

# **Schriftliche Hausarbeit zur Abschlussprüfung der erweiternden Studien für Lehrer im Fach Informatik**

**VII. Weiterbildungskurs in Zusammenarbeit mit der Fernuniversität Hagen**

Eingereicht dem Amt für Lehrerbildung - Außenstelle Gießen -  
Vorsitzender des Prüfungsausschusses: Gerhard Jungermann

---

## **Modellierung eines webbasierten Informationssystems**

Einsatz eines LAMP-Systems in der Schule am Beispiel einer Software  
zur Verwaltung von Weiterbildungskursen für Lehrerinnen und Lehrer

---

Verfasser: Alexander Hauptmann

Gutachter: StD Gerhard Röhner

# Inhaltsverzeichnis

<b>1</b>	<b>VORWORT</b>	<b>4</b>
<b>2</b>	<b>EINLEITUNG</b>	<b>6</b>
<b>3</b>	<b>TEIL 1: ANALYSE</b>	<b>8</b>
<b>3.1</b>	<b>Thematische Einordnung in den Lehrplan</b>	<b>8</b>
<b>3.2</b>	<b>Technische Voraussetzungen</b>	<b>9</b>
3.2.1	Installation des Betriebssystems	10
3.2.2	Webserver Apache mit PHP-Modul	11
3.2.3	Datenbank MySQL	11
3.2.4	Fileserver Samba	12
3.2.5	FTP-Server	16
3.2.6	Clients	17
<b>3.3</b>	<b>Analyse eines Informatiksystems</b>	<b>18</b>
3.3.1	Anforderungen	18
3.3.2	Vorgaben	19
3.3.3	Pflichtenheft	19
3.3.4	UML und use cases	20
<b>4</b>	<b>TEIL 2: DESIGN</b>	<b>24</b>
<b>4.1</b>	<b>Modellierung der Datenbank</b>	<b>24</b>
4.1.1	Entity-Relationship-Modell	24
4.1.2	Datenmodell	29
4.1.3	Relationales Modell	30
4.1.4	Relationales Modell mit Datentypen	31
4.1.5	Vom Modell zur Datenbank	32
<b>4.2</b>	<b>Case-Tools</b>	<b>33</b>
4.2.1	DeZign for Databases	33
4.2.2	DBDesigner4	34
4.2.3	ERWin	34
<b>4.3</b>	<b>Das relationale Datenbanksystem MySQL</b>	<b>34</b>
4.3.1	SQL	34
4.3.2	DBMS	37
4.3.3	PhpMyAdmin	39
<b>5</b>	<b>TEIL 3: IMPLEMENTIERUNG</b>	<b>40</b>
<b>5.1</b>	<b>Einführung in HTML</b>	<b>40</b>
5.1.1	HTML	40
5.1.2	CSS	41

<b>5.2</b>	<b>PHP</b>	<b>42</b>
5.2.1	Verbindung zur Datenbank	42
5.2.2	Anzeigen von Einträgen	43
5.2.3	Einfügen von Daten	44
5.2.4	Löschen von Daten	45
5.2.5	Ändern von Daten	46
<b>5.3</b>	<b>Umsetzung</b>	<b>47</b>
5.3.1	Konventionen	47
5.3.2	Programmstruktur	49
<b>6</b>	<b>ZUSAMMENFASSUNG</b>	<b>57</b>
<b>7</b>	<b>LITERATURVERZEICHNIS</b>	<b>58</b>
<b>8</b>	<b>INTERNETADRESSEN</b>	<b>60</b>
<b>9</b>	<b>ANHANG</b>	<b>61</b>

# 1 Vorwort

Die Informatik repräsentiert in unserer Gesellschaft einen Bereich, der permanent an Wichtigkeit gewinnt. „Unsere Gesellschaft befindet sich an der Schwelle des Übergangs von der Industriegesellschaft zur Informations- und Wissensgesellschaft, in der aus Informationen generiertes Wissen zum entscheidenden Produktionsfaktor wird“ heißt es in den Empfehlungen der Gesellschaft für Informatik für ein Gesamtkonzept zur informatischen Bildung an allgemein bildenden Schulen (GESELLSCHAFT FÜR INFORMATIK, 2000). Und weiter heißt es dort: „Der Informatik als Wissenschaft kommt dabei eine Schlüsselrolle zu, da sie systematisch Möglichkeiten der automatischen Informationsverarbeitung und Wissensrepräsentation untersucht und in Informatiksystemen nutzbar macht“ (GESELLSCHAFT FÜR INFORMATIK, 2000).

Eines dieser häufig genutzten Informatiksysteme sind Datenbanken, die ein wichtiger Bestandteil unserer heutigen Lebenswelt sind. Datenbanken gehören zu den wichtigsten Produkten der Software-Industrie; kaum eine Informatikanwendung ist ohne Datenbankunterstützung denkbar (HÄRDER und RAHM, 2001). Beispiele dafür finden sich in den unterschiedlichsten Lebensbereichen. So spielen Datenbanken unter anderem eine Rolle bei der Verwaltung von Daten in einem Organizer, Warenbeständen einer Firma, den Patienten in einem Krankenhaus oder bei der Verwendung von Bestellsystemen im Internet.

Welche Ziele verfolgt der Informatikunterricht im Gymnasium? Im hessischen Lehrplan für das Fach Informatik im gymnasialen Bildungsgang (HESSISCHES KULTUSMINISTERIUM, 2003) heißt es dazu: „Wesentliches Ziel des Informatikunterrichts ist die Ausbildung der Sach-, Methoden-, Sozial- und Selbstkompetenz im Umgang mit Informations- und Kommunikationssystemen, welche die Schülerinnen und Schüler zur gleichberechtigten und aktiven Teilhabe am gesellschaftlichen Leben befähigen, sowie auf Studium, Beruf und Gestaltung des individuellen Lebens in der Informationsgesellschaft vorzubereiten.“ Der Informatikunterricht orientiert sich dabei an vier Leitlinien, die der Gestaltung der Jahrgangsstufe zu Grunde gelegt werden: Umgang mit Informationen, Wirkprinzipien von Informatiksystemen, Informatische Modellierung und Wechselwirkungen zwischen Informatiksystemen, Individuum und Gesellschaft. Dabei erstreckt sich jede Leitlinie auf die Jahrgangsstufen 11 bis 13 und wird spiralartig mehrfach durchlaufen. Die Ziele und Kursinhalte treten dabei in allen Jahrgangsstufen auf, wobei ihre Komplexität zunimmt und sich das Abstraktionsniveau erhöht.

Im Bereich „Umgang mit Informatiksystemen“ eignen sich die Schülerinnen und Schüler einen Vorrat an Grundstrategien und -methoden an, um Information zu beschaffen, zu strukturieren, zu bearbeiten, aufzubewahren und wiederzuverwenden, darzustellen, zu interpretieren, zu bewerten und zu präsentieren. Die Schülerinnen und Schüler lernen „Wirkprinzipien von Informatiksystemen“ kennen und verstehen, wie Informatiksysteme aufgebaut sind, nach welchen Funktionsprinzipien ihre Systemkomponenten zusammenwirken und wie sich diese in größere Systemzusammenhänge einordnen lassen. Ein wesentliches Ziel dabei ist die „Entmystifizierung von Informatiksystemen“ (GESELLSCHAFT FÜR INFORMATIK, 2000). Ein weiterer zentraler Themenkomplex ist die „Informatische Modellierung“, bei der die Schülerinnen und Schüler lernen, dass jedes Informatiksystem als Kombination von Hard- und Software-Komponenten das Ergebnis eines informatischen Modellierungsvorgangs ist. Durch die Kenntnis von Voraussetzungen und Folgen, Chancen und Risiken des Einsatzes komplexer Informatiksysteme, werden Schülerinnen und Schüler in die Lage versetzt, sich mit den „Wechselwirkungen zwischen Informatiksystemen, Individuum und Gesellschaft“ auseinanderzusetzen und sie lernen, sich verantwortungsbewusst an der Gestaltung und am Einsatz dieser Technologie zu beteiligen und ihre Zukunft menschengerecht zu gestalten.

Zusammenfassend lässt sich sagen: „Informatische Bildung ist das Ergebnis von Lernprozessen, in denen Grundlagen, Methoden, Anwendungen, Arbeitsweisen und die gesellschaftliche Bedeutung von Informatiksystemen erschlossen werden“ (GESELLSCHAFT FÜR INFORMATIK, 2000).

Die vorliegende Arbeit beschäftigt sich mit der Modellierung eines webbasierten Informationssystems. Sie orientiert sich dabei an der Empfehlung des hessischen Lehrplans: „Als Leitkonzept für die Gestaltung geeigneter Lernumgebungen eignet sich die Problemorientierung, die eine Balance zwischen Instruktion und Konstruktion einfordert“ (HESSISCHES KULTUSMINISTERIUM, 2003).

Der Begriff Informatiksystem wird im Folgenden häufiger verwendet und orientiert sich dabei an der von BAUMANN (1996) gegebenen Definition: „Unter einem Informatiksystem versteht man ein verteiltes, heterogenes technisches System, das Wissen unterschiedlicher Art und Herkunft repräsentiert, diese Wissensrepräsentation in Gestalt von Daten und Programmen verarbeitet und den Benutzern in geeigneter Form zur Verfügung stellt.“

## 2 Einleitung

Aufgabe ist die Erstellung eines webbasierten Kursverwaltungssystems für Veranstaltungen in der Lehrerfortbildung. Vorgegeben sind bereits Strukturen eines lokalen Kursverwaltungssystems (Vera) des **Hessischen Landesinstituts für Pädagogik (HeLP)**. Die im Zusammenhang mit diesem Datenbanksystem behandelten Fragestellungen betreffen die Strukturierung, die Darstellung und die Codierung von Daten. Sie berücksichtigen aber auch Aspekte der Sicherheit und des Schutzes dieser Daten vor fehlerhaftem oder unzulässigem Zugriff.

„Die Umsetzung und Bearbeitung der ER-Modelle (ERM) sollte realitätsnah in einem modernen Datenbanksystem erfolgen“ (HESSISCHES KULTUSMINISTERIUM, 2003). Die Forderung im Unterricht Beispiele aus dem Umfeld der Schülerinnen und Schüler zu wählen, die zudem noch interessant und praxisnah sind, zieht sich wie ein roter Faden durch die didaktische Literatur. Es ist allerdings schwer erprobte Beispiele zu finden. Die vorliegende Arbeit skizziert ein solches Beispiel von der Analyse über das Design bis zur Implementierung und zeigt alle Phasen der Entwicklung beginnend bei der Installation der Rechner und der Konfiguration der verwendeten Beispielprogramme auf. Da sich einerseits die verwendeten Hard- und Software-Komponenten stetig weiterentwickeln und andererseits eine große Anzahl an Alternativen existiert, das vorgestellte Informatiksystem zu realisieren, kann es sich hierbei nur um eine Momentaufnahme handeln.

Als Server für das vorliegende Beispiel dient ein handelsüblicher PC mit einem Linux-Betriebssystem (SuSE 8.2) und ausgewählten Komponenten (Webserver Apache mit PHP-Modul und Datenbank MySQL) der bei dieser Distribution mitgelieferten Software. Das vorgestellte Informatiksystem kann auch mit anderen Betriebssystemen (z.B. FreeBSD, Windows, MacOS), Webservern (z.B. IIS, OmniHTTPd, Tomcat), Scriptsprachen (z.B. ASP, ColdFusion, Perl, Java-Script) und Datenbanken (z.B. mSQL, PostgreSQL, ADABAS) verwirklicht werden. Grundsätzlich notwendig ist die Verwendung eines Webserver mit Unterstützung einer Scriptsprache und eine Datenbank. Bei der Verwendung anderer, als der im Beispiel genannten Softwarekomponenten, müssen entsprechende, systemspezifische Anpassungen vorgenommen werden.

Die vorliegende Arbeit ist in drei Teile gegliedert:

Im ersten Teil erfolgt zunächst eine Einordnung des Beispiels in den aktuellen Lehrplan. Es schließt sich eine Auflistung der Anforderungen an Hard- und Software und Hinweise zur Konfiguration der Einzelkomponenten an. Den Abschluss bildet die Gesamtplanung des

Informatiksystem mit der Erstellung eines Pflichtenhefts und den Anwendungsfällen (use cases).

Im zweiten Teil werden die Ergebnisse der Anforderungsermittlung analysiert und in das Entity-Relationship-Modell (ERM) übertragen. Danach erfolgt eine Einführung in das Datenbankmanagementsystem von MySQL und die Übertragung des ER-Modells in eine Datenbank.

Im dritten Teil erfolgt die eigentliche Implementierung der Anwendung unter Verwendung von HTML und der Scriptsprache PHP. Dazu wird ein Überblick über die wichtigsten Sprachelemente gegeben und anhand von Beispielen die Anbindung an die Datenbank und die Manipulation der gespeicherten Daten gezeigt. Da die grundlegenden Operationen mit einfachen Beispielen demonstriert werden, sind die Listings des kompletten Codes im Anhang aufgeführt.

Alle Programmbeispiele, Modelle und der Quellcode der Anwendung werden ebenfalls im Anhang und auf der beiliegenden CD zur Verfügung gestellt.

### 3 Teil 1: Analyse

#### 3.1 Thematische Einordnung in den Lehrplan

Im Folgenden sind stichwortartig die Themengebiete des derzeit gültigen hessischen Lehrplans für das Fach Informatik im gymnasialen Bildungsgang für die Jahrgangsstufe 11 bis 13 (gültig ab 01.08.2003) aufgelistet. Die Umsetzung des vorgeschlagenen Beispiels im Unterricht greift hierbei relevante Inhalte der unterschiedlichen Halbjahre auf (kursiv und unterstrichen markiert) und wiederholt diese im Sinne eines Spiralcurriculums. Insbesondere werden die Halbjahre 11.I, 11.II, 12.I und 12.II sowie bei geeignetem Wahlthema 13.II berücksichtigt.

#### Themen laut Lehrplan (HESSISCHES KULTUSMINISTERIUM, 2003):

##### 11.I - Internet

HTML, CSS

Struktur und Dienste

TCP/IP-Protokollfamilie

Client-Server-Architektur

Suchmaschinen

Internet-Datenbanken

##### 11.II - Grundlagen der Programmierung

Variablen

Datentypen

Kontrollstrukturen

Struktogramme

Modularisierung

Benutzeroberflächen

Modellierung

##### 12.I - Objektorientierte Modellierung

Objektmodell

Klassen

Abstrakte Datentypen

Standardalgorithmen

Software-Engineering

Komplexität

##### 12.II - Datenbanken

ER-Modell

Relationenmodell

Relationenalgebra

QBE/SQL

Normalformen

DDL, DCL, DML

##### 13.I - Theoretische Informatik

Formale Sprachen

Grammatiken

Automatentheorie

Berechenbarkeit

Kellerautomat

Turingmaschine

##### 13.II - Wahlthema

Betriebssysteme

Rechnernetze

Computergrafik

PROLOG als Sprache der KI

Simulationen, Chaostheorie

Technische Informatik



Analyse, Design und Implementierung des vorgestellten Beispiels kann im Rahmen eines Projekts am Ende des Halbjahres der 12.II oder bei entsprechendem Wahlthema am Ende des Halbjahres der 13.II erfolgen. Auch in einer Projektwoche könnte die Thematisierung eines solchen Informatiksystems sinnvoll sein.

### **3.2 Technische Voraussetzungen**

Serverseitig werden das Betriebssystem Linux, der Webserver Apache inklusive des Moduls zur Unterstützung der Scriptsprache PHP und die Datenbank MySQL benötigt. Um den Zugriff der Clientsysteme während der Systementwicklung gewährleisten zu können, bietet es sich zusätzlich an einen FTP- und/oder Fileserver (Samba) zu verwenden. Es ist nicht nötig eine grafische Benutzeroberfläche auf dem Server zu installieren. Dadurch sind die Anforderungen an die Hardware geringer und es genügt ein älteres, nicht dem neuesten Stand der Technik entsprechendes Gerät. Da sich die Linux-Distributionen hinsichtlich der Anforderungen stark unterscheiden, ist es kaum möglich Richtwerte anzugeben. Es sollte jedoch kein Problem sein geeignete Systeme zu finden, die bei einer textbasierten Installation mit 486 DX Prozessoren, 32 MB RAM und einer 400 MB Festplatte auskommen (z.B. SuSE 7.0 und älter).

Soll eine grafische Benutzeroberfläche installiert werden, muss der Server entsprechend besser ausgestattet sein. Die hier verwendete Linux-Distribution von SuSE (Version 8.2) verlangt bei einer normalen Installation bereits Rechner mit Prozessoren ab Pentium 1, mindestens 64 MB RAM und eine 1,5 GB Festplatte (dringend empfohlen).

Da es eine große Auswahl an Betriebssystemen und Softwarekomponenten gibt, mit der das vorgestellte Beispiel realisiert werden kann, und eine erschöpfende Beschreibung aller verfügbaren Systeme hier nicht möglich ist, soll im Folgenden nur die Installation und Konfiguration der bei der SuSE-Distribution mitgelieferten Software beschrieben werden. Hierbei handelt es sich lediglich um eine Ergänzung der jeweiligen Dokumentationen und nicht um eine lückenlose Ersetzung. Grundsätzlich sei hier auf die Handbücher und Dokumentationen der SuSE 8.2-Distribution verwiesen (URL: siehe Kapitel 8).

Für das weitere Vorgehen wird ein ausreichend dimensionierter Server vorausgesetzt.

### **3.2.1 Installation des Betriebssystems**

Mittlerweile lassen sich Linux-Systeme dank komfortabler Setup-Routinen ähnlich leicht installieren, wie man es aus der Microsoft-Welt gewohnt ist. Die Installation von SuSE 8.2 kann direkt durch Einlegen der DVD und Einschalten des Rechners gestartet werden.

Bei einer Neuinstallation des Systems wird in einem ersten Dialogfenster nach der gewünschten Sprache gefragt, die automatisch für das Tastaturlayout übernommen wird. Die Setup-Routine schlägt schließlich eine Konfiguration bestehend aus angeschlossenen Peripheriegeräten, Partitionen und einer Softwareauswahl vor. Hier kann eine Feinanpassung vorgenommen werden. Für das vorgestellte System müssen lediglich in der Rubrik Software die zusätzlichen Komponenten mitinstalliert werden (Webserver Apache, PHP-Modul, Datenbank MySQL, Fileserver Samba, FTP-Server). Bei allen anderen Rubriken können die Voreinstellungen belassen werden.

Im weiteren Verlauf der Konfiguration muss noch ein Passwort für den root-Benutzer angegeben werden. Anschließend wird das Netzwerk konfiguriert. Die Netzwerkkarte sollte mit einer statischen IP-Adresse ausgestattet werden, die in ein bestehendes Netzwerk integriert werden kann. Im einfachsten Fall kann dies die IP-Adresse „192.168.1.1“ mit der Subnetzmaske „255.255.255.0“ sein. Zusätzlich wird dem Server an dieser Stelle ein Name zugewiesen.

Da das Arbeiten als Superuser (root) nur zu Administrationszwecken genutzt werden sollte, ist es nötig mindestens einen weiteren Benutzer anzulegen. Bereits während der Installation wird dazu ein Dialogfenster bereitgestellt, in dem die Angaben für einen ersten Benutzer angegeben werden sollten.

Schließlich werden die erkannten Peripheriegeräte konfiguriert. Die Erkennung der Hardware erfolgt bei Standardgeräten in der Regel problemlos und die vom System gemachten Vorschläge zur Konfiguration können im Allgemeinen übernommen werden.

Nach einem Neustart des Systems muss noch überprüft werden, ob die installierten Dienste mitgestartet wurden. Ist dies nicht der Fall müssen im Runlevel-Editor die gewünschten Dienste (MySQL, Samba) zum Start vorgesehen werden.

### 3.2.2 Webserver Apache mit PHP-Modul

Anfang 1995 war der von Rob McCool entwickelte NCSA-Webserver der meistgenutzte Webserver im Internet (EILEBRECHT, 1997). Die Weiterentwicklung durch Hinzufügen von Verbesserungen (patches) einer Vielzahl unterschiedlicher Entwickler führte schließlich zu einem gepatchten (a patchy server) Server. Daraus entstand der Name Apache (URL: siehe Kapitel 8).

Der Webserver wird nach einer Neuinstallation standardmäßig beim Hochfahren des Systems gestartet. Im Wurzelverzeichnis des Webserver (in der SuSE-Distribution 8.2 ist dies das Verzeichnis „/srv/www/htdocs“) befindet sich bereits eine Index-Datei (index.htm), die vom Webserver als Startseite angezeigt wird. Wenn der Webserver erfolgreich gestartet ist, kann über einen Client mit einem Browser auf die Startseite des neu installierten Webserver zugegriffen werden. Dies kann über das Netzwerk mit einem anderen Rechner, dessen Netzwerkdienste korrekt konfiguriert sein müssen, oder über den Standardbrowser Konqueror des Servers selbst geschehen. Dazu wird in der Adresszeile des Browsers die IP-Adresse des Servers (in diesem Fall 192.168.1.1) eingetragen. Ob der Webserver läuft kann mit dem Befehl „ps -x | grep http“ über die Kommandozeile herausgefunden werden.

```
p15106343:~ # ps -x | grep http
12141 ?          _S      2:08 /usr/sbin/httpd -f /etc/httpd/httpd.conf -D SSL
```

### 3.2.3 Datenbank MySQL

Die Wurzeln von MySQL gehen auf das Jahr 1979 und das Datenbank-Tool UNIREG zurück, das Michael „Monty“ Widenius für die schwedische Firma TcX erschaffen hat (DUBOIS, 2000). Für diese Firma entwickelte Widenius einen neuen SQL-Server, der eine Programmierschnittstelle zur existierenden Datenbank mSQL hatte. Ab dem Jahr 1995 wurde die Veröffentlichung von MySQL im Internet vorangetrieben (URL: siehe Kapitel 8).

Die Datenbank MySQL wird nicht automatisch beim Hochfahren des Systems gestartet. Daher muss zunächst im Runlevel-Editor ein entsprechender Eintrag vorgenommen werden. Ob die Datenbank erfolgreich gestartet wurde kann über die Kommandozeile mit dem Befehl „ps -x | grep mysql“ überprüft werden.

```
p15106343:~ # ps -x | grep mysql
321 ?          S      0:00 /bin/sh /usr/bin/safe_mysqld --user=mysql
```

Genauere Informationen über die Datenbank MySQL liefert das Kommando „mysqladmin version“. Anhand der erscheinenden Bildschirmmeldung ist die verwendete Version der Datenbank, die verwendete Protokollversion und der Status der Datenbank erkennbar.

```
mysqladmin Ver 8.19 Distrib 3.23.37, for suse-linux on i686
Copyright (C) 2000 MySQL AB & MySQL Finland AB & TCX DataKonsult AB
This software comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to modify and redistribute it under the GPL license

Server version      3.23.37-log
Protocol version    10
Connection          Localhost via UNIX socket
UNIX socket         /var/lib/mysql/mysql.sock
Uptime:             74 days 21 hours 47 min 20 sec

Threads: 1 Questions: 619673 Slow queries: 0 Opens: 74 Flush tables: 1 Open tables: 64 Queries per second avg: 0.096
```

### 3.2.4 Fileserver Samba

Andrew Tridgell begann die Entwicklung eines SMB-Servers im Dezember 1991. Den späteren Namen erhielt die Software aufgrund eines Rechtsproblems. Andrew Tridgell suchte für seinen SMB-Server in einem Wörterbuch alle Namen, die der Kombination aus S\*M\*B entsprachen, wobei das „\*“ für beliebige Buchstaben steht. Aus den Treffern wählte er den Namen „Samba“, der ihm am Besten gefiel (KÜHNEL, 1999).

Samba ist ein Paket von Programmen, das es Betriebssystemen wie Unix oder Linux möglich macht, Dienste über das Microsoft-eigene Netzwerk-Protokoll SMB anzubieten (RAISON und KIRSCH, 1999). Samba kann so die Datei- und Druckerressourcen eines Linux-Servers anderen Rechnern zur Verfügung zu stellen. Darüber hinaus verfügt Samba über die Fähigkeit noch weitere Funktionalitäten eines Windows-Servers in einem Netzwerk zu übernehmen. Auch der Fileserver Samba wird nach einer Neuinstallation nicht automatisch beim Hochfahren des Systems gestartet. Daher muss dieser Dienst im Runlevel-Editor für einen Start im jeweiligen Runlevel vorgemerkt werden.

Die Hauptaufgabe von Samba ist die Freigabe von Netzlaufwerken für Windows-Clients. Damit können diese Clients auf Verzeichnisse und Dateien des Linux-Servers zugreifen. Da aufgrund der unterschiedlichen Konzepte der Betriebssysteme Linux und Windows unter anderem Differenzen im Bereich der Systemsicherheit und Benutzerführung existieren, sind zunächst einige Vorarbeiten nötig, die einen fehlerfreien Betrieb garantieren. Bei der Installation werden die grundlegenden Dateien für Konfiguration und Wartung sowie die notwendigen Verzeichnisse bereits angelegt.

Damit ein Benutzer von einem Windows-Client auf eine Freigabe des Servers zugreifen kann, muss auf dem Zielsystem ebenfalls ein Benutzer mit gleichem Anmeldenamen und gleichem Passwort existieren. Dieser Benutzer muss zunächst auf dem Linux-System über die Benutzerverwaltung angelegt werden. Da die Passwörter eines Windows- und eines Linux-Servers nicht identisch verwaltet werden, muss zusätzlich eine Passwortliste für den Fileserver angelegt werden. Dies kann über die Kommandozeile mit dem Befehl „smbpasswd -a Benutzername“ erledigt werden. In einem Dialog wird der Benutzer für Samba angelegt und ein Passwort angegeben.

```
obiwan:/etc/samba # smbpasswd -a alex
New SMB password:
Retype new SMB password:
Added user alex.
```

Der Fileserver Samba wird mit Hilfe einer Konfigurationsdatei verwaltet, in der alle Angaben über Freigaben enthalten sind. Die zentrale Konfigurationsdatei für Linux als Samba-Server ist die Datei „/etc/samba/smb.conf“ (BORKNER-DELCARLO, 1999). In dieser Datei muss das Verzeichnis des Webservers freigegeben werden. Die Datei „smb.conf“ (in der SuSE-Distribution 8.2 liegt die Datei im Verzeichnis „/etc/samba“) besteht aus verschiedenen Sektionen, die jeweils mit einem Sektionsnamen in eckigen Klammern beginnen. In der Regel stellt jede dieser Sektionen eine Freigabe dar. Alle Optionen bis zum Beginn der nächsten Freigabe gehören zur aktuellen Freigabe. Mit Hilfe von Parametern können die Eigenschaften der Freigaben beeinflusst werden.

Um zum Beispiel das Verzeichnis des Webservers unter dem Namen „www“ freizugeben genügt die Angabe folgender Parameter in der Datei „smb.conf“.

[www]:	Freigabename
comment:	Kommentar, der in der Netzwerkumgebung angezeigt wird
path:	Pfad für die Dateien des Webservers
browsable:	Die Freigabe wird in der Netzwerkumgebung angezeigt
writable:	Die Freigabe ist beschreibbar
create mask:	Berechtigung für neu erzeugte Dateien
directory mask:	Berechtigung für neu erzeugte Verzeichnisse

```
[www]
comment = Webverzeichnis
path = /srv/www/htdocs
browsable = yes
writable = yes
create mask = 0775
directory mask = 0775
```

Nachdem die geänderte Datei abgespeichert wurde muss der Fileserver über die Kommandozeile mit den folgenden Befehlen neu gestartet werden. Dies ist notwendig, damit die geänderte Konfigurationsdatei neu eingelesen wird.

rcsmb restart: Neustart des Fileservers  
rcnmb restart Neustart des netbios-Nameservers

```
obiwan:/etc/samba # rcsmb restart
Shutting down Samba classic SMB daemon           done
Starting Samba classic SMB daemon                 done
obiwan:/etc/samba # rcnmb restart
Shutting down Samba classic NMB daemon            done
Starting Samba classic NMB daemon                 done
obiwan:/etc/samba #
```

Obwohl die entsprechenden Benutzer angelegt wurden und die Freigabe „www“ bereits über Clients erreichbar ist, führen Schreibversuche in der Freigabe zu einer Fehlermeldung mit dem Hinweis, dass keine Berechtigung existiert. Dies liegt daran, dass die freigegebenen Verzeichnisse und Dateien noch keine Schreibberechtigung für den angelegten Benutzer besitzen.

Um dies zu überprüfen wechselt man mit dem Befehl „cd /srv/www“ in das Verzeichnis des Webservers und erfragt mit dem Kommando „ls -al“ die aktuellen Rechte der Verzeichnisse und Dateien.

```
obiwan:/etc/samba # cd /srv/www/
obiwan:/srv/www # ls -al
total 5
drwxr-xr-x  6 root  root    144 Nov  1  2002 .
drwxr-xr-x  4 root  root     96 Nov  1  2002 ..
drwxr-xr-x  3 root  root   896 Apr 26 19:13 cgi-bin
drwxr-xr-x  5 root  root   152 Nov  1  2002 htdig
drwxr-xr-x  6 root  root   392 Apr 26 19:12 htdocs
drwxr-xr-x  3 root  root  4744 Apr 26 19:12 icons
obiwan:/srv/www #
```

In der Anzeige sind mehrere Spalten erkennbar. Die erste Spalte besteht aus 10 Zeichen und macht Angaben über die Rechte von Dateien und Verzeichnissen. Das erste Zeichen der ersten Spalte gibt an, ob es sich um ein Verzeichnis („d“) oder eine Datei („-“) handelt. Die nachfolgenden Zeichen sind in „Dreier-Päckchen“ organisiert, wobei das erste Päckchen Angaben über die Rechte des Besitzers macht, das zweite Päckchen die Rechte der Gruppe anzeigt und das letzte Päckchen die Rechte aller übrigen Benutzer beschreibt. Das Kürzel „r“ steht für eine Leseberechtigung, das Kürzel „w“ für eine Schreibberechtigung und das Kürzel „x“ für das Ausführen von Dateien. Es folgen die Anzahl der Einträge, der Name des Besitzer, die Gruppe, Größe, Datum der letzten Änderung und Name der Datei bzw. des Verzeichnisses.

