



**BÜCHERRAUSCH.DE**

## **UMSETZUNG EINES BÜCHER-WEBSHOPS UNTER VERWENDUNG VON LINUX, APACHE, MYSQL UND PHP**

Projektarbeit von Steffen Krauß, Dominik Morio, Matthias Trauth  
@FSM 02



## Inhaltverzeichnis

Seite 3	Vorwort
Seite 4, 5	Teammitglieder
Seite 6	Für und wider E-Commerce
Seite 7	Herangehensweise Projekt
Seite 8	Entity Relation-Modell
Seite 9	Apache
Seite 10	MySQL
Seite 11	Installation MySQL
Seite 16	phpMyAdmin
Seite 22	Rechte
Seite 23	Datenbankspezifische Rechte
Seite 24	HTTP
Seite 25	HTTPS
Seite 26	HTML
Seite 27	Dreamweaver
Seite 27	Gestaltung und Aufbau eines Webshops
Seite 29	PHP
Seite 34	Abschließendes Fazit
Seite 35	Webshop
Seite 37	Quellenangaben
Seite 38	Abschließende Erklärung



## Vorwort, Erwartungen an das Projekt:

### **„Umsetzung eines Bücher-Webshops unter Verwendung von Linux, Apache, MySQL und PHP.“**

Für uns war es die Projektbeschreibung mit dem größten Anreiz – ausschlaggebend war die Einbeziehung und die praktische Anwendungsmöglichkeit verschiedener Elemente, die wir im Unterricht des Moduls „Multimediale Arbeitsplätze in Datennetzen“ (MMADN) behandelten. Ferner erhofften wir uns durch die Realisierung des Projekts die Möglichkeit, ein wenig hinter die Kulissen der Anwendungen zu blicken, welche uns im alltäglichen Umgang mit der Internet-Nutzung umgeben.

Unsere größte Herausforderung bei der Realisierung des Projekts mit der für uns komplexen Aufgabenstellung war die Herangehensweise (s. dazu S.XX) – Berufserfahrungen und technische, bzw. programmierspezifische Kenntnisse lagen uns nicht vor.

Doch **„...der Weg ist das Ziel“**.

Nach Befassung mit der Materie und einer Phase der Ideenfindung nahm das Projekt Gestalt an und wir entschieden uns für die themenspezifischen Projektierung:

**Konzeption und Gestaltung einer Internetseite mit Webshop-Anbindung für den Bereich Buch und Literatur unter Verwendung eines Apache-Webservers und MySQL mit Einbeziehung spezifischer Komponenten.**

Unsere Ausarbeitung soll einen kurzen Ein- und Überblick über den Aufbau unseres Projekts geben, Erarbeitetes transparent machen und die Vorgehensweise einer datenbankgestützten Anwendung und die damit verbundenen Komponenten und Spezifikationen erläutern.

Neustadt an der Weinstraße,  
Mai 2005

Steffen Krauß  
Dominik Morio  
Matthias Trauth



## Teammitglieder:

Unser Projektteam setzte sich aus Steffen Krauß, Dominik Morio und Matthias Trauth zusammen. Nachfolgend eine Kurzvita und die Rollen- bzw. Aufgabenverteilung der einzelnen Personen in Bezug auf unser Projekt.

### **Steffen Krauß**

Jahrgang 1976  
Wohnhaft in Kirrweiler/Pfalz,  
Ausbildung zum Druckformhersteller

**Beruflich eingesetzt in den Bereichen:**  
CTP-Druckvorstufe und Datenübernahme

### **Projektbezogene Arbeitsbereiche:**

Fotografie der eingebundenen Produkte  
Formulierung der produktspezifischen Beschreibungstexte  
Einrichtung und Konfiguration des Apache-Webservers  
Datenbankstrukturierung  
Erstellung der Datenbanken unter Verwendung von PhpMyAdmin

### **Für Dokumentation mit Schwerpunkten:**

MySQL-Dokumentation  
phpMyAdmin-Dokumentation

### **Dominik Morio**

Jahrgang 1978  
Wohnhaft in Germersheim am Rhein,  
Ausbildung zum Schriftsetzer

### **Beruflich eingesetzt in den Bereichen:**

Kundenberatung  
Datenübernahme  
Projektrealisation  
Digitaldruck  
Schnittstelle Druckvorstufe >< Druck

### **Projektbezogene Arbeitsbereiche:**

Einrichtung und Konfiguration des Apache-Webservers  
Datenbankstrukturierung  
Definition und Entwicklung der benötigten PHP-Skripte  
Durchführung der PHP-Abfragen mit MySQL  
Einbinden der PHP-Skripte in HTML

### **Für Dokumentation mit Schwerpunkten:**

PHP-Dokumentation



### **Matthias Trauth**

Jahrgang 1981

Wohnhaft in Herxheim bei Landau,

Ausbildung zum Mediengestalter, Fachrichtung Mediendesign

#### **Beruflich eingesetzt in den Bereichen:**

Kundenberatung

Datenübernahme

Grafik- und Kommunikationsdesign

Digitaldruck

Schnittstelle Druckvorstufe >< Druck

#### **Projektbezogene Arbeitsbereiche:**

Strukturierung des Projektablaufs

(Definition von Themenblöcken und Arbeitspaketen)

Datenbankstrukturierung

Gestaltung der verwendeten Elemente (Logo, Internetseite, Ausarbeitung)

Aufbau der Internetseite in HTML

Einbinden der PHP-Skripte in HTML

#### **Für Dokumentation mit Schwerpunkten:**

Apache-Dokumentation

HTML-Dokumentation

Gestaltung der Dokumentation



## Einleitung, Für und wider e-Commerce:

Die Projektierung verdeutlicht die Bedeutung und das Vorkommen von Internetanwendungen mit Webshop-Anbindung (e-Commerce) im medialen Zeitalter. Viele Unternehmen nutzen diesen zusätzlichen Distributionsweg für ein breiter gefächertes Absatzspektrum Ihrer Produkte.

### +

Der e-Commerce bietet dem Unternehmen einige, wesentliche Vorteile gegenüber dem konventionellen Verkauf und Vertrieb von Produkten. Einsparung von Verkaufsräumen und Reduzierung von Verkaufspersonal (bei Firmen, die sich auf e-Commerce spezialisieren), Präsenz rund um die Uhr, einfache Aktualisierung und Präsentation neuer Produkte. Für den Kunden ist die bequeme Art und Weise, „online zu bummeln und zu kaufen“ ein gewichtiger Faktor – höhere Angebotsvielfalt, direkte Onlinepreisvergleiche und die Möglichkeit der direkten Produkthanlieferung stehen dem Medium ebenfalls positiv zu Buche.

Selbstverständlich gehört es in einer objektiven Ausarbeitung auch dazu, die Kehrseite einer Medaille aufzuzeigen:

### -

Das Anbieten, der Verkauf und Vertrieb von Produkten über das Internet führt auch zu personellen Einsparungen, was aus Arbeitnehmersicht negativ zu Buche steht. Die Möglichkeit des „Online-Shoppings“ „fördert“ ferner das Aussterben von Geschäften und dem Einzelhandel und somit die Verarmung von Stadtzentren. Durch direkten Vergleich, sowohl produkt- als auch preis-technisch, sehen sich kleinere Firmen oftmals im Abseits gegenüber Firmen mit einer größeren Produktpalette und günstigeren Konditionen.

In Bezug auf unser Projekt könnte man sich folgendes vor Augen führen: Ein gesuchtes Buch wird in Amazon schnell gefunden und auch gleich bestellt. Der örtliche Buchhändler (regisitriert dieses Kaufverhalten) geht leer aus. Beratung und Kommunikation sind sekundär.

Wie diese Entwicklung in beiden Richtungen gehen wird, bleibt abzuwarten. **Abschließend jedoch sind wir unabhängig der geschilderten Vor- und Nachteile der Ansicht, dass eine Synergie zwischen konventionellem Vertrieb und der Anwendung von Webshops große Vorteile für Verkäufer und Verbraucher erzielen kann.**



## Herangehensweise Projekt:

Nach der Wahl des Projekts und einem kurzen Austausch (teamintern) wurden Themenblöcke definiert und Phasen für das Projekt festgelegt. Daraus ergibt sich folgende Struktur und Projektablauf:

### Phase 0:

Themenwahl („Konzeption und Gestaltung einer Internetseite mit Webshop-Anbindung für den Bereich Buch und Literatur unter Verwendung eines Apache-Webservers und MySQL mit Einbeziehung spezifischer Komponenten“).

