</TH>
<TH><INPUT TYPE="TEXT" NAME="name"></TH>
</TR>
<TR>
<TH>Telefon:</TH>
<TH><INPUT TYPE="TEXT" NAME="tel"></TH>
<TH><INPUT TYPE="SUBMIT" NAME="Speichern"
VALUE="absenden"></TH>
</TR>
</TABLE> </FORM> <?

)
?>
</BODY>
</HTML>
1
```

# Allgemeine Informationen zur Studienarbeit

- Termine:

1. Vorlesung:	06.11.2015	Architekturfestlegung
2. Vorlesung:	13.11.2015	
3. Vorlesung:	20.11.2015	Daten parsen
4. Vorlesung:	27.11.2015	
5. Vorlesung:	04.12.2015	Googlmepsanwendung
6. Vorlesung:	11.12.2015	Vorführung am Rechner
7. Vorlesung:	18.12.2015	Vorführung am Rechner