Das Verzeichnis „htdocs“ gehört im angegebenen Beispiel dem Benutzer „root“ und weist die Berechtigungen „rwx“ für den Besitzer (root), „rx“ für die Gruppe (root) und alle anderen Benutzer aus. Somit hat nur der Besitzer „root“ zurzeit das Recht zum Ändern (Schreiben, Löschen, Anlegen) von Dateien im Verzeichnis „htdocs“. Um den Benutzern eines Windows-Rechners die Möglichkeit zu geben über eine Freigabe bequem auf das Webserver-Verzeichnis zugreifen zu können, müssen einige Änderungen vorgenommen werden:

- Das Verzeichnis wird einem normalen Benutzer zugeordnet:  
chown Benutzername Verzeichnisname
- Das Verzeichnis wird einer normalen Gruppe zugeordnet:  
chgrp Gruppenname Verzeichnisname
- Die Zugriffsrechte der Gruppe werden geändert:  
chmod Rechte Verzeichnisname

Die entsprechenden Befehle werden in die Kommandozeile eingegeben und anschließend wird das Ergebnis überprüft:

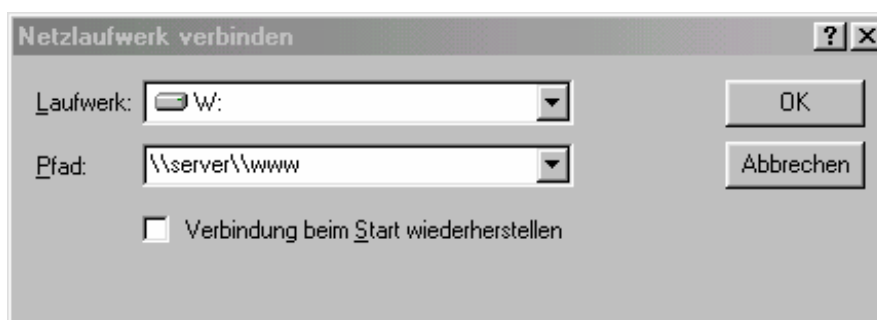
```
obiwan:/srv/www # chown alex htdocs
obiwan:/srv/www # chgrp users htdocs
obiwan:/srv/www # chmod g+w htdocs
obiwan:/srv/www # ls -al
total 5
drwxr-xr-x  6 root    root      144 Nov  1  2002 .
drwxr-xr-x  4 root    root        96 Nov  1  2002 ..
drwxr-xr-x  3 root    root      896 Apr 26 19:13 cgi-bin
drwxr-xr-x  5 root    root      152 Nov  1  2002 htdig
drwxrwxr-x  6 alex    users    392 Apr 26 19:12 htdocs
drwxr-xr-x  3 root    root     4744 Apr 26 19:12 icons
obiwan:/srv/www #
```

Das Ergebnis der Änderungen der Berechtigungen zeigt, dass das Verzeichnis „htdocs“ jetzt dem Benutzer „alex“ gehört und der Gruppe „users“ zugewiesen wurde. Zusätzlich haben Angehörige der Gruppe „users“ Schreibrechte in diesem Verzeichnis. Da alle neu angelegten Benutzer zunächst dieser Gruppe zugeordnet werden, haben jetzt alle Benutzer,

- die Zugang zu einem Windows-Client haben,
- auf dem Linux-Server als Benutzer geführt sind und
- für die eine Benutzerkennung unter Samba angelegt wurde,

Schreibzugriff auf die Freigabe mit dem Namen „www“.

Jetzt kann von einem Windows-Rechner über den Explorer auf die Freigabe des Linux-Servers mit der Option „Netzlaufwerk verbinden“ zugegriffen werden. Dazu müssen ein Laufwerksbuchstabe, der frei gewählt werden kann, der Name des Servers (dieser Name des Servers wurde bei der Netzwerkkonfiguration festgelegt) und der Name der Freigabe (www) angegeben werden. Dann wird das Verzeichnis des Webservers (im Beispiel als Laufwerk „W“) eingebunden und HTML-Seiten können dort bequem abgespeichert werden.



Die jeweils neueste Samba-Version ist im Internet verfügbar (URL: siehe Kapitel 8).

### 3.2.5 FTP-Server

Eine weitere Möglichkeit auf das Verzeichnis des Webservers zuzugreifen stellt das **File Transfer Protokoll (FTP)** dar. Ist ein funktionsfähiger FTP-Server installiert, kann man sich mit einem auf dem Linux-System vorhandenen Benutzernamen anmelden. Nach erfolgreicher Anmeldung steht das Home-Verzeichnis des jeweiligen Benutzers zur Verfügung.

Um Dateien auf den Server verschieben zu können wechselt man in das Verzeichnis des Webservers (/srv/www/htdocs).



Eine bequeme Variante ist das Setzen eines Links vom Home-Verzeichnis des Benutzers auf das Verzeichnis des Webserver.

Einige FTP-Server bieten die Möglichkeit als Startverzeichnis ein anderes als das Home-Verzeichnis des Benutzers zur Verfügung zu stellen. So kann in der Konfigurationsdatei des FTP-Servers „ProFTPD“ auch das Verzeichnis des Webserver als Startverzeichnis angegeben werden. ProFTPD ist nicht Teil der SuSE-Distribution. Das Programm kann jedoch aus dem Internet heruntergeladen werden (URL: siehe Kapitel 8)

Auch bei diesen Methoden müssen die entsprechenden Rechte für den angemeldeten Benutzer korrekt gesetzt werden.

### **3.2.6 Clients**

Auf den Client-Rechnern benötigt man einen Editor zur Erstellung der Webseiten und Scripte. Die Verbindung zum Server wird über den Explorer oder einen geeigneten FTP-Client hergestellt. Die Webseiten selbst werden über einen Browser angezeigt.

Zur Programmierung des vorgestellten Systems werden keine zusätzlichen Programme benötigt. Es ist möglich den Editor von Windows (notepad) als Werkzeug zur Erstellung der HTML-Seiten zu verwenden und die Verbindung zum Server über den Explorer zu realisieren. Trotzdem kann man sich mit einigen Hilfsmitteln die Arbeit erleichtern. Zu empfehlen ist der kostenlose HTML-Editor „Phase 5“ von Ulli Meybohm (URL: siehe Kapitel 8) zur Erstellung der Scripte.

### 3.3 Analyse eines Informatiksystems

“A successful software organization is one that consistently deploys quality software that meets the needs of its users” (BOOCH ET AL. 2001). Jedes Informatiksystem bedarf zu Beginn einer gründlichen Planung. In dieser Phase werden die Anforderungen, die an die Software gestellt werden, ermittelt und in einer Spezifikation (Pflichtenheft) festgeschrieben. Die Anforderungen an die zu erstellende Software entstammen dem unmittelbaren Anwendungsbereich. Als Basis dienen beispielsweise Gespräche mit den Auftraggebern, existierende Formulare und eventuell bereits vorhandene Softwarelösungen. In einem Pflichtenheft werden die aus der Anforderungsermittlung stammenden Ergebnisse genau festgeschrieben. Dieses Pflichtenheft enthält lediglich Anweisungen was die Software leisten soll, macht jedoch noch keine Vorgaben auf welchem Weg dies realisiert werden soll. Am Ende der Softwareentwicklung erfolgt die Endabnahme nach den Vorgaben des Pflichtenheftes.

Hilfreich bei der Erstellung komplexer Informatiksysteme ist die genaue Betrachtung der Anwendungsfälle (use cases), die in der Lage sind Arbeitsabläufe zu modellieren. Insbesondere bei der Planung der Benutzeroberfläche spielen diese Anwendungsfälle eine große Rolle.

#### 3.3.1 Anforderungen

Die geplante Anwendung soll Informationen der realen Welt verwalten und speichern. Dazu ist es notwendig sich auf bestimmte, nur für die Anwendung relevante Merkmale zu beschränken. Um die Objekte, die zu diesem Ausschnitt der Realität gehören, erfassen zu können, benötigt man zunächst Informationen darüber, was das Anwendungssystem leisten soll:

- Es soll ein Kursverwaltungssystem erstellt werden, mit dem Weiterbildungskurse für Lehrerinnen und Lehrer organisiert werden können.
- Diese Kursverwaltung soll über das Internet verfügbar und mit möglichst beliebigen Browsern bedienbar sein.
- Das Verwaltungssystem muss Daten für Schulen, Kurse, Teilnehmer, Referenten und Kursorte verwalten und Listen mit den erforderlichen Daten zur Verfügung stellen.
- Der Zugriff auf das System muss für Teilnehmer, Referenten und Administratoren unterschiedlich geregelt sein.

- Der Datenaustausch mit dem offline-Verwaltungssystem des HeLP soll durch einheitliche Datenstrukturen möglich sein.

### **3.3.2 Vorgaben**

Das offline-Verwaltungssystem "Vera" des HeLP stellt eine Importfunktion für Teilnehmer, Referenten, Kurse und Schulen zur Verfügung. Der Abgleich der Feldnamen erfolgt über verschiedene Eingabemasken. Es müssen folgende Daten verwaltet werden:

- Teilnehmer/innen (Anrede, Vorname, Name, Lehramt, Schulnummer, Straße, Plz, Ort, Telefon, Fax, E-mail)
- Kursleiter/innen (Anrede, Vorname, Name, Straße, Plz, Ort, Telefon-privat, Telefon-beruflich, Fax-privat, Fax-beruflich, E-Mail)
- Schulen (Schulnummer, Schulname, Schulform, Rechtsstellung, Schulträger, Staatliches Schulamt, Lehrerzahl, Straße, Plz, Ort, Telefon, Fax, Anrede Schulleiter/in, Name Schulleiter/in)
- Kurse (Halbjahr, Bereich, Kursnummer, Kursthema, Kurzthema, Zielgruppe, Termin, Veranstaltungsort)

### **3.3.3 Pflichtenheft**

Aus der allgemeinen Aufgabenstellung und den Anforderungen, die das offline-Programm des HeLP an die Funktionalität der Anwendung stellt, kann ein Pflichtenheft erstellt werden, aus dem hervorgeht, was das System zu leisten hat. In diesem Pflichtenheft wird auch festgelegt, welche Software der Anwendung zugrunde liegen soll. Das Pflichtenheft gliedert sich in die sechs Bereiche Zielgruppen, Produktfunktionen, Produktdaten, Benutzerprofile, Qualitätsanforderungen und technische Anforderungen.

#### **I. Zielgruppen**

- Es soll ein Informationssystem zur Planung von Weiterbildungskursen von Lehrern erstellt werden.
- Zielgruppe sind Lehrer, Referenten und Administratoren.

**II. Produktfunktionen**

- Erfassung, Änderung und Löschen von Daten (Schule, Kurse, Kursort, Lehrer, Referent, Kompetenz, Kursbelegung).
- Es müssen diverse Listen (Kurse, Lehrer) zur Verfügung gestellt werden.
- Benachrichtigungen per E-Mail müssen ermöglicht werden.
- Zugriff muss beschränkt werden können.

**III. Produktdaten**

- Alle relevanten Daten müssen gespeichert werden.

**IV. Benutzerprofile**

- Nicht angemeldete Benutzer erhalten lediglich Informationen über angebotene Kurse.
- Teilnehmer erhalten Informationen über angebotene Kurse und können Kurse belegen.
- Referenten erhalten Informationen über angebotene Kurse und Listen von Teilnehmern.
- Administratoren dürfen alle Daten manipulieren.

**V. Qualitätsanforderungen**

- Nur angemeldete Personen dürfen Daten ändern.
- Je nach Status stehen unterschiedliche Optionen zur Verfügung.
- Die Daten werden durch Standardroutinen vor unbefugtem Zugriff geschützt.

**VI. Technische Anforderungen**

- Dem System liegt eine internetbasierte Client/Server-Architektur zugrunde.
- Zugriff der Clients erfolgt über Standardbrowser.
- Alle Daten sind mit einem relationalen Datenbankmanagementsystem auf einem Server zu verwalten.
- Die Anwendung besteht aus HTML-Dokumenten mit integrierten PHP-Skripten.

**3.3.4 UML und use cases**

Mit der „Unified Modeling Language“ (UML) haben sich in den letzten Jahren auf dem Gebiet der Softwareentwicklung Modellierungstechniken durchgesetzt, die sich durch Anschaulichkeit und Beschreibungsmächtigkeit auszeichnen. Diese Modellierungstechniken eignen sich auch sehr gut für den Einsatz im schulischen Kontext. So heißt es bei

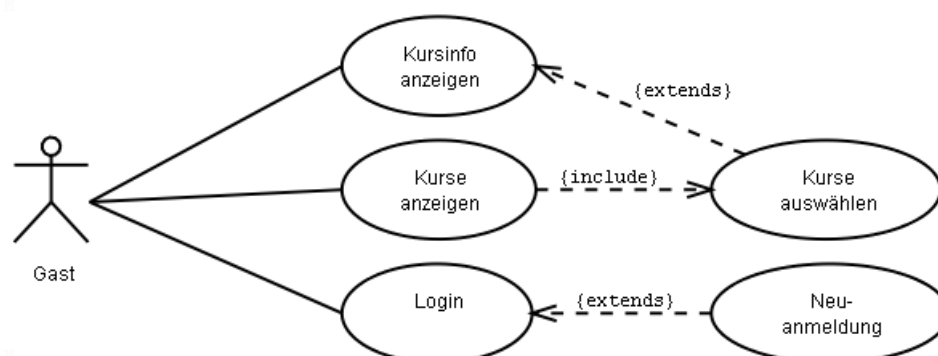
HUBWIESER (2001), dass einige dieser relativ neuen Techniken eine durchaus altergemäße Modellierung einfacher Sachverhalte ermöglichen. Das primäre Ziel des Einsatzes von UML sollte aber nicht das Erstellen optisch perfekter Diagramme sein, sondern UML sollte als Hilfsmittel zur schnellen Visualisierung von Ideen und Gedanken im Zuge der Modellierung verwendet werden. „It is not so helpful to learn syntactically correct UML diagramming and perhaps a UML CASE tool, but then not be able to create an excellent design, or evaluate and improve an existing one“ (LARMAN, 2002).

Anwendungsfalldiagramme stellen eine Möglichkeit dar, um Funktionen zu beschreiben, die ein Anwendungssystem aus Sicht der Benutzer bereitstellen muss. Dies schafft eine Ausgangsbasis für den Entwurf eines Systems (ROFF, 2003). Anwendungsfalldiagramme bestehen aus Akteuren, den Anwendungsfällen und den Beziehungen zwischen beiden. Ein Akteur repräsentiert etwas, das ein System nutzt. Dabei kann ein Akteur eine Person oder ein Teil des Systems sein. Anwendungsfälle sind die Aktionen, die ein Akteur mit einem System ausführt. Die Beziehungen verbinden die Akteure mit den Anwendungsfällen. Nah verwandte Anwendungsfälle können in einer „include“- oder „extend“-Beziehung stehen. Benutzt ein Anwendungsfall die Funktionalität eines anderen Anwendungsfalles, so kann dies durch die „include“-Beziehung verdeutlicht werden. Die „extend“-Beziehung wird benutzt um anzuzeigen, dass ein Anwendungsfall optional durch einen weiteren Anwendungsfall erweitert werden kann (ROFF, 2003).

Jeder Anwendungsfall (use case) formuliert eine Teilfunktionalität, die für mindestens einen Akteur ein bestimmtes Ergebnis erbringt. Gäste, Teilnehmer, Referenten und Administratoren können auf das Kursverwaltungssystem zugreifen und sind somit Akteure des Anwendungssystems. Jede der Personengruppen stellt unterschiedliche Anforderungen an das System.

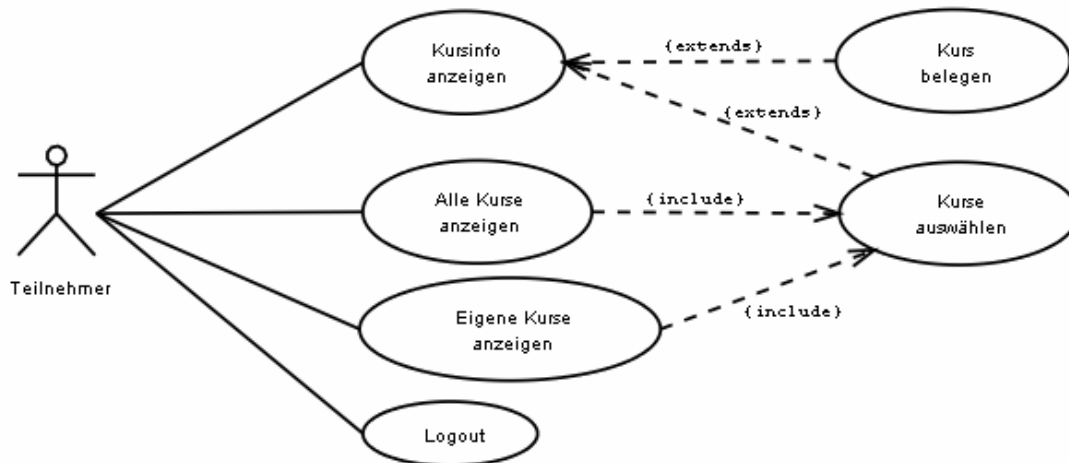
### Gäste können

- Informationen über bestehende Kurse abrufen,
- sich am System neu anmelden.

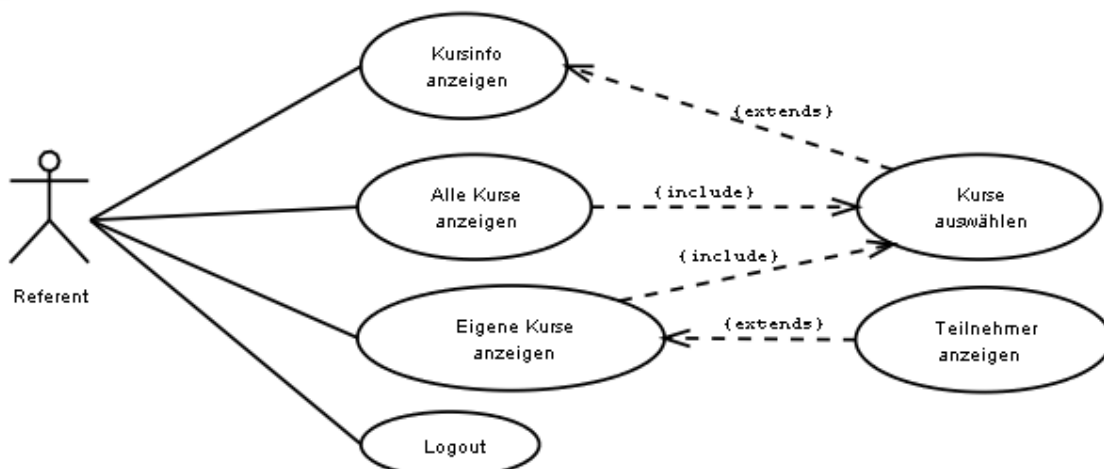


**Teilnehmer können**

- Informationen über bestehende Kurse abrufen,
- sich in bestehende Kurse einwählen,
- alle Kurse abrufen, in die sie sich eingewählt haben.

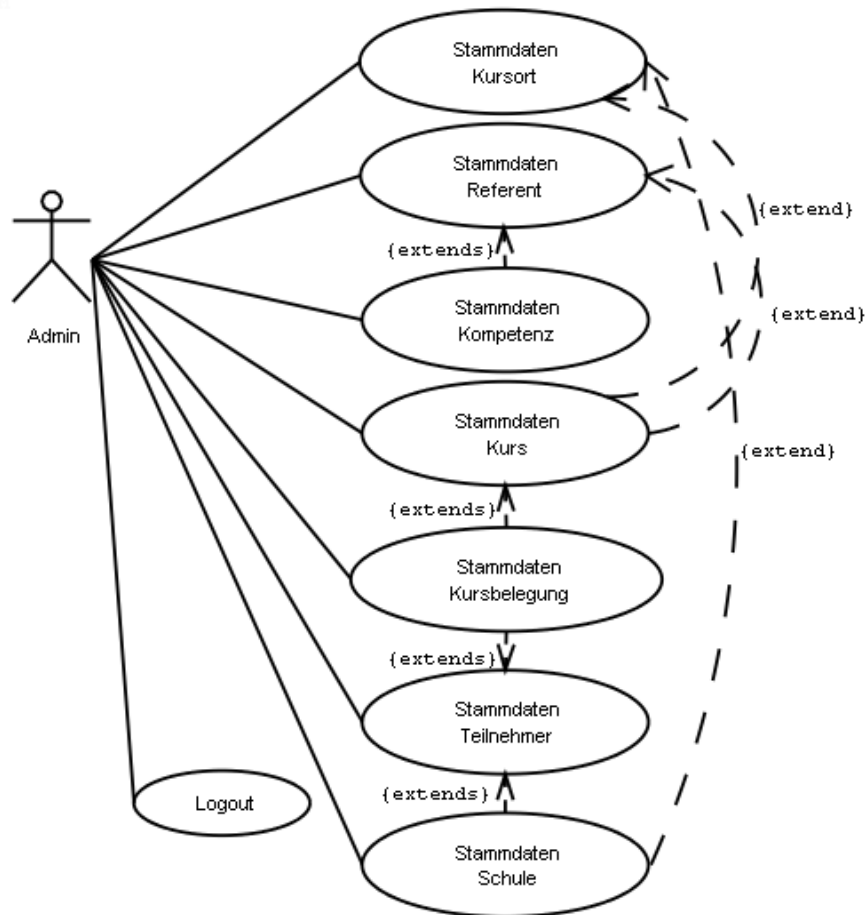
**Referenten können**

- Informationen über bestehende Kurse abrufen,
- alle Kurse abrufen, die sie leiten,
- die Anzahl der Teilnehmer ihrer Kurse abrufen,
- Kurse mit ausreichender Teilnehmerzahl beginnen,
- die Teilnehmer ihrer Kurse informieren.



### Administratoren können

- Stammdaten der Kurse, Kursorte, Teilnehmer, Schulen, Kompetenzen und Referenten manipulieren,
- Kursbelegungen bearbeiten,
- den Status von Kursen ändern.



Da das System Personengruppen mit unterschiedlichen Rechten verwaltet, muss eine Zugangskontrolle eingerichtet werden, die dafür sorgt, dass am System angemeldete Personen nur die jeweils erlaubten Aktionen ausführen können.

## 4 Teil 2: Design

Die Wichtigkeit des Modellierungsvorgangs wird von Seiten der Informatikdidaktiker immer wieder betont. So wird dieses Thema zum Beispiel in den GI-Empfehlungen von 1993 (Schulz-Zander et al. 1993 in Hubwieser, 2001) als erster Lerninhalt aufgeführt. Dort heißt es: „Die Schülerinnen und Schüler sollen: A1 Exemplarische Methoden und Verfahren der Modellierung eines Ausschnittes der Wirklichkeit kennen lernen, diese anwenden und kritisch hinterfragen, z.B. im Hinblick auf die Grenzen dieser Methoden und auf das Interesse am Einsatz.“

### 4.1 Modellierung der Datenbank

Das Thema Modellierung von Datenbanken ist auch ein wesentlicher Themenschwerpunkt des hessischen Lehrplans (HESSISCHES KULTUSMINISTERIUM, 2003). Was genau ist mit dem Begriff „Modell“ gemeint? BAUMANN (1996) schreibt: „Modell ist eine konkrete, wegen idealisierender Reduktion auf relevante Züge fasslichere oder leichter realisierbare, Darstellung unübersichtlicher oder abstrakter Gegenstände“. Er fasst seine Aussage zusammen: „Modell = vereinfachte struktur- und verhaltenstreue Beschreibung eines realen Systems“. Ein solches Modell ist der erste Schritt zur Erstellung eines komplexen Informatiksystems.

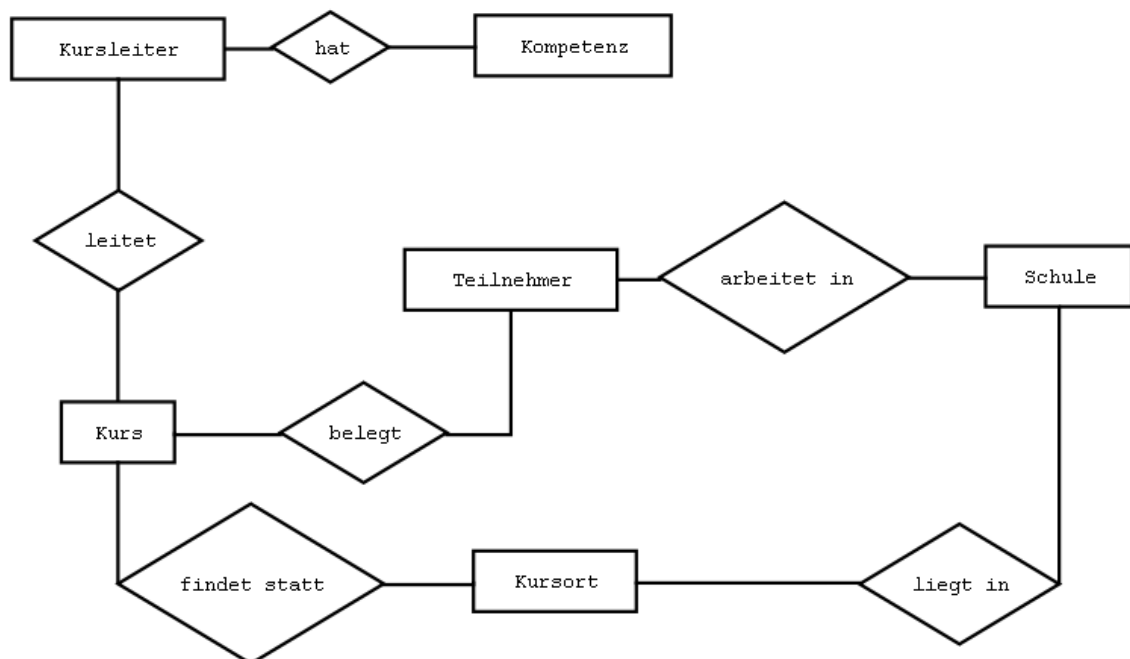
#### 4.1.1 Entity-Relationship-Modell

Ein Datenbankmodell ist ein System von Konzepten zur Beschreibung von Datenbanken. „Es legt somit Syntax und Semantik von Datenbankbeschreibungen für ein Datenbanksystem fest, den sogenannten Datenbankschemata“ (HEUER und SAAKE, 2000). Das Entity-Relationship-Modell (ER-Modell, ERM) stellt ein gutes Hilfsmittel dar, um Datenbanken zu modellieren. Es wurde erstmals von P. P. Chen im Jahr 1976 vorgeschlagen und eignet sich hervorragend als Standardmodell für frühe Entwurfsphasen der Datenbankentwicklung. Das ER-Modell benutzt die drei Grundkonzepte Entity, Relationship und Attribut, die folgendermaßen charakterisiert sind.

- **Entity** ist ein Objekt der realen oder der Vorstellungswelt (z.B. Teilnehmer, Kursleiter, Kursort, Schule), wobei auch Informationen über Ereignisse oder Fähigkeiten (z.B. Kurs, Kompetenz) Objekte im Sinne des ER-Modells sein können.
- **Relationship** beschreibt eine Beziehung zwischen Entities.
- **Attribut** stellt eine Eigenschaft eines Entities oder einer Beziehung dar.



Zunächst werden die Entities erfasst und die Beziehungen zwischen ihnen ermittelt. Im vorliegenden Beispiel können die tatsächlich existierenden Objekte der realen Welt als Entity aufgefasst werden (Kursleiter, Teilnehmer, Schule, Kursort). Auf der Suche nach weiteren Entity-Kandidaten werden noch Kurse und Kompetenzen ausgewählt. Zwischen diesen Entities existieren Beziehungen, die als Verb formuliert in die Rauten zwischen den Entities geschrieben werden.



25

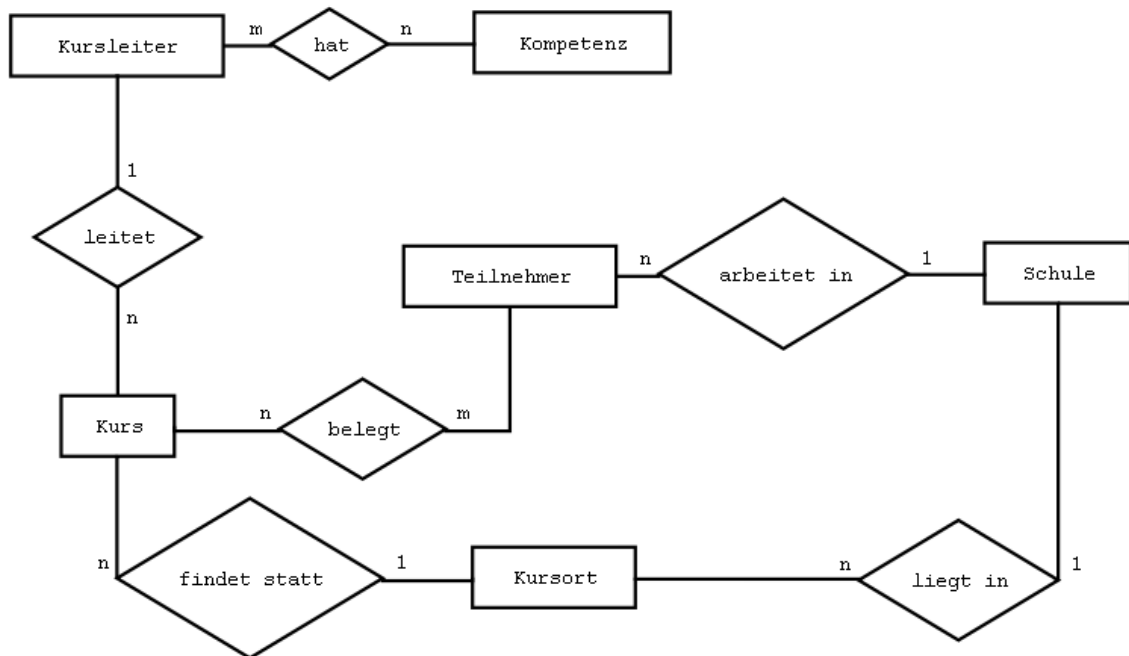
beteiligten Entities an einer Beziehung geklärt wird. **Kann** eine Entity eine **Mehrfachbeziehung** zu einer anderen Entity haben („KaMe“-Frage)? Die Frage wird für jede Entity, die an einer Beziehung beteiligt ist, gestellt (RÖHNER, 2003).

Beispiel: Kann ein Kursleiter mehrere Kurse leiten? (ja)

Kann ein Kurs von mehreren Kursleitern geleitet werden? (nein)

Wird eine Frage mit „ja“ beantwortet, wird an das Ende der Beziehung ein „n“ bzw. ein „m“ notiert. Sollte die Frage mit „nein“ beantwortet werden, so wird an das Ende der Beziehung eine 1 geschrieben.

Da auf die Frage „Kann ein Kursleiter mehrere Kurse leiten?“ mit „ja“ geantwortet wurde, muss an das Ende der Beziehung (Kurs-Seite) ein „n“ notiert werden. Umgekehrt wurde auf die Frage „Kann ein Kurs von mehreren Kursleitern geleitet werden?“ mit „nein“ geantwortet. Aus diesem Grund wird jetzt am Ende der Beziehung (Kursleiter-Seite) eine „1“ notiert. Dieses Verfahren wird nun analog auf alle Entities und Relationships angewendet. Damit entsteht das folgende Diagramm:



Im letzten Schritt wird die Frage untersucht, ob eine Instanz einer Entity zwangsläufig an einer Beziehung zu einer anderen Entity teilnehmen muss und diese Beziehung fakultativ ist oder optional sein kann. Auch in diesem Fall ist die Formulierung einer Frage hilfreich. **Muss**

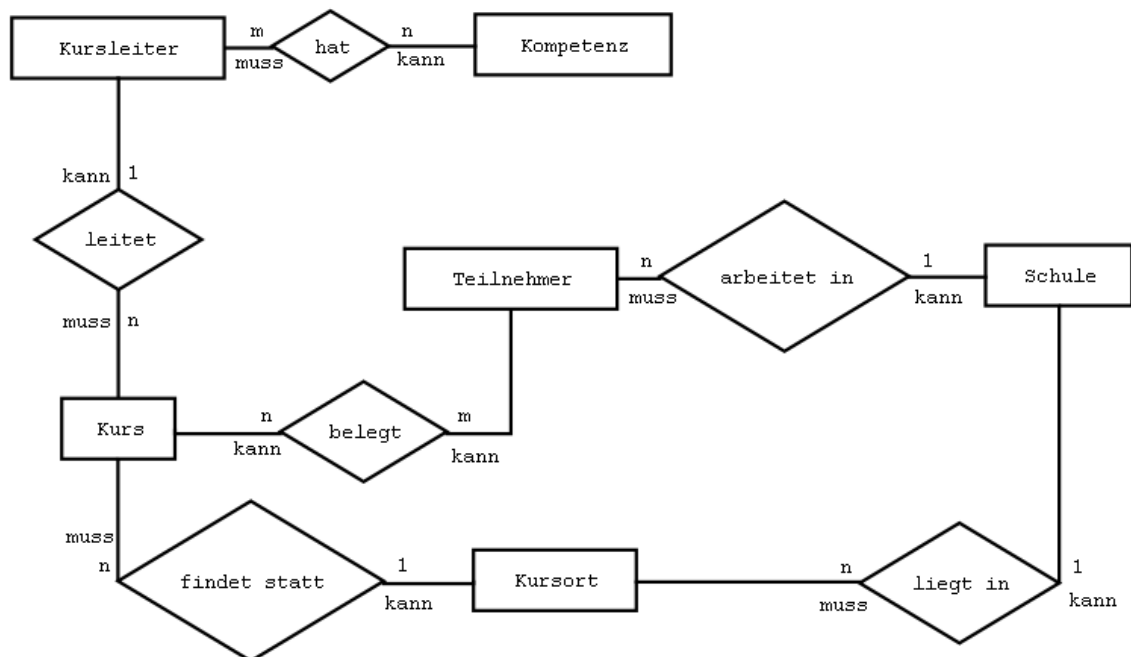
eine Entity **mindestens** mit einer anderen Entity in Beziehung stehen („MuMi“-Frage)? Auch diese Frage wird für jede Entity gestellt (RÖHNER, 2003).

Beispiel: Muss ein Kursleiter mindestens einen Kurs leiten? (nein)

Muss ein Kurs von mindestens einem Kursleiter geleitet werden? (ja)

Wird eine Frage mit „ja“ beantwortet, wird an den Anfang der Beziehung ein „muss“ notiert. Sollte die Frage mit „nein“ beantwortet werden, so wird an den Anfang der Beziehung ein „kann“ geschrieben.

Da auf die Frage „Muss ein Kursleiter mindestens einen Kurs leiten?“ mit „nein“ geantwortet wurde, muss an den Anfang der Beziehung (Kursleiter-Seite) ein „kann“ notiert werden. Umgekehrt wurde auf die Frage „Muss ein Kurs von mindestens einem Kursleiter geleitet werden?“ mit „ja“ geantwortet. Aus diesem Grund wird jetzt am Anfang der Beziehung (Kurs-Seite) eine „muss“ notiert. Dieses Verfahren wird nun analog auf alle Entities und Relationships angewendet. Damit entsteht das folgende Diagramm:

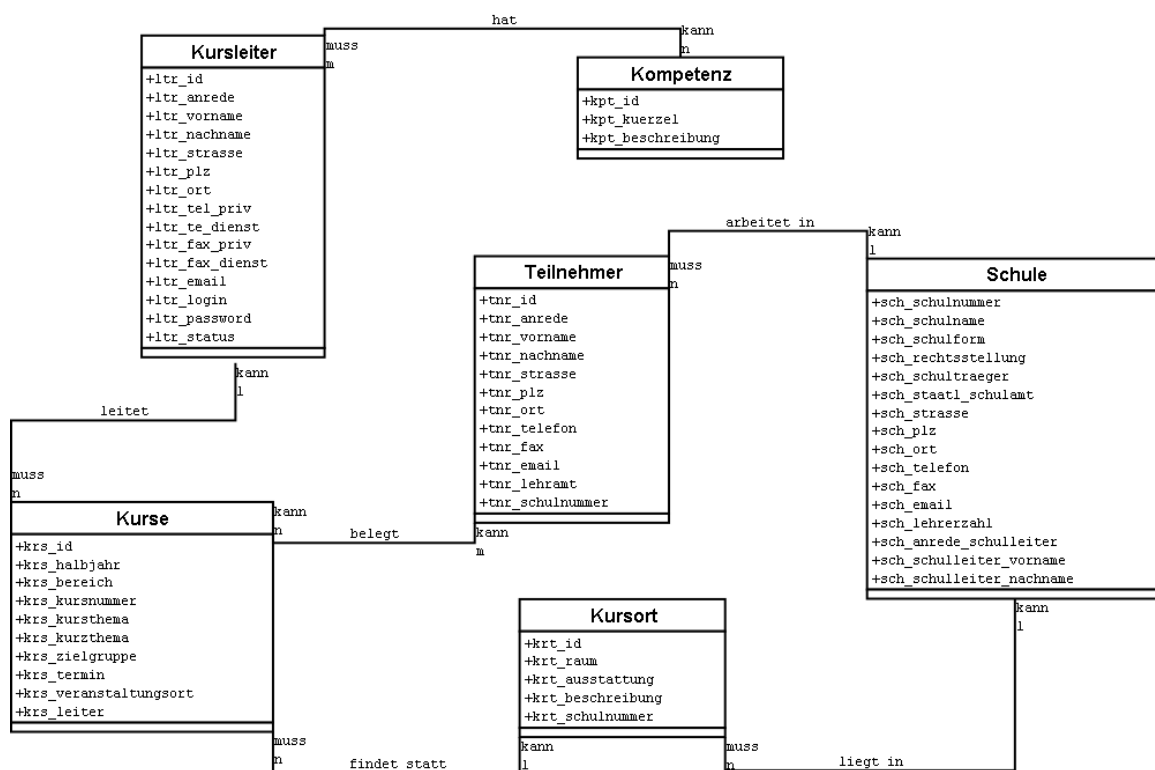


Zur Überprüfung kann nun aus den Namen der Entities, den Angaben zu Kardinalität und Optionalität und dem Verb, das die Art der Beziehung ausdrückt für jede Beziehung zwischen den Entities ein vollständiger Satz gebildet werden:

<b>Unbest. Artikel</b>	<b>Entity 1</b>	<b>Optionalität</b>	<b>Kardinalität</b>	<b>Entity 2</b>	<b>Verb</b>
Ein	Kursleiter	kann	mehrere (n)	Kurse	leiten
Ein	Kurs	muss	von einem (1)	Kursleiter	geleitet werden

### 4.1.2 Datenmodell

Die Darstellung aller Attribute im ER-Modell kann dazu führen, dass die Übersichtlichkeit eingeschränkt wird. Eine alternative und übersichtliche Möglichkeit zur Darstellung der Entities, Relationships und der beteiligten Attribute liefert das sogenannte Datenmodell in UML-konformer Notation. Hierbei werden die Entities ebenfalls in Form von Rechtecken dargestellt. Die Relationships werden nur als Verbindungslinien zwischen den Entities gezeichnet, wobei die Angaben zur Kardinalität und Optionalität analog an den Anfang und das Ende der jeweiligen Beziehung notiert werden. Die Attribute werden unterhalb der Entities aufgelistet. Hier können auch weitere Spezifikationen der Attribute beispielsweise bezüglich des Wertebereichs, des Datentyps oder der Sichtbarkeit angegeben werden. Die Attribute der Entities ergeben sich aus der Anforderungsspezifikation, die zu Beginn der Softwareentwicklung erstellt wurde.



Es ist möglich, die einzelnen Instanzen eines Entity-Typs durch die Kombination oft mehrerer verschiedener Attribute zu identifizieren, die dann Schlüsselkandidat oder Schlüssel genannt werden. Existieren mehrere solcher Schlüsselkandidaten, wählt man einen geeigneten aus und bezeichnet diesen als Primärschlüssel. Eine weitere Möglichkeit ist es, trotz der Existenz von Schlüsselkandidaten, ein Attribut „ID“ zur Identifikation der Instanzen eines Entity-Typs einzuführen, das dann als Primärschlüssel dient. Die Definition eines künstlichen, nicht

zusammengesetzten Primärschlüssels erleichtert die spätere Programmierarbeit. Im vorliegenden Beispiel wurde so verfahren.

### 4.1.3 Relationales Modell

Die Daten liegen bisher in einem abstrakten, datenbankunabhängigen Modell (ER-Modell) vor. MySQL ist ein Datenbankmanagementsystem, das auf dem relationalen Modell beruht. Weitere Beispiele für Datenbankmodelle sind das hierarchische Modell, das Netzwerkmodell und das objektorientierte Modell. Daher muss das ER-Modell in ein relationales Modell übersetzt werden. Dieses Modell basiert auf dem mathematischen Begriff der Relation  $R$ , die laut Definition eine Teilmenge des kartesischen Produkts einer Liste von Wertbereichen  $W_1, W_2, \dots, W_n$  ist.

Die Übersetzung des ER-Modells in das relationale Modell verläuft in mehreren Schritten. Hierbei wird zunächst der Entity-Typ zum Namen der Relation und die jeweils zugehörigen Attribute werden übernommen. Dazu wird der Name des Entity-Typs notiert und die Namen der Attribute durch Kommata getrennt in Klammern hinter den Entity-Typ geschrieben. Der Primärschlüssel wird unterstrichen.

```
kurs (krs_id, krs_halbjahr, krs_bereich, krs_kursnummer, krs_kursthema,
      krs_kurzthema, krs_zielgruppe, krs_termin)
```

Die Beziehungen werden in eigenen Relationenschemata abgebildet. Hierbei sind die Primärschlüssel der beteiligten Entities die Attribute. Gleichzeitig stellt die Kombination der Attribute einen geeigneten Schlüsselkandidaten dar. Im vorliegenden Beispiel wurde auch bei den Beziehungsrelationen ein künstlicher Primärschlüssel gewählt.

```
teilnehmer_kurs (tk_id, tnr_id, krs_id)
```

Die Anzahl der Relationen im relationalen Modell ergibt sich aus der Anzahl der Entities und der Anzahl der Beziehungen zwischen den Entities. Das relationale Modell kann jedoch bezüglich der Anzahl der Relationen optimiert werden. Dies betrifft beispielsweise alle Beziehungen mit der Kardinalität 1:n mit Optionalität "muss" auf der n-Seite. In diesem Fall kann der Primärschlüssel der 1-Seite als Fremdschlüssel in die Relation der n-Seite aufgenommen werden. Zur Verdeutlichung wird der Fremdschlüssel markiert (kursiv, fett)

```
kurs (krs_id, krs_halbjahr, krs_bereich, krs_kursnummer, krs_kursthema,
      krs_kurzthema, krs_zielgruppe, krs_termin, krs_veranstaltungsort,
      krs_leiter)
```

#### 4.1.4 Relationales Modell mit Datentypen

Die Darstellung des relationalen Modells kann jetzt um die Datentypen und eventuelle Besonderheiten des später verwendeten Datenbankmanagementsystems ergänzt werden. Im vorliegenden Fall wurden die von MySQL unterstützten Datentypen „VARCHAR“, „MEDIUMTEXT“ und „BIGINT“ verwendet.

**BIGINT:** vorzeichenlose Werte von 0 bis  $2^{64} - 1$  und  
 vorzeichenbehaftete Werte von  $-2^{63} - 1$  bis  $+2^{63} - 1$


**VARCHAR:** Zeichenkette variabler Länge (1 - 255 Zeichen)

**MEDIUMTEXT:** Beliebiger Text ( $2^{24} - 1$  Zeichen)

Zusätzlich wurde vermerkt, dass es sich beim Attribut „krs\_id“ um den Primärschlüssel handelt, der vom System automatisch verwaltet und beim Einfügen neuer Datensätze inkrementiert werden soll.

Die Attribute „krs\_veranstaltungsort“ und „krs\_leiter“ sind Fremdschlüssel, wobei das Attribut „krs\_veranstaltungsort“ auf den Primärschlüssel „krt\_id“ der Relation „Kursort“ und das Attribut „krs\_leiter“ auf den Primärschlüssel „ltr\_id“ der Relation Kursleiter verweist.

Beispiel: Relation Kurs

Key	Feldname	Datentyp	Besonderheit
	krs_id	BIGINT(21)	NOT NULL, AUTOINCREMENT
	krs_halbjahr	VARCHAR(50)	
	krs_bereich	VARCHAR(50)	
	krs_kursnummer	BIGINT(21)	
	krs_kurzthema	VARCHAR(100)	
	krs_kursthema	MEDIUMTEXT	
	krs_zielgruppe	VARCHAR(50)	
	krs_termin	VARCHAR(50)	
	krs_veranstaltungsort	BIGINT(21)	-> Kursort{krt_id}
	krs_leiter	BIGINT(21)	-> Kursleiter{ltr_id}
	krs_status	VARCHAR(10)	

### 4.1.5 Vom Modell zur Datenbank

Im Folgenden wird exemplarisch gezeigt, wie aus einem relationalen Modell eine Datenbanktabelle erzeugt werden kann. Detailliertere Erläuterungen zum Sprachumfang von SQL folgen im Kapitel 4.3.1.

Zur Erzeugung von Tabellen aus dem relationalen Modell dient eine einfache Anweisung aus dem Datendefinitionsteil (DDL - Data Definition Language) von SQL (Structured Query Language). Die Definition einer Tabelle geschieht mittels der „CREATE TABLE“-Anweisung. Die Syntax entspricht in ihrer einfachsten Form:

```
CREATE TABLE Tabellename (
    Spaltenname1, Wertebereich1,
    Spaltenname2, Wertebereich2,
    ...
    Spaltennamen, Wertebereichn)
```

In SQL kann durch die Angabe „PRIMARY KEY“ eine bestimmte Spalte als Schlüsselattribut gekennzeichnet werden. Der Primärschlüssel darf keine Null-Werte annehmen (NOT NULL). Wurde eine Spalte mit „AUTO\_INCREMENT“ markiert, so erzeugt MySQL automatisch die nächste Folgenummer bei Einfügen eines neuen Datensatzes. Es ist außerdem möglich „Default-Werte“ anzugeben (default NULL).

#### Beispiel: Relation Kurs

```
CREATE TABLE kurs (
    krs_id bigint(21) NOT NULL AUTO_INCREMENT,
    krs_halbjahr varchar(50) default NULL,
    krs_bereich varchar(50) default NULL,
    krs_kursnummer bigint(21) default NULL,
    krs_kurzthema varchar(100) default NULL,
    krs_kursthema mediumtext,
    krs_zielgruppe varchar(50) default NULL,
    krs_termin varchar(50) default NULL,
    krs_ort bigint(21) default NULL,
    krs_leiter bigint(21) default NULL,
    krs_status varchar(10) default NULL,
    PRIMARY KEY (krs_id))
```

Mit der oben stehenden Anweisung kann eine Tabelle in einer MySQL-Datenbank definiert werden. Die Syntax ist MySQL-spezifisch und muss gegebenenfalls bei Verwendung einer anderen relationalen Datenbank angepasst werden.

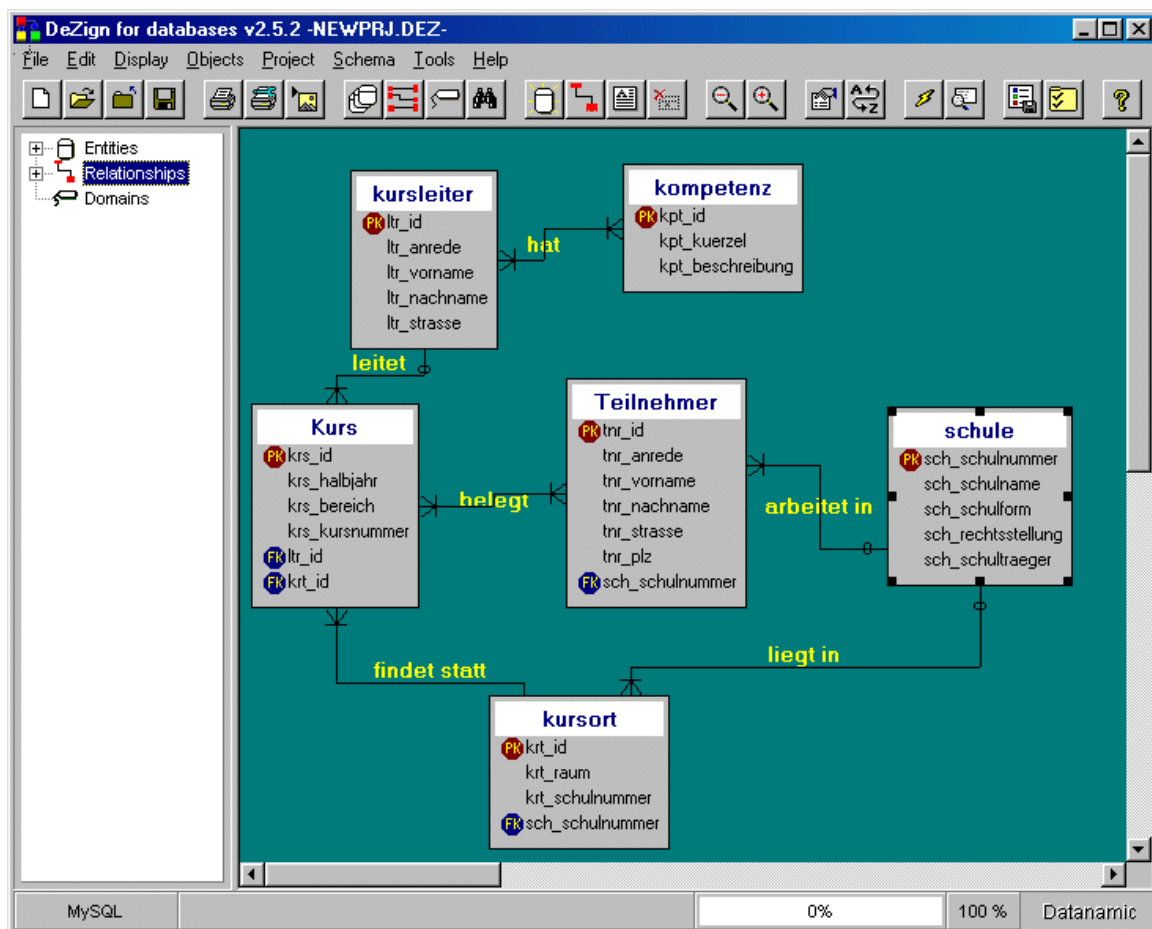


## 4.2 Case-Tools

Auch zur Modellierung von Datenbanken gibt es Hilfsmittel in Form von sogenannten CASE-Tools (**C**omputer **A**ided **S**oftware **E**ngineering). Zusätzlich zur Visualisierung von Entities und den entsprechenden Beziehungen erzeugen diese Programme SQL-Anweisungen zur Erstellung der jeweiligen Relationen. Auf die Besonderheiten der zugrundeliegenden Datenbank kann in entsprechenden Konfigurationsoptionen Einfluss genommen werden. Für die fast ausnahmslos kommerziellen (und teuren) Produkte gibt es in Einzelfällen Evaluationsversionen, die mit Einschränkungen kostenlos genutzt werden können.

### 4.2.1 DeZign for Databases

DeZign for Databases ist ein Datenbankentwicklungswerkzeug, das bei der Visualisierung das Entity-Relationship-Modell verwendet. Aus dem ER-Modell generiert das Programm fertigen SQL-Code, der an die Syntax verschiedener Datenbankmanagementsysteme angepasst werden kann (URL: siehe Kapitel 8).



Dieses Programm erzeugt nach erfolgreicher Modellierung der Datenbank auf Knopfdruck SQL-Anweisungen zum Erzeugen einer Datenbank unter MySQL. Die Evaluationsversion hat allerdings den Nachteil, dass nur 5 Attribute pro Relation möglich sind.

#### **4.2.2 DBDesigner4**

DBDesigner 4 ist ein grafisches Entwicklungstool, das Datenbankdesign, Modellierung von Datenbanken und die Erstellung von Datenbanken und Tabellen automatisiert. Es handelt sich bei dieser Software um ein open source-Projekt, das für die Betriebssysteme Windows und Linux zur Verfügung steht. DBDesigner 4 wurde entwickelt und optimiert für die Verwendung von MySQL. Diese Software ist kostenlos erhältlich (URL: siehe Kapitel 8).

#### **4.2.3 ERWin**

Bei der Software „AllFusion ERwin Data Modeler“ handelt es sich um ein Entwicklungswerkzeug, das in der Lage ist, Datenbanken und Data Warehouse-Systeme zu erzeugen und zu verwalten (URL: siehe Kapitel 8).

### **4.3 Das relationale Datenbanksystem MySQL**

Eine Datenbank (DB) ist eine Ansammlung von Daten, die entsprechend den natürlichen Zusammenhängen in der Alltagswelt strukturiert ist, und die allen Anwendern eines Anwendungsbereiches als gemeinsame Basis aktueller Informationen zur Verfügung steht (STOLL und LEIERER, 2000). Das Datenbankmanagementsystem (DBMS) ist ein Softwaresystem, das den Zugriff auf die Daten einer Datenbank ermöglicht und so die Funktionalität eine Datenbank zu definieren, Daten zu speichern, zu löschen und zu manipulieren sowie Anfragen an die Datenbank zu senden zur Verfügung stellt. Datenbank und Datenbankmanagementsystem ergeben zusammen ein Datenbanksystem (DBS).

#### **4.3.1 SQL**

Die **Structured Query Language (SQL)** ist die meistgenutzte gewerbliche Sprache für relationale Datenbanken. Ursprünglich wurde SQL in den SEQUEL-XRM und System-R Projekten bei IBM entwickelt (1974-1977). Beinahe gleichzeitig verwendeten auch andere Hersteller SQL bei der Einführung ihrer auf relationalen Datenbanken basierenden Systeme

und so entwickelte sich SQL zu einem de facto Standard (RAMAKRISHNAN und GEHRKE, 2003). Der zurzeit gültige Standard für SQL nennt sich SQL:1999, der jedoch noch nicht von allen Herstellern komplett unterstützt wird. Der zurzeit noch nicht verabschiedete nächste gültige Standard ist unter dem Kürzel SQL:2003 bekannt.

Das Datenbankmanagementsystem MySQL hat nicht alle Sprachelemente von SQL:1999 implementiert. MySQL lehnt sich weitgehend an den SQL Standard SQL:1992 an. Es fehlen verschachtelte SELECT-Anweisungen, Foreign Keys, Views, Transaktionen, gespeicherte Prozeduren und Trigger. Dagegen verfügt MySQL über einige Zusatzfunktionen, die nicht im Standard SQL enthalten sind. Dies betrifft hauptsächlich die Verwendung von String-Operationen und regulären Ausdrücken (DUBOIS, 2000).

Die primäre Funktion der Sprache SQL besteht in der Unterstützung der Definition, Manipulation und Kontrolle von Daten in SQL-Datenbanken (MATTES, 2001). Die Sprache SQL besteht aus mehreren Teilen, die jeweils unterschiedliche Funktionalitäten zur Verfügung stellen. Beispielhaft seien hier die Data Definition Language (DDL), die Data Control Language (DCL) und die Data Manipulation Language (DML) aufgeführt.

#### 4.3.1.1 Data Definition Language

Diese Untermenge von SQL unterstützt das Erstellen, Löschen und Verändern von Definitionen für Datenbanken, Tabellen und Sichten. Datenbank Anwender haben mit diesen SQL-Anweisungen im Normalfall wenig zu tun. Sie dienen hauptsächlich den Systemadministratoren dazu Datenbanken, Tabellen, Sichten, etc. zu erstellen. Mit den Anweisungen aus der DDL muss im genannten Beispiel zunächst eine Datenbank erstellt und ein aussagekräftiger Name gewählt werden (kursverwaltung). Anschließend werden die Tabellen mittels der aus dem relationalen Modell übersetzten SQL-Anweisungen erzeugt.

```
CREATE          Datenbankname;
CREATE TABLE  Tabellename
               (Spaltenname1, Wertebereich1,
               Spaltenname2, Wertebereich2,
               ...
               Spaltennamen, Wertebereichn)

CREATE          kursverwaltung
CREATE TABLE  kompetenz
               (kpt_id bigint(21) NOT NULL auto_increment,
               kpt_kuerzel varchar(255) default NULL,
               kpt_beschreibung mediumtext,
               PRIMARY KEY (kpt_id))
```

#### 4.3.1.2 Data Control Language

Die DCL ist eine Untermenge aus dem Sprachumfang von SQL. Sie unterstützt Mechanismen um den Benutzerzugriff auf Tabellen und Sichten zu beschränken. Der Systemadministrator kann den Zugriff auf Daten mit den Sprachkonstrukten der DCL kontrollieren. Dabei kann der Zugriffsschutz auf die gesamte Datenbank oder auf Teilbereiche erfolgen.

```
GRANT SELECT, INSERT, DELETE, UPDATE ON datenbankname.* TO user@%
IDENTIFIED BY 'passwort';
```

```
GRANT SELECT, INSERT, DELETE, UPDATE ON kursverwaltung.* TO alex@%
IDENTIFIED BY 'geheim';
```

#### 4.3.1.3 Data Manipulation Language

Die Verfügbarkeit, bzw. der Zugang zu den eigentlichen Daten wird durch die DML bereitgestellt. Eine ihrer Hauptaufgaben ist die Extraktion von Daten. Die DML benötigt man aber auch zum Einfügen, Löschen und Aktualisieren der Datensätze.

##### Einfügen von Datensätzen

```
INSERT INTO Tabellename(Spaltenname1, Spaltenname2) VALUES (Wertebereich1,
Wertebereich2);
```

```
INSERT INTO kursleiter(vorname, nachname) VALUES('Alexander', 'Hauptmann');
```

##### Auswählen von Datensätzen

```
SELECT Spaltenname1, Spaltenname2, FROM Tabellename WHERE Spaltenname1=
Wertebereich1 ORDER BY Spaltenname2;
```

```
SELECT vorname, nachname FROM kursleiter WHERE vorname='Alexander' ORDER BY
nachname;
```

##### Ändern von Datensätzen

```
UPDATE Tabellename SET Spaltenname1=Wertebereich1 WHERE Spaltenname2=
Wertebereich2;
```

```
UPDATE kursleiter SET vorname ='Ulla' WHERE nachname ='Hauptmann';
```

##### Löschen von Datensätzen

```
DELETE FROM Tabellename WHERE Spaltenname1= Wertebereich1;
```

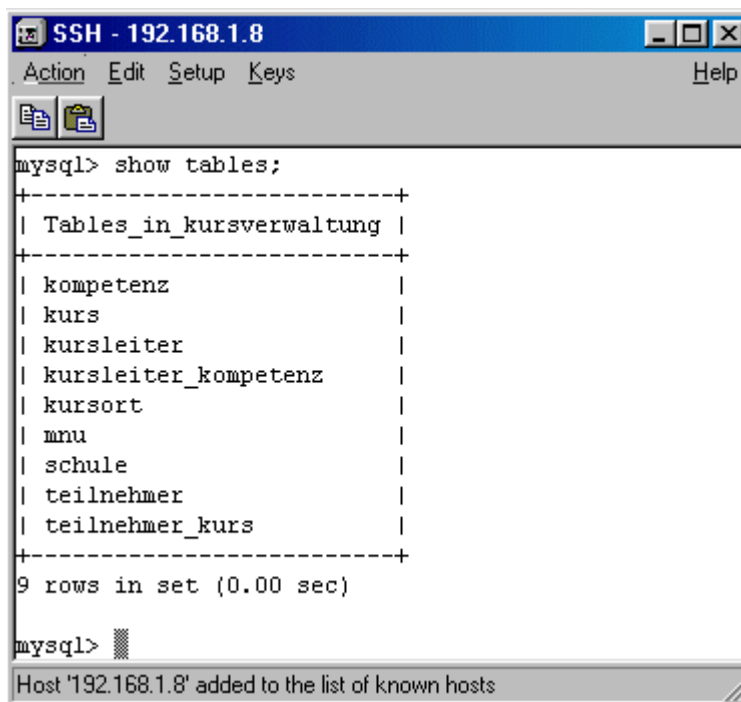
```
DELETE FROM kursleiter WHERE nachname ='Hauptmann';
```

### 4.3.2 DBMS

Das Datenbankmanagementsystem von MySQL ist als Konsolenanwendung unter Linux verfügbar. Es gibt allerdings auch eine Reihe von Tools, mit denen man MySQL auch mit einer grafischen Benutzeroberfläche verwalten kann. Mit dem Befehl „mysql Datenbankname“ wird das DBMS von MySQL gestartet. Mit den Befehlen der DDL, DCL und DML kann die Datenbank angelegt und manipuliert werden.

Die folgenden Beispiele zeigen die Auswirkungen einiger SQL-Befehle auf die Datenbank „kursverwaltung“. Mit dem Befehl „SHOW TABLES“ werden die Tabellen der aktuellen Datenbank aufgelistet. Mit der SELECT-Anweisung werden Daten angezeigt. Die INSERT-Anweisung fügt Daten in Tabellen ein.

Befehl "show tables;"

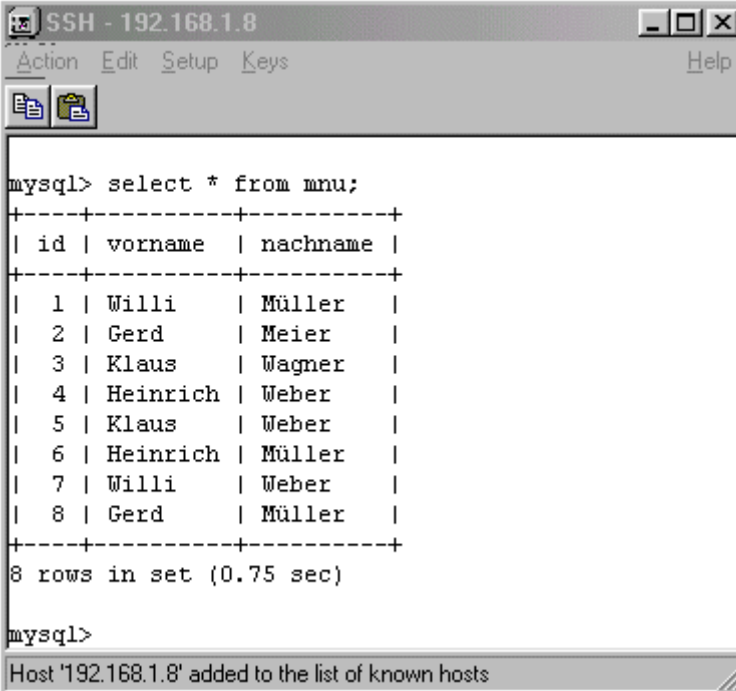


```
SSH - 192.168.1.8
Action Edit Setup Keys Help
mysql> show tables;
+-----+
| Tables_in_kursverwaltung |
+-----+
| kompetenz                |
| kurs                     |
| kursleiter               |
| kursleiter_kompetenz     |
| kursort                  |
| mnu                      |
| schule                   |
| teilnehmer              |
| teilnehmer_kurs         |
+-----+
9 rows in set (0.00 sec)

mysql>
```

Host '192.168.1.8' added to the list of known hosts

Befehl "select \* from mnu;"



SSH - 192.168.1.8

Action Edit Setup Keys Help

```
mysql> select * from mnu;
```

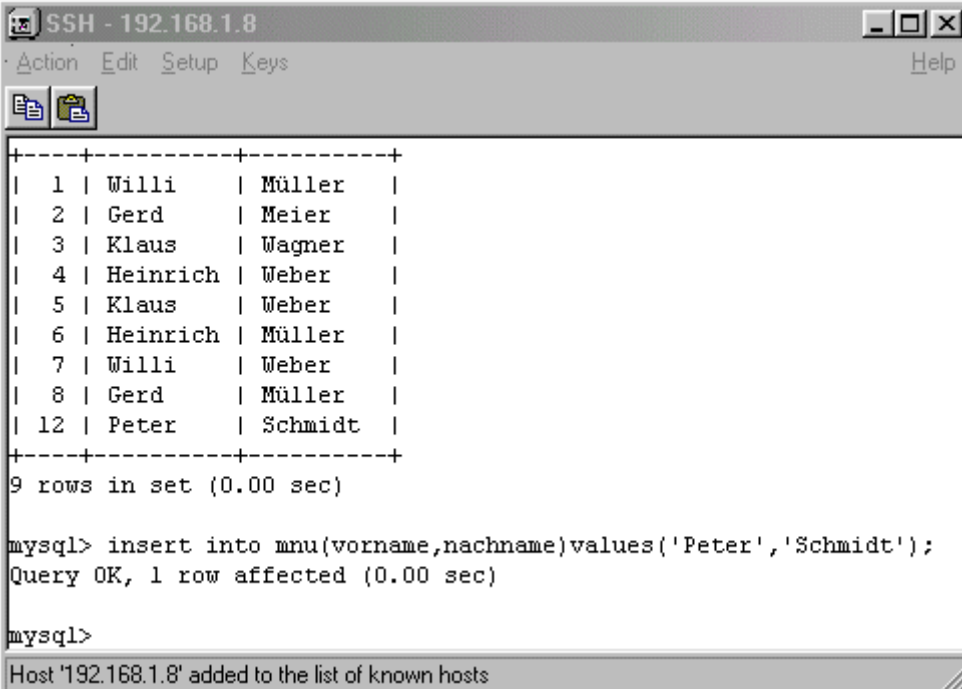
id	vorname	nachname
1	Willi	Müller
2	Gerd	Meier
3	Klaus	Wagner
4	Heinrich	Weber
5	Klaus	Weber
6	Heinrich	Müller
7	Willi	Weber
8	Gerd	Müller

```
8 rows in set (0.75 sec)

mysql>
```

Host '192.168.1.8' added to the list of known hosts

Befehl "insert into mnu(vorname,nachname)values('Peter','Schmidt');"



SSH - 192.168.1.8

Action Edit Setup Keys Help

```
mysql> select * from mnu;
```

id	vorname	nachname
1	Willi	Müller
2	Gerd	Meier
3	Klaus	Wagner
4	Heinrich	Weber
5	Klaus	Weber
6	Heinrich	Müller
7	Willi	Weber
8	Gerd	Müller
12	Peter	Schmidt

```
9 rows in set (0.00 sec)

mysql> insert into mnu(vorname,nachname)values('Peter','Schmidt');
Query OK, 1 row affected (0.00 sec)

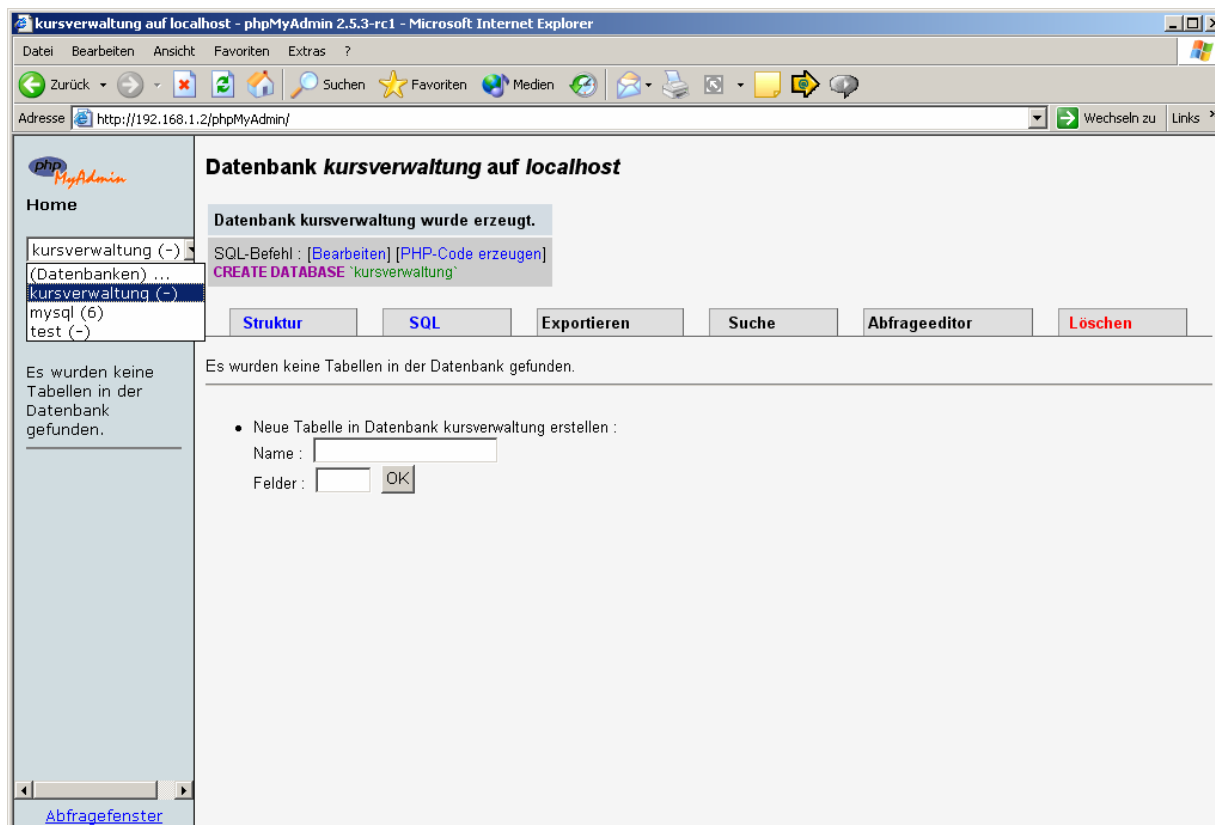
mysql>
```

Host '192.168.1.8' added to the list of known hosts

### 4.3.3 PhpMyAdmin

PhpMyAdmin ist eine auf PHP basierende, kostenfrei erhältliche Scriptsammlung (URL: siehe Kapitel 8), mit der man über einen gewöhnlichen Browser MySQL administrieren kann. Zur Installation kann man einfach die gesamte Scriptsammlung in ein eigenes Verzeichnis kopieren, das über den Webserver erreichbar ist. Konfiguriert wird die Scriptsammlung über das Script „config.inc.php“. Für eine einfache Installation genügt es, den Pfad der Scriptsammlung sowie den Server, den Benutzernamen und das Passwort für den Zugriff auf die Datenbank einzustellen.

Mit PhpMyAdmin können beispielsweise neue Datenbanken erzeugt, Tabellen angelegt, Zugriffsrechte vergeben und Daten manipuliert werden.



## 5 Teil 3: Implementierung

### 5.1 Einführung in HTML

#### 5.1.1 HTML

HTML steht für **HyperText Markup Language**. Es handelt sich dabei um eine Sprache, die mit Hilfe von SGML (**S**tandard **G**eneralized **M**arkup **L**anguage) definiert wird. SGML ist als ISO-Norm 8879 festgeschrieben (MÜNZ und NEFZGER, 1999). HTML ist eine sogenannte Auszeichnungssprache (Markup Language). Als Auszeichnungssprache enthält HTML daher Befehle zum Markieren typischer Elemente eines Dokuments, wie Überschriften, Textabsätze, Listen, Tabellen oder Grafikreferenzen. HTML-Seiten bestehen aus einem Kopf (<head>...</head>) und einem Körper (<body>...</body>). Elemente innerhalb des Körpers bilden den Inhalt der eigentlichen Webseiten. Die zu formatierenden Textbereiche werden von sogenannten „tags“ (Anweisungen in spitzen Klammern) eingerahmt. Die gute Dokumentation (selfhtml) von S. Münz lässt sich im Internet finden (URL: siehe Kapitel 8).

```
<html>
  <head></head>
  <body>
    <h1>
      Hallo Welt
    </h1>
    <hr>
    <b>
      Und hier geht es fett weiter...
    </b>
  </body>
</html>
```

zeigt in einem Browser die Überschrift "Hallo Welt" (<h1>...</h1>), eine waagrechte Linie (<hr>) und den Text "Und hier geht es fett weiter..." in fetter Schrift (<b>...</b>) an.





### 5.1.2 CSS

CSS steht für **Cascading Style Sheets**. CSS werden wie HTML vom W3C ständig weiterentwickelt. Bereits 1996 wurde die erste CSS Norm (Level 1) verabschiedet. CSS Level 2 ist der derzeit offizielle Standard (KLOSS, 2000).

Anwendungen bestehen meist aus sehr vielen Seiten, die wiederum viele verschiedene Formatierungen enthalten. Um ein einheitliches Aussehen von Seiten zu erreichen, kann man zentrale Formatierungsanweisungen in einer CSS-Datei definieren. Dazu muss man in jeder HTML-Seite im Kopfbereich auf die Datei verweisen.

```
<html>
  <head>
    <link rel="stylesheet" type="text/css" href="formate.css">
  </head>
  <body>
    <h1>
      Hallo Welt
    </h1>
    <hr>
    <b>
      Und hier geht es fett weiter...
    </b>
  </body>
</html>
```

In der CSS-Datei "formate.css" steht beispielsweise die Anweisung, dass im Körperbereich die Schriftart "VERDANA" verwendet und die Überschrift in der Farbe "grün" angezeigt werden soll.

```
body{
font-family:Verdana;
}
h1{
color:green;
}
```

Im beschriebenen Beispiel wird im Browser der Text innerhalb der HTML-Seite in der Schriftart „Arial“ und die Überschrift in grüner Farbe angezeigt.

## 5.2 PHP

PHP wurde ursprünglich von Rasmus Lerdorf entwickelt und stand ursprünglich für „**P**ersonal **H**ome **P**age“ (URL: siehe Kapitel 8). Mit der Weiterentwicklung der Sprache änderte sich auch die Bedeutung des Akronymes zur rekursiven Version „**P**HP **H**ypertext **P**reprocessor“ (KRAUSE, 2000). PHP ist eine Scriptsprache zur Erstellung dynamischer Webseiten, die beim Aufruf einer Seite vom Webserver interpretiert und ausgeführt wird. Das Ergebnis des ausgeführten Quellcodes wird in HTML-Anweisungen umgewandelt und an den aufrufenden Browser gesendet. Skripte, die mit PHP erstellt werden, sind daher browserunabhängig. PHP unterstützt insbesondere die einfache Bearbeitung von Formularen und die Zusammenarbeit mit vielen verschiedenen Datenbanksystemen. Besonders im Zusammenhang mit MySQL-Datenbanken wird PHP im Internet häufig eingesetzt (THEIS, 2000).

Da HTML-Seiten mit integriertem PHP-Code von einem Webserver interpretiert werden, müssen die entsprechenden Dateien auf dem Server abgelegt und über den Webserver abgerufen werden. Ein einfaches Öffnen der Seiten mit einem Browser bewirkt, dass der PHP-Code nicht ausgeführt wird.

Damit der Webserver erkennt, ob sich innerhalb einer HTML-Seite PHP-Anweisungen befinden, enthalten Seiten mit PHP-Code eine besondere Dateiendung (z.B. .php oder .phtml). Der Quellcode selbst wird von bestimmten Befehlen (tags) eingerahmt (z.B. <?...?>). Folgende HTML-Seite mit PHP-Anweisungen:

```
<html>
  <body>
    <?
      echo "Hallo Welt";
    ?>
  </body>
</html>
```

zeigt in einem Browser den Text "Hallo Welt" an.

### 5.2.1 Verbindung zur Datenbank

Der erste Schritt zur Manipulation von Daten ist der Aufbau einer Verbindung zur Datenbank mit anschließender Authentifizierung. Dazu muss in einem PHP-Script eine Funktion mit dem

Namen des Servers (wahlweise IP-Adresse), dem Benutzernamen und dem Passwort als Parameter aufgerufen werden.

Syntax:

```
mysql_connect("Server", "Benutzer", "Passwort");
```

Um zu überprüfen, ob die Verbindung tatsächlich erfolgreich war, kann man das Ergebnis des Verbindungsaufbaus einer Variable zuweisen und den Wahrheitswert überprüfen. Das Script wird etwas erweitert:

```
<html>
  <body>
    <?
      $conn=mysql_connect("Server", "Benutzer", "Passwort");
      if($conn){
        echo "Verbindung steht!";
      }
      else{
        echo "Verbindungsaufbau fehlgeschlagen!";
      };
      mysql_close($conn);
    ?>
  </body>
</html>
```

### 5.2.2 Anzeigen von Einträgen

Datensätze können per SQL-Anweisung aus der Datenbank ausgelesen werden. Dazu wird zunächst die Verbindung zur Datenbank hergestellt. Die SQL-Anweisung und der Datenbankname werden in einer Variablen gespeichert. Mit diesen Variablen als Parameter wird die Funktion zur Übergabe an den Server aufgerufen. Das Ergebnis dieser Anweisung wird in einer Variablen in Form eines Arrays gespeichert. Die Länge des Arrays wird abgefragt und ebenfalls in einer Variablen gespeichert. Die einzelnen Datensätze werden nun innerhalb einer For-Schleife ausgegeben.

Im ersten Beispiel werden alle Datensätze hintereinander ausgegeben, da bisher keine HTML-Anweisungen im PHP-Code vorhanden sind. Eine kleine Ergänzung trennt die Felder durch ein Leerzeichen (echo " "); und die Datensätze durch eine horizontale Linie und einen Zeilenumbruch (echo "<hr><br>").

**Datensätze hintereinander anzeigen:**

```
<?
mysql_connect("Srv", "User", "PW");
$sql="select vorname, nachname from tabelle";
$database="Datenbankname";
$result = mysql($database,$sql);
$num = mysql_numrows($result);
for($i=0;$i<$num;$i++){
    echo mysql_result($result,$i,"vorname");
    echo mysql_result($result,$i,"nachname");
};
?>
```

**Datensätze getrennt anzeigen:**

```
<?
mysql_connect("Srv", "User", "PW");
$sql="select vorname, nachname from tabelle";
$database="Datenbankname";
$result = mysql($database,$sql);
$num = mysql_numrows($result);
for($i=0;$i<$num;$i++){
    echo mysql_result($result,$i,"vorname");
    echo " "; //Ausgabe eines Leerzeichens
    echo mysql_result($result,$i,"nachname");
    echo "<HR><BR>"; //Linie und Zeilenumbruch
};
?>
```

**5.2.3 Einfügen von Daten**

Damit Datensätze eingefügt werden können, müssen die Daten zunächst über ein HTML-Formular erfasst und das Ergebnis an ein PHP-Skript übergeben werden. Dabei werden die Eingabefelder mit einem eindeutigen Namen versehen. Die Inhalte der Felder werden in Form von Variablen, die den Namen der Eingabefelder entsprechen (edtVorname, edtNachname), gespeichert und können vom Skript ausgewertet werden. Die Daten werden durch eine SQL-Anweisung mit einer Funktion an die Datenbank übergeben (mysql(\$database,\$sql);).

**Formular (Auszug):**

```
<form action="script.phtml" method="POST">
    <input type="Text" name="edtVorname">
    <input type="Text" name="edtNachname">
    <input type="Submit" value="Einfügen">
</form>
```

Script „script.phtml“ (PHP-Anweisungen):

```
<?
mysql_connect("Srv", "User", "PW");
$sql="insert into tabelle(vorname, nachname) values ('$edtVorname',
'$edtNachname')";
$databasename="Datenbankname";
$result = mysql($databasename,$sql);
?>
```

#### 5.2.4 Löschen von Daten

Das Löschen von einzelnen Datensätzen erfordert eine eindeutige Identifizierung. Daher müssen die Datensätze zunächst mit einer SELECT-Anweisung ausgewählt und in einer Liste angezeigt werden. Zur Identifikation eignet sich der Primärschlüssel.

Aus dieser Liste kann der entsprechende Datensatz ausgewählt werden, indem man auf einen Hyperlink klickt, der das Script zum Löschen der Daten aufruft. Der Primärschlüssel muss dabei als Variable übergeben werden. Im Gegensatz zu Formularen, bei denen die Inhalte von Feldern mit der Post-Methode automatisch übergeben werden, muss bei einem Hyperlink die Variable an die URL angehängt werden.

Im aufgerufenen Script wird der Datensatz mit einer SQL-Anweisung gelöscht. Als Liste verwenden wir das Beispiel aus der SELECT-Anweisung, das ein wenig modifiziert wird.

Auswahlliste (Auszug):

```
<?
mysql_connect("Srv", "User", "PW");
$sql="select id, vorname, nachname from tabelle";
$databasename="Datenbankname";
$result = mysql($databasename,$sql);
$num = mysql_numrows($result);
for($i=0;$i<$num;$i++){
    $id=mysql_result($result,$i,"id");
    echo "<a href='delete_action.phtml?id=".$id."'>".$id."</a>";
    echo ". ";
    echo mysql_result($result,$i,"vorname");
    echo " ";
    echo mysql_result($result,$i,"nachname");
    echo "<HR><P>";
};
?>
```

Script zum Löschen "delete\_action.phtml" (Auszug):

```
<?
mysql_connect("Srv", "User", "PW");
$sql="delete from tabelle where id=$id";
$databasename="Datenbankname";
$result = mysql($databasename,$sql);
?>
```

### 5.2.5 Ändern von Daten

Auch beim Aktualisieren von Datensätzen müssen zunächst Datensätze aus einer Liste ausgewählt und der Primärschlüssel an ein Script übergeben werden. Der gewünschte Datensatz wird mit einer SQL-Anweisung ausgewählt. Im Gegensatz zum Löschen von Daten müssen die einzelnen Feldinhalte in ein Formular eingegeben (VALUE-Attribut eines INPUT-TAG) und der Primärschlüssel im Formular als verstecktes Feld eingefügt werden. Der SUBMIT-Button ruft ein zweites Script auf, das mit einer SQL-Anweisung den Datensatz aktualisiert. Insgesamt benötigt man zum Einfügen, Löschen und Aktualisieren drei verschiedene Scripte. Diese sind im Folgenden aufgelistet:

#### 1. Auswahl-Script:

```
mysql_connect("Srv", "User", "PW");
$sql="select id, vorname, nachname from tabelle";
$databasename="Datenbankname";
$result = mysql($databasename,$sql);
$num = mysql_numrows($result);
for($i=0;$i<$num;$i++){
    $id=mysql_result($result,$i,"id");
    echo "<a href='update_kursleiter.phtml?id=".$id."'>".$id."</a>";
    echo ". ";
    echo mysql_result($result,$i,"vorname");
    echo " ";
    echo mysql_result($result,$i,"nachname");
    echo "<HR><P>";
};
```

#### 2. Formular „update\_kursleiter.phtml“ (Beispiel INSERT erweitert):

```
<?
mysql_connect("Srv", "User", "PW");
$sql="select id, vorname, nachname from tabelle where id=$id";
$databasename="Datenbankname";
$result = mysql($databasename,$sql);
$wert1=mysql_result($result,0,"id");
$wert2=mysql_result($result,0,"vorname");
$wert3=mysql_result($result,0,"nachname");
?>
<form action="update_kursleiter_action.phtml" method="POST">
    <input type="Hidden" name="edtId" value="<? echo $wert1; ?>">
    <input type="Text" name="edtVorname" value="<? echo $wert2; ?>">
    <input type="Text" name="edtNachname" value="<? echo $wert3; ?>">
    <input type="Submit" value="Aktualisieren">
</form>
```

#### 3. Update-Script "update\_kursleiter\_action.phtml" (PHP-Anweisungen):

```
<?
mysql_connect("Srv", "User", "PW");
$sql="update tabelle set vorname='$edtVorname', nachname='$edtNachname'
where id=$edtId";
$databasename="kursverwaltung";
$result = mysql($databasename,$sql);
?>
```

## 5.3 Umsetzung

### 5.3.1 Konventionen

Nach der Fertigstellung der Datenbank muss die grafische Benutzeroberfläche in Form von HTML-Seiten mit integrierten PHP-Skripten realisiert werden. Da die endgültige Anwendung aus einer Reihe von HTML-Seiten besteht, die unterschiedliche Funktionalitäten zur Verfügung stellen, ist es sinnvoll sich auf einige Konventionen zur Dateiablage und Benennung von Dateien und Verzeichnissen zu einigen:

- Für die unterschiedlichen Funktionalitäten werden eigene Unterverzeichnisse angelegt, in denen die jeweiligen Webseiten abgespeichert werden (kompetenz, kurs, kursleiter, kursort, schule, teilnehmer).
- Grafiken werden in einem eigenen Unterverzeichnis untergebracht (grafik).
- Gemeinsam genutzte Skripte und Dateien, die zur Steuerung des Anwendungssystems benötigt werden, erhalten ein eigenes Unterverzeichnis (system).
- Dateien für den Superuser werden in einem eigenen Verzeichnis abgespeichert (useradmin).
- Zur Unterscheidung der Webseiten wird folgende Konvention festgelegt: Allen HTML-Seiten, die Listen generieren, wird ein „show\_“ vorangestellt. Skripten, die zur Ausführung kommen, wird ein „\_action“ nachgestellt. Ein Skript zum Anzeigen von Kursen, die gelöscht werden sollen, heißt daher „delete\_kurs.phtml“. Das entsprechende Skript, welches das Löschen vornimmt, wird unter dem Namen „delete\_kurs\_action.phtml“ abgespeichert.
- Großbuchstaben, Sonderzeichen und Leerzeichen sind in Datei- und Verzeichnisnamen nicht gestattet. Als Trennzeichen ist der Unterstrich erlaubt.
- Im root-Verzeichnis befinden sich nur die Dateien zur Erzeugung des Framesets (index.phtml), die HTML-Seiten, die innerhalb der Frames angezeigt werden (main.phtml, menue.phtml, logo.phtml), HTML-Seiten zur Verwaltung der Login-Angaben (login.phtml, login\_action.phtml, login\_neu\_action.phtml) und eine Datei mit den Einstellungen zu Passwort- und Serverangaben (common.phtml).
- Im Verzeichnis „system“ gibt es Dateien mit PHP-Code, die von jeder Seite eingebunden werden können. Die Seiten „kopf.phtml“, „fuss.phtml“ und „stylesheet.phtml“ werden an der entsprechenden Stelle von jeder HTML-Seite mittels der „include-Anweisung“ eingebunden.

Durch diese Konventionen ist sichergestellt, dass alle Seiten über ein einheitliches Layout verfügen und Verweise auf andere Webseiten problemlos möglich sind. Auch die Anbindung an die Datenbank (kopf.phtml) und die Steuerung der Anwendung über die durch PHP zur Verfügung gestellten Session-Variablen (session.phtml) kann so zentral zur Verfügung gestellt werden.

Anhaltspunkte für die Eingabeformulare und Ausgabelisten liefern die zu Beginn erstellten use cases. Grundsätzlich sind Formulare zur Pflege der Stammdaten und Listen zur Ausgabe der Daten notwendig. Um redundante Eingabemasken zu vermeiden kann mit Hilfe der use cases entschieden werden, welche Funktionalitäten zur Verfügung gestellt werden müssen.

Analog zu den Beispielen zur Programmierung von Scripten mit PHP werden für die folgenden Tabellen Formulare zur Eingabe, zum Ändern, zum Anzeigen und Löschen von Datensätzen erstellt.

- Kursleiter, Kompetenz, Teilnehmer, Schule, Kursort, Kurs

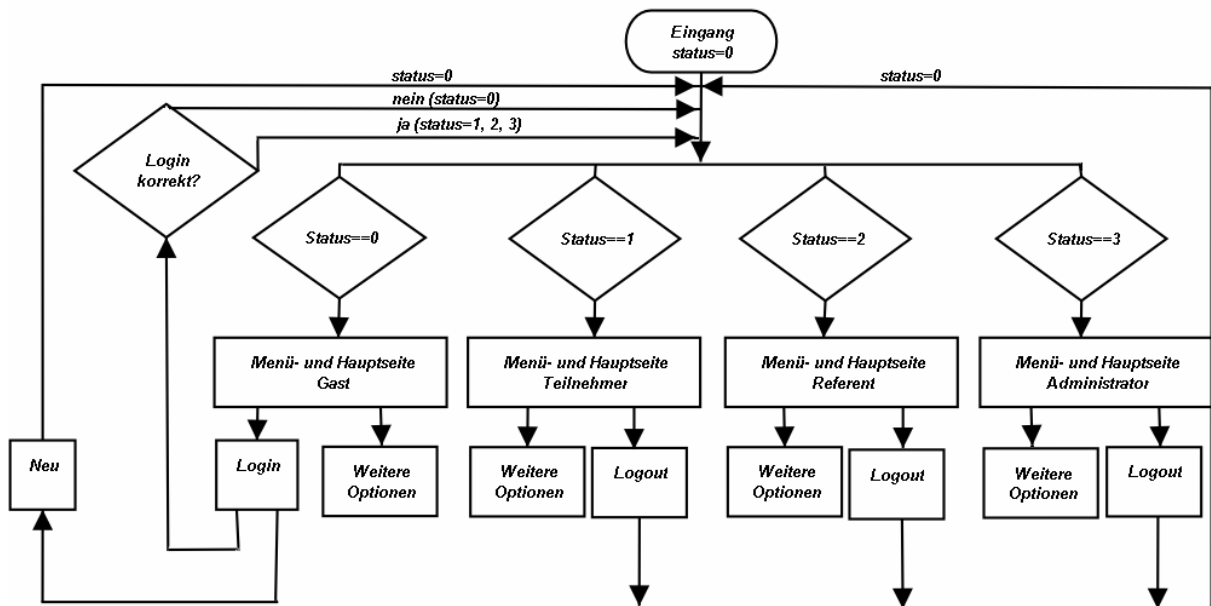
Da Eingabeformulare und Updateformulare ähnlich aufgebaut sind, kann diese Funktionalität in einem einzigen Formular realisiert werden. Die Unterscheidung erfolgt über eine Variablenbelegung (edit==true).

Die Zugangskontrolle wird über die PHP-Scripte mit Hilfe von Session-Variablen realisiert.



### 5.3.2 Programmstruktur

Die Benutzerführung wird über PHP-Sessions kontrolliert. Die Session wird bereits mit der Anzeige der ersten Seite initialisiert (session\_start) und in der Sessionvariable „status“ (system/session.phtml) der Status des aktuell angemeldeten Benutzers gespeichert. Zu Beginn der Anwendung ist nur ein Gastzugang zum System möglich (status=0).



Die Anwendung wird durch ein Frameset in vier Bildschirmbereiche unterteilt.

```

<?
session_start;
?>
<html>
  <head>
    <title>VSW Kursverwaltung</title>
  </head>
  <frameset rows="120,1*">
    <frameset cols="170,831*" rows="*">
      <frame src="logo.phtml" name="nav">
      <frame src="titel.phtml" name="titel">
    </frameset>
    <frameset cols="170,831*" rows="*">
      <frame src="menue.phtml" name="menue">
      <frame src="main.phtml" name="home">
    </frameset>
  </frameset>
  <noframes>
    <body bgcolor="#FFF8D3" background="grafik/background.jpg">
    </body>
  </noframes>
</html>

```

In der linken oberen Ecke wird das Logo angezeigt (logo.phtml). Im oberen rechten Bildschirmbereich befindet sich der Titel der Anwendung (titel.phtml). Über das Menü (menue.phtml) auf der linken Seite sind die verschiedenen Funktionalitäten erreichbar, die im rechten unteren Bereich angezeigt werden (main.phtml). Je nach Status des Benutzers werden in der Menüseite nur bestimmte Menüpunkte eingeblendet.

Die Startseite des Informatiksystems besteht aus Titelbereich, Menüführung und Hauptseite. Zu Beginn der Sitzung sind die Optionen „Login“ zur Anmeldung am System und „Kursangebote“ zur Anzeige von aktuellen Kursangeboten verfügbar. Auf der Hauptseite befindet sich ein Begrüßungsbildschirm.



### 5.3.2.1 Login

Bei Betätigen der Option „Login“ wird im Hauptfenster ein Formular (login.phtml) aufgerufen, in das im oberen Bereich vorhandene Login-Daten zur Authentifizierung eingegeben werden können (login\_action.phtml).

Benutzername	<input type="text"/>	Passwort	<input type="password"/>
<input type="button" value="Anmelden"/>			

Die eingegebenen Daten werden mit den Einträgen der Datenbank verglichen.

```
$result = mysql($dbdatabase,"select * from kursleiter where ltr_login
='$benutzername' AND ltr_password = '$password'");
```

```
$result = mysql($dbdatabase,"select * from teilnehmer where tnr_login
='$benutzername' AND tnr_password = '$password'");
```

Bei erfolgreicher Anmeldung wird ein dem Status entsprechend erweitertes Menü angezeigt, mit dem weitere Optionen erreichbar sind. Gleichzeitig werden auch die Sessionvariablen mit den korrekten Werten initialisiert.

Bei einem Treffer in der Tabelle „kursleiter“

```
if($num==1){
    $login=mysql_result($result,0,"ltr_login");
    $status=mysql_result($result,0,"ltr_status");
    $id=mysql_result($result,0,"ltr_id");
    $angemeldet=1;
}
```

Bei einem Treffer in der Tabelle „teilnehmer“

```
if($num==1){
    $login=mysql_result($result,0,"tnr_login");
    $status=1;
    $id=mysql_result($result,0,"tnr_id");
    $angemeldet=1;
}
```

Im unteren Bereich kann, falls noch kein Benutzerkonto existiert, eine Neuanmeldung mit anschließender Neugenerierung von Zugangsdaten vorgenommen werden (login\_neu\_action.phtml). Die Neugenerierung von Zugangsdaten erzeugt einen Benutzernamen und ein Passwort, die im Hauptfenster angezeigt werden.

Anrede/Titel	Herr ▾		
Vorname/Nachname	<input type="text"/>	Telefon	<input type="text"/>
Straße	<input type="text"/>	Fax	<input type="text"/>
PLZ/Ort	<input type="text"/>	Email	<input type="text"/>
Lehramt	GYM ▾	Schulnummer	5152 Goethe-Gymnasium ▾
<div style="text-align: center;"> <input type="button" value="Abschicken"/> <input type="button" value="Daten löschen"/> </div>			

Das Passwort wird per Zufallsgenerator aus einer Folge von Ziffern, Klein- und Großbuchstaben generiert und besteht aus sechs Zeichen.

```
$tnr_password=randString(6);

function randString($length){
    mt_srand((double)microtime()*1000000);
    $newstring="";
    if($length>0){
        while(strlen($newstring)<$length){
            switch(mt_rand(1,3)){
                case 1:    $newstring.=chr(mt_rand(49,57));
                           break;    //Rückgabe einer Ziffer 0-9
                case 2:    $newstring.=chr(mt_rand(65,90));
                           break;    //Rückgabe eines Zeichens A-Z
                case 3:    $newstring.=chr(mt_rand(97,122));
                           break;    // Rückgabe eines Zeichens a-z
            }
        }
    }
    return $newstring;
}
```

Der Benutzername besteht aus einem vorangestellten „t\_“, den jeweils ersten beiden Buchstaben des Vor- und Nachnamens sowie der Schulnummer.

```
$tnr_login='t_'.substr($tnr_vorname,0,2).substr($tnr_nachname,0,2).$tnr_schulnummer;
```

Diese Daten werden in die Datenbank aufgenommen, auf einer HTML-Seite angezeigt und gleichzeitig an die angegebene E-Mail-Adresse gesendet.

Ihre Daten wurde aufgenommen! Folgende Zugangsdaten wurden für Sie erstellt:

Login: t\_NuFR5157

Passwort: 7ZFGb2

Sie sollten sich diese Zugangsdaten notieren!!!

Bitte loggen Sie sich nun mit diesen Zugangsdaten über den Menüpunkt 'Login' in das System ein!

**Vielen Dank,**

Ihre Zugangsdaten wurden an die angegebene Email-Adresse versendet.

### 5.3.2.2 Optionen für den Gastzugang

Beim Anklicken der einzigen Option „Kursangebote“ wird eine Liste (show\_kurs.phtml) mit Kurzinformationen zu allen aktuellen Kursen angezeigt.

Nr.	Kurzthema	Belegung	Zielgruppe	Halbjahr	Aktion
3	Softwareeinsatz im Mathematikunterricht	1	Anfänger	03/04-1	<a href="#">Info</a>
4	Internetrecherche	0	Fortgeschrittene	03/04-1	<a href="#">Info</a>
5	BSCW, Projektarbeit	0	Anfänger	03/04-1	<a href="#">Info</a>
6	HotPotatoes 5.5	4	Fortgeschrittene	03/04-1	<a href="#">Info</a>
7	HTML im Kunstunterricht	3	Anfänger	03/04-1	<a href="#">Info</a>
8	Grafikbearbeitung	0	Fortgeschrittene	03/04-1	<a href="#">Info</a>
9	Grafiksoftware im Kunstunterricht	3	Profis	03/04-1	<a href="#">Info</a>
10	Mikroskopieren mit Kamera	3	Anfänger	03/04-1	<a href="#">Info</a>

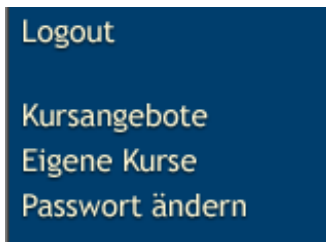
Der Link „Info“ öffnet eine weitere HTML-Seite (kurs\_info.phtml) auf der weitere verfügbare Informationen und die aktuelle Anzahl der Teilnehmer angezeigt werden. Weitere Optionen sind nicht verfügbar.

Kurs			
Kurshalbjahr	03/04-1	Bereich	Schulverbund
Kursnummer	6	Zielgruppe	Fortgeschrittene
Termin	Herbst 2003	Ort	Martin-Luther-Schule, Raum: A 105
Kurzthema	HotPotatoes 5.5	Kursleiter	Hauptmann, Ulla
Kursthema	HotPotatoes 5.5 für die Erstellung von Arbeitsmaterialien für den offenen Unterricht im sprach- und geisteswissenschaftlichen Unterricht)  Dieser Kurs baut auf den Inhalten der Basisqualifikation 'IntelLehren' auf.		

Aktuelle Teilnehmerzahl: 4

### 5.3.2.3 Teilnehmer

Ein erfolgreiches Login stellt im Menü die Optionen „Logout“ zur Abmeldung vom System (logout.phtml), „Kursangebote“ zum Anzeigen der aktuellen Kursangebote (show\_kurs.phtml), „Eigene Kurse“ zur Anzeige der bereits angewählten Kurse (show\_kurs\_eigen.phtml) und „Passwort ändern“ zur Änderung des aktuell gespeicherten Passwortes (user.phtml) zur Verfügung.



Im Gegensatz zum Status Gast gibt es nun auf der Informationsseite (kurs\_info.phtml) der einzelnen Kurse einen weiteren Link, mit dem man sich in den aktuell eingewählten Kurs eintragen kann (kurs\_belegen.phtml), sofern noch keine eigene Buchung für diesen Kurs vorliegt. Mehrfachbuchungen eines Kurses sind nicht möglich.

Kurs			
Kurshalbjahr	03/04-1	Bereich	Schulverbund
Kursnummer	6	Zielgruppe	Fortgeschrittene
Termin	Herbst 2003	Ort	Martin-Luther-Schule, Raum: A 105
Kurzthema	HotPotatoes 5.5	Kursleiter	Hauptmann, Ulla
Kursthema	HotPotatoes 5.5 für die Erstellung von Arbeitsmaterialien für den offenen Unterricht im sprach- und geisteswissenschaftlichen Unterricht)  Dieser Kurs baut auf den Inhalten der Basisqualifikation 'IntelLehren' auf.		

Aktuelle Teilnehmerzahl: 4

**Kurs belegen**

#### 5.3.2.4 Referent

Referenten können sich lediglich Informationen zu aktuellen Kursen ansehen (show\_kurs.phtml, kurs\_info.phtml), Einbuchungen sind nicht möglich.

Über die Option „Eigene Kurse“ erhalten Referenten jedoch einen Überblick über die bereits eingewählten Personen (kurs\_belegung.phtml).

Kurs			
Kurshalbjahr	03/04-1	Bereich	Schulverbund
Kursnummer	7	Zielgruppe	Anfänger
Termin	Herbst 2003	Ort	Martin-Luther-Schule, Raum: A 105
Kurzthema	HTML im Kunstunterricht	Kursleiter	Nürnberger, Norbert
Kursthema	Erstellen von Webseiten im Kunstunterricht		

Aktuelle Teilnehmerzahl: 3

[Personen anzeigen](#)

Die eingewählten Personen werden in Form einer Liste ausgegeben.

Name	Lehramt	Email	Sch.Nr.	Schule
Müller, Marius	GYM	a.hauptmann@t- online.de	5152	Goethe-Gymnasium
Meyer, Martin	GYM	a.hauptmann@t- online.de	5157	Martin-Luther-Schule
Eigen, Elvira	GYM	a.hauptmann@t- online.de	5157	Martin-Luther-Schule

#### 5.3.2.5 Administrator

Administratoren verfügen über einen Vollzugriff auf alle Formulare. Dadurch ist es möglich Kompetenzen, Kursleiter, Kurse, Kursorte, Schulen und Teilnehmer neu anzulegen (kompetenz.phtml, kursleiter.phtml, kurs.phtml, kursort.phtml, schule.phtml, teilnehmer.phtml) und vorhandene zu bearbeiten (show\_kompetenz.phtml, show\_kursleiter.phtml, show\_kurs.phtml, show\_kursort.phtml, show\_schule.phtml, show\_teilnehmer.phtml). Die auf der Hauptseite angezeigten Formulare zum Bearbeiten von vorhandenen Daten ermöglichen noch zusätzliche Optionen wie „Löschen“ oder „Passwort

ändern“ (Teilnehmer, Kursleiter). Über die Option „Passwort ändern“ kann auch das Administratorpasswort geändert werden (user.phtml).



Zusätzliche Funktionalitäten in HTML-Seiten, die je nach Status verfügbar sind, werden in den Scripten durch Kontrollfunktionen anhand der Belegung der Sessionvariablen angezeigt oder weggeblendet. Die Übergabe von Variablen erfolgt über Formulare mittels der Post-Methode, durch Anhängen an die URL bei Hyperlinks oder zentral über die Änderung der Sessionvariablen.

Nr.	Kurzthema	Belegung	Zielgruppe	Halbjahr	Aktion
3	Softwareeinsatz im Mathematikunterricht	1	Anfänger	03/04-1	<a href="#">Bearbeiten</a>   <a href="#">Löschen</a>   <a href="#">Info</a>
4	Internetrecherche	3	Fortgeschrittene	03/04-1	<a href="#">Bearbeiten</a>   <a href="#">Löschen</a>   <a href="#">Info</a>
5	BSCW, Projektarbeit	0	Anfänger	03/04-1	<a href="#">Bearbeiten</a>   <a href="#">Löschen</a>   <a href="#">Info</a>



## 6 Zusammenfassung

Das vorgestellte Datenbankprojekt wiederholt im Sinne eines Spiralcurriculums viele Aspekte obligater Kursinhalte. Die Erstellung einer komplexen Anwendung kann am Ende des Halbjahres der 12 II, 13 II oder im Rahmen einer Projektwoche erfolgen. Aufgrund der kostenlos erhältlichen open source-Software und den geringen Anforderungen an die Hardware ist ein LAMP-System kostengünstig in der Schule zu etablieren. Die aktuellen Standarddistributionen enthalten ausreichend vorkonfigurierte Software mit einer Reihe gut dokumentierter Werkzeuge. Provider vermieten preiswerte Server mit einem lauffähigen LAMP-System. Grundsätzlich ist open source-Software sehr flexibel im Einsatz. Zum vorgestellten System existieren zahlreiche Alternativen (PostgreSQL, mSQL, Adabas, DB2, Informix, Perl, Cocoon, Java-Servlets, Tomcat Server, usw.).

Die Entwicklung einer komplexen Anwendungssoftware gliedert sich in die drei Phasen Analyse, Design und Implementierung. Die Analyse setzt sich hauptsächlich mit der Anforderungsermittlung und Erstellung eines Pflichtenhefts auseinander. Wesentlicher Bestandteil ist die Erstellung von use cases. In der Designphase wird die Anwendung unter Berücksichtigung der Ergebnisse der Analysephase modelliert. Ein gutes Hilfsmittel zur Modellierung von Datenbanken ist das ER-Modell. Die Übersetzung des ER-Modells in das relationale Modell führt schließlich unter Verwendung der Datenbanksprache SQL zu einer Datenbank mit Tabellen. Die hauptsächliche Programmierarbeit wird in der Implementierungsphase geleistet. PHP eignet sich hervorragend als Vermittlungssprache zwischen HTML-Formularen und Datenbanken im Internet.

Das vorliegende Beispiel zeigt, dass auch in der Schule anspruchsvolle internetbasierte Datenbanklösungen modelliert und programmiert werden können, die den Vergleich mit professionellen Informatiksystemen nicht scheuen müssen. Die Schülerinnen und Schüler lernen neben intensiver Projektarbeit auch moderne Techniken kennen, die zurzeit auch im kommerziellen Bereich im Einsatz sind.

## 7 Literaturverzeichnis

BAUMANN, Rüdiger: *Didaktik der Informatik*. Stuttgart: Ernst Klett Verlag, 1996 (2. Auflage)

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar: *The Unifies Modeling Language User Guide*. Boston: Addison-Wesley, 1999

BORKNER-DELCARLO, Olaf: *Das Samba-Buch*. Nürnberg: SuSE GmbH, 1999

DUBOIS, Paul: *MySQL – Entwicklung, Implementierung und Referenz*. München: Markt + Technik Verlag, 2000

EILEBRECHT, Lars: *Apache Web-Server*. Bonn: ITP Verlag, 1997

GESELLSCHAFT FÜR INFORMATIK: *Empfehlungen für ein Gesamtkonzept zur informatischen Bildung an allgemein bildenden Schulen*. LOG IN 20, 2000 (Beilage)

HÄRDER, Theo; RAHM, Erhard: *Datenbanken – Konzepte und Techniken der Implementierung*. Berlin: Springer Verlag, 2001 (2., überarbeitete Auflage)

HESSISCHES KULTUSMINISTERIUM: *Lehrplan Informatik – gymnasialer Bildungsgang*. 2003 (<http://www.hessisches-kultusministerium.de/downloads/lehrpl/gymnasium/Informatik.pdf>)

HEUER, Andreas; SAAKE, Gunter: *Datenbanken: Konzepte und Sprachen*. Bonn: mitp-Verlag, 2000

HUBWIESER, Peter: *Didaktik der Informatik*. Berlin: Springer Verlag, 2001 (1. korrigierter Nachdruck)

KLOSS, Birgit: *CSS und DHTML*. München: Markt + Technik Verlag, 2002

KRAUSE, Jörg: *PHP Grundlagen und Lösungen – Webserver Programmierung unter Windows und Linux*. München: Carl Hanser Verlag, 2000

KÜHNEL, Jens: *SAMBA – Intergration von Linux/Unix-Rechnern in Windows-Netzwerke*. Bonn: mitp-Verlag, 1999

- LARMAN, Craig: *Applying UML and Patterns*. Upper Saddle River: Prentice Hall PTR, 2002 (2<sup>nd</sup> edition)
- MATTES, Daniel H.: *SQL – Der Einsatz im Intra- und Internet*. Böblingen: C & L Computer- und Literaturverlag, 2001
- MÜNZ, Stefan; NEFZGER, Wolfgang: *HTML 4.0 Handbuch*. Poing: Franzis' Verlag, 1999 (3., neubearbeitete Auflage)
- RAISON, André von; KIRSCH, Christian (Hrsg.): *Linux im Einsatz – Praxistips für Pinguine*. Heidelberg: dpunkt-Verlag, 1999
- RAMAKRISHNAN, Raghu; GEHRKE, Johannes: *Database Management Systems*. New York: McGraw-Hill, 2003 (3<sup>rd</sup>. edition)
- RÖHNER, GERHARD: *Zur Rolle der Sprache bei der Modellierung von Datenbanken*. Frankfurt/Main: 94. MNU-Kongress, 13 - 17.4.2003
- ROFF, Jason T.: *UML IT-Tutorial*. Bonn: mitp-Verlag, 2003
- STOLL, Rolf D.; LEIERER, Gudrun Anna: *PHP 4 + MySQL – Internet intern*. Düsseldorf: DATA Becker GmbH, 2000
- THEIS, Thomas: *PHP 4 – Webserverprogrammierung für Um- und Einsteiger*. Bonn: Galileo Press GmbH, 2000

## 8 Internetadressen

Apache	<a href="http://www.apache.org/">http://www.apache.org/</a>
Datenbankprojekt	<a href="http://kursverwaltung.vs-weschnitztal.de">http://kursverwaltung.vs-weschnitztal.de</a>
DBDesigner4	<a href="http://www.fabforce.net/dbdesigner4/">http://www.fabforce.net/dbdesigner4/</a>
DeZign for Databases	<a href="http://www.datanamic.com/">http://www.datanamic.com/</a>
ERwin Data Modeller	<a href="http://www3.ca.com/Solutions/Product.asp?ID=260">http://www3.ca.com/Solutions/Product.asp?ID=260</a>
HTML-Editor	<a href="http://www.qhaut.de/">http://www.qhaut.de/</a>
mySQL	<a href="http://www.mysql.de/">http://www.mysql.de/</a>
PHP	<a href="http://www.php.net/">http://www.php.net/</a>
phpMyAdmin	<a href="http://www.phpmyadmin.net/">http://www.phpmyadmin.net/</a>
ProFTPD	<a href="http://proftpd.linux.co.uk/">http://proftpd.linux.co.uk/</a>
SAMBA	<a href="http://de.samba.org/samba/samba.html">http://de.samba.org/samba/samba.html</a>
SELFHTML 8.0	<a href="http://selfhtml.teamone.de/">http://selfhtml.teamone.de/</a>
SuSE	<a href="http://www.suse.de/">http://www.suse.de/</a>

## 9 Anhang

## Anhang A: VERA - Bildschirmfotos der Import-Dialogfenster

**Teilnehmer/innen importieren**

dBase - Importdatei **Keine Importdatei für Teilnehmer/innen ausgewählt**

Anrede	<input type="text"/>	C4	Straße	<input type="text"/>	C30
Vorname	<input type="text"/>	C15	PLZ	<input type="text"/>	C5
Name	<input type="text"/>	C25	Ort	<input type="text"/>	C25
Lehramt	<input type="text"/>	C3	Telefon	<input type="text"/>	C15
Schulnummer	<input type="text"/>	N	Fax	<input type="text"/>	C15
			E-Mail	<input type="text"/>	C50

Ohne die Felder 'Anrede', 'Vorname' und 'Name' ist ein Import nicht möglich!  
Die Schulnummer ist numerisch!

Wenn Sie die Schuldaten bereits importiert oder erfasst haben und wenn bei den Teilnehmerdaten die Schulnummer ebenfalls importiert wurde, kann Vera die Informationen verknüpfen. Klicken Sie dazu auf 'Schulen zuordnen'.

Importdatei wählen   Daten importieren   Felder leeren   Schulen zuordnen   Schließen

**Kursleiter/innen importieren**

dBase - Importdatei **Keine Importdatei für Kursleiter/innen ausgewählt**

Anrede	<input type="text"/>	C4	Telefon privat	<input type="text"/>	C15
Vorname	<input type="text"/>	C15	Telefon beruflich	<input type="text"/>	C15
Name	<input type="text"/>	C25			
			Fax privat	<input type="text"/>	C15
Straße	<input type="text"/>	C30	Fax beruflich	<input type="text"/>	C15
PLZ	<input type="text"/>	C5			
Ort	<input type="text"/>	C25	E-Mail	<input type="text"/>	C50

Ohne die Felder 'Vorname' und 'Name' ist ein Import nicht möglich!

Importdatei wählen   Daten importieren   Felder leeren   Schließen

**Schulen importieren**

dBase - Importdatei **Keine Importdatei für Schulen ausgewählt**

Schulnummer	<input type="text"/>	N	Straße	<input type="text"/>	C30
Schulname	<input type="text"/>	C50	PLZ	<input type="text"/>	C5
			Ort	<input type="text"/>	C25
Schulform	<input type="text"/>	C4			
Rechtsstellung	<input type="text"/>	C4	Telefon	<input type="text"/>	C15
Schulträger	<input type="text"/>	C4	Fax	<input type="text"/>	C15
Staatliches Schulamt	<input type="text"/>	C4			
			Anrede Schulleiter/in	<input type="text"/>	C4
Lehrerzahl	<input type="text"/>	N3	Name Schulleiter/in	<input type="text"/>	C30

Alle Felder mit Ausnahme von 'Anrede Schulleiter/in' und 'Schulleiter/in' sind für einen Import erforderlich!  
Schulnummer und Lehrerzahl sind numerisch!

Wenn Sie auf '20% errechnen' klicken, ermittelt Vera anhand der Lehrerzahl jeder Schule den jeweiligen Anteil von 20%.

Importdatei wählen   Daten importieren   Felder leeren   20% errechnen   Schließen

**Kurse importieren**

dBase - Importdatei **Keine Importdatei für Kurse ausgewählt**

Halbjahr	<input type="text"/>	C7	Kurzthema	<input type="text"/>	C25
Bereich	<input type="text"/>	C6	Zielgruppe	<input type="text"/>	C100
Kursnummer	<input type="text"/>	C6	Termin	<input type="text"/>	C50
Kurzthema	<input type="text"/>	C100	Veranstaltungsort	<input type="text"/>	C50

Ohne die Felder 'Halbjahr', 'Kursnummer' und 'Kurzthema' ist ein Import nicht möglich!  
Beim Feld 'Halbjahr' müssen die Daten in der Importdatei das Format '01/02-1' (2001/2002, 1. Halbjahr) haben!

Importdatei wählen   Daten importieren   Felder leeren   Schließen

## Anhang B: Relationales Modell

```
Teilnehmer (tnr_id, tnr_anrede, tnr_vorname, tnr_nachname, tnr_strasse,
            tnr_plz, tnr_ort, tnr_telefon, tnr_fax, tnr_email, tnr_lehramt,
            tnr_schulnummer)

Kursleiter (ltr_id, ltr_anrede, ltr_vorname, ltr_nachname, ltr_strasse,
            ltr_plz, ltr_ort, ltr_tel_priv, ltr_tel_dienst, ltr_fax_priv,
            ltr_fax_dienst, ltr_login, ltr_password, ltr_status)

Kurs (krs_id, krs_halbjahr, krs_bereich, krs_kursnummer, krs_kursthema,
      krs_kurzthema, krs_zielgruppe, krs_termin, krs_veranstaltungsort,
      krs_leiter)

Schule (sch_schulnummer, sch_schulname, sch_schulform, sch_rechtsstellung,
        sch_schultraeger, sch_staatl_schulamt, sch_strasse, sch_plz, sch_ort,
        sch_telefon, sch_fax, sch_email, sch_lehrerzahl,
        sch_anredeSchulleiter, sch_schulleiterVorname,
        sch_schulleiterNachname)

Kompetenz (kpt_id, kpt_kuerzel, kpt_beschreibung)


Kursort (krt_id, krt_raum, krt_ausstattung, krt_beschreibung,
         krt_schulnummer)

Kursleiter_Kompetenz (kk_id, ltr_id, kpt_id)


Teilnehmer_Kurs (tk_id, tnr_id, krs_id)
```

## Anhang C: Relationales Modell mit Datentypen

## Relation Teilnehmer:


Key	Feldname	Datentyp	Besonderheit
	tnr_id	BIGINT (21)	NOT NULL, AUTOINCREMENT
	tnr_anrede	VARCHAR (10)	
	tnr_vorname	VARCHAR (50)	
	tnr_nachname	VARCHAR (50)	
	tnr_strasse	VARCHAR (100)	
	tnr_plz	VARCHAR (5)	
	tnr_ort	VARCHAR (100)	
	tnr_telefon	VARCHAR (50)	
	tnr_fax	VARCHAR (50)	
	tnr_email	VARCHAR (100)	
	tnr_lehramt	VARCHAR (50)	
	tnr_schulnummer	BIGINT (21)	
	tnr_login	VARCHAR (100)	
	tnr_password	VARCHAR (100)	

## Relation Kursleiter:


Key	Feldname	Datentyp	Besonderheit
	ltr_id	BIGINT (21)	NOT NULL, AUTOINCREMENT
	ltr_anrede	VARCHAR (10)	
	ltr_vorname	VARCHAR (50)	
	ltr_nachname	VARCHAR (50)	
	ltr_strasse	VARCHAR (100)	
	ltr_plz	VARCHAR (5)	
	ltr_ort	VARCHAR (100)	
	ltr_tel_priv	VARCHAR (50)	
	ltr_tel_dienst	VARCHAR (50)	
	ltr_fax_priv	VARCHAR (50)	
	ltr_fax_dienst	VARCHAR (50)	
	ltr_email	VARCHAR (100)	
	ltr_login	VARCHAR (100)	
	ltr_password	VARCHAR (100)	
	ltr_status	INT (11)	




**Relation Kurs:**

Key	Feldname	Datentyp	Besonderheit
	krs_id	BIGINT(21)	NOT NULL, AUTOINCREMENT
	krs_halbjahr	VARCHAR(50)	
	krs_bereich	VARCHAR(50)	
	krs_kursnummer	BIGINT(21)	
	krs_kurzthema	VARCHAR(100)	
	krs_kursthema	MEDIUMTEXT	
	krs_zielgruppe	VARCHAR(50)	
	krs_termin	VARCHAR(50)	
	krs_veranstaltungsort	BIGINT(21)	-> Kursort{krt_id}
	krs_leiter	BIGINT(21)	-> Kursleiter{ltr_id}
	krs_status	VARCHAR(10)	


**Relation Schule:**

Key	Feldname	Datentyp	Besonderheit
	sch_schulnummer	BIGINT(21)	
	sch_schulname	VARCHAR(255)	
	sch_schulform	VARCHAR(100)	
	sch_rechtsstellung	VARCHAR(100)	
	sch_schultraeger	VARCHAR(100)	
	sch_staatl_schulamt	VARCHAR(100)	
	sch_strasse	VARCHAR(100)	
	sch_plz	VARCHAR(5)	
	sch_ort	VARCHAR(100)	
	sch_telefon	VARCHAR(50)	
	sch_fax	VARCHAR(50)	
	sch_email	VARCHAR(100)	
	sch_lehrerzahl	INT(11)	
	sch_anredeSchulleiter	VARCHAR(10)	
	sch_vorname_schulleiter	VARCHAR(50)	
	sch_nachname_schulleiter	VARCHAR(50)	


**Relation Kompetenz:**

Key	Feldname	Datentyp	Besonderheit
	kpt_id	BIGINT(21)	NOT NULL, AUTOINCREMENT
	kpt_kuerzel	VARCHAR(255)	
	kpt_beschreibung	MEDIUMTEXT	


**Relation Kursort:**

Key	Feldname	Datentyp	Besonderheit
	krt_id	BIGINT(21)	NOT NULL, AUTOINCREMENT
	krt_raum	VARCHAR(255)	
	krt_ausstattung	MEDIUMTEXT	
	krt_schulnummer	BIGINT(21)	-> Schule{sch_schulnummer}

**Relation Kursleiter\_Kompetenz:**

Key	Feldname	Datentyp	Besonderheit
	kk_id	BIGINT(21)	NOT NULL, AUTOINCREMENT
	ltr_id	BIGINT(21)	-> Kursleiter{ltr_id}
	kpt_id	BIGINT(21)	-> Kompetenz{kpt_id}

**Relation Teilnehmer\_Kurs:**

Key	Feldname	Datentyp	Besonderheit
	tk_id	BIGINT(21)	NOT NULL, AUTOINCREMENT
	tnr_id	BIGINT(21)	-> Teilnehmer{tnr_id}
	krs_id	BIGINT(21)	-> Kurs{krs_id}

## Anhang D: SQL-Anweisungen

```
CREATE TABLE teilnehmer (  
    tnr_id bigint(21) NOT NULL auto_increment,  
    tnr_anrede varchar(10) default NULL,  
    tnr_vorname varchar(50) default NULL,  
    tnr_nachname varchar(50) default NULL,  
    tnr_strasse varchar(100) default NULL,  
    tnr_plz varchar(5) default NULL,  
    tnr_ort varchar(100) default NULL,  
    tnr_telefon varchar(50) default NULL,  
    tnr_fax varchar(50) default NULL,  
    tnr_email varchar(100) default NULL,  
    tnr_lehramt varchar(50) default NULL,  
    tnr_schulnummer bigint(21) default NULL,  
    tnr_login varchar(100) default NULL,  
    tnr_password varchar(100) default NULL,  
    PRIMARY KEY (tnr_id)  
) TYPE=ISAM PACK_KEYS=1;
```

```
CREATE TABLE kursleiter (  
    ltr_id bigint(21) NOT NULL auto_increment,  
    ltr_anrede varchar(10) default NULL,  
    ltr_vorname varchar(50) default NULL,  
    ltr_nachname varchar(50) default NULL,  
    ltr_strasse varchar(100) default NULL,  
    ltr_plz varchar(5) default NULL,  
    ltr_ort varchar(100) default NULL,  
    ltr_tel_priv varchar(50) default NULL,  
    ltr_tel_dienst varchar(50) default NULL,  
    ltr_fax_priv varchar(50) default NULL,  
    ltr_fax_dienst varchar(50) default NULL,  
    ltr_email varchar(100) default NULL,  
    ltr_login varchar(100) default NULL,  
    ltr_password varchar(100) default NULL,  
    ltr_status int(11) default NULL,  
    PRIMARY KEY (ltr_id)  
) TYPE=ISAM PACK_KEYS=1;
```

```
CREATE TABLE kurs (  
    krs_id bigint(21) NOT NULL auto_increment,  
    krs_halbjahr varchar(50) default NULL,  
    krs_bereich varchar(50) default NULL,  
    krs_kursnummer bigint(21) default NULL,  
    krs_kurzthema varchar(100) default NULL,  
    krs_kursthema mediumtext,  
    krs_zielgruppe varchar(50) default NULL,  
    krs_termin varchar(50) default NULL,  
    krs_ort bigint(21) default NULL,  
    krs_leiter bigint(21) default NULL,  
    krs_status varchar(10) default NULL,  
    PRIMARY KEY (krs_id)  
) TYPE=ISAM PACK_KEYS=1;
```

```
CREATE TABLE schule (  
    sch_schulname varchar(255) default NULL,  
    sch_schulform varchar(100) default NULL,  
    sch_rechtsstellung varchar(100) default NULL,  
    sch_schultraeger varchar(100) default NULL,  
    sch_strasse varchar(100) default NULL,  
    sch_plz varchar(5) default NULL,  
    sch_ort varchar(100) default NULL,  
    sch_tel_priv varchar(50) default NULL,  
    sch_tel_dienst varchar(50) default NULL,  
    sch_fax_priv varchar(50) default NULL,  
    sch_fax_dienst varchar(50) default NULL,  
    sch_email varchar(100) default NULL,  
    sch_lehreranzahl int(11) default NULL,  
    sch_anrede_schulleiter varchar(10) default NULL,  
    sch_vorname_schulleiter varchar(50) default NULL,  
    sch_nachname_schulleiter varchar(50) default NULL,  
    sch_schulnummer bigint(21) default NULL,  
    sch_id bigint(21) NOT NULL auto_increment,  
    PRIMARY KEY (sch_id)  
) TYPE=ISAM PACK_KEYS=1;  
  
CREATE TABLE kompetenz (  
    kpt_id bigint(21) NOT NULL auto_increment,  
    kpt_kuerzel varchar(255) default NULL,  
    kpt_beschreibung mediumtext,  
    PRIMARY KEY (kpt_id)  
) TYPE=ISAM PACK_KEYS=1;  
  
CREATE TABLE kursort (  
    krt_id bigint(21) unsigned NOT NULL auto_increment,  
    krt_raum varchar(255) default NULL,  
    krt_ausstattung mediumtext,  
    krt_schulnummer bigint(21) default NULL,  
    PRIMARY KEY (krt_id)  
) TYPE=ISAM PACK_KEYS=1;  
  
CREATE TABLE kursleiter_kompetenz (  
    kk_id bigint(21) NOT NULL auto_increment,  
    ltr_id bigint(21) default NULL,  
    kpt_id bigint(21) default NULL,  
    PRIMARY KEY (kk_id)  
) TYPE=ISAM PACK_KEYS=1;  
  
CREATE TABLE teilnehmer_kurs (  
    tk_id bigint(21) NOT NULL auto_increment,  
    tnr_id bigint(21) default NULL,  
    krs_id bigint(21) default NULL,  
    PRIMARY KEY (tk_id)  
) TYPE=ISAM PACK_KEYS=1;
```

## Anhang E: Grundlegende HTML-Befehle

### Überschriften:

```
<H1>...</H1>, <H2>...</H2>, <H3>...</H3>, <H4>...</H4>, <H5>...</H5>,
<H6>...</H6>
```

### Textformatierungen:

```
<B>fett</B>, <I>kursiv</I>, <U>unterstrichen</U>
<BIG>Größer</BIG>, <SMALL>Kleiner</SMALL>
<SUB>tiefergestellt</SUB>, <SUP>höhergestellt</SUP>
<FONT COLOR="Farbe (englisch)">Text</FONT>
< FONT FACE="VERDANA, ARIAL, HELVETICA">Texte</ FONT >
<FONT SIZE="+3">Text</FONT>
```

### Absätze und Linien:

```
<HR>
<BR>, <P>
<P ALIGN="LEFT">Text</P> linksbündig
<P ALIGN="RIGHT">Text</P> rechtsbündig
<P ALIGN="CENTER">Text</P> zentriert
<P ALIGN="JUSTIFY">Text</P> Blocksatz
```

### Formulare:

```
<form action="dateiname.html" method="POST">
    <input type="Text" name="" value="" size="" maxlength="">
    <input type="Password" name="" value="" size=""
maxlength="">
    <textarea name="" cols="20" rows="5"></textarea>
    <input type="Radio" name="" value="v">
    <input type="Checkbox" name="" value="v">
    <select name="" size="">
        <OPTION VALUE="">Text
    </select>
    <input type="Submit" name="" value="">
    <input type="reset">
</form>
```

Hierbei müssen unterschiedliche Namen vergeben werden. Value ist ein einzutragender Grundwert, Size bezieht sich auf die Größe des Feldes.

### Verweise und Bilder:

```
<A HREF="dateiname.html">Link</A>
<IMG SRC="bildname.gif">
<A HREF="dateiname.html"> <IMG SRC="bildname.gif" BORDER="0"></A>
```

**Frames:**

```

<HTML>
<head>
    <title>
        Titel in der Titelleiste
    </title>
</head>
<frameset rows="30%,70%">
    <frame src="oben.html" name="oben">
    <frameset cols="30%,70%">
        <frame src="links.html" name="links">
        <frame src="rechts.html" name="rechts">
    </frameset>
</frameset>
</HTML>

```

**Verweise, die in einem anderen Frame angezeigt werden sollen:**

```

<A HREF="dateiname.html" TARGET="framename">Link</A>

```

**Tabellen:**

Diese Tabelle hat 4 Zeilen und 5 Spalten


```

<table >
<tr> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td></tr>
<tr> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td></tr>
<tr> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td></tr>
<tr> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td></tr>
</table>

```

<TR> beginnt eine neue Zeile, die durch </TR> abgeschlossen werden muss.

<TD>beginnt eine Zelle, die durch </TD> abgeschlossen werden muss.

Zwischen <TD> und </TD> steht der Text, der in dem Tabellenfeld angezeigt werden soll.

```
index.phtml
<?
session_start;
?>
<html>
<head>
<title>VSW Kursverwaltung
</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<frameset rows="120,1*" frameborder="NO" border="0">
  <frameset cols="170,831*" rows="*" frameborder="NO" border="0" framespacing="0">
    <frame src="logo.phtml" name="nav" scrolling="NO" noresize frameborder="NO"
marginwidth="0" marginheight="0">
    <frame src="titel.phtml" name="titel" noresize frameborder="NO" marginwidth="0"
marginheight="0">
  </frameset>
  <frameset cols="170,831*" rows="*" frameborder="NO" border="0" framespacing="0">
    <frame src="menue.phtml" name="menue" scrolling="NO" noresize frameborder="NO"
marginwidth="0" marginheight="0">
    <frame src="main.phtml" name="home" noresize frameborder="NO" marginwidth="0"
marginheight="0">
  </frameset>
</frameset>
<noframes>
<body bgcolor="#FFF8D3" background="grafik/background.jpg">
</body></noframes>
</html>
```

```
logo.phtml
<html>
<head>
<title>logo</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body background="grafik/background.jpg">

</body>
</html>
```

```
titel.phtml
<html>
<head>
<title>logo</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body background="grafik/background.jpg">
<div align="center"></div>
</body>
</html>
```

```
main.phtml
<? require('system/session.phtml');?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<link rel="stylesheet" type="text/css" href="system/stylesheet.css">
</head>
<body background="grafik/background.jpg">
<Table border=0 width=100%>
<tr>
<td width=100 valign=top>&nbsp; </td>
<td valign=top align=center>
```

```

<h2>Herzlich Willkommen</h2><P>
<p align="justify">Sie sind zur Zeit als<B>
<?
switch($status){
    case 0:
        echo "Gast </b>registriert.</p> <p align=justify> Sie können sich am System
anmelden, indem Sie im Menü die Option 'Login' wählen. Auf der Login-Seite können
Sie sich entweder mit einem Login-Namen und Passwort anmelden oder, falls Sie noch
nicht registriert sind, sich mit Ihren persönlichen Angaben registrieren lassen.
Sie erhalten dann einen gültigen Login-Namen und ein Passwort mit dem Sie sich
anmelden können.    </p> ";
        break;
    case 1:
        echo "Anwender </b>registriert.</p> <p align=justify>Sie können sich im Menü
alle angebotenen Kurse oder diejenigen, in die sich bereits eingewählt haben,
anzeigen lassen. Falls Sie sich in einen weiteren Kurs einwählen wollen, lassen Sie
sich zunächst alle Kurse anzeigen und wählen Sie dann den Punkt 'Info'. Hier
besteht die Möglichkeit, sich einzuwählen.</p>";
        break;
    case 2:
        echo "Referent </b>registriert.</p> Sie können sich im Menü alle angebotenen
Kurse oder diejenigen, in denen Sie Kursleiter sind anzeigen lassen. Zu den eigenen
Kursen können Sie sich die entsprechenden Teilnehmerlisten anzeigen lassen.";
        break;
    case 3:
        echo "Administrator </b>registriert.</p>";
        break;
}
?>
</td>
<td width=100 valign=top>&nbsp; </td>
</TR>
<tr>
<td width=100 valign=top>&nbsp; </td>
<td valign=top align=center>
<p align="justify"><BR>Falls Sie ein Kursthema wünschen, das nicht angeboten wird,
können Sie uns gerne einen <a href="mailto:a.hauptmann@mls-
rimbach.de">Vorschlag</a> per Email unterbreiten.    </p>
</td>
<td width=100 valign=top>&nbsp; </td>
</TR>
</TABLE>
</body>
</html>

menue.phtml
<?
require('system/session.phtml');
?>
<html>
<head>
<title></title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<style type="text/css">
a:link { font-weight:bold; color:#E06000; text-decoration:none }
a:visited { font-weight:bold; color:#E06000; text-decoration:none }
a:hover { font-weight:bold; color:#E06000; text-decoration:none }
a:active { font-weight:bold; color:#E06000; text-decoration:underline }
a:focus { font-weight:bold; color:#E06000; text-decoration:underline }
<!--
/* ... Hier werden die Formate definiert ... */
-->
</style>

</head>
<body bgcolor="#FFFFFF" background="grafik/nav_back.gif" text="#FFC000">
<?
if ($angemeldet==1){

```



```

    echo "<a href=logout.phtml target=_top><img src='grafik/logout.gif'
border=0></a><P>";
}
else{
    echo "<a href=login.phtml target=home><img src='grafik/login.gif'
border=0></a><P>";
}

if($status==0){?>
<a href="kurs/show_kurs.phtml" Target="home"><img src='grafik/kurs_alle.gif'
border=0></a><BR>
<?
}
else if($status==1){ ?>
<a href="kurs/show_kurs.phtml" Target="home"><img src='grafik/kurs_alle.gif'
border=0></a><BR>
<a href="kurs/show_kurs_eigen.phtml" Target="home"><img
src='grafik/kurs_eigene.gif' border=0></a><BR>
<a href="useradmin/user.phtml" Target="home"><img src='grafik/pwd.gif'
border=0></a><BR>
<?
}
else if($status==2){ ?>
<a href="kurs/show_kurs.phtml" Target="home"><img src='grafik/kurs_alle.gif'
border=0></a><BR>
<a href="kurs/show_kurs_eigen.phtml" Target="home"><img
src='grafik/kurs_eigene.gif' border=0></a><p>
<a href="useradmin/user.phtml" Target="home"><img src='grafik/pwd.gif'
border=0></a><BR>
<?
}
else if($status==3){ ?>
<BR>
<img src='grafik/kompetenz.jpg' border=0><BR>
<a href="kompetenz/kompetenz.phtml" Target="home"><img src='grafik/neu.jpg'
border=0></a>
<a href="kompetenz/show_kompetenz.phtml" Target="home"><img
src='grafik/bearbeiten.jpg' border=0></a><BR>

<img src='grafik/referent.jpg' border=0><BR>
<a href="kursleiter/kursleiter.phtml" Target="home"><img src='grafik/neu.jpg'
border=0></a>
<a href="kursleiter/show_kursleiter.phtml" Target="home"><img
src='grafik/bearbeiten.jpg' border=0></a></a><BR>
<img src='grafik/kurs.jpg' border=0><BR>
<a href="kurs/kurs.phtml" Target="home"><img src='grafik/neu.jpg' border=0></a>
<!-- <a href="kurs/show_kurs.phtml?filter=geplant" Target="home">Geplante
bearbeiten</a><BR>
<a href="kurs/show_kurs.phtml?filter=aktuell" Target="home">Aktuelle
bearbeiten</a><BR>
<a href="kurs/show_kurs.phtml?filter=fertig" Target="home">Fertige
bearbeiten</a><BR>
<a href="kurs/show_kurs.phtml?filter=storniert" Target="home">Stornierte
bearbeiten</a><BR> -->
<a href="kurs/show_kurs.phtml?filter='alle'" Target="home"><img
src='grafik/bearbeiten.jpg' border=0></a></a><BR>
<img src='grafik/kursort.jpg' border=0><BR>
<a href="kursort/kursort.phtml" Target="home"><img src='grafik/neu.jpg'
border=0></a>
<a href="kursort/show_kursort.phtml" Target="home"><img src='grafik/bearbeiten.jpg'
border=0></a></a><BR>
<img src='grafik/schule.jpg' border=0><BR>
<a href="schule/schule.phtml" Target="home"><img src='grafik/neu.jpg' border=0></a>
<a href="schule/show_schule.phtml" Target="home"><img src='grafik/bearbeiten.jpg'
border=0></a></a><BR>
<img src='grafik/teilnehmer.jpg' border=0><BR>
<a href="teilnehmer/teilnehmer.phtml" Target="home"><img src='grafik/neu.jpg'
border=0></a>

```

```

<a href="teilnehmer/show_teilnehmer.phtml" Target="home"><img
src='grafik/bearbeiten.jpg' border=0></a></a><p>
<a href="useradmin/user.phtml" Target="home"><img src='grafik/pwd.gif'
border=0></a><BR>
<?
}
else{
echo "Anmeldung Fehlgeschlagen";
};
?>
</body>
</html>

login.phtml
<?
require('./common.phtml');
require('./system/session.phtml');
$conn=mysql_connect($dbserver, $dbuser, $dbpasswd);
$schule = mysql($dbdatabase,"select * from schule");
$anz_schule = mysql_numrows($schule);
?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<link rel="stylesheet" type="text/css" href="system/stylesheet.css">
<SCRIPT LANGUAGE="JavaScript">
function check_data(){
var vorname = window.document.sendform.tnr_vorname.value;
var nachname = window.document.sendform.tnr_nachname.value;
var strasse = window.document.sendform.tnr_strasse.value;
var plz = window.document.sendform.tnr_plz.value;
var ort = window.document.sendform.tnr_ort.value;
var email = window.document.sendform.tnr_email.value;
var re_email = /^[_a-zA-Z0-9-]+)(\.[_a-zA-Z0-9-]+)*@[a-zA-Z0-9-]+\.[a-zA-Z]{2,3}$/;
var checked = true;
var adresse = true;

if((re_email.test(email)==false || email.length<1)){
if(vorname.length<1){
alert ("Das Feld Vorname muss ausgefüllt sein");
checked = false;
adresse = false;
}
if(nachname.length<1){
alert ("Das Feld Nachname muss ausgefüllt sein");
checked = false;
adresse = false;
}
if(strasse.length<1){
alert ("Das Feld Straße muss ausgefüllt sein");
checked = false;
adresse = false;
}
if(plz.length<1){
alert ("Das Feld Postleitzahl muss ausgefüllt sein");
checked = false;
adresse = false;
}
if(ort.length<1){
alert ("Das Feld Ort muss ausgefüllt sein");
checked = false;
adresse = false;
}
}
if(adresse == false){
if((re_email.test(email)==false || email.length<1)){

```



```
login_action.phtml
<?
require('./common.phtml');
require('system/session.phtml');
$conn=mysql_connect($dbserver, $dbuser, $dbpasswd);
//Login als alt oder Neuanmeldung
    $password = md5($password);
    $result = mysql($dbdatabase,"select * from kursleiter where ltr_login
='$benutzername' AND ltr_password = '$password'");
    $num = mysql_numrows($result);
    if($num==1){
        $login=mysql_result($result,0,"ltr_login");
```

```

        $status=mysql_result($result,0,"ltr_status");
        $id=mysql_result($result,0,"ltr_id");
        $angemeldet=1;
    }
    $result = mysql($dbdatabase,"select * from teilnehmer where tnr_login
='$benutzername' AND tnr_password = '$password'");
    $num = mysql_numrows($result);
    if($num==1){
        $login=mysql_result($result,0,"tnr_login");
        $status=1;
        $id=mysql_result($result,0,"tnr_id");
        $angemeldet=1;
    }
?>
<HTML><HEAD>
<meta http-equiv="refresh" content="0 ; URL=index.phtml">
</head>
<body background="grafik/background.jpg">
<?
mysql_close($conn);
?>
</body>
</HTML>

login_neu_action.phtml
<?
require('./common.phtml');
require('system/session.phtml');
function randString($length){
    mt_srand((double)microtime()*1000000);
    $newstring="";
    if($length>0){
        while(strlen($newstring)<$length){
            switch(mt_rand(1,3)){
                case 1: $newstring.=chr(mt_rand(49,57)); break; // 0-9
                case 2: $newstring.=chr(mt_rand(65,90)); break; // A-Z
                case 3: $newstring.=chr(mt_rand(97,122)); break; // a-z
            }
        }
    }
    return $newstring;
}
?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body background="grafik/background.jpg">
<Table border=0 width=100%>
<tr>
<td width=100 valign=top>&nbsp; </td>
<TD valign=top align=center>
<?
$conn=mysql_connect($dbserver, $dbuser, $dbpasswd);
//Password erzeugen
$tnr_password=randString(6);
$encrypt=md5($tnr_password);
//login erzeugen
$tnr_login='t_'.substr($tnr_vorname,0,2).substr($tnr_nachname,0,2).$tnr_schulnummer
;
$result= mysql($dbdatabase,"SELECT * FROM teilnehmer WHERE tnr_login LIKE
'$tnr_login' and tnr_vorname LIKE '$tnr_vorname' and tnr_nachname LIKE
'$tnr_nachname'");
$num = mysql_numrows($result);
if ($num > 0){
    echo "Dieser Datensatz existiert bereits";
}
else {

```

```

$insert = mysql($dbdatabase,"insert into teilnehmer(tnr_anrede, tnr_vorname,
tnr_nachname, tnr_strasse, tnr_plz, tnr_ort, tnr_telefon, tnr_fax, tnr_email,
tnr_lehramt, tnr_schulnummer, tnr_login, tnr_password) values ('$tnr_anrede',
'$tnr_vorname', '$tnr_nachname', '$tnr_strasse', '$tnr_plz', '$tnr_ort',
'$tnr_telefon', '$tnr_fax', '$tnr_email','$tnr_lehramt','$tnr_schulnummer,
'$tnr_login', '$encrypt')");
echo "Ihre Daten wurde aufgenommen! ";
echo "Folgende Zugangsdaten wurden für Sie erstellt:";
echo "<P>";
echo "Login: ".$tnr_login;
echo "<BR>";
echo "Passwort: ".$tnr_password;
echo "<P>";
echo "Sie sollten sich diese Zugangsdaten notieren!!!";
echo "<BR>";
echo "Bitte loggen Sie sich nun mit diesen Zugangsdaten über den Menüpunkt
'Login' in das System ein!";
if ($tnr_email!=""){
//Wenn die Email-Adresse oder beides angegeben wurde.
$mailbody="Automatisch generierte Email: Zugangsdaten zum Kurswahlssystem
-----
Persoenliche Daten:
-----
";
if($tnr_anrede!=""){
$mailbody=$mailbody.$tnr_anrede."
";
}
if($tnr_vorname!=""){
$mailbody=$mailbody.$tnr_vorname." ";
}
if($tnr_nachname!=""){
$mailbody=$mailbody.$tnr_nachname."
";
}
if($tnr_strasse!=""){
$mailbody=$mailbody.$tnr_strasse."
";
}
if($tnr_plz!=""){
$mailbody=$mailbody.$tnr_plz;
}
if($tnr_ort!=""){
$mailbody=$mailbody." ".$tnr_ort."
";
}
if($tnr_telefon!=""){
$mailbody=$mailbody."Telefon: ".$tnr_telefon."
";
}
if($tnr_fax!=""){
$mailbody=$mailbody."Fax: ".$tnr_fax."
";
}
if($tnr_email!=""){
$mailbody=$mailbody."Email: ".$tnr_email."
";
}
$mailbody=$mailbody."
-----
Angaben zur Schule:
-----
";
if($tnr_lehramt!=""){
$mailbody=$mailbody."Lehramt: ".$tnr_lehramt."
";
}

```



## Verzeichnis kompetenz

## kompetenz.phtml

```

<?
require('../system/kopf.phtml');
if (!isset($edit)){
    $edit=0;
};
if($edit==1){
    $result = mysql($dbdatabase,"select * from kompetenz where kpt_id=$kpt_id");
    $num = mysql_numrows($result);
}
?>
<h2>Kompetenzen
<?
if($edit==1){
    echo "bearbeiten";
}else{
echo "eingeben";
}
?>
</h2>
<form action="kompetenz_action.phtml" method="Post">
<input type="hidden" name="edit" value="<? echo $edit; ?>">
<input type="hidden" name="kpt_id" value="<? echo $kpt_id; ?>">
<table >
<tr>
    <td valign="TOP">Kürzel </td>
    <td><input type="Text" name="kpt_kuerzel" value="<?
if($edit==1){echo mysql_result($result,0,"kpt_kuerzel");} ?>"> </td>
</tr>
<tr>
    <td valign="TOP">Beschreibung </td>
    <td><textarea name="kpt_beschreibung" cols="50" rows="10"><? if($edit==1){echo
mysql_result($result,0,"kpt_beschreibung");} ?></textarea></td>
</tr>
<tr>
    <td colspan="2" align="center"><input type="Submit" name="einfuegen"
value="Einfügen"> </td>
</tr>
</table>
</form>
<?
require('../system/fuss.phtml');
?>

```

## kompetenz\_action.phtml

```

<?
require('../system/kopf.phtml');
if ($edit==0){
    $result = mysql($dbdatabase,"insert into
kompetenz(kpt_kuerzel,kpt_beschreibung)values('$kpt_kuerzel','$kpt_beschreibung')")
;
}
else if($edit==1){
    $result=mysql($dbdatabase,"update kompetenz set kpt_kuerzel='$kpt_kuerzel',
kpt_beschreibung='$kpt_beschreibung' where kpt_id=$kpt_id");
}
?>
eingetragen

<?
require('../system/fuss.phtml');
?>

```

## show\_kompetenz.phtml

```

<?
require('../system/kopf.phtml');

```



```

    $result = mysql($dbdatabase,"select * from kompetenz");
    $num = mysql_numrows($result);
    ?>
    <h2>Kompetenzen anzeigen</h2>
    <?
    $i=0;
    if ($num ==0){
        echo "<FONT SIZE=2 FACE=Verdana,Arial>Keine Kompetenzen eingetragen</FONT>";
    }
    else {
        echo "<TABLE BORDER=0 WIDTH=100%>";
        echo"<TR>
        <TH VALIGN=TOP width=20 align=left>Nr.</TH>
        <TH width=100 align=left>Kürzel</TH>
        <TH width=300 align=left>Beschreibung</TH>
        <TH align=left>Aktion</TH>
        </TR>";
        while($i<$num){
            echo "<TR>";
            echo "<TD VALIGN=TOP>";
            echo $i+1;
            echo "</TD><TD VALIGN=TOP>";
            if (mysql_result($result,$i,"kpt_kuerzel")!=""){
                echo mysql_result($result,$i,"kpt_kuerzel");
            }
            echo "</TD><TD VALIGN=TOP>";
            if (mysql_result($result,$i,"kpt_beschreibung")!=""){
                echo mysql_result($result,$i,"kpt_beschreibung");
            }
            echo "</TD><TD VALIGN=TOP>";
        }
        ?>
        <a href="kompetenz.phtml?kpt_id=
        <? echo mysql_result($result,$i,"kpt_id"); ?>
        &edit=1">Bearbeiten</a><BR>
        <a href="delete_kompetenz_action.phtml?kpt_id=
        <? echo mysql_result($result,$i,"kpt_id"); ?>
        ">Löschen</a>
        <?
            echo "</TD></TR>";
            $i++;
        };
        echo "</TABLE>";
    };
    require('../system/fuss.phtml');
    ?>

delete_kompetenz.phtml
<?
    require('../system/kopf.phtml');
    $result = mysql($dbdatabase,"delete from kompetenz where kpt_id=$kpt_id");
    ?>
    Gelöscht

    <?
        require('../system/fuss.phtml');
    ?>

```

## Verzeichnis kurs

kurs.phtml

```

<?
require('../system/kopf.phtml');
$kursort = mysql($dbdatabase,"select * from kursort");
$anz_kursort = mysql_numrows($kursort);
$kursleiter = mysql($dbdatabase,"select * from kursleiter");
$anz_kursleiter = mysql_numrows($kursleiter);
if (!isset($edit)){
    $edit=0;
};
if($edit==1){
    $result = mysql($dbdatabase,"select * from kurs where krs_id=$krs_id");
    $num = mysql_numrows($result);
}
?>
<form action="kurs_action.phtml" method=post>
<input type="hidden" name="edit" value="<? echo $edit; ?>">
<input type="hidden" name="krs_id" value="<? echo $krs_id; ?>">
<center>
<?
if($edit==1){
    echo "<h2>Kurs bearbeiten</h2>";
}else{
echo "<h2>Neuen Kurs eintragen</h2>";
}
?>
<table BORDER=1 CELSPACING=0 CELLPADDING=4 BGCOLOR=#DDDDDD Width="100%">
<tr BGCOLOR=#CCCCCC>
<td><b>Kurs</b> </td>
<td>&nbsp; </td>
<td>&nbsp; </td>
<td>&nbsp; </td>
</tr>
<tr BGCOLOR=#CCCCCC>
<td WIDTH="50">Kurshalbjahr</td>
<td>
<Select name="krs_halbjahr">
<? if($edit==1){
    echo "<Option Value=";
    echo mysql_result($result,0,"krs_halbjahr");
    echo ">";
    echo mysql_result($result,0,"krs_halbjahr");
    echo"</option>";
}
?>
<Option Value="02/03-2">02/03-2</option>
<Option Value="03/04-1">03/04-1</option>
<Option Value="03/04-2">03/04-2</option>
<Option Value="04/05-1">04/05-1</option>
<Option Value="04/05-2">04/05-2</option>
</SELECT>
</td>
<td>Bereich</td>
<td>
<Select name="krs_bereich">
<? if($edit==1){
    echo "<Option Value=";
    echo mysql_result($result,0,"krs_bereich");
    echo ">";
    echo mysql_result($result,0,"krs_bereich");
    echo"</option>";
}
?>
<Option Value="Schulverbund">Schulverbund</option>
<Option Value="Bergstraße">Bergstraße</option>
<Option Value="Odenwald">Odenwald</option>

```

```

</SELECT>
</td>
</tr>
<tr BGCOLOR=#CCCCCC>
  <td>Kursnummer </td>
  <td><input type="text" name="krs_kursnummer" value="<?
if($edit==1){echo mysql_result($result,0,"krs_kursnummer");} ?>"></td>
  <td>Zielgruppe</td>
  <td>
<Select name="krs_zielgruppe">
  <? if($edit==1){
    echo "<Option Value=";
    echo mysql_result($result,0,"krs_zielgruppe");
    echo ">";
    echo mysql_result($result,0,"krs_zielgruppe");
    echo "</option>";
  }
  ?>
  <Option Value="Anfänger">Anfänger</option>
  <Option Value="Fortgeschrittene">Fortgeschrittene</option>
  <Option Value="Profis">Profis</option>
  <Option Value="Administratoren">Administratoren</option>
  <Option Value="IT-Beauftragte">IT-Beauftragte</option>
</SELECT>
</td>
</tr>
<tr BGCOLOR=#CCCCCC>
  <td>Termin</td>
  <td>
<input type="text" name="krs_termin" value="<?
if($edit==1){echo mysql_result($result,0,"krs_termin");} ?>">
  </td>
  <td>Ort</td>
  <td><select name="krs_ort">
  <?
  if($edit==1){
    $akt_kursort=mysql_result($result,0,"krs_ort");
    $erg_kursort = mysql($dbdatabase,"select * from kursort where
krt_id=$akt_kursort");
    echo "<option value='".mysql_result($erg_kursort,0,"krt_id").'"
selected'>".mysql_result($erg_kursort,0,"krt_id")."
".mysql_result($erg_kursort,0,"krt_raum");
  }
  $i=0;
  while($i<$anz_kursort){
    echo "<option
value='".mysql_result($kursort,$i,"krt_id")."'>".mysql_result($kursort,$i,"krt_id")
." ".mysql_result($kursort,$i,"krt_raum");
    $i++;
  }
  ?>
</select>
  </td>
</tr>
<tr BGCOLOR=#CCCCCC>
  <td> Kurzthema</td>
  <td><input type="text" name="krs_kurzthema" value="<?
if($edit==1){echo mysql_result($result,0,"krs_kurzthema");} ?>"> </td>
  <td>Kursthema</td>
  <td rowspan=3><textarea name="krs_kursthema" cols="30" rows="5"><?
if($edit==1){echo mysql_result($result,0,"krs_kursthema");} ?></textarea></td>
</tr>
<tr BGCOLOR=#CCCCCC>
  <td>Kursleiter</td>
  <td>
<select name="krs_leiter">
  <?
  if($edit==1){

```

```

    $akt_kursleiter=mysql_result($result,0,"krs_leiter");
    $erg_kursleiter = mysql($dbdatabase,"select * from kursleiter where
ltr_id=$akt_kursleiter");
    echo "<option value='".mysql_result($erg_kursleiter,0,"ltr_id")."'
selected>".mysql_result($erg_kursleiter,0,"ltr_id")."
".mysql_result($erg_kursleiter,0,"ltr_nachname");
}
$i=0;
while($i<$anz_kursleiter){
    echo "<option
value='".mysql_result($kursleiter,$i,"ltr_id")."'>".mysql_result($kursleiter,$i,"lt
r_id")." ".mysql_result($kursleiter,$i,"ltr_nachname");
    $i++;
}
?>
</select>
</td>
<td>&nbsp;</td>
</tr>
<tr BGCOLOR=#CCCCCC>
    <td>Kursstatus</td>
    <td>
        <Select name="krs_status">
        <? if($edit==1){
            echo "<Option Value=";
            echo mysql_result($result,0,"krs_status");
            echo ">";
            echo mysql_result($result,0,"krs_status");
            echo"</option>";
        }
        ?>
        <Option Value="geplant">geplant</option>
        <Option Value="aktuell">aktuell</option>
        <Option Value="fertig">fertig</option>
        <Option Value="storniert">storniert</option>
        </SELECT>
    </td>
<td>&nbsp;</td>
</tr>
</table>
<P ALIGN="CENTER">
<input type="submit" name="Button" value="Abschicken">
<input type="reset" name="Button" value="Daten löschen"></p>
</form><BR>

<?
    require('../system/fuss.phtml');
?>

kurs_action.phtml
<?php
require ('../system/kopf.phtml');
if ($edit==0){
    $insert = mysql($dbdatabase,"insert into kurs(krs_halbjahr, krs_bereich,
krs_kursnummer, krs_kurzthema, krs_kursthema, krs_zielgruppe, krs_termin, krs_ort,
krs_leiter,krs_status) values ('$krs_halbjahr', '$krs_bereich', $krs_kursnummer,
'$krs_kurzthema', '$krs_kursthema', '$krs_zielgruppe', '$krs_termin', $krs_ort,
$krs_leiter, '$krs_status')");
    Print "Neuer Kurs wurde eingetragen!";
}
else if ($edit=1){
    $result= mysql($dbdatabase,"update kurs set krs_halbjahr='$krs_halbjahr',
krs_bereich='$krs_bereich', krs_kursnummer=$krs_kursnummer,
krs_kurzthema='$krs_kurzthema', krs_kursthema='$krs_kursthema',
krs_zielgruppe='$krs_zielgruppe', krs_termin='$krs_termin', krs_ort=$krs_ort,
krs_leiter=$krs_leiter, krs_status='$krs_status' where krs_id=$krs_id");
    Print "Kurs wurde geändert!";
}

```

```

require ('../system/fuss.phtml');
?>

delete_kurs_action.phtml
<?
require('../system/kopf.phtml');
$belegung = mysql($dbdatabase,"select * from teilnehmer_kurs where
krs_id=$krs_id");
$anz_belegung=mysql_numrows($belegung);
if($anz_belegung>0){
    echo "Achtung!<BR>Dieser Kurs ist zurzeit mit ".$anz_belegung." Teilnehmern
belegt und kann nicht gelöscht werden.<P>Nehmen Sie bitte zunächst die Teilnehmer
aus dem Kurs oder weisen Sie andere Kurse zu!";
    echo "
<TABLE BORDER=0 WIDTH=100%>
<TR>
<TH VALIGN=TOP width=100 align=left>Name</TH>
<TH width=100 align=left>Lehramt</TH>
<TH width=10 align=left>Email</TH>
<TH width=10 align=left>Sch.Nr.</TH>
<TH align=left>Aktion</TH>
</TR>";
    $i=0;
    while($i<$anz_belegung){
        $akt_tnr=mysql_result($belegung,$i,"tnr_id");
        $erg_tnr = mysql($dbdatabase,"select * from teilnehmer where tnr_id=$akt_tnr");
        echo "<TR>";
        echo "<TD VALIGN=TOP>";
        if (mysql_result($erg_tnr,0,"tnr_nachname")!=""){
            echo mysql_result($erg_tnr,0,"tnr_nachname").",
".mysql_result($erg_tnr,0,"tnr_vorname");
        }
        echo "</TD><TD VALIGN=TOP>";
        if (mysql_result($erg_tnr,0,"tnr_lehramt")!=""){
            echo mysql_result($erg_tnr,0,"tnr_lehramt");
        }
        echo "</TD><TD VALIGN=TOP>";
        if (mysql_result($erg_tnr,0,"tnr_email")!=""){
            echo mysql_result($erg_tnr,0,"tnr_email");
        }
        echo "</TD><TD VALIGN=TOP>";
        if (mysql_result($erg_tnr,0,"tnr_schulnummer")!=""){
            echo mysql_result($erg_tnr,0,"tnr_schulnummer");
        }
        echo "</TD><TD VALIGN=TOP>";
    }
    ?>
    <a href="kurs_zuweisung_delete.phtml?tnr_id=
    <? echo mysql_result($erg_tnr,0,"tnr_id"); ?>
    ">Herausnehmen</a> |
    <a href="kurs_zuweisung_edit.phtml?tnr_id=
    <? echo mysql_result($erg_tnr,0,"tnr_id"); ?>
    ">Zuweisen</a>
    <?
        echo "</TD></tr>";
        $i++;
    }
    echo"</Table>";
}
else{
    $result = mysql($dbdatabase,"delete from kurs where krs_id=$krs_id");
    echo "Der Kurs Nr. ".$krs_id." wurde erfolgreich gelöscht!";
}
require('../system/fuss.phtml');
?>

kurs_belegen.phtml
<?
    require('../system/kopf.phtml');

```

```

    $result = mysql($dbdatabase,"select* from teilnehmer_kurs where tnr_id=$id and
krs_id=$krs_id");
    $num = mysql_numrows($result);
    if($num>0){
        echo "Kursbelegung fehlgeschlagen. Sie haben sich bereits in diesen Kurs
eingetragen!";
    }
    else{
        $result = mysql($dbdatabase,"insert into
teilnehmer_kurs(tnr_id,krs_id)values($id,$krs_id)");
        echo "Kursbelegung erfolgreich!";
    }
    require('../system/fuss.phtml');
?>

kurs_belegung.phtml
<?
    require('../system/kopf.phtml');
    $result = mysql($dbdatabase,"select teilnehmer.tnr_id, teilnehmer.tnr_vorname,
teilnehmer.tnr_nachname, teilnehmer.tnr_lehramt, teilnehmer.tnr_email,
teilnehmer.tnr_schulnummer, teilnehmer_kurs.krs_id, teilnehmer_kurs.tnr_id from
teilnehmer, teilnehmer_kurs where teilnehmer.tnr_id=teilnehmer_kurs.tnr_id and
teilnehmer_kurs.krs_id=$krs_id");
    $num = mysql_numrows($result);
?>
<h2>Teilnehmer anzeigen</h2>
<?
$i=0;
if ($num ==0){
    echo "<FONT SIZE=2 FACE=Verdana,Arial>Keine Teilnehmer eingetragen</FONT>";
}
else {
    echo "<TABLE BORDER=0 WIDTH=100%>";
    echo"<TR>
<TH VALIGN=TOP width=100 align=left>Name</TH>
<TH width=15 align=left>Lehramt</TH>
<TH width=100 align=left>Email</TH>
<TH width=15 align=left>Sch.Nr.</TH>
<TH width=180 align=left>Schule</TH>";
    if($status==3){
        echo "<TH width=150 align=left>Aktion</TH>";
    };
    echo"</TR>";
    while($i<$num){
        echo "<TR>";
        echo "<TD VALIGN=TOP>";
        if (mysql_result($result,$i,"tnr_nachname")!=""){
            echo mysql_result($result,$i,"tnr_nachname").",
".mysql_result($result,$i,"tnr_vorname");
        }
        echo "</TD><TD VALIGN=TOP>";
        if (mysql_result($result,$i,"tnr_lehramt")!=""){
            echo mysql_result($result,$i,"tnr_lehramt");
        }
        echo "</TD><TD VALIGN=TOP>";
        if (mysql_result($result,$i,"tnr_email")!=""){
            echo mysql_result($result,$i,"tnr_email");
        }
        echo "</TD><TD VALIGN=TOP>";
        if (mysql_result($result,$i,"tnr_schulnummer")!=""){
            echo mysql_result($result,$i,"tnr_schulnummer");
        }
        echo "</TD><TD VALIGN=TOP>";
        if (mysql_result($result,$i,"tnr_schulnummer")!=""){
            $akt_schule=mysql_result($result,$i,"tnr_schulnummer");
            $erg_schulname = mysql($dbdatabase,"select sch_schulname from schule where
sch_schulnummer=$akt_schule");
            $anzahl=mysql_numrows($erg_schulname);

```

```

        if($anzahl>0){
            echo mysql_result($erg_schulname,0,"sch_schulname");
        }
        else{
            echo "Schulname unbekannt";
        }
    }
    if($status==3){
        echo "</TD><TD VALIGN=TOP>";
        $tnr_id=mysql_result($result,$i,"tnr_id");
        echo "<a
href='kurs_zuweisung_delete.phtml?tnr_id=$tnr_id&krs_id=$krs_id&loesche=konkret'>Ab
melden</a>";
    };
    echo "</TD></TR>";
    $i++;
};
echo "</TABLE>";
};

require('../system/fuss.phtml');
?>
kurs_info.phtml
<?
    require('../system/kopf.phtml');
    $result = mysql($dbdatabase,"select * from kurs where krs_id=$krs_id");
    $num = mysql_numrows($result);
    $result_anzahl = mysql($dbdatabase,"select * from teilnehmer_kurs where
krs_id=$krs_id");
    $num_anzahl = mysql_numrows($result_anzahl);
?>
<center>
<h2>Kursinfo</h2>
<table BORDER=1 CELLSPACING=0 CELLPADDING=4 BGCOLOR=#DDDDDD Width="100%">
<tr BGCOLOR=#CCCCCC>
    <th>Kurs</th>
    <th>&nbsp;</th>
    <th>&nbsp;</th>
    <th>&nbsp;</th>
</tr>
<tr BGCOLOR=#CCCCCC>
    <td WIDTH="50" valign=top>Kurshalbjahr</td>
    <td WIDTH="200"><? echo mysql_result($result,0,"krs_halbjahr"); ?></td>
    <td WIDTH="50" valign=top>Bereich</td>
    <td WIDTH="200"><? echo mysql_result($result,0,"krs_bereich"); ?></td>
</tr>
<tr BGCOLOR=#CCCCCC>
    <td valign=top>Kursnummer </td>
    <td><? echo mysql_result($result,0,"krs_kursnummer"); ?></td>
    <td valign=top>Zielgruppe</td>
    <td><? echo mysql_result($result,0,"krs_zielgruppe"); ?></td>
</tr>
<tr BGCOLOR=#CCCCCC>
    <td valign=top>Termin</td>
    <td><? echo mysql_result($result,0,"krs_termin"); ?></td>
    <td valign=top>Ort</td>
    <td>
<?
        $akt_ort=mysql_result($result,0,"krs_ort");
        $erg_ort= mysql($dbdatabase,"select krt_schulnummer,krt_raum from kursort where
krt_id=$akt_ort");
        $akt_schule=mysql_result($erg_ort,0,"krt_schulnummer");
        $erg_schule=mysql($dbdatabase,"select sch_schulname from schule where
sch_schulnummer=$akt_schule");
        echo mysql_result($erg_schule,0,"sch_schulname");
        echo", Raum: ";
        echo mysql_result($erg_ort,0,"krt_raum");

```

```

?>
</td>
</tr>
<tr BGCOLOR=#CCCCCC>
  <td valign=top> Kurzthema</td>
  <td><? echo mysql_result($result,0,"krs_kurzthema"); ?></td>
  <td valign=top>Kursleiter</td>
  <td>
    <?
      $akt_leiter=mysql_result($result,0,"krs_leiter");
      $serg_leiter= mysql($dbdatabase,"select ltr_nachname,ltr_vorname from kursleiter
where ltr_id=$akt_leiter");
      echo mysql_result($serg_leiter,0,"ltr_nachname");
      echo", ";
      echo mysql_result($serg_leiter,0,"ltr_vorname");
    ?>
  </td>
</tr>
<tr BGCOLOR=#CCCCCC>
  <td valign=top>Kursthema</td>
  <td valign=top colspan=3><? echo mysql_result($result,0,"krs_kursthema"); ?></td>
</tr>
</table>
<p align="left">
Aktuelle Teilnehmerzahl: <? echo $num_anzahl; ?>
<BR>
<?
if($status==1){ ?>
  <a href="kurs_belegen.phtml?krs_id=<? echo mysql_result($result,0,"krs_id"); ?>">
  Kurs belegen</a>
<?
}
else if($status>1){ ?>
  <a href="kurs_belegung.phtml?krs_id=<? echo mysql_result($result,0,"krs_id");
?>">
  Personen anzeigen</a>
<?
}
require('../system/fuss.phtml');
?>

kurs_zuweisung_delete.phtml
<?
require('../system/kopf.phtml');
if($loesche=='kurs'){
  //Alle Kurszuweisungen eines Kurses l&ouml;schen. Kurs leeren!!!
  $result = mysql($dbdatabase,"delete from teilnehmer_kurs where krs_id=$krs_id");
  echo "Alle Kurszuordnungen des Kurses Nr. ".$krs_id." wurden erfolgreich
gel&ouml;scht!";
}
else if($loesche=='teilnehmer'){
  //Alle Kurszuweisungen eines Teilnehmers l&ouml;schen!!!
  $result = mysql($dbdatabase,"delete from teilnehmer_kurs where tnr_id=$tnr_id");
  echo "Die Kurszuordnungen von Teilnehmer Nr. ".$tnr_id." wurden erfolgreich
gel&ouml;scht!";
}
else if($loesche=='konkret'){
  //Kurszuweisung f&uuml;r 1 Buchung l&ouml;schen!
  $result = mysql($dbdatabase,"delete from teilnehmer_kurs where tnr_id=$tnr_id and
krs_id=$krs_id");
  echo "Die Kurszuordnung von Teilnehmer Nr. ".$tnr_id." zum Kurs Nr. ".$krs_id."
wurde erfolgreich gel&ouml;scht!";
}
require('../system/fuss.phtml');
?>

show_kurs.phtml
<?

```



```

require('../system/kopf.phtml');
if (!isset($filter)){
    $filter='geplant';
};
//Kurse für Kursleiter
if($filter=='geplant'){
    $result = mysql($dbdatabase,"select * from kurs where krs_status='geplant'");
}
else if($filter=='aktuell'){
    $result = mysql($dbdatabase,"select * from kurs where krs_status='aktuell'");
}
else if($filter=='fertig'){
    $result = mysql($dbdatabase,"select * from kurs where krs_status='fertig'");
}
else if($filter=='storniert'){
    $result = mysql($dbdatabase,"select * from kurs where krs_status='storniert'");
}
else{
    $result = mysql($dbdatabase,"select * from kurs");
}
$num = mysql_numrows($result);
?>
<h2>Kurse anzeigen</h2>
<?
$i=0;
if ($num ==0){
    echo "<FONT SIZE=2 FACE=Verdana,Arial>Keine Kurse eingetragen</FONT>";
}
else {
    echo "<TABLE BORDER=1 WIDTH=100% BGCOLOR=#CCCCCC CELLSPACING=0 CELLPADDING=4>";
    echo"<TR>
    <TH VALIGN=TOP width=10 align=left>Nr.</TH>
    <TH width=100 align=left>Kurzthema</TH>
    <TH width=10 align=left>Belegung</TH>
    <TH width=100 align=left>Zielgruppe</TH>
    <TH width=50 align=left>Halbjahr</TH>
    <TH width=100 align=left>Aktion</TH>
    </TR>";
    while($i<$num){
        $akt_kurs= mysql_result($result,$i,"krs_id");
        $erg_anzahl= mysql($dbdatabase,"select count(*) as summe from teilnehmer_kurs
where krs_id=$akt_kurs ");
        $anzahl = mysql_result($erg_anzahl,0,"summe");
        echo "<TR>";
        echo "<TD VALIGN=TOP>";
        if (mysql_result($result,$i,"krs_kursnummer")!=""){
            echo mysql_result($result,$i,"krs_kursnummer");
        }
        echo "</TD><TD VALIGN=TOP>";
        if (mysql_result($result,$i,"krs_kurzthema")!=""){
            echo mysql_result($result,$i,"krs_kurzthema");
        }
        echo "</TD><TD VALIGN=TOP>";
        echo $anzahl;
        echo "</TD><TD VALIGN=TOP>";
        if (mysql_result($result,$i,"krs_zielgruppe")!=""){
            echo mysql_result($result,$i,"krs_zielgruppe");
        }
        echo "</TD><TD VALIGN=TOP>";
        if (mysql_result($result,$i,"krs_halbjahr")!=""){
            echo mysql_result($result,$i,"krs_halbjahr");
        }
        echo "</TD><TD VALIGN=TOP>";
        //Zugriffe werden beschränkt
        //Admin
        if($status>2){
            echo "<a href='kurs.phtml?krs_id=".mysql_result($result,$i,"krs_id");
            echo "&edit=1'>Bearbeiten</a>";

```

```

        echo " | ";
        echo "<a
href='delete_kurs_action.phtml?krs_id=".mysql_result($result,$i,"krs_id");
echo "'>Löschen</a>";
        echo " | ";
        echo "<a href='kurs_info.phtml?krs_id=".mysql_result($result,$i,"krs_id");
        echo "&edit=1'>Info</a>";
    }
    //Kursleiter
    elseif($status>1){
        echo "<a href='kurs_info.phtml?krs_id=".mysql_result($result,$i,"krs_id");
        echo "&edit=1'>Info</a>";
    }
    //Teilnehmer
    elseif($status>0){
        echo "<a href='kurs_info.phtml?krs_id=".mysql_result($result,$i,"krs_id");
        echo "&edit=1'>Info</a>";
    }
    else{
        echo "<a href='kurs_info.phtml?krs_id=".mysql_result($result,$i,"krs_id");
        echo "&edit=1'>Info</a>";
    }
    echo "</TD></TR>";
    $i++;
};
echo "</TABLE>";
};
require('../system/fuss.phtml');
?>

show_kurs_eigene.phtml
<?
require('../system/kopf.phtml');
if($status>1){
    $result = mysql($dbdatabase,"select * from kurs where krs_leiter=$id");
    $num = mysql_numrows($result);
}
else if($status==1){
    $result = mysql($dbdatabase,"select kurs.krs_id, kurs.krs_kursnummer,
kurs.krs_kurzthema, kurs.krs_halbjahr, kurs.krs_zielgruppe, teilnehmer_kurs.tnr_id,
teilnehmer_kurs.krs_id from kurs, teilnehmer_kurs where teilnehmer_kurs.tnr_id=$id
and kurs.krs_id=teilnehmer_kurs.krs_id");
    $num = mysql_numrows($result);
}
else{
    $result = mysql($dbdatabase,"select * from kurs");

    $num = mysql_numrows($result);
}
?>
<h2>Kurse anzeigen</h2>
<?
$i=0;
if ($num==0){
    echo "<FONT SIZE=2 FACE=Verdana,Arial>Keine Kurse eingetragen</FONT>";
}
else {
    echo "<TABLE BORDER=1 WIDTH=100% BGCOLOR=#CCCCC CELLSPACING=0 CELLPADDING=4>";
    echo"<TR>
<TH VALIGN=TOP width=10 align=left>Nr.</TH>
<TH width=100 align=left>Kurzthema</TH>
<TH width=10 align=left>Belegung</TH>
<TH width=100 align=left>Zielgruppe</TH>
<TH width=50 align=left>Halbjahr</TH>
<TH width=100 align=left>Aktion</TH>
</TR>";
    while($i<$num){
        $akt_kurs= mysql_result($result,$i,"krs_id");

```

```

$erg_anzahl= mysql($dbdatabase,"select count(*) as summe from teilnehmer_kurs
where krs_id=$akt_kurs ");
$anzahl = mysql_result($erg_anzahl,0,"summe");
echo "<TR>";
echo "<TD VALIGN=TOP>";
if (mysql_result($result,$i,"krs_kursnummer")!=""){
    echo mysql_result($result,$i,"krs_kursnummer");
}
echo "</TD><TD VALIGN=TOP>";
if (mysql_result($result,$i,"krs_kurzthema")!=""){
    echo mysql_result($result,$i,"krs_kurzthema");
}
echo "</TD><TD VALIGN=TOP>";
echo $anzahl;
echo "</TD><TD VALIGN=TOP>";
if (mysql_result($result,$i,"krs_zielgruppe")!=""){
    echo mysql_result($result,$i,"krs_zielgruppe");
}
echo "</TD><TD VALIGN=TOP>";
if (mysql_result($result,$i,"krs_halbjahr")!=""){
    echo mysql_result($result,$i,"krs_halbjahr");
}
echo "</TD><TD VALIGN=TOP>";
//Zugriffe werden beschränkt
//Admin
if($status>2){
    echo "<a href='kurs.phtml?krs_id=".mysql_result($result,$i,"krs_id");
    echo "&edit=1'>Bearbeiten</a>";
    echo "<BR>";
    echo "<a
href='delete_kurs_action.phtml?krs_id=".mysql_result($result,$i,"krs_id");
    echo "'>Löschen</a>";
    echo "<BR>";
    echo "<a href='kurs_info.phtml?krs_id=".mysql_result($result,$i,"krs_id");
    echo "&edit=1'>Info</a>";
}
//Kursleiter
elseif($status>1){
    echo "<a href='kurs_info.phtml?krs_id=".mysql_result($result,$i,"krs_id");
    echo "&edit=1'>Info</a>";
}
//Teilnehmer
elseif($status>0){
    echo "<a href='kurs_info.phtml?krs_id=".mysql_result($result,$i,"krs_id");
    echo "&edit=1'>Info</a>";
}
else{
    echo "<a href='kurs_info.phtml?krs_id=".mysql_result($result,$i,"krs_id");
    echo "&edit=1'>Info</a>";
}
echo "</TD></TR>";
$i++;
};
echo "</TABLE>";
};
require('../system/fuss.phtml');
?>

```

## Verzeichnis kursleiter

kursleiter.phtml

```

<?
require('../system/kopf.phtml');
if (!isset($edit)){
    $edit=0;
};
if($edit==1){
    $result = mysql($dbdatabase,"select * from kursleiter where ltr_id=$ltr_id");
    $num = mysql_numrows($result);
}
?>

<form action="kursleiter_action.phtml" method=post>
<input type="hidden" name="edit" value="<? echo $edit; ?>">
<input type="hidden" name="ltr_id" value="<? echo $ltr_id; ?>">
<center>
<?
if($edit==1){
    echo "<h2>Kursleiter bearbeiten</h2>";
}else{
    echo "<h2>Neuen Kursleiter eintragen</h2>";
}
?>
<table BORDER=1 CELSPACING=0 CELLPADDING=4 BGCOLOR=#DDDDDD Width="100%">
<tr BGCOLOR=#CCCCCC>
    <td><b>Adresse</b> </td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
</tr>
<tr >
<td>Anrede/Titel </td>
<td><Select name="ltr_anrede">
    <Option Value="Herr">Herr</option>
    <Option Value="Frau">Frau</option>
    <Option Value="Firma">Firma</option>
</SELECT>
</td>
<td></td>
<td><b>Privat</b></td>
<td><b>Geschäftlich</b> </td>
</tr>
<tr BGCOLOR=#CCCCCC>
<td WIDTH="50">Vorname/Nachname</td>
<td>
    <input type="text" name="ltr_vorname" SIZE="15" value="<?
if($edit==1){echo mysql_result($result,0,"ltr_vorname");} ?>">
    <input type="text" name="ltr_nachname" SIZE="15" value="<?
if($edit==1){echo mysql_result($result,0,"ltr_nachname");} ?>">
</td>
<td>Telefon</td>
<td><input type="text" name="ltr_tel_priv" value="<?
if($edit==1){echo mysql_result($result,0,"ltr_tel_priv");} ?>"> </td>
<td><input type="text" name="ltr_tel_dienst" value="<?
if($edit==1){echo mysql_result($result,0,"ltr_tel_dienst");} ?>"> </td>
</tr>
<tr BGCOLOR=#CCCCCC>
<td>Straße </td>
<td><input type="text" name="ltr_strasse" SIZE="34" value="<?
if($edit==1){echo mysql_result($result,0,"ltr_strasse");} ?>"></td>
<td>Fax</td>
<td><input type="text" name="ltr_fax_priv" value="<?
if($edit==1){echo mysql_result($result,0,"ltr_fax_priv");} ?>"></td>
<td><input type="text" name="ltr_fax_dienst" value="<?
if($edit==1){echo mysql_result($result,0,"ltr_fax_dienst");} ?>"></td>

```

```

</tr>
<tr BGCOLOR=#CCCCCC>
  <td>PLZ/Ort </td>
  <td>
    <input type="text" name="ltr_plz" SIZE="5" value="<?
    if($edit==1){echo mysql_result($result,0,"ltr_plz");} ?>"><input type="text"
    name="ltr_ort" SIZE="27" value="<?
    if($edit==1){echo mysql_result($result,0,"ltr_ort");} ?>">
  </td>
  <td>Email</td>
  <td><input type="text" name="ltr_email" value="<?
  if($edit==1){echo mysql_result($result,0,"ltr_email");} ?>"> </td>
  <td></td>
</tr>
<tr BGCOLOR=#DDDDDD>
  <td><b>Loginangaben</b> </td>
  <td>&nbsp; </td>
  <td>&nbsp; </td>
  <td>&nbsp; </td>
  <td>&nbsp; </td>
</tr>
<tr BGCOLOR=#CCCCCC>
  <td> Login</td>
  <td><input type="text" name="ltr_login" value="<?
  if($edit==1){echo mysql_result($result,0,"ltr_login");} ?>"> </td>
  <td>Status</td>
  <td><input type="text" name="ltr_status" value="<?
  if($edit==1){echo mysql_result($result,0,"ltr_status");} ?>"> </td>
  <td></td>
</tr>
<tr BGCOLOR=#CCCCCC>
  <td>Passwort</td>
  <td>
    <? if($edit==1){echo "Passwort kann hier nicht geändert werden!";}else{ ?>
      <input type="text" name="ltr_password">
      <?>;?>
    </td>
  <td></td>
  <td></td>
  <td></td>
</tr>
</table>
<P ALIGN="CENTER">
<input type="submit" name="Button" value="Abschicken">
<input type="reset" name="Button" value="Daten löschen"></p>
</form><BR>
<?
  require('../system/fuss.phtml');
?>

kursleiter_action.phtml
<?php
require ('../system/kopf.phtml');
if ($edit==0){
  $encrypt = md5($ltr_password);
  $result= mysql($dbdatabase,"SELECT * FROM kursleiter WHERE ltr_login LIKE
  '$ltr_login' and ltr_password LIKE '$encrypt'");
  $num = mysql_numrows($result);
  if ($num > 0){
    Print "Dieser Datensatz existiert bereits";
  }
  else {
    $insert = mysql($dbdatabase,"insert into kursleiter(ltr_anrede, ltr_vorname,
    ltr_nachname, ltr_strasse, ltr_plz, ltr_ort, ltr_tel_priv, ltr_tel_dienst,
    ltr_fax_priv, ltr_fax_dienst, ltr_email, ltr_login, ltr_password, ltr_status)
    values ('$ltr_anrede', '$ltr_vorname', '$ltr_nachname', '$ltr_strasse', '$ltr_plz',
    '$ltr_ort', '$ltr_tel_priv', '$ltr_tel_dienst', '$ltr_fax_priv', '$ltr_fax_dienst',
    '$ltr_email', '$ltr_login', '$encrypt', $ltr_status)");
  }
}

```

```

    Print "Neuer Benutzer wurde eingetragen!";
}
}
else if ($edit=1){
    $result= mysql($dbdatabase,"update kursleiter set ltr_anrede='$ltr_anrede',
ltr_vorname='$ltr_vorname', ltr_nachname='$ltr_nachname',
ltr_strasse='$ltr_strasse', ltr_plz='$ltr_plz', ltr_ort='$ltr_ort',
ltr_tel_priv='$ltr_tel_priv', ltr_tel_dienst='$ltr_tel_dienst',
ltr_fax_priv='$ltr_fax_priv', ltr_fax_dienst='$ltr_fax_dienst',
ltr_email='$ltr_email',ltr_login='$ltr_login', ltr_status=$ltr_status where
ltr_id=$ltr_id");
    Print "Benutzer wurde geändert!";
}
require ('../system/fuss.phtml');
?>

delete_kursleiter.phtml
<?
    require('../system/kopf.phtml');
    $result = mysql($dbdatabase,"delete from kursleiter where ltr_id=$ltr_id");
?>
Gelöscht

<?
    require('../system/fuss.phtml');
?>

show_kursleiter.phtml
<?
    require('../system/kopf.phtml');
    $result = mysql($dbdatabase,"select * from kursleiter");
    $num = mysql_numrows($result);
?>
<h2>Kursleiter anzeigen</h2>
<?
    $i=0;
    if ($num ==0){
        echo "<FONT SIZE=2 FACE=Verdana,Arial>Keine Kursleiter eingetragen</FONT>";
    }
    else {
        echo "<TABLE BORDER=0 WIDTH=100%>";
        echo"<TR>
<TH VALIGN=TOP width=200 align=left>Name</TH>
<TH width=100 align=left>Login</TH>
<TH width=10 align=left>Status</TH>
<TH align=left>Aktion</TH>
</TR>";
        while($i<$num){
            echo "<TR>";
            echo "<TD VALIGN=TOP>";
            if (mysql_result($result,$i,"ltr_nachname")!=""){
                echo mysql_result($result,$i,"ltr_nachname").",
".mysql_result($result,$i,"ltr_vorname");
            }
            echo "</TD><TD VALIGN=TOP>";
            if (mysql_result($result,$i,"ltr_login")!=""){
                echo mysql_result($result,$i,"ltr_login");
            }
            echo "</TD><TD VALIGN=TOP>";
            if (mysql_result($result,$i,"ltr_status")!=""){
                echo mysql_result($result,$i,"ltr_status");
            }
            echo "</TD><TD VALIGN=TOP>";
        }
        echo "</TR>";
    }
?>
<a href="kursleiter.phtml?ltr_id=
<? echo mysql_result($result,$i,"ltr_id"); ?>
&edit=1">Bearbeiten</a><BR>
<a href="delete_kursleiter_action.phtml?ltr_id=

```

```
<? echo mysql_result($result,$i,"ltr_id"); ?>
">Löschen</a>
<?
    echo "</TD></TR>";
    $i++;
};
echo "</TABLE>";
};

require('../system/fuss.phtml');
?>
```

## Verzeichnis kursort

kursort.phtml

```

<?
require('../system/kopf.phtml');
$schule = mysql($dbdatabase,"select * from schule");
$anz_schule = mysql_numrows($schule);
if (!isset($edit)){
    $edit=0;
};
if($edit==1){
    $result = mysql($dbdatabase,"select * from kursort where krt_id=$krt_id");
    $num = mysql_numrows($result);
}
?>
<h2>Kursort
<?
if($edit==1){
    echo "bearbeiten";
}else{
    echo "eingeben";
}
?>
</h2>
<form action="kursort_action.phtml" method="Post">
<input type="hidden" name="edit" value="<? echo $edit; ?>">
<input type="hidden" name="krt_id" value="<? echo $krt_id; ?>">
<table >
<tr>
<td valign="TOP">Raum </td>
<td><input type="Text" name="krt_raum" value="<?
if($edit==1){echo mysql_result($result,0,"krt_raum");} ?>"> </td>
</tr>
<tr>
<td valign="TOP">Ausstattung </td>
<td><textarea name="krt_ausstattung" cols="50" rows="10"><? if($edit==1){echo
mysql_result($result,0,"krt_ausstattung");}?></textarea></td>
</tr>
<tr>
<td valign="TOP">Schulnummer </td>
<td><select name="krt_schulnummer">
<?
if($edit==1){
    $akt_schule=mysql_result($result,0,"krt_schulnummer");
    $erg_schule = mysql($dbdatabase,"select * from schule where
sch_schulnummer=$akt_schule");
    echo "<option value='".mysql_result($erg_schule,0,"sch_schulnummer")."'
selected'>".mysql_result($erg_schule,0,"sch_schulnummer")."
".mysql_result($erg_schule,0,"sch_schulname");
}
$zi=0;
while($zi<$anz_schule){
    echo "<option
value='".mysql_result($schule,$zi,"sch_schulnummer")."'>".mysql_result($schule,$zi,"s
ch_schulnummer")." ".mysql_result($schule,$zi,"sch_schulname");
    $zi++;
}
?>
</select>

</td>
</tr>
<tr>
<td colspan="2" align="center"><input type="Submit" name="einfuegen"
value="Einfügen"> </td>
</tr>
</table>
</form>

```



```

<?
require('../system/fuss.phtml');
?>

kursort_action.phtml
<?
require('../system/kopf.phtml');
if ($edit==0){
$result = mysql($dbdatabase,"insert into
kursort(krt_raum,krt_ausstattung,krt_schulnummer)values('$krt_raum',
'$krt_ausstattung', $krt_schulnummer)");
}
else if($edit==1){
$result=mysql($dbdatabase,"update kursort set krt_raum='$krt_raum',
krt_ausstattung='$krt_ausstattung',krt_schulnummer=$krt_schulnummer where
krt_id=$krt_id");
}
?>
eingetragen

<?
require('../system/fuss.phtml');
?>

delete_kursort_action.phtml
<?
require('../system/kopf.phtml');
$result = mysql($dbdatabase,"delete from kursort where krt_id=$krt_id");
?>
Gelöscht

<?
require('../system/fuss.phtml');
?>

show_kursort.phtml
<?
require('../system/kopf.phtml');
$result = mysql($dbdatabase,"select * from kursort");
$num = mysql_numrows($result);
?>
<h2>Kursorte anzeigen</h2>
<?
$i=0;
if ($num ==0){
echo "<FONT SIZE=2 FACE=Verdana,Arial>Keine Kursorte eingetragen</FONT>";
}
else {
echo "<TABLE BORDER=0 WIDTH=100%>";
echo"<TR>
<TH VALIGN=TOP width=50 align=left>Raum</TH>
<TH width=200 align=left>Ausstattung</TH>
<TH width=20 align=left>Schulnummer</TH>
<TH align=left>Aktion</TH>
</TR>";
while($i<$num){
echo "<TR>";
echo "<TD VALIGN=TOP>";
if (mysql_result($result,$i,"krt_raum")!=""){
echo mysql_result($result,$i,"krt_raum");
}
echo "</TD><TD VALIGN=TOP>";
if (mysql_result($result,$i,"krt_ausstattung")!=""){
echo mysql_result($result,$i,"krt_ausstattung");
}
echo "</TD><TD VALIGN=TOP>";
if (mysql_result($result,$i,"krt_schulnummer")!=""){
echo mysql_result($result,$i,"krt_schulnummer");
}
}
}

```

```
        }
        echo "</TD><TD VALIGN=TOP>";
    ?>
    <a href="kursort.phtml?krt_id=
    <? echo mysql_result($result,$i,"krt_id"); ?>
    &edit=1">Bearbeiten</a><BR>
    <a href="delete_kursort_action.phtml?krt_id=
    <? echo mysql_result($result,$i,"krt_id"); ?>
    ">Löschen</a>
    <?
        echo "</TD></TR>";
        $i++;
    };
    echo "</TABLE>";
};

require('../system/fuss.phtml');
?>
```

## Verzeichnis schule

schule.phtml

```

<?
require('../system/kopf.phtml');
if (!isset($edit)){
    $edit=0;
};
if($edit==1){
    $result = mysql($dbdatabase,"select * from schule where sch_id=$sch_id");
    $num = mysql_numrows($result);
}
?>

<form action="schule_action.phtml" method=post>
<input type="hidden" name="edit" value="<? echo $edit; ?>">
<input type="hidden" name="sch_id" value="<? echo $sch_id; ?>">
<center>
<?
if($edit==1){
    echo "<h2>Schule bearbeiten</h2>";
}else{
    echo "<h2>Neue Schule eintragen</h2>";
}
?>

<table BORDER=0 CELSPACING=0 CELLPADDING=4 BGCOLOR=#DDDDDD Width="100%">
<tr BGCOLOR=#CCCCCC>
<td><b>Adresse</b></td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
</tr>
<tr >
<td>Schulform</td>
<td><input type="text" name="sch_schulform" SIZE="22" value="<?
if($edit==1){echo mysql_result($result,0,"sch_schulform");} ?>">
</td>
<td>Rechtstellung</td>
<td><input type="text" name="sch_rechtsstellung" SIZE="15" value="<?
if($edit==1){echo mysql_result($result,0,"sch_rechtsstellung");} ?>"></td>
<td>Träger</td>
<td><input type="text" name="sch_schultraeger" SIZE="15" value="<?
if($edit==1){echo mysql_result($result,0,"sch_schultraeger");} ?>"></td>
</tr>
<tr BGCOLOR=#CCCCCC>
<td WIDTH="50">Nummer/Name</td>
<td>
    <input type="text" name="sch_schulnummer" SIZE="5" value="<?
if($edit==1){echo mysql_result($result,0,"sch_schulnummer");} ?>"><input
type="text" name="sch_schulname" SIZE="15" value="<?
if($edit==1){echo mysql_result($result,0,"sch_schulname");} ?>">
</td>
<td>Tel. priv.</td>
<td><input type="text" name="sch_tel_priv" SIZE="15" value="<?
if($edit==1){echo mysql_result($result,0,"sch_tel_priv");} ?>"> </td>
<td>Tel. dienst.</td>
<td><input type="text" name="sch_tel_dienst" SIZE="15" value="<?
if($edit==1){echo mysql_result($result,0,"sch_tel_dienst");} ?>"> </td>
</tr>
<tr BGCOLOR=#CCCCCC>
<td>Straße </td>
<td><input type="text" name="sch_strasse" SIZE="22" value="<?
if($edit==1){echo mysql_result($result,0,"sch_strasse");} ?>"></td>
<td>Fax priv.</td>
<td><input type="text" name="sch_fax_priv" SIZE="15" value="<?

```

```

if($edit==1){echo mysql_result($result,0,"sch_fax_priv");} ?>"></td>
<td>Fax dienst.</td>
<td><input type="text" name="sch_fax_dienst" SIZE="15" value="<?
if($edit==1){echo mysql_result($result,0,"sch_fax_dienst");} ?>"></td>
</tr>
<tr BGCOLOR=#CCCCCC>
<td>PLZ/Ort </td>
<td>
<input type="text" name="sch_plz" SIZE="5" value="<?
if($edit==1){echo mysql_result($result,0,"sch_plz");} ?>"><input type="text"
name="sch_ort" SIZE="15" value="<?
if($edit==1){echo mysql_result($result,0,"sch_ort");} ?>">
</td>
<td>Email</td>
<td><input type="text" name="sch_email" SIZE="15" value="<?
if($edit==1){echo mysql_result($result,0,"sch_email");} ?>"> </td>
<td>Lehrer</td>
<td><input type="text" name="sch_lehreranzahl" SIZE="15" value="<?
if($edit==1){echo mysql_result($result,0,"sch_lehreranzahl");} ?>"></td>
</tr>
<tr BGCOLOR=#DDDDDD>
<td><b>Schulleiter</b> </td>
<td>&nbsp; </td>
<td>&nbsp; </td>
<td>&nbsp; </td>
<td>&nbsp; </td>
<td>&nbsp; </td>
</tr>
<tr BGCOLOR=#CCCCCC>
<td>Anrede</td>
<td><Select name="sch_anrede_schulleiter">
<Option Value="Herr">Herr</option>
<Option Value="Frau">Frau</option>
<Option Value="Firma">Firma</option>
</SELECT></td>
<td>Vorname</td>
<td><input type="text" name="sch_vorname_schulleiter" SIZE="15" value="<?
if($edit==1){echo mysql_result($result,0,"sch_vorname_schulleiter");} ?>"> </td>
<td>Nachname</td>
<td><input type="text" name="sch_nachname_schulleiter" SIZE="15" value="<?
if($edit==1){echo mysql_result($result,0,"sch_nachname_schulleiter");} ?>"></td>
</tr>
</table>
<P ALIGN="CENTER">
<input type="submit" name="Button" value="Abschicken">
<input type="reset" name="Button" value="Daten löschen"></p>
</form><BR>

```

```

<?
require('../system/fuss.phtml');
?>

```

```

schule_action.phtml
<?php
require('../system/kopf.phtml');
if ($edit==0){
    $insert = mysql($dbdatabase,"insert into schule(sch_schulnummer, sch_schulname,
sch_schulform, sch_rechtsstellung, sch_schultraeger, sch_strasse, sch_plz, sch_ort,
sch_tel_priv, sch_tel_dienst, sch_fax_priv, sch_fax_dienst, sch_email,
sch_lehreranzahl, sch_anrede_schulleiter, sch_vorname_schulleiter,
sch_nachname_schulleiter) values ($sch_schulnummer, '$sch_schulname',
'$sch_schulform', '$sch_rechtsstellung', '$sch_schultraeger', '$sch_strasse',
'$sch_plz', '$sch_ort', '$sch_tel_priv', '$sch_tel_dienst', '$sch_fax_priv',
'$sch_fax_dienst', '$sch_email', '$sch_lehreranzahl', '$sch_anrede_schulleiter',
'$sch_vorname_schulleiter', '$sch_nachname_schulleiter')");
    Print "Neue Schule wurde eingetragen!";
}
else if ($edit=1){

```

```

    $result= mysql($dbdatabase,"update schule set sch_schulnummer=$sch_schulnummer,
sch_schulname='$sch_schulname', sch_schulform='$sch_schulform',
sch_rechtsstellung='$sch_rechtsstellung', sch_schultraeger='$sch_schultraeger',
sch_strasse='$sch_strasse', sch_plz='$sch_plz', sch_ort='$sch_ort',
sch_tel_priv='$sch_tel_priv', sch_tel_dienst='$sch_tel_dienst',
sch_fax_priv='$sch_fax_priv', sch_fax_dienst='$sch_fax_dienst',
sch_email='$sch_email',sch_lehreranzahl=$sch_lehreranzahl,
sch_anrede_schulleiter='$sch_anrede_schulleiter',
sch_vorname_schulleiter='$sch_vorname_schulleiter',
sch_nachname_schulleiter='$sch_nachname_schulleiter' where sch_id=$sch_id");
    Print "Schule wurde geändert!";
}
require ('../system/fuss.phtml');
?>

delete_schule_action.phtml
<?
    require('../system/kopf.phtml');
    $result = mysql($dbdatabase,"delete from schule where sch_id=$sch_id");
?>
Gelöscht

<?
    require('../system/fuss.phtml');
?>

show_schule.phtml
<?
    require('../system/kopf.phtml');
    $result = mysql($dbdatabase,"select * from schule");
    $num = mysql_numrows($result);
?>
<h2>schule anzeigen</h2>
<?
$i=0;
if ($num ==0){
    echo "<FONT SIZE=2 FACE=Verdana,Arial>Keine Schule eingetragen</FONT>";
}
else {
    echo "<TABLE BORDER=0 WIDTH=100%>";
    echo"<TR>
<TH VALIGN=TOP width=20 align=left>Nummer</TH>
<TH width=200 align=left>Name</TH>
<TH width=100 align=left>Email</TH>
<TH align=left>Aktion</TH>
</TR>";
    while($i<$num){
        echo "<TR>";
        echo "<TD VALIGN=TOP>";
        if (mysql_result($result,$i,"sch_schulnummer")!= ""){
            echo mysql_result($result,$i,"sch_schulnummer");
        }
        echo "</TD><TD VALIGN=TOP>";
        if (mysql_result($result,$i,"sch_schulname")!= ""){
            echo mysql_result($result,$i,"sch_schulname");
        }
        echo "</TD><TD VALIGN=TOP>";
        if (mysql_result($result,$i,"sch_email")!= ""){
            echo mysql_result($result,$i,"sch_email");
        }
        echo "</TD><TD VALIGN=TOP>";
    }
}
?>
<a href="schule.phtml?sched_id=
<? echo mysql_result($result,$i,"sch_id"); ?>
&edit=1">Bearbeiten</a><BR>
<a href="delete_schule_action.phtml?sched_id=
<? echo mysql_result($result,$i,"sch_id"); ?>
">Löschen</a>

```

```
<?
    echo "</TD></TR>";
    $i++;
};
echo "</TABLE>";
};

require('../system/fuss.phtml');
?>
```

## Verzeichnis teilnehmer

teilnehmer.phtml

```

<?
require('../system/kopf.phtml');
$schule = mysql($dbdatabase,"select * from schule");
$anz_schule = mysql_numrows($schule);
if (!isset($edit)){
    $edit=0;
};
if($edit==1){
    $result = mysql($dbdatabase,"select * from teilnehmer where tnr_id=$tnr_id");
    $num = mysql_numrows($result);
}
?>
<SCRIPT LANGUAGE="JavaScript">
function check_data(){
    var vorname = window.document.sendform.tnr_vorname.value;
    var nachname = window.document.sendform.tnr_nachname.value;
    var strasse = window.document.sendform.tnr_strasse.value;
    var plz = window.document.sendform.tnr_plz.value;
    var ort = window.document.sendform.tnr_ort.value;
    var email = window.document.sendform.tnr_email.value;
    var re_email = /^[_a-zA-Z0-9-]+\.[_a-zA-Z0-9-]+@[a-zA-Z0-9-]+\.[a-zA-Z]{2,3}$/;
    var checked = true;
    var adresse = true;

    if((re_email.test(email)==false || email.length<1)){
        if(vorname.length<1){
            alert ("Das Feld Vorname muss ausgefüllt sein");
            checked = false;
            adresse = false;
        }
        if(nachname.length<1){
            alert ("Das Feld Nachname muss ausgefüllt sein");
            checked = false;
            adresse = false;
        }
        if(strasse.length<1){
            alert ("Das Feld Straße muss ausgefüllt sein");
            checked = false;
            adresse = false;
        }
        if(plz.length<1){
            alert ("Das Feld Postleitzahl muss ausgefüllt sein");
            checked = false;
            adresse = false;
        }
        if(ort.length<1){
            alert ("Das Feld Ort muss ausgefüllt sein");
            checked = false;
            adresse = false;
        }
        if(adresse == false){
            if((re_email.test(email)==false || email.length<1)){
                alert ("Das Feld Email enthält einen Fehler (entspricht nicht den Regeln) oder keinen Eintrag");
                checked = false;
            }
        }
        return checked;
    }
}
</script>
<form name="sendform" action="teilnehmer_action.phtml" method=post>
<input type="hidden" name="edit" value="<? echo $edit; ?>">
<input type="hidden" name="tnr_id" value="<? echo $tnr_id; ?>">

```

```

<center>
<?
if($edit==1){
    echo "<h2>Teilnehmer bearbeiten</h2>";
}else{
echo "<h2>Neuen Teilnehmer eintragen</h2>";
}
?>

<table BORDER=0 CELSPACING=0 CELLPADDING=4 BGOLOR=#DDDDDD Width="100%">
<tr BGOLOR=#CCCCCC>
    <td><b>Adresse</b> </td>
    <td>&nbsp; </td>
    <td>&nbsp; </td>
    <td>&nbsp; </td>
</tr>
<tr >
<td>Anrede/Titel </td>
    <td><Select name="tnr_anrede">
        <Option Value="Herr">Herr</option>
        <Option Value="Frau">Frau</option>
        <Option Value="Firma">Firma</option>
    </SELECT>
    </td>
    <td></td>
    <td><b>Privat</b></td>
</tr>
<tr BGOLOR=#CCCCCC>
<td WIDTH="50">Vorname/Nachname</td>
    <td>
        <input type="text" name="tnr_vorname" SIZE="15" value="<?
if($edit==1){echo mysql_result($result,0,"tnr_vorname");} ?>">
        <input type="text" name="tnr_nachname" SIZE="15" value="<?
if($edit==1){echo mysql_result($result,0,"tnr_nachname");} ?>">
    </td>
    <td>Telefon</td>
    <td><input type="text" name="tnr_telefon" value="<?
if($edit==1){echo mysql_result($result,0,"tnr_telefon");} ?>"> </td>
</tr>
<tr BGOLOR=#CCCCCC>
    <td>Straße </td>
    <td><input type="text" name="tnr_strasse" SIZE="34" value="<?
if($edit==1){echo mysql_result($result,0,"tnr_strasse");} ?>"></td>
    <td>Fax</td>
    <td><input type="text" name="tnr_fax" value="<?
if($edit==1){echo mysql_result($result,0,"tnr_fax");} ?>"></td>
</tr>
<tr BGOLOR=#CCCCCC>
    <td>PLZ/Ort </td>
    <td>
        <input type="text" name="tnr_plz" SIZE="5" value="<?
if($edit==1){echo mysql_result($result,0,"tnr_plz");} ?>"><input type="text"
name="tnr_ort" SIZE="27" value="<?
if($edit==1){echo mysql_result($result,0,"tnr_ort");} ?>">
    </td>
    <td>Email</td>
    <td><input type="text" name="tnr_email" value="<?
if($edit==1){echo mysql_result($result,0,"tnr_email");} ?>"> </td>
</tr>
<tr BGOLOR=#DDDDDD>
    <td><b>Schul- und Loginangaben</b> </td>
    <td>&nbsp; </td>
    <td>&nbsp; </td>
    <td>&nbsp; </td>
</tr>
<tr BGOLOR=#CCCCCC>
    <td>Fächer</td>

```



```

        <td><input type="text" name="tnr_lehramt" value="<?
if($edit==1){echo mysql_result($result,0,"tnr_lehramt");} ?>"> </td>
        <td>Login</td>
        <td><input type="text" name="tnr_login" value="<?
if($edit==1){echo mysql_result($result,0,"tnr_login");} ?>"></td>
    </tr>
    <tr BGCOLOR=#CCCCCC>
        <td>Schulnummer</td>
        <td>
            <select name="tnr_schulnummer">
            <?
            if($edit==1){
                $akt_schule=mysql_result($result,0,"tnr_schulnummer");
                $erg_schule = mysql($dbdatabase,"select * from schule where
sch_schulnummer=$akt_schule");
                echo "<option value='".mysql_result($erg_schule,0,"sch_schulnummer")."'
selected'>".mysql_result($erg_schule,0,"sch_schulnummer")."
".mysql_result($erg_schule,0,"sch_schulname");
            }
            $i=0;
            while($i<$anz_schule){
                echo "<option
value='".mysql_result($schule,$i,"sch_schulnummer")."'>".mysql_result($schule,$i,"s
ch_schulnummer")." ".mysql_result($schule,$i,"sch_schulname");
                $i++;
            }
            ?>
            </select>
        </td>
        <td>Passwort</td>
        <td><? if($edit==1){echo "Passwort kann hier nicht geändert werden!";}else{ ?>
            <input type="text" name="tnr_password">
            <?>>?>
        </tr>
    </table>
    <P ALIGN="CENTER">
    <input type="submit" name="Button" value="Abschicken" onclick="return
check_data()">
    <input type="reset" name="Button" value="Daten löschen"></p>
</form><BR>

<?
require('../system/fuss.phtml');
?>

teilnehmer_action.phtml
<?php
require ('../system/kopf.phtml');
if ($edit==0){
    $encrypt = md5($tnr_password);
    $result= mysql($dbdatabase,"SELECT * FROM teilnehmer WHERE tnr_login LIKE
'$tnr_login' and tnr_password LIKE '$encrypt'");
    $num = mysql_numrows($result);
    if ($num > 0){
        Print "Dieser Datensatz existiert bereits";
    }
    else {
        $insert = mysql($dbdatabase,"insert into teilnehmer(tnr_anrede, tnr_vorname,
tnr_nachname, tnr_strasse, tnr_plz, tnr_ort, tnr_telefon, tnr_fax, tnr_email,
tnr_lehramt, tnr_schulnummer, tnr_login, tnr_password) values ('$tnr_anrede',
'$tnr_vorname', '$tnr_nachname', '$tnr_strasse', '$tnr_plz', '$tnr_ort',
'$tnr_telefon', '$tnr_fax', '$tnr_email','$tnr_lehramt','$tnr_schulnummer,
'$tnr_login', '$encrypt')");
        Print "Neuer Teilnehmer wurde eingetragen!";
    }
}
else if ($edit=1){

```

```

    $result= mysql($dbdatabase,"update teilnehmer set tnr_anrede='$tnr_anrede',
tnr_vorname='$tnr_vorname', tnr_nachname='$tnr_nachname',
tnr_strasse='$tnr_strasse', tnr_plz='$tnr_plz', tnr_ort='$tnr_ort',
tnr_telefon='$tnr_telefon', tnr_fax='$tnr_fax',
tnr_email='$tnr_email',tnr_lehramt='$tnr_lehramt',
tnr_schulnummer=$tnr_schulnummer, tnr_login='$tnr_login' where tnr_id=$tnr_id");
    Print "Teilnehmer wurde geändert!";
}
require ('../system/fuss.phtml');
?>

delete_teilnehmer_action.phtml
<?
    require('../system/kopf.phtml');
    $result = mysql($dbdatabase,"delete from teilnehmer_kurs where tnr_id=$tnr_id");
    $result = mysql($dbdatabase,"delete from teilnehmer where tnr_id=$tnr_id");
?>
Gelöscht

<?
    require('../system/fuss.phtml');
?>

show_teilnehmer.phtml
<?
    require('../system/kopf.phtml');
    $result = mysql($dbdatabase,"select * from teilnehmer order by tnr_schulnummer,
tnr_nachname");
    $num = mysql_numrows($result);
?>
<h2>Teilnehmer anzeigen</h2>
<?
$i=0;
if ($num ==0){
    echo "<FONT SIZE=2 FACE=Verdana,Arial>Keine Teilnehmer eingetragen</FONT>";
}
else {
    echo "<TABLE BORDER=0 WIDTH=100%>";
    echo"<TR>
<TH VALIGN=TOP width=100 align=left>Name</TH>
<TH width=100 align=left>Lehramt</TH>
<TH width=10 align=left>Email</TH>
<TH width=10 align=left>Sch.Nr.</TH>
<TH align=left>Aktion</TH>
</TR>";
    while($i<$num){
        echo "<TR>";
        echo "<TD VALIGN=TOP>";
        if (mysql_result($result,$i,"tnr_nachname")!=""){
            echo mysql_result($result,$i,"tnr_nachname").",
".mysql_result($result,$i,"tnr_vorname");
        }
        echo "</TD><TD VALIGN=TOP>";
        if (mysql_result($result,$i,"tnr_lehramt")!=""){
            echo mysql_result($result,$i,"tnr_lehramt");
        }
        echo "</TD><TD VALIGN=TOP>";
        if (mysql_result($result,$i,"tnr_email")!=""){
            echo mysql_result($result,$i,"tnr_email");
        }
        echo "</TD><TD VALIGN=TOP>";
        if (mysql_result($result,$i,"tnr_schulnummer")!=""){
            echo mysql_result($result,$i,"tnr_schulnummer");
        }
        echo "</TD><TD VALIGN=TOP>";
    }
?>
<a href="teilnehmer.phtml?tnr_id=
<? echo mysql_result($result,$i,"tnr_id"); ?>

```

```

&edit=1">Bearbeiten</a>|
<a href="delete_teilnehmer_action.phtml?tnr_id=
<? echo mysql_result($result,$i,"tnr_id"); ?>
">Löschen</a>
|
<a href="teilnehmer_login.phtml?tnr_id=
<? echo mysql_result($result,$i,"tnr_id"); ?>
">Passwort</a>
|
<a href="teilnehmer_kurs.phtml?tnr_id=
<? echo mysql_result($result,$i,"tnr_id"); ?>
">Kurse</a>
<?
    echo "</TD></TR>";
    $i++;
};
echo "</TABLE>";
};

require('../system/fuss.phtml');
?>

teilnehmer_kurs.phtml
<?
    require('../system/kopf.phtml');
    $result = mysql($dbdatabase,"select krs_id as akt_kurs from teilnehmer_kurs where
tnr_id=$tnr_id");
    $num = mysql_numrows($result);
?>
<h2>Kurse der Teilnehmer anzeigen</h2>
<?
$i=0;
if ($num ==0){
    echo "<FONT SIZE=2 FACE=Verdana,Arial>Keine Kurse eingetragen</FONT>";
}
else {
    echo "<TABLE BORDER=0 WIDTH=100%>";
    echo"<TR>
<TH VALIGN=TOP width=10 align=left>Nr.</TH>
<TH width=200 align=left>Kurzthema</TH>
<TH width=50 align=left>Halbjahr</TH>
<TH width=100 align=left>Aktion</TH>
</TR>";
    while($i<$num){
        $akt_kurs=mysql_result($result,$i,"akt_kurs");
        $erg_kurs = mysql($dbdatabase,"select * from kurs where krs_id=$akt_kurs");
        echo "<TR>";
        echo "<TD VALIGN=TOP>";
        if (mysql_result($erg_kurs,0,"krs_kursnummer")!=""){
            echo mysql_result($erg_kurs,0,"krs_kursnummer");
        }
        echo "</TD><TD VALIGN=TOP>";
        if (mysql_result($erg_kurs,0,"krs_kurzthema")!=""){
            echo mysql_result($erg_kurs,0,"krs_kurzthema");
        }
        echo "</TD><TD VALIGN=TOP>";
        if (mysql_result($erg_kurs,0,"krs_halbjahr")!=""){
            echo mysql_result($erg_kurs,0,"krs_halbjahr");
        }
        echo "</TD><TD VALIGN=TOP>";
        if ($status==3){
            $krs_id=mysql_result($erg_kurs,0,"krs_id");
            echo "<a
href='../kurs/kurs_zuweisung_delete.phtml?tnr_id=$tnr_id&krs_id=$krs_id&loesche=kon
kret'>Abmelden</a>";
        };
        echo "</TD></TR>";
    }
}

```

```

        $i++;
    }
    echo "</table>";
}
require('../system/fuss.phtml');
?>

teilnehmer_login.phtml
<?
    require('../system/kopf.phtml');
?>
<form action="teilnehmer_login_action.phtml" method=post>
<input type="hidden" name="tnr_id" value="<? echo $tnr_id; ?>">
<center>
<h2>Passwort und Login ändern</h2>
<p>
<table border=0>
    <tr>
        <td>Neues Login</td>
        <td><input type=text size=20 name="neu_login"></td>
    </tr>
    <tr>
        <td>Neues Passwort</td>
        <td><input type=password size=20 name="neu_password"></td>
    </tr>
</table>
<input type=submit value="Eintragen">

<?
    require('../system/fuss.phtml');
?>

teilnehmer_login_action.phtml
<?php
require ('../system/kopf.phtml');
$encrypt=md5($neu_password);
$result= mysql($dbdatabase,"update teilnehmer set tnr_password='$encrypt',
tnr_login='$neu_login' where tnr_id=$tnr_id");
Print "Passwort wurde geändert!";
echo "<BR>";
echo $tnr_id;
echo "<BR>";
echo $neu_password;
echo "<BR>";
echo $neu_login;
echo "<BR>";
echo $encrypt;

require ('../system/fuss.phtml');
?>

Verzeichnis useradmin

user.phtml
<?
    require('../system/kopf.phtml');
?>
<form action="user_action.phtml" method=post>
<center>
<h2>Passwort ändern</h2>
<p>
<table border=0>
    <tr>
        <td>Altes Passwort</td>
        <td><input type=password size=20 name="password1"></td>
    </tr>
    <tr>
        <td>Altes Passwort</td>

```

```

        <td><input type=password size=20 name="password2"></td>
    </tr>
    <tr>
        <td>Neues Passwort:</td>
        <td><input type=password size=20 name="password"></td>
    </tr>
</table>
<input type=submit value="Eintragen">

<?
    require('../system/fuss.phtml');
?>

user_action.phtml
<?php
require ('../system/kopf.phtml');
if ($status==1){
    $result = mysql($dbdatabase,"select tnr_password as password from teilnehmer
where tnr_id=$id");
    $num = mysql_numrows($result);
}
else if($status==2){
    $result = mysql($dbdatabase,"select ltr_password as password from kursleiter
where ltr_id=$id");
    $num = mysql_numrows($result);
}
else if($status==3){
    $result = mysql($dbdatabase,"select ltr_password as password from kursleiter
where ltr_id=$id");
    $num = mysql_numrows($result);
}
if($num==0){
echo "Falsches Passwort";
}
else{
    $old_password=mysql_result($result,0,"password");
    if(($password1==$password2) AND ($old_password==md5($password1))){
        $encrypt=md5($password);
        if ($status==1){
            $result= mysql($dbdatabase,"update teilnehmer set
tnr_password='$encrypt'where tnr_id=$id");
        }
        else{
            $result= mysql($dbdatabase,"update kursleiter set
ltr_password='$encrypt'where ltr_id=$id");
        }
        Print "Passwort wurde geändert!";
    }
}
require ('../system/fuss.phtml');
?>

delete_user_action.phtml
<?
    require('../system/kopf.phtml');
    $result = mysql($dbdatabase,"delete from kursleiter where ltr_id=$ltr_id");
?>
Gelöscht

<?
    require('../system/fuss.phtml');
?>

show_user.phtml
<?
    require('../system/kopf.phtml');
    $result = mysql($dbdatabase,"select * from kursleiter");
    $num = mysql_numrows($result);

```

```

?>
<h2>Kursleiter anzeigen</h2>
<?
$i=0;
if ($num ==0){
    echo "<FONT SIZE=2 FACE=Verdana,Arial>Keine Kursleiter eingetragen</FONT>";
}
else {
    echo "<TABLE BORDER=0 WIDTH=100%>";
    echo"<TR>
    <TH VALIGN=TOP width=200 align=left>Name</TH>
    <TH width=100 align=left>Login</TH>
    <TH width=10 align=left>Status</TH>
    <TH align=left>Aktion</TH>
    </TR>";
    while($i<$num){
        echo "<TR>";
        echo "<TD VALIGN=TOP>";
        if (mysql_result($result,$i,"ltr_nachname")!=""){
            echo mysql_result($result,$i,"ltr_nachname").",
".mysql_result($result,$i,"ltr_vorname");
        }
        echo "</TD><TD VALIGN=TOP>";
        if (mysql_result($result,$i,"ltr_login")!=""){
            echo mysql_result($result,$i,"ltr_login");
        }
        echo "</TD><TD VALIGN=TOP>";
        if (mysql_result($result,$i,"ltr_status")!=""){
            echo mysql_result($result,$i,"ltr_status");
        }
        echo "</TD><TD VALIGN=TOP>";
    }
    ?>
    <a href="user.phtml?ltr_id=
    <? echo mysql_result($result,$i,"ltr_id"); ?>
    &edit=1">Bearbeiten</a><BR>
    <a href="delete_user_action.phtml?ltr_id=
    <? echo mysql_result($result,$i,"ltr_id"); ?>
    ">Löschen</a>
    <?
        echo "</TD></TR>";
        $i++;
    };
    echo "</TABLE>";
};

require('../system/fuss.phtml');
?>

```

Verzeichnis system  
kopf.phtml

```

<?
require('../common.phtml');
require('../system/session.phtml');
$conn=mysql_connect($dbserver, $dbuser, $dbpasswd);
?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<link rel="stylesheet" type="text/css" href="../system/stylesheet.css">

</head>
<body background="../grafik/background.jpg">
<Table border=0 width=100%>
<tr>
<td width=100 valign=top>&nbsp; </td>
<TD valign=top align=center>

```

```
stylesheet.css
body{
    font-family:Verdana;
    font-size:6pt;
}
table{
    empty-cells:show;
}
th{
    background-color:#909090;
}
a:link { font-weight:bold; color:#E06000; text-decoration:none }
a:visited { font-weight:bold; color:#800000; text-decoration:none }
a:hover { font-weight:bold; color:#0000A0; text-decoration:none }
a:active { font-weight:bold; color:#E00000; text-decoration:underline }
a:focus { font-weight:bold; color:#00E000; text-decoration:underline }
```

Ich versichere, diese Arbeit selbständig verfasst, keine anderen als die angegebenen Hilfsmittel verwendet und die Stellen, die in anderen Werken im Wortlaut, als Grafik oder dem Sinne nach entnommen sind, mit Quellenangaben kenntlich gemacht zu haben.

---

Alexander Hauptmann