### Phase 1:

- Allgemeines Brainstorming bzgl. Thematik
- Ordnen der Brainstormingpunkte
- Wahl des Webshop-Anwendungsbereiches (Buch und Literatur)
- Allgemeines Selbststudium über Themenbereiche

### Phase 2:

- Festlegung der Projektpunkte
- Definition von Themenblöcken
- Verteilung der Themen im Team
- Persönliche Befassung mit der Materie jeweils gewählter Themenbereiche
- Definition von Arbeitspaketen  
(individuell je Teammitglied, persönliche Erläuterung zu Arbeitsschritten)
- Zeitplanung
- Rückkopplung mit Modullehrer Herr Alexander Scheib über Themenwahl, Tiefe der Projektrealisation, geplante Vorgehensweise, geforderte Themeninhalte

### Phase 3:

- Datenbankaufbau- und Datenbankinhalte
- Datenbankstruktur
- Einrichtung und Konfiguration des Webservers und der benötigten Softwarepakete
- Konzeption und Design der grafischen Benutzeroberfläche
- Seitenaufbau nach Konzeptvorlage in HTML
- Datenbankerstellung und Strukturierung
- Benutzerrechte und Rechtevergabe
- PHP-Skriptentwicklung
- Testphasen und Feinabstimmung
- Dokumentationsplanung und Dokumentationsvorbereitung

#### Phase 4:

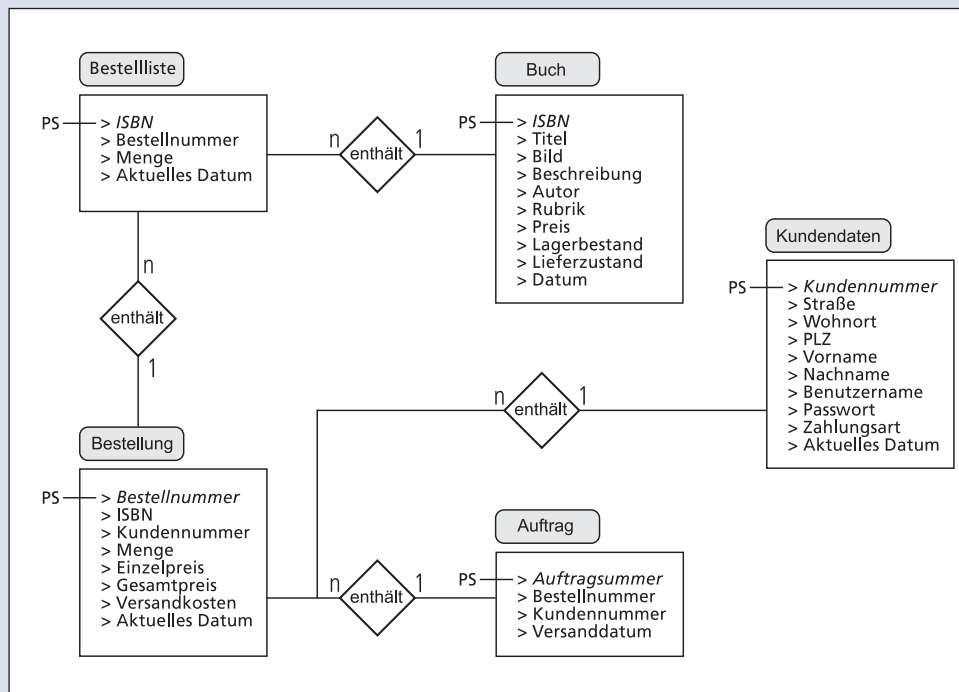
- Prüfung des Designs und der Funktionalität der Internetanwendung (visuell-technisch)
- Dokumentationserstellung
- Dokumentationsprüfung
- Präsentationsplanung und Erstellung
- Überprüfung und Wirkung von Umsetzung der Anwendung im Gesamten

#### Info:

In die Phasen wurde jeweils Herr Scheib mit eingebunden, um ein Feedback über Geplantes zu erhalten bzw. konstruktive Kritik aufzunehmen. Diese Form des Austausches und die Art der Zwischenbriefings haben uns gewährleistet, die Erwartungen und die Zielsetzung des Lehres einschätzen zu können.

## ER-Modell

### (Entity Relation Modell)

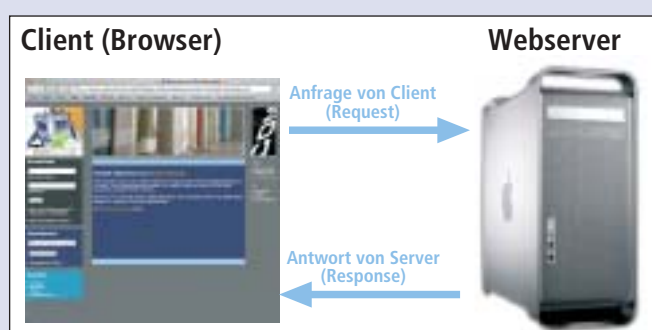




# Apache

## Zuerst einmal - was ist ein Webserver?

Generell kann man sich einen Webserver als physisch vorhandenen Server (Rechner) vorstellen, der auf Anfrage eines Clients Daten zur Verfügung stellt. Die Anfrage (Request) vom Client zum Webserver erfolgt über eine HTTP-URL, beispielsweise durch Anklicken eines Hyperlinks. Die angeforderten Daten, die der Webserver über das HTTP-Protokoll an den Client sendet, sind HTML-Seiten, die der Browser interpretiert. Diese HTML-Seiten sind auf dem (zuvor freigegebenen Verzeichnis) Webserver statisch abgelegt oder können dynamisch auf eine Anfrage generiert werden (durch den Einsatz von PHP, ASP, PERL oder CGI-Skripten). Dann spricht man von dynamischen Seiten.



## Apache HTTP Webserver

Der Apache HTTP Webserver ist der meistverbreiteste Webserver im Internet und kommt mit einem Anteil von über 65% noch vor Microsofts IIS (Internet Information Center) oder etwa Servern aus

dem Hause Sun. Das Produkt der Apache Software Foundation ist Open-Source-Software, d.h. der Programm-code ist frei verfügbar – was Basis zur stetigen Verbesserung und Optimierung ist. Apache läuft unter Unix und Linux, doch auch unter Windows, Netware oder anderen Betriebssystemen.

## Der Apache HTTP Webserver ist modular aufgebaut:

Module, die man sich auch als „Aktivitäten“ vorstellen kann, sind z.B. die Verschlüsselung der Kommunikation zwischen Browser und Webserver (mod\_ssl) oder die Verwendung als Proxy-Server (mod\_proxy), um nur einige zu nennen. Je nach Bedarf können unterschiedliche Module eingebunden werden. Der Apache HTTP Webserver bietet die Möglichkeit, dynamisch Webseiten zu generieren mit Hilfe serverseitiger Skriptsprachen wie PHP, ASP oder PERL (wie im obigen Text schon genannt). In unserem Projekt werden die durch Abfragen dynamisch generierten Seiten mit PHP realisiert (welches im HTML-Code eingebunden ist, serverseitig interpretiert wird und die Ergebnisse über HTML an den Client wieder zurücksendet).

### Info:

Das Verzeichnis der Webseite muss auf dem Rechner des Webserver drei elementare Verzeichnisse aufweisen:

- **config:** enthält die Config.-Datei; diese teilt dem Apache mit, wie auf verschiedene Anfragen zu reagieren ist (die Hauptkonfigurationsdatei, um den Apache zu konfigurieren, ist die httpd.config).
- **htdocs:** enthält die Dokumente wie HTML-Seiten, Bilder etc.
- **logs:** enthält die Logdaten zur Aktivitätsprotokollierung



# MySQL

Zur Erstellung unserer Datenbank haben wir uns für MySQL entschieden. MySQL stellt die am meisten verbreitete Datenbanksoftware da. Ferner ist MySQL, wie der Apache auch, Open Source. Das heißt, sie kann frei genutzt werden, ist für jedermann zugänglich und jeder der die Möglichkeit hat, kann sie auch weiterentwickeln.

## Begriffserklärung

MySQL > Der Ursprung des My ist nicht genau beschrieben, ist aber auch nicht weiter relevant. SQL bedeutet Structured Query Language. Was soviel heißt wie Strukturierte Abfragesprache.

## Woher kommt MySQL?

MySQL stammt von dem Unternehmen MySQL AB dessen Ursprung in Schweden liegt. Zur Beschreibung der Firma kann man sagen, dass sie sich über die ganze Welt erstreckt. Die Firma selbst setzt sich aus Entwicklern zusammen, welche an der Software arbeiten und sie weiterentwickeln. Zur Philosophie der Firma gehört es, MySQL ständig weiterzuentwickeln und weiterhin zu verbreiten.

## Als Unternehmensziele nennt MySQL AB:

1. Die beste und meistgenutzte Datenbank der Welt zu entwickeln
2. MySQL für alle verfügbar zu machen
3. Eine einfache Benutzung zu erreichen
4. MySQL kontinuierlich zu verbessern

## Was ist eine Datenbank?

Eine Datenbank stellt eine Sammlung von Daten dar. Die Art der Daten kann beliebig sein, bspw. Text oder Bilder. Die Datenmenge einer Datenbank ist sehr unterschiedlich, so gibt es auch Datenbanken in riesigen Dimensionen mit mehreren Gigabyte.

## Datenbankmanagementsystems DBMS

Da eine Datenbank nun ein Pool aus möglichen Daten darstellt, kann dies sehr unübersichtlich werden. Ziel der Datenbank ist es schließlich auch, dass verschiedene Benutzer auf einzelne Daten zugreifen können um Daten auslesen. Hier kommt nun MySQL zum Einsatz. Man bezeichnet MySQL als Relationales Datenbanksystem. Nun, was zeichnet dies aus? Diese Art der Datenbank zeichnet sich dahingehend aus, dass die Daten nicht einfach gesammelt werden, sondern in separaten Tabellen abgelegt werden. Somit ist eine strukturierte Abfrage möglich. Die einzelnen Tabellen sind weiterhin nicht isoliert voneinander, sondern stellen untereinander Verbindungen her. Dies ermöglicht eine Abfrage über die ganze Datenbank hinweg und den einzelnen Tabellen.



## Installation von MySQL

### rpm

rpm (RedHat Packet Manager) installiert und verwaltet Softwarepakete. Der RedHat Packet Manager rpm installiert neue Softwarepakete im RPM-Format auf einem Linux-System oder entfernt vorher installierte Pakete vollständig.

rpm stellt bei der Installation neuer Pakete die Integrität der bereits installierten Pakete sicher. rpm überprüft vor der Installation eines neuen Pakets die durch Abhängigkeit von anderen Paketen gegebenen Voraussetzungen. Beim Update eines installierten Pakets auf eine neuere Version werden die Konfigurationsdateien des alten Pakets automatisch gesichert.

Unter Linux wird empfohlen eine Installation mit einer RPM-Datei durchzuführen. Diese Art der Installation lässt es zu, einzelne Bausteine zu installieren. So gibt es separate RPM-Dateien für eine Client-Version oder eine Server-Version. Weiter gibt es komplette Pakete zum Installieren die alle Bestandteile beinhalten.

Unter [www.mysql.com](http://www.mysql.com) findet man Installationspakete für viele verschiedene Plattformen.

In unserem Fall haben wir unter YAST vorhandene Pakete mit MySQL gesucht und diese installiert.

Will man manuell unter [Shell](#) (Konsole Linux) eine RPM-Datei installieren, so sieht dies folgendermaßen aus: `rpm -i MySQLversion.i386.rpm  
MySQLclient version.rpm`

Durch die Installation wird ein Verzeichnis angelegt in dem Daten der Datenbank abgelegt werden, bspw. die Datenbank-Datei und Benutzertabellen.

Nun ist MySQL Datenbank schon einsatzfähig. In den folgenden Abschnitten erfährt man grundlegende Funktionen zum Arbeiten mit MySQL. Zum Einen auf Kommandozeilenebene und zweitens „benutzerfreundlicher“ mit phpMyAdmin.

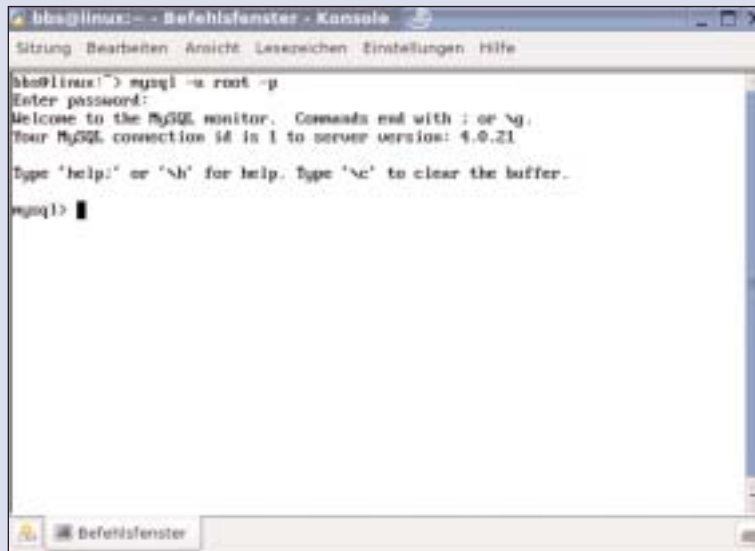
## Arbeiten mit MySQL auf Kommandozeilenebene

Wird MySQL zum ersten Mal aufgerufen, bekommt man den Hinweis, dass noch kein Benutzer angelegt worden ist. Standardmäßig existiert ein Benutzer ohne Passwort, somit ist es ratsam als erstes einen Benutzer mit allen Rechten und Passwort anzulegen.

Das Erstellen des Benutzers sieht folgendermaßen aus:

Hier wird unterschieden ob lokal zugegriffen wird oder über das Netzwerk. Bei lokalem Zugriff wird die Hostnummer nicht benötigt.

Wird die Hostnummer benötigt sieht die Eingabe folgendermaßen aus:



```
bls@linux:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1 to server version: 4.0.21

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

Shell:

`mysql -h Hostnummer -u root -p`

Enter password:

Bei einigen MySQL-Versionen ist es nun auch möglich lokal MySQL ohne Passwort zu starten. Einfach mit der Eingabe: **Shell: mysql**

### Der Befehl

Ein Befehl besteht aus einem MySQL Statement und einem Semikolon, der den Befehl sozusagen abschließt. Hierbei gibt es ein paar Ausnahmen, wie bei dem Befehl **Quit**, der kein Semikolon benötigt. Um weitere Informationen zu Befehlen zu bekommen, kann man sich mit **help** weiterhelfen. Werden mehrere Statements hintereinander geschrieben, so müssen diese mit Semikolon getrennt werden. Groß- und Kleinschreibung wird hier nicht berücksichtigt. Befehle können auch über mehrere Zeilen geschrieben werden:

```
mysql > Select
      -> User
      -> From usw.
```

Der Pfeil -> gibt an, dass MySQL auf einen weiteren Eintrag wartet, der Befehl wird erst mit dem Semikolon als beendet erkannt. Möchte man einen Befehl abbrechen, so geht dies ganz einfach mit folgender Eingabe: `-> \c`

## Datenbank erstellen

Um am Anfang festzustellen, welche Datenbanken vorhanden sind, kann man sich diese mit **Show** anzeigen lassen.

Der Befehl um unsere Datenbank **buch** zu erstellen, sieht so aus:

```
mysql > create database buch;
```

Der Befehl besteht aus dem Teil, der MySQL sagt „erstelle mir einen Datenbank“ und dem

Anhängen des gewünschten Namens der Datenbank. Um nun auch mit dieser Datenbank arbeiten zu können, das heißt sie mit Inhalt zu füllen und Abfragen auszuführen, muss diese sozusagen angesprochen werden.

Es genügt also nicht, eine Datenbank zu erstellen, sondern man muss MySQL explizit mitteilen, welche Datenbank man benutzen will. Dies geschieht mit dem Befehl **use**.


Eine weitere Möglichkeit ist es, direkt bei der Anmeldung in MySQL die gewünschte Datenbank anzugeben. Dies geht folgendermaßen:

```
shell> mysql -h Hostnummer -u user -p buch  
Enter password:
```

Somit kann man mit der Datenbank **buch** arbeiten.



```
bb@linux:~ - Befehlsfenster - Konsole  
Sitzung Bearbeiten Ansicht Leerezeichen Einstellungen Hilfe  
bb@linux:~$ mysql -u root -p  
Enter password:  
Welcome to the MySQL monitor. Commands end with ; or ^g.  
Your MySQL connection id is 64 to server version: 4.0.21  
Type 'help;' or '^k' for help. Type '^c' to clear the buffer.  
mysql> show databases;  
+-----+  
| Database |  
+-----+  
| buch     |  
| mysql   |  
+-----+  
2 rows in set (0.00 sec)  
mysql>
```



```
bb@linux:~ - Befehlsfenster - Konsole  
Sitzung Bearbeiten Ansicht Leerezeichen Einstellungen Hilfe  
bb@linux:~$ mysql -u root -p  
Enter password:  
Welcome to the MySQL monitor. Commands end with ; or ^g.  
Your MySQL connection id is 69 to server version: 4.0.21  
Type 'help;' or '^k' for help. Type '^c' to clear the buffer.  
mysql> use buch;  
Reading table information for completion of table and column names  
You can turn off this feature to get a quicker startup with -B  
Database changed  
mysql>
```

## Tabellen

Der nächste Schritt ist nun, die Datenbank zu füllen. Als Grundlage zur Erstellung dient das vorher verfasste ER-Modell (s. Seite XX), in welchem alle Inhalte definiert sind – welche Tabellen benötigt werden, wie die Felder bezeichnet werden und über welche Werte eine Verbindung unter den Tabellen hergestellt wird, den sogenannten **Primärschlüssel**.

Der Primärschlüssel gibt einen festen Wert an, der sich nicht verändert. Durch ihn wird eine genaue Abfrage ermöglicht.

Mit **use** hat man nun die Datenbank **buch** ausgewählt. Um nun festzustellen ob eventuell schon Tabellen existieren, kann man sich diese auch wieder zeigen lassen mit **show**.

Um eine Tabelle in der Datenbank **buch** zu erzeugen, wird folgendermaßen geschrieben:



```
mysql> mysql
mysql> show tables;
ERROR 1046: No database selected
mysql> use buch;
Database changed
mysql> show tables;
+-----+
| Tables_in_buch |
+-----+
| auftrag        |
| bestellliste   |
| bestellung     |
| buch           |
| kundenaten    |
+-----+
5 rows in set (0.00 sec)

mysql>
```

```
mysql> Create Table buch
(Titel Varchar(30), Preis Float (0.00), Beschreibung Text, Autor Varchar (30),
Lieferzustand Text);
```

Hier ein paar Erklärungen zu verschiedenen Datentypen:

**Varchar** = Eine variable Zeichenkette mit einer Länge von 1 bis 4000

**Char** = Feste Zeichenkette mit einer Länge von 1 bis 2000

**Date** = Datumsangabe

**Long** = Variable Zeichenkette bis 2GB

**Blob** = Binärdaten bis 4GB

**Integer** = Ganzzahl

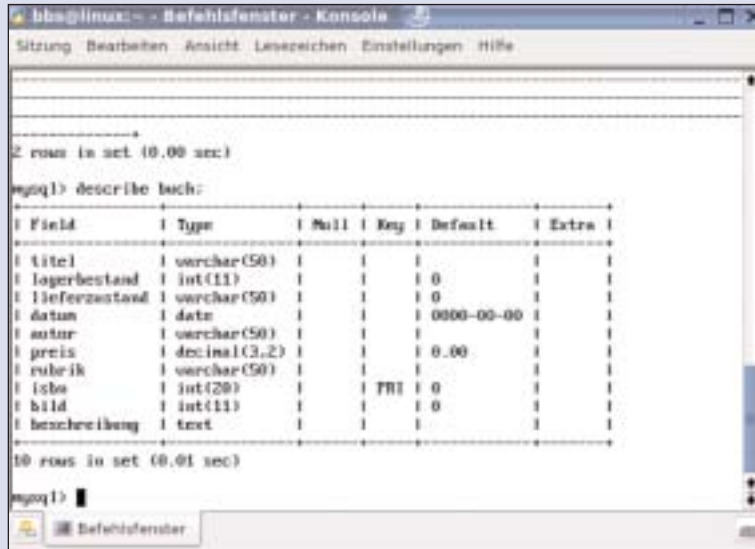
**Float** = Fließkommazahl

**Text** = erlaubt Eingabe variablen Texts

Um sich nun die Tabelle `buch` anzeigen zu lassen, wird der Befehl `describe` verwendet:

In der Ausgabe erhält man die Tabelle in gezeigter Ansicht und sieht die einzelnen Spaltennamen, deren Typ, Schlüssel usw.

Als Ergebnis ist nun eine Datenbank `buch` mit einer Tabelle `buch` vorhanden.



```
mysql> describe buch;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| title | varchar(50) | | | | |
| lagerbestand | int(11) | | | 0 | |
| lieferzustand | varchar(50) | | | 0 | |
| datum | date | | | 0000-00-00 | |
| autor | varchar(50) | | | | |
| preis | decimal(3,2) | | | 0.00 | |
| rubrik | varchar(50) | | | | |
| isbn | int(20) | | | 0 | |
| bild | int(11) | | | 0 | |
| beschreibung | text | | | | |
+-----+-----+-----+-----+-----+-----+
10 rows in set (0.01 sec)
```

Als nächster Schritt muss die Tabelle mit Datensätzen gefüllt werden. Die Daten eines Buchs in die Tabelle `buch` zu schreiben sieht nun folgendermaßen aus:

```
mysql> insert buch values
('Der Hobbit','20','In einem Loch im Boden da lebte ein Hobbit...'),'J.R.R
Tolkien','Lieferbar innerhalb 14 Tagen');
```

Zu beachten ist hierbei, dass die Reihenfolge der Feldinhalte mit denen der vorher festgelegten Feldnamen übereinstimmt.

### Abfrage von Inhalten

`Select * From`

Zur Durchführung einer Datenbankabfrage wird SQL benötigt. Hierbei benötigt man den Befehl `select`. Mit diesem Befehl wird eine Abfrage verlangt. Es muss noch angegeben werden, wo diese ausgeführt werden soll, dies geschieht mit dem Zusatz `from`. Um sich z. B. alle Titel der Tabelle `buch` anzeigen zu lassen, sieht die Eingabe folgendermaßen aus:

```
select titel from buch;
Where
```

Um eine spezielle Anfrage durchzuführen wird der Zusatz `where` benutzt. Dieser lässt es zu, eine spezielle Option anzugeben, wie z. B. „welches buch teurer ist als 15,00“. Die Befehlszeile sieht dann so aus:

```
select * From buch Where preis>15
```



## phpMyAdmin

Bis jetzt wurde nur mit MySQL auf Kommandozeilenebene gearbeitet. Um eine einfachere Bedienung von Datenbanken zu erreichen, gibt es die Software phpMyAdmin. Diese kann ebenfalls (wie Apache und MySQL) kostenlos aus dem Internet geladen werden unter [www.phpmyadmin.net](http://www.phpmyadmin.net).

### Was ist phpMyAdmin?

phpMyAdmin stellt ein Programm dar, das komplett in PHP geschrieben ist und zur Aufgabe hat, MySQL Datenbanken zu verwalten.

### Installation

Nach dem Download des phpMyAdmin-Pakets muss dieses entpackt werden. Wichtig hierbei ist dass der Ordner im richtigen Verzeichnis steht.

Unter Linux ist dies: `/srv/www/htdocs/phpmyadmin`

Damit PhpMyAdmin auch einsatzfähig ist, müssen in der `config.inc.php` Datei einige Einstellungen geändert werden. Da das Programm auf MySQL zugreift, muss der Benutzer jeweils in der `config.inc.php Datei` und in MySQL angelegt werden. Benutzer können direkt in MySQL oder auch über phpMyAdmin angelegt werden und mit speziellen Rechten ausgestattet werden. Die wichtigsten Einstellungen, damit phpMyAdmin läuft, sind folgende:

> Datei: `config.inc.php`

```
$cfg[¥Servers`][$i][`host`]= "localhost"
```

Hier wird der Hostname oder die IP hinterlegt

```
$cfg[¥Servers`][$i][`host`] [¥auth_type`= „http“
```

muss http sein, damit auch von anderen Rechnern per http zugegriffen werden kann

```
$cfg[¥Servers`][$i][`host`] [¥user`= „root“
```

```
$cfg[¥Servers`][$i][`host`] [¥password`= `bbsnw`
```



## Arbeiten mit phpMyAdmin

phpMyAdmin lässt sich einfach über den Browser starten.

Als URL muss eingegeben werden: <http://localhost/phpMyAdmin>

Anstatt der Bezeichnung localhost ist natürlich auch die Eingabe der IP des eigenen Rechners möglich: <http://127.0.0.1/phpMyAdmin>

Wichtig hierbei ist dass die Schreibweise von phpMyAdmin in der URL mit der des Ordners unter htdocs übereinstimmt, ansonsten findet der Browser diesen

nicht. Ist ein Benutzer mit Passwort angelegt, so wird hier Benutzer und Passwort abgefragt, um überhaupt mit PhpMyAdmin arbeiten zu können. Das Startfenster von PhpMyAdmin sieht folgendermaßen aus -->



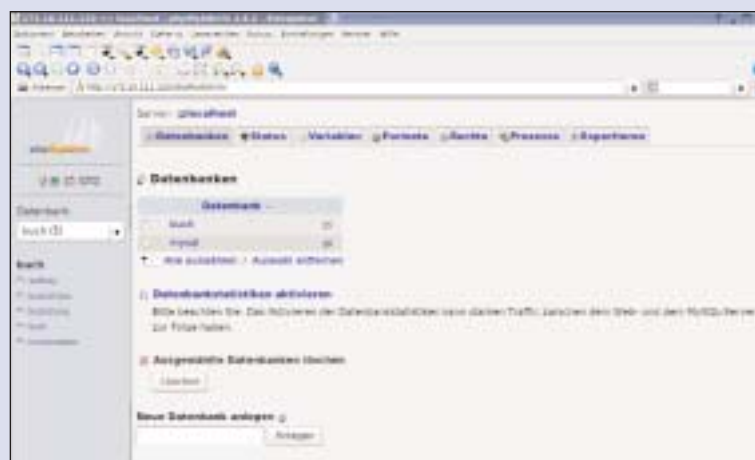
Links befindet sich eine Navigationsleiste, mit der es möglich ist, zwischen verschiedenen Datenbanken auszuwählen.

Im rechten Bereich gibt es zwei Bereiche:

Zum einen den Bereich MySQL, zum anderen den Bereich phpMyAdmin.

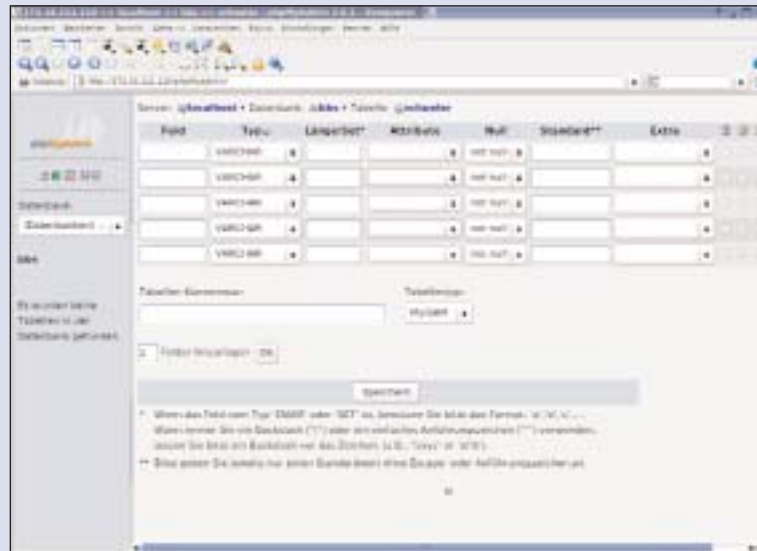
Unter dem Bereich „Rechte“ können nun weitere Benutzer angelegt und Rechte vergeben werden. Dazu später mehr.

Als erstes starten wir mit dem Anlegen einer neuen Datenbank **buch**. Hierzu gibt man unter **Neue Datenbank anlegen** den gewünschten Namen an. Diese kann daraufhin unter Datenbanken angewählt werden.



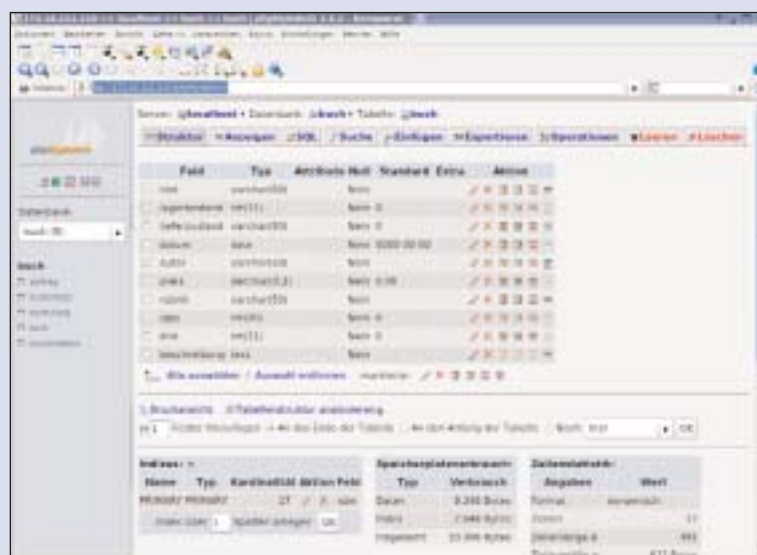
Wie in MySQL auch, muss die Datenbank mit den Tabellen ausgestattet werden die im ER-Modell erarbeitet wurden. Der Tabelle können nun wieder Felder (Spalten) zugewiesen werden. Im Nachhinein ist es auch möglich, eine Tabelle um Felder zu erweitern oder zu verringern.

Die unbearbeitete Tabelle sieht so aus:

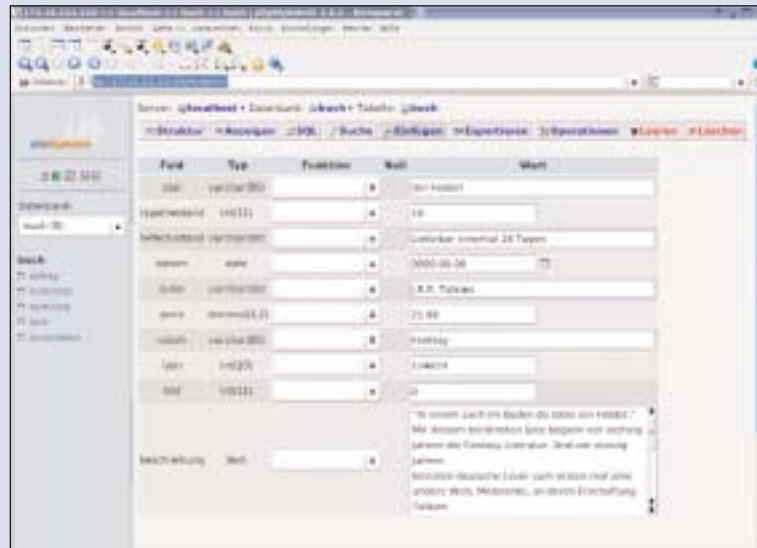


Wie in MySQL wird für jedes Feld eine Bezeichnung eingetragen, der Datentyp festgelegt und welches Feld den Primärschlüssel darstellt, um eine Verbindung zu anderen Tabellen herstellen zu können.

Die Struktur einer Tabelle sieht dann so aus:

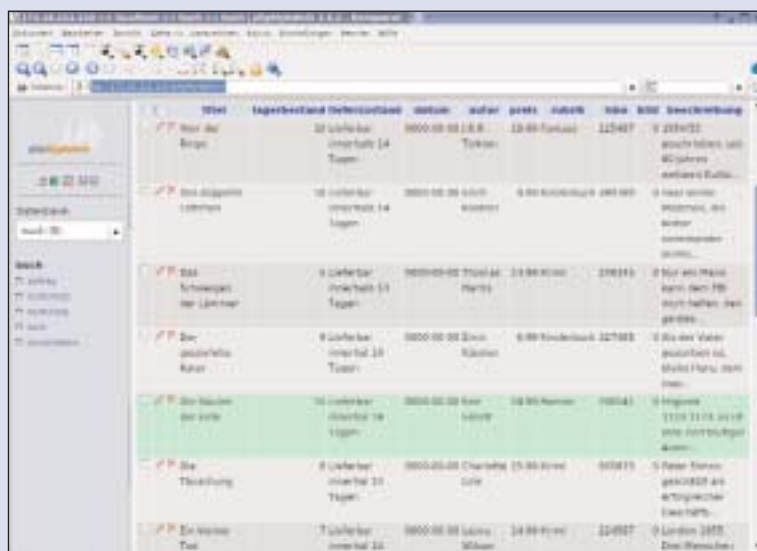


Hier ist es natürlich weiterhin möglich die Tabellen zu bearbeiten. Wählt man einen Tabelleninhalt aus, um diesen zu bearbeiten, oder will man einen neuen Inhalt in die Tabelle einfügen, so sieht das Fenster folgendermaßen aus. Man sieht die Feldnamen, die Datentypen der Felder und unter dem Bereich „Wert“ ist es nun möglich, Inhalte einzugeben oder zu ändern.

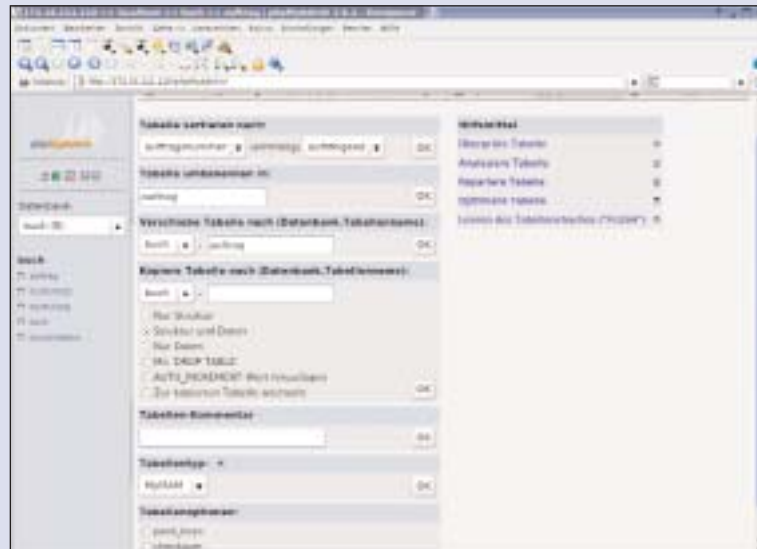


Sind alle Inhalte erstellt, so kann man sich unter „Anzeigen“ den Tabelleninhalt anzeigen lassen.

Eine Gesamtübersicht über den Inhalt wird hier dargestellt.



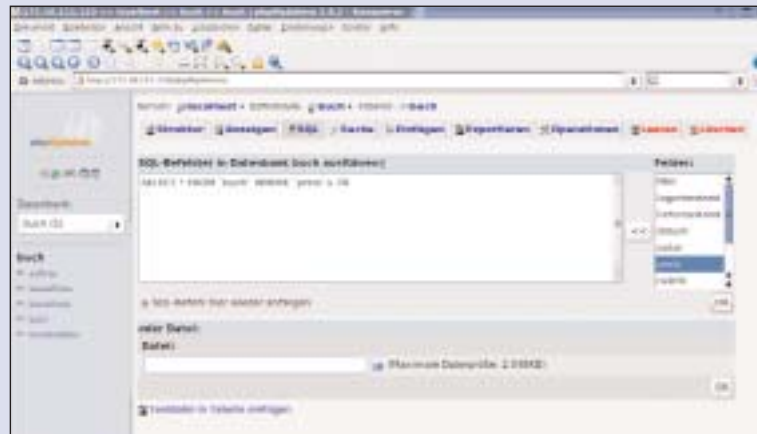
Der Bereich „Operationen“ lässt verschiedene Aktionen zu. Tabellennamen können geändert werden oder nach Inhalten sortiert werden. Tabellen können auf verschiedene Weise in andere Datenbanken kopiert werden.



In der „Struktur“ wird der Inhalt der angewählten Datenbank angezeigt. Einzelne Tabellen werden hier angezeigt. Unter „Aktionen“ können die Tabellen bearbeitet werden oder angezeigt werden. Weiter ist zu erkennen, ob schon Inhalte in die Tabellen eingetragen wurden und wieviele. Auch die Datengröße der einzelnen Tabellen kann hier abgelesen werden.



Nun ist eine Datenbank mit mehreren Tabellen angelegt. Gefüllt ist die Tabelle **buch** mit mehreren Büchern. Über das Feld „SQL“ ist hier eine Abfrage möglich. Der Befehl ist identisch mit dem MySQL-Befehl. Im rechten Bereich sind die einzelnen Felder anzuwählen, die in der Tabelle angelegt wurden. Somit kann eine gezielte, feldbezogene Abfrage durchgeführt werden.



Unter dem Punkt „Exportieren“ erhält man mehrere Möglichkeiten, eine Datenbank zu exportieren – entweder nur als SQL oder auch der Export in verschiedenen Programmen ist möglich, wie bspw. Microsoft Excel oder Microsoft Word.



## Rechte

Ein weiterer wichtiger Punkt ist das Verwalten von Benutzern und deren Rechte (auf die Datenbank bezogen). Wie schon zuvor erwähnt, befindet sich auf der Startseite von phpMyAdmin der Punkt „Rechte“, unter welchem neue Benutzer angelegt werden. Geschieht dies, so muss gleichzeitig festgelegt werden, welche Rechte der Benutzer haben soll.

Hier wird in verschiedenen Punkten unterschieden:

### Grant und Revoke

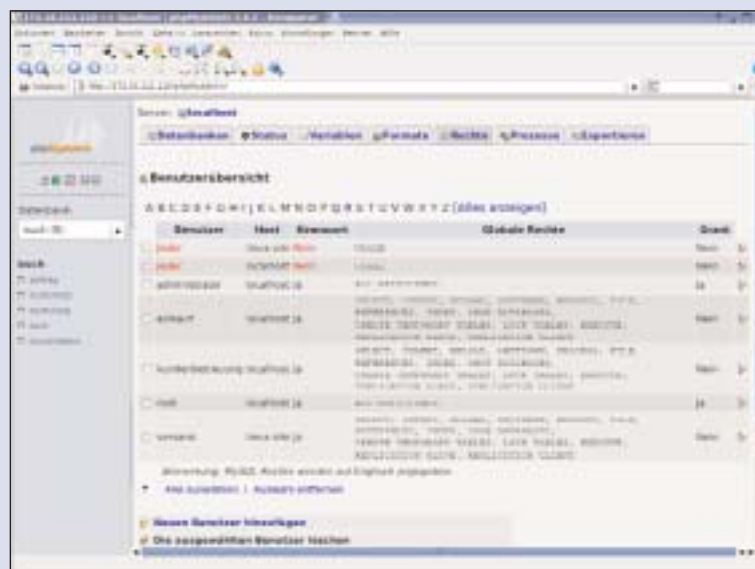
Diese Anweisungen treten bei der Rechtevergabe auf.

Grant bedeutet, dass ein Recht vergeben werden kann. Man kann hier für jeden Benutzer wählen, ob er diese Möglichkeit der Rechtevergabe bekommen darf, oder nicht. Zu beachten ist hierbei, dass ein Benutzer, der die Grant-Option besitzt auch nur die Rechte an andere Benutzer weitergeben darf, die er selbst besitzt.

Die Option Revoke bedeutet in gleicher Art und Weise, das Entziehen von Rechten.

### Globale Rechte

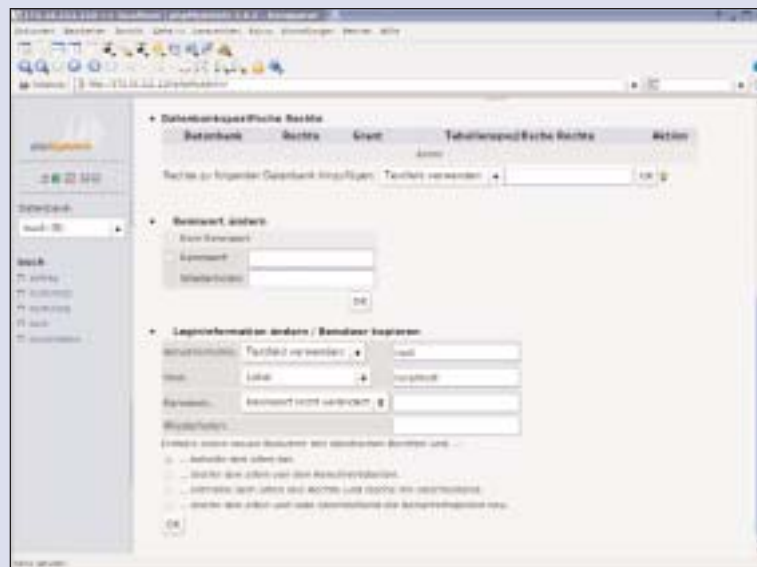
Unter diesem Punkt findet man Rechte wie **select**, **insert**, **shutdown** und **show databases**. Hier wird festgelegt, welche Aktionen ein Benutzer grundsätzlich durchführen darf.





## Datenbankspezifische Rechte

Wie der Name schon sagt, wird hier ausgewählt auf welche Datenbank ein Benutzer zugreifen und die Aktionen durchführen darf, die unter den globalen Rechten aufgeführt sind. Unter weiteren Punkten kann festgelegt werden, ob z. B. nur auf die Tabelle **buch** zugegriffen werden darf. Wählt man eine einzelne Tabelle aus, so können speziell einzelne Felder der Tabelle ausgewählt werden und festgelegt, welche Aktionen der jeweilige Benutzer hier durchführen darf. Somit ist eine sehr detaillierte Benutzererstellung möglich. Als abschließenden Punkt der Benutzererstellung wird ein Benutzername und Passwort eingetragen, unter dem ein Benutzer Zugriff auf phpMyAdmin mit seinen persönlichen Rechten erhält. So ist die Funktion eines CMS (Content Management Systems) gegeben. Oft werden solche Funktionen zum Pflegen einer Datenbank über HTML ausgeführt, in unserem Fall hier direkt über phpMyAdmin.



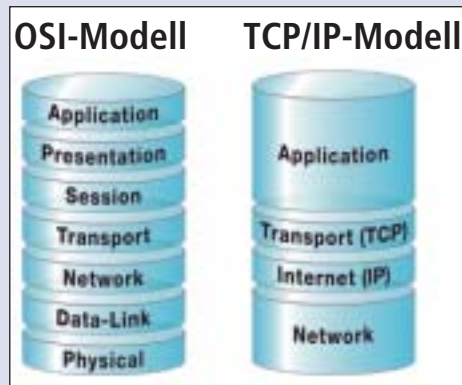
Der letzte wichtige Punkt ist die Option „Host“. Ist hier die Option localhost angegeben, können Benutzer nur lokal auf phpMyAdmin zugreifen.

In unserem Fall ist es nun so, dass von anderen Rechnern auf die Datenbank und somit auf phpMyAdmin zugegriffen werden soll. Um dies zu ermöglichen ist der Eintrag der Rechner-IP des Servers möglich. Um zu gewährleisten, dass über den erstellten Shop Abfragen ermöglicht werden, ist der Eintrag **linux.site** nötig. Dies wird benötigt, da der Rechnername **root@linux.site** lautet. Lautet die Bezeichnung anders, so erhält der Kunde in seinem Browser die Meldung, dass er keine Zugriffsrechte besitzt.

# HTTP

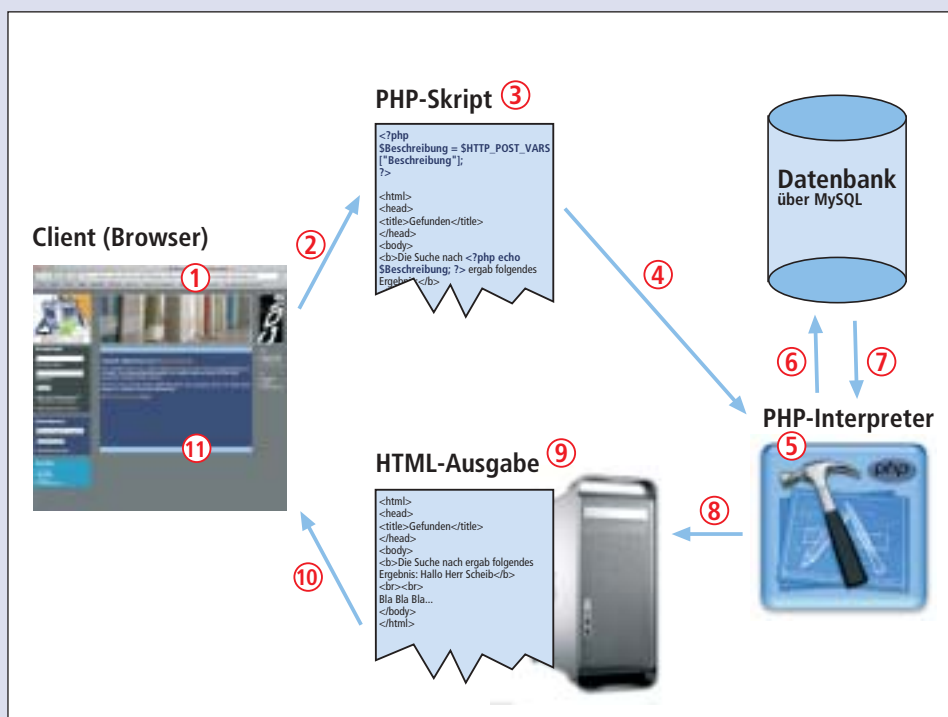
HTTP steht für HyperText Transfer Protocoll und ist eines der elementaren Protokolle im Internet. Es dient zur Übertragung und Regelung von Daten zwischen Client und Server.

Das HTTP-Protokoll wird vom TCP/IP-Modell bereitgestellt und ist der Anwendungsschicht (Application Layer) zugeordnet.



## Nochmals zur Vertiefung des Kommunikationsvorgangs:

Die Kommunikation zwischen Client und Server splittet sich in den Anfrage- und den Antwortvorgang (Response and Request). Der Client startet die Anfrage an den Webserver, der diese bearbeitet und die relevanten Daten an den Client zurücksendet.







**Beispiel**, um das HTTP-Kommunikationsschema detaillierter begreifbar zu machen: Client klickt auf der Bücherrausch-Internetseite auf den Link [www.buecherrausch.de/Seiten/kontakt.html](http://www.buecherrausch.de/Seiten/kontakt.html).

Diese Anfrage wird an den Rechner geschickt, der Webserver für die URL [www.buecherrausch.de](http://www.buecherrausch.de) ist. (*Übertragungsart HTTP-GET über TCP-Protokoll, Standardport für HTTP-Übertragungen: 80*)

**Anfrage:**GET/kontakt.html HTTP/1.1  
Host: www.buecherrausch.de : 80

**Antwort:**HTTP/1.1 200 ok  
Server: Apache/Versionsnummer (Linux)  
Content-length: (Bytegröße von kontakt.html)  
Content-language: de  
Content-Type: text/html  
Connection: close

[Hier steht nun der Inhalt der Seite kontakt.html](#)

Für unser Projekt ist es wichtig zu erwähnen, dass über das Setzen von Cookies in den Headerinformationen, die in den Anfragen mitgeschickt werden, Statusinformationen zugeordnet werden können, die Anwendungen wie Benutzereinträge, Benutzerauthentifizierung und Warenkörbe erfordern. Diese Daten, die durchaus empfindliche Inhalte enthalten (können), sollten jedoch nicht unverschlüsselt übertragen werden . . .

## HTTPS

HTTPS steht für HyperText Transfer Protocoll **Secure** und ist ebenfalls ein Netzwerkprotokoll – mit dem Unterschied, dass eine gesicherte Verbindung zwischen Client und Server ermöglicht wird. Die Daten werden hierbei über SSL/TLS verschlüsselt. (*Übertragung über TCP-Protokoll, Standardport für HTTPS-Übertragungen: 443*)

Die HTTPS-Verbindung sieht man im Browser über das Schlosssymbol. SSL (Secure Socket Layer) ist ein Übertragungsprotokoll zur verschlüsselten Kommunikation zwischen Client und Browser und baut ebenfalls auf TCP/IP in der Transport-Schicht auf – das SSL-Protokoll befindet sich zwischen der Transport-Schicht und der Anwendungsschicht.

Das SSL-Protokoll gliedert sich in zwei wichtige Bestandteile bzw. Ebenen:

**Obere Ebene:** SSL-Handshake-Protokoll zur gegenseitigen Authentifizierung (zwischen Client / Server) während des Verbindungsauf- und Abbaus.

**Untere Ebene:** SSL-Record Protocoll. Zerlegt Datenblöcke in einheitliche Größen, komprimiert und verschlüsselt diese und gibt sie an die tieferliegende TCP-Schicht.



# HTML

HTML (HyperText Markup Language) ist ein Dokumentenformat zur Auszeichnung von Hypertexten im World Wide Web, die mit Hilfe von SGML (Standard Generalized Markup Language) definiert wird. Somit ist HTML eine Auszeichnungssprache zur Beschreibung von Hypertexten.

HTML weist aufgrund der ursprünglichen Entwicklungsabsichten nur wenige Formatierungsmöglichkeiten auf, was für den visuellen Anspruch negativ zu Buche steht. Abhilfe können hier CSS (Cascading Style Sheets) oder auch JavaScript schaffen, die Elemente, Elementgruppen und ganze HTML-Seiten formatieren können.

Ein HTML-Dokument besteht aus drei grundlegenden Bereichen:

**1.) Die Doctype Declaration**, also die Definition des Dokumenten-Typs:

```
<!DOCTYPE HTML PUBLIC „-//W3C//DTD HTML 4.01 Transitional//EN“>
```

**2.) Der HTML-Head:** Enthält hauptsächlich technische Informationen zur Dokumentation. Der Headinhalt wird nicht bei der Ausgabe im Browser angezeigt, ist jedoch über den Quelltext einsehbar. Im Headbereich können Informationen wie title / meta-tags / script oder style enthalten sein.

```
<HTML>  
<HEAD>  
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">  
<TITLE>::: Willkommen auf B&uuml;lcherrausch :::</TITLE>  
</HEAD>  
</HTML>
```

**3.) Der HTML-Body:** Der eigentliche Körper, in welchem alle Informationen enthalten sind, die auch bei der späteren Ausgabe im Browser angezeigt werden.

```
<html>  
<head>  
<title>Unbenanntes Dokument</title>  
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"><link href="typografie.css" rel="stylesheet" type="text/css">  
</head>  
<body>  
<tr><td>Willkommen auf unserer Internetseite!</td></tr>  
</table>  
</body>  
</html>
```



## Dreamweaver

Zur Erstellung der Internetseiten im design- und programmiertechnischen Bereich haben wir den Webeditor Macromedia Dreamweaver MX verwendet. Da es sich um einen WYSIWIG-Editor (What You See Is What You Get) handelt, ist Arbeiten unter einer grafischen Benutzeroberfläche als auch direkt im (HTML-) Quelltext möglich. Dreamweaver ermöglicht das Testen von Webseiten unter verschiedenen Browserversionen sowie unter Einbeziehung eines lokalen Webservers. Neben der HTML-Programmierung werden auch aktuelle Skriptsprachen wie PHP, JavaScript oder PERL unterstützt.

### **Info:**

Die Erstellung des HTML-Quelltexts wäre auch in einem reinen Texteditor möglich.

## Gestaltung und Aufbau des Webshops

Bei der Planung und der Konzeption unserer Internetseiten haben wir uns für ein Design mit drei Frames entschieden.

Der linke, vertikale Navigationsbereich (naviframe\_links) bleibt immer bestehen, in diesem sind alle Hauptnavigationselemente aufgeführt. Die Aufführung in vertikaler Form trägt wesentlich zur Übersichtlichkeit und schnellen Orientierung bei.

Im mittleren Frame (contentframe\_mitte) werden die jeweils themenspezifischen Daten bzw. HTML-Dokumente geladen. Die Dokumente sind entweder statisch im Verzeichnis des Webservers abgelegt und werden über direkte Verweise geladen – oder sie werden über Datenbankausgaben dynamisch generiert.

Der rechte Frame (logoframe\_rechts) bietet Raum für eine Subnavigation und dient der gestalterischen Abrundung. Auch dieser Frame bleibt, wie auch der linke Navigationsframe, inhaltlich unverändert.

Der Auftritt orientiert sich am linken Browserfenster und ist auf eine Monitorauflösung von 800x600 Pixel angepasst. Ist eine höhere Auflösung gewählt, verlängert sich die Hintergrund-Farbe des rechten Frames jeweils bis zum rechten Bildschirmbereich. Es ist zwar, den Standards der heutigen Zeit entsprechend, anzunehmen, dass sich die Auflösung von 1024x768 Pixel als Standard durchgesetzt hat, doch möchten wir gewährleisten, dass auch User mit einer Auflösung von 800x600 Pixel die Seite betrachten können.




Das Logo steht zentriert über der Navigation und ist ein deutlicher Eyecatcher. Über dem contentframe\_mitte ist ein Bild mit einem alten Bücherregal platziert, das dem Besucher das Webshop-Thema auch dauerhaft kommuniziert,

Mit dem kleinen Bild (com) im rechten Framee entsteht eine horizontale Bildleiste, die das visuelle Gleichgewicht zu den Inhalten bildet.

### Farben

Die verwendeten Farben sollen leuchtend und fröhlich, jedoch harmonisch und seriös wirken.

Orange:	# FF 99 00	
Dunkelblau:	# 33 33 66	
Hellblau:	# 99 CC FF	
Dunkelgrau:	# 33 33 33	

### Texte

Alle verwendeten Texte sind mit CSS definiert, um einen grafischen Anspruch zu gewährleisten und den Text besser zu visualisieren und zu strukturieren.

#### CSS Definition für Hyperlinks

```
A:link { color: #FFFFFF; text-decoration: none; }
A:visited { color: #FFFFFF; text-decoration: none; }
A:hover { color: #FF9900; text-decoration: none; }
A:active { color: #FF9900; text-decoration: none; }
```

#### CSS Definition für Schriftformat

```
.navibold {
  font-family: Verdana, Arial, Helvetica, sans-serif;
  font-size: 12px;
  font-weight: bold;
  color: #FFFFFF;
  margin: auto;
}
.navinormal {
  font-family: Verdana, Arial, Helvetica, sans-serif;
  font-size: 11px;
  color: #FFFFFF;
}
```

Die gesamten HTML-Dokumente (bzw. PHP-Dokumente), die in den Frames geladen werden, sind mit Tabellen (tables) aufgebaut.

Auf Ebenen bzw. Layers (div) wurde aus Kompatibilitätsgründen verzichtet.



## PHP

PHP (rekursives Akronym für „PHP: Hypertext Preprocessor“, ursprünglich „Personal Home Page Tools“) ist eine Skriptsprache mit einer an C bzw. Perl angelehnten Syntax, die hauptsächlich zur Erstellung dynamischer Webseiten verwendet wird. **Bei PHP handelt es sich ebenfalls um Open-Source-Software.**

PHP zeichnet sich besonders durch die leichte Erlernbarkeit, die breite Datenbankunterstützung und Internet-Protokolleinbindung sowie die Verfügbarkeit zahlreicher, zusätzlicher Funktionsbibliotheken aus.

**PHP ist eine serverseitig interpretierte Sprache**, die den Quelltext nicht an den Browser übermittelt, sondern an einen Interpreter auf dem Webserver. Erst dessen Ausgabe wird an den Browser geschickt. In den meisten Fällen ist dies eine HTML-Seite, wobei es mit PHP möglich ist, andere Datentypen, wie z. B. Bilder oder PDF-Dateien, zu generieren.

Um eine PHP-Datei ausführen zu können, benötigt man ein System, das mit den in der Datei enthaltenen Anweisungen umzugehen weiß (Apache). Das auf Servern übliche Programmpaket für Linux/Windows – Apache als Webserver, MySQL-Server als Datenbank und PHP – wird LAMP bzw. WAMP genannt. (**L**inux **A**pache **M**ySQL **P**HP bzw. **W**indows **A**pache **M**ySQL **P**HP)

Die Vorteile der serverseitigen Ausführung sind, dass beim Browser (z. B. Safari, Firefox, ...) keine speziellen Fähigkeiten erforderlich sind oder Probleme (zusätzliche Tools) auftreten können. Außerdem bleibt der PHP-Quelltext der Seite auf dem Server und nur der generierte Code ist für den User sichtbar. Gleiches gilt für andere Ressourcen wie z. B. Datenbanken, die daher auch keine direkte Verbindung zum Client benötigen.



## Automatische Datenbankverbindung

Über das unten aufgeführte, einfache PHP-Script wird der Server „172.16.111.110“ angesprochen, der „Username“, das „Passwort“ und die angesprochene Datenbank übergeben. Falls der Verbindungsaufbau fehlschlägt, wird eine Fehlermeldung ausgegeben, ansonsten die Verbindung mit einer „echo“ Ausgabe bestätigt.

```
<?php
    $server = „172.16.111.110“;
    $user = „root“;
    $password = „bbsnw“;
    $datenbank = „buch“;

    $conn = @mysql_connect($server, $user, $password);
    @mysql_select_db($datenbank, $conn);

    if ($conn!=TRUE)
    {
        echo „Verbindungsfehler: „.mysql_error().“! Versuchen Sie es
zu einem späteren Zeitpunkt nochmals. Danke.“;
        die;
    }

    if ($conn==TRUE)
    {
        echo „Verbindung steht.“;
    }
?>
```



## Büchersuche

In dieser PHP-Sequenz wird ein `mysql_query` auf die Tabelle `buch` der Datenbank ausgeführt und die entsprechenden Daten in einem `echo` ausgegeben. Am Ende wird die Datenbank mit `mysql_close` wieder geschlossen.

```
<?php
$query = „SELECT * FROM buch WHERE titel LIKE '%$Beschreibung%' „;
$result=mysql_query($query);

while ($daten=mysql_fetch_array($result, MYSQL_ASSOC))
{
echo '<table width="500" border=" „ class="navinormal" cellpadding="0" cellspacing="0">
  <tr>
  <p>' . $daten[titel] . '</p>
  <p>' . $daten[autor] . '</p>
  <p>' . $daten[beschreibung] . '</p>
  <p>ISBN: ' . $daten[isbn] . '</p>
  <p>' . $daten[lieferzustand] . '</p>
  </tr>
</table>';
echo „<br /><br />“;
}
mysql_close($conn);
?>
```



## Eintragung von Kundendaten

Über das Anmeldeformular „benutzer\_anmelden.html“ kann sich ein Kunde neu registrieren lassen. Die in dieser HTML-Datei eingetragenen Daten werden entsprechend in der „eintragen\_submit.php“ Datei überprüft und in die Tabelle „kundendaten“ der Datenbank eingepflegt.

```
<?php
    $strasse = $HTTP_POST_VARS[„strasse“];
    $wohntort = $HTTP_POST_VARS[„wohntort“];
    $plz = $HTTP_POST_VARS[„plz“];
    $benutzername = $HTTP_POST_VARS[„benutzername“];
    $passwort = $HTTP_POST_VARS[„passwort“];
    $vorname = $HTTP_POST_VARS[„vorname“];
    $nachname = $HTTP_POST_VARS[„nachname“];
?>
...

<?php include(„connect.php“); ?>
<br>
<br>
Vorname: <b><?php echo $vorname; ?></b><br>
<br>
Nachname: <b><?php echo $nachname; ?></b>
<br>
Strasse: <b><?php echo $strasse; ?></b>
<br>
PLZ: <b><?php echo $plz; ?></b>
<br>
Wohnort: <b><?php echo $wohntort; ?></b>
<br>
<?php
if ($strasse == „“ || $wohntort == „“ || $plz == „“ || $benutzername == „“ ||
$password == „“ || $vorname == „“ || $nachname == „“) {
    echo „Leider wurden von Ihnen nicht alle Felder ausgefüllt!“;
    die(„Bitte überprüfen Sie alle Felder!“);
}
$insert=„INSERT INTO kundendaten (strasse, wohntort, plz, benutzername,
password, vorname, nachname)
VALUES ('$strasse', '$wohntort', '$plz', '$benutzername', '$password',
'$vorname', '$nachname)'“;
//@mysql_select_db($datenbank, $conn);
$r_insert=mysql_query($insert);
if ($r_insert==TRUE) {
    mysql_close($conn);
}
?>
</body>
</html>
```



## Wählen einzelner Rubriken

Wie bei der Schnellsuche hat hier der Kunde die Möglichkeit, einzelne Rubriken speziell auszuwählen und diese werden dann aufgelistet.

```
<?php
$query = „SELECT * FROM buch WHERE rubrik =‘roman‘“;
$result=mysql_query($query);

while ($daten=mysql_fetch_array($result, MYSQL_ASSOC))
{
echo ‘<table width=“500“ border=“ „ class=“navinormal“ cellspacing=“0“ cell-
padding=“0“>
  <tr>
  <p>‘ . $daten[titel] . ‘</p>
  <p>‘ . $daten[autor] . ‘</p>
    <p>‘ . $daten[beschreibung] . ‘</p>
  <p>ISBN: ‘ . $daten[isbn] . ‘</p>
    <p>‘ . $daten[lieferzustand] . ‘</p>
  </tr>
</table>‘;
echo „<br /><br />“;
}
mysql_close($conn);
?>
</body>
</html>
```



## Abschließendes Fazit

Zu Beginn des Projektes unterteilten wir die Aufgabengebiete und jedes Mitglied erarbeitet selbstständig die definierten Aufgaben. Bei der Zusammenführung dieser ergaben sich einige Überschneidungen und komplexen Aufgabenstellungen, die leider zu terminlichen Engpässen führten. Anschließend wollen wir diese aufzeigen und unsere Gedanken zur Realisation hierzu darbringen.

### **Formatierung von php-Abfragen:**

Bei der Erstellung der Dynamischen Seiten durch PHP erkannten wir schnell die Tücken der Ausgabe von Schrift in festgelegten Größen und Schnitten. Unser Lösungsansatz war über die Formatierung mit css zu gehen, wobei diese in das PHP-Skript mit einzubinden seien.

### **Kundenlogin:**

Im Bereich Kundenlogin, sollte dem hier schon registrierten Benutzer die Möglichkeit geboten werden, sich in seinem eigenen Bereich (persönliche Kundendaten, Änderung Kundendaten, Bestellungen beobachten) zu informieren. Über die Eingabe seines Benutzernamens und Passwortes würden diese mit der Tabelle kundendaten abgeglichen und ihm der Zugang gewährt oder verweigert. War die Eingabe korrekt kommt er in einen eigenen Bereich um die oben genannten Optionen einzusehen. Dieser Bereich wiederum wird durch die entsprechenden Abfragen auf die Tabellen kundendaten oder bestellung erstellt.

### **Warenkorb:**

Die vom Kunden ausgewählten Produkte werden über ein PHP-Skript in eine separate Tabelle bestellung eingetragen und beim Aufrufen des Warenkorbes aus dieser wieder ausgelesen und aufgelistet.

### **Bestellung:**

Nachdem der Kunde die Liste seines Warenkorbes mit einem Klick auf den Link Bestellung abschicken bestätigt hat erhält er als Übersicht seine Bestellung und seine Kundendaten auf dem Bildschirm zur letzten Ansicht. Ein im Hintergrund ausführendes PHP-Skript fügt die aus der Tabelle bestellung eingetragene Liste mit dem aus der Tabelle kundendaten zugehörigen Daten zusammen und stellt diese den User dar. Sofern es keine weiteren Änderungen gibt bestätigt der Kunde die Angaben abermals und die Daten werden aus der Tabelle warenkorb gelöscht. Die Informationen aus der Tabelle bestellung werden an die zuständige Stelle als Email oder Übergabe an eine andere Datenbank weitergeleitet. Der Benutzer erhält als Bestellbestätigung ebenfalls eine Email in sein Postfach.



# Webshop

## Allgemein

Im Internet existieren verschiedene Formen von Websites, die auf unterschiedliche Weise Informationen aus Datenbanken auslesen und Benutzerdaten in Datenbanken speichern.

Im Bereich E-Commerce unterscheidet man folgende Arten:

**Webshop:** Ein einzelner Laden bietet einen bestimmten Produktschwerpunkt (z.B. MusikCDs) dem Endverbraucher über einen Produktkatalog online an aus dem der Benutzer einkaufen kann.

**Malls:** Verschiedene Läden mit unterschiedlichen Produktschwerpunkten werden über eine Portalseite zusammengefasst

**Werbefinanzierte Portalseiten:** Kostenlose E-Mail-Briefkästen, Verzeichnisse, Chats (z.B. web.de)

**Maklerseiten:** Auktionen, Tauschbörsen (z.B. ebay)

Je nach Größe haben diese verschiedene Charaktereigenschaften, die aber alle auf einem identischen Kern aufbauen.

## Vorgehensweise

### Beginn:

Über eine Startseite (index.html) kann der Besucher auf diverse html bzw. php-basierende Unterseiten (Kontakt, Rubriken, etc.) zugreifen. Auf diesen Seiten erhält der Besucher die Möglichkeiten, Kontakt mit dem Unternehmen aufzunehmen, sich Informationen über Lieferbedingungen und -konditionen herauszusuchen und durch Eingabe seiner persönlichen Daten, diese in einer Tabelle der Datenbank eintragen zu lassen.

### Rubrik:

Unter dem Feld Rubrik kann der Benutzer sich bestimmte Themengruppen der Bücher schon vorselektiert ansehen. Durch den entsprechenden Link auf der Gruppe wird eine Abfrage ausgelöst, die zur Aufgabe hat die entsprechenden Begriffe aus der Tabelle der Datenbank darzustellen.

### Schnellsuche:

Das Feld Schnellsuche ermöglicht dem User einfach über die Eingabe eines Wortes alle gefundenen Artikel der Abfrage aufzulisten. Prinzipiell erfolgt die Abfrage wie in Rubrik beschrieben.



#### Warenkorb:

Sofern sich der Benutzer für ein Produkt entschieden hat, wird dieses in den Warenkorb gelegt, wobei dieser gleichzusetzen ist mit einem Einkaufswagen. Das bedeutet, es können weiterhin Produkte hinzugefügt oder entfernt und deren Menge geändert werden. Der Warenkorb ist noch keine Bestellung, sondern nur eine Auflistung der in Frage kommenden Produkte. In diesem Warenkorb erhält der User zusätzlich Überblick über die Preise und den Lieferzustand.

#### Bestellung:

Der angelegte Warenkorb wird über einen gesonderten Link zu einer neuen Seite "Bestellung" hinzugefügt, wobei der nun "werdende Kunde" seine persönlichen Daten überprüfen und die gewünschte Zahlungs- bzw. Lieferart festlegen kann.

Nach dem Abschicken der Bestellung wird dem Kunden eine Bestellbestätigung entweder auf dem Bildschirm angezeigt oder per Email zugesendet.



## Quellenangaben

Literaturverzeichnis

PHP und MySQL

Marco Skulschus, Marcus Wiederstein – Markt + Technik

Das LAMP Buch

Heiko Goller – SuSE PRESS

[www.wikipedia.org](http://www.wikipedia.org)

[www.php.net](http://www.php.net)

[www.phpinfo.de](http://www.phpinfo.de)

[www.smial.de/apache.html](http://www.smial.de/apache.html)

[www.netmirror.org](http://www.netmirror.org)

[www.linux-fuer-alle.de](http://www.linux-fuer-alle.de)

[www.phpmyadmin.net](http://www.phpmyadmin.net)

[www.mysql.com](http://www.mysql.com)

[www.f4.fhtw-berlin.de](http://www.f4.fhtw-berlin.de)

[www.ammler.com](http://www.ammler.com)

[www.apachefriends.org/de](http://www.apachefriends.org/de)

<http://de.selfhtml.org>



## Abschließende Erklärung

### Erklärung

Hiermit erklären wir, dass die Projektarbeit selbstständig und ohne fremde Hilfe verfasst wurde und keine anderen, als die angegebenen Mittel verwendet wurden.

Neustadt an der Weinstraße  
Donnerstag, 19. Mai 2005

---

Steffen Krauß

---

Dominik Morio

---

Matthias Trauth