

# **Framework für einen Webshop in PHP**

## **Praktikumsarbeit PA Fei01/2**

Dozent: Patrick Feisthammel

Studenten: Reto Glanzmann, José Fontanil

März bis Mai 2001

## 1. Vorwort

Im Rahmen unserer ersten Projektarbeit im dritten Studienjahr erstellten wir ein Shop-Framework auf Basis von PHP4 welcher unter der GPL jedem kostenlos zur Verfügung gestellt wird.

Wir haben in diesen 7½ Wochen einen Shop erschaffen, der vom Design her praktisch unbegrenzt erweiterbar ist. Der Shop wurde userseitig komplett fertig gestellt und in der Administration fehlt nur noch die Versandkostenkalkulation. Von der Aufgabenstellung her konnten wir sogar ein optionales Element, unbegrenzte Anzahl Optionen, implementieren.

Trotz einiger Probleme verloren wir unser Ziel und unsere Motivation nicht und können heute voll hinter unserem Produkt stehen.

## 2. Inhaltsverzeichnis

Framework für einen Webshop in PHP.....	1
Praktikumsarbeit PA Fei01/2.....	1
1. Vorwort.....	2
3. Aufgabenstellung.....	5
4. Analyse bestehender Webshops.....	5
5. Wahl der verwendeten Technologien.....	7
6. Zeitplanung.....	8
8. Shopaufbau.....	9
8.1 Security.....	9
8.1.1 PHP.....	9
8.1.2 htaccess.....	9
8.1.3 MySQL.....	9
8.2 Module und Funktionen.....	10
8.2.1 Übersicht.....	10
8.2.2 Module und ihre Funktionen.....	11
8.3 Session-Management.....	19
8.3.1 Maximale Session Dauer.....	19
8.3.2 Session-Management-Header.....	19
8.4 Datenbank - shopdb.....	20
8.4.1 Idee und Konzept.....	20
8.4.2 Implementation.....	20
8.4.3 shopdb – Relationen.....	21
8.4.4 Entity Relationship Diagramm der Shop Datenbank.....	22
8.4.4.1 ER-Diagramm Teil 1.....	22
8.4.4.2 ER-Diagramm Teil 2.....	23
8.4.5 Tabellen und ihre Attribute im Detail.....	23
8.5 SQL-Query Handling.....	25
8.6 DB-Wrapper (database.php).....	25
8.7 Datenbank Installations- / Deinstallations-SQL-Skripts.....	26
8.8 Shoplayout (HTML / CSS / Javascript).....	26
8.8.1 HTML Shopbenutzer.....	26
8.8.2 HTML Shopadministrator.....	27
8.8.3 CSS.....	28
8.10 Test.....	28
9. Bedienung.....	33
9.1 Bedienung Shop.....	33
9.1.1 Bedienung Shop.....	33
9.1.2 Bedienung Administrationstool.....	33
9.2 Installation des Shops.....	34
9.3 Backup.....	36
10. Erfahrungen.....	36
10.1 Datenbanken allgemein.....	36
10.2 Bilder in der Datenbank speichern.....	37
10.3 Das Projekt.....	37
10.4 PHP Programmierung.....	38
11. Was sollte noch gemacht werden.....	38
11.1 Probleme, die vor einer Shopinbetriebnahme unbedingt behoben werden sollten. .	39
11.2 Funktionen und Probleme die zur optischen Einheitlichkeit und Usability gelöst	

werden sollten.....	39
11.3 Sinnvolle Zusatzfunktionalität.....	40
12. Schlusswort.....	42
13. Literaturverzeichnis.....	43
13.1 PHP.....	43
13.2 Apache Webserver.....	43
13.3 Datenbanken und im speziellen MySQL.....	44
13.4 HTML, JavaScript und Cascading Style Sheets.....	44
14. Software Versionen.....	45
14.1 Auf dem Server eingesetzte Software.....	45
14.2 Auf den Arbeitsstationen eingesetzte Software.....	45
15. GPL – GNU General Public Licence.....	46
15.1 Deutsche Fassung der GPL (Auszug).....	46
16. Anhang.....	49
A Source Code.....	49
B Zusammenfassungen.....	49
A Source-Code	
B Zusammenfassungen	

### 3. Aufgabenstellung

#### PA1 Fei 01/2: Framework für PHP Webshop

- **Studierende:** [Fontanil Jose, IT3a](#)  
[Glanzmann Reto, IT3a](#)
- **Dozent:** Feisthammel Patrick
- **Kurzbeschreibung:**

Es gibt viele Webshops, jedoch haben die preiswerteren den gravierenden Nachteil, dass sie von der Produktpräsentation her sehr statisch sind. Das Ziel ist ein allgemein gehaltener Webshop, der mit relativ wenig Programmieraufwand an die Bedürfnisse der Shopbetreiber angepasst werden kann.

Es sollen die Grundfunktionen eines in weiten Bereichen konfigurierbaren Webshops mit Warenkorbsystem zur Verfügung gestellt werden. Die Produktdaten werden in einer SQL Datenbank gehalten. Die Pflege des Artikelsstamms soll über MS-Access möglich sein.

Die Architektur ist als typische 3-Tier Architektur auszulegen.

Die Software und Dokumentation wird unter Open Source entwickelt.

### 4. Analyse bestehender Webshops

Eine Studie von PricewaterhouseCoopers hat ergeben, dass 77 Prozent der Surfer in Online-Shops eine Suchfunktion benutzen. 43 Prozent der Befragten erklärten sogar, Suchfunktionalität sei das wichtigste Feature von eShops. 44 Prozent der Befragten sagten, ihnen seien Detailansichten des Produktes besonders wichtig

Diese Studie bewog uns dazu, einige erfolgreiche Webshops und Webshopprogramme etwas genauer unter die Lupe zu nehmen. Da Suchfunktionalität einen wichtigeren Stellenwert einnimmt, als wir eigentlich annahmen, machten wir uns auch auf die Suche nach intuitiven Suchfunktionen.

Folgende Webshops und Sites haben wir genauer unter die Lupe genommen:

<b>Programm / Site</b>	<b>Beschreibung</b>	<b>Link</b>
Shopfactory	Kommerzielles Webshopprogramm mit Offline-Bearbeitungstool, welches bei Shoptests gut abgeschlossen hat.	<a href="http://www.shopfactory.com">www.shopfactory.com</a>
Steg Computer GmbH	Webshop eines Computerteileverkäufers, der sehr gute Umsätze per Web erreicht.	<a href="http://www.stegpc.ch">www.stegpc.ch</a>
Glarotech Informatikdienstl.	ISP mit guter Suchfunktion.	<a href="http://www.glaro.ch">www.glaro.ch</a>
3COM	Hierarchische Suchfunktion im Treiber-Downloadbereich.	<a href="http://infodeli.3com.com/infodeli/tools/nic/index.htm">http://infodeli.3com.com/infodeli/tools/nic/index.htm</a>

Google	Benutzerfreundliche Suchmaschine	<a href="http://www.google.com">www.google.com</a>
--------	----------------------------------	--

Alle Vor- und Nachteile jedes Shops und jeder Suchmaschine hier aufzulisten, würde den Rahmen dieses Kapitels sprengen. Wir fassen hier die Funktionen und Eigenschaften der verschiedenen Shops und Suchmaschinen zusammen, die uns besonders angesprochen haben. In der vorderen Spalte ist jeweils die Funktionalität an sich beschrieben, in der mittleren, wo wir die Funktion gefunden haben und in der rechten, ob sie in unserem Shop implementiert ist (☑), der Shop dafür vorbereitet, aber noch nicht implementiert wurde(☐), oder wir darauf gänzlich Verzichtet haben (☒).

<b>Suchfunktionen</b>		
Sehr einfach gehalten, Advanced Search für erfahrene Anwender	Shopfactory	☒
Hierarchisch angeordnete Pulldown Menus. Je eines für Artikelkategorie, Produktfamilie und Produktbeschreibung.	3COM	☒
Suche von mehreren Wörtern mit Booleschen Operatoren z.B (+artikel +detail)	Glarotech	☐
Volltextsuche	Shopfactory	☑
Wenn die Suche kein Resultat ergibt, Ausgabe dass nichts gefunden wurde an Benutzer	Alle	☑
Ausgabe des Suchresultates formatiert, mit Highlighting des Suchwortes im Text und Möglichkeit, direkt zur Fundstelle zu springen	Google	☐
Lange Suchresultatlisten auf mehrere Seiten verteilt	Google	☐
<b>Produkte-Darstellung</b>		
Buttons eindeutig in Ihrer Funktionalität erkennbar (Piktogramm und Text)	Steg	☑
Eindeutige Produkteabtrennung mit Balken oder in Tabelle	Shopfactory Steg	☑
Wenn eine Artikelkategorie ausgegeben wird, oben Kategoriebeschreibung und eventuell vorhandenes Kategoriebild.	Shopfactory	☐
Detailansicht vom Produkt, wenn man auf das Thumbnail klickt.	Shopfactory	☑
Link zum Hersteller oder zu weiteren Informationen	Steg	☑
Verfügbarkeit des Artikels	Steg	☒
<b>Home-Screen Darstellung</b>		
Navigationsbaum	Steg	☑
Kontakt zum Unternehmen knüpfen können	Steg Shopfactory	☑
Aktionen, Top-Seller, Monatshit	Steg	☐
Bedienungshilfe	Steg	☑

Letztes Preisupdate des Shops	Steg	<input checked="" type="checkbox"/>
Versandkostenpreise ersichtlich	Steg Shopfactory	<input type="checkbox"/>
<b>Bestellung</b>		
Vor endgültigem Abschluss der Bestellung, AGB einblenden	Steg	<input checked="" type="checkbox"/>
Permanent ersichtlicher Warenkorb Button	Steg Shopfactory	<input checked="" type="checkbox"/>
Jederzeit die Möglichkeit bieten, in der Bestellung einen Schritt zurück zu gehen.	Steg	<input checked="" type="checkbox"/>
Bestellung, vor absenden ausdrückbar am Browser anzeigen	Steg Shopfactory	<input checked="" type="checkbox"/>

## 5. Wahl der verwendeten Technologien

Ein Framework für einen Webshop lässt sich grundsätzlich auf viele Arten realisieren, in den folgenden Absätzen wollen wir begründen, wieso wir uns für die Kombination PHP4 als Apache-Modul und MySQL als Datenbank entschieden haben.

Ein vielfältiger Webshop sollte ein hohes Mass an Dynamik bieten und seine Daten, wenn möglich, aus einer zentralen Datenbank beziehen können. Damit hat man auch alle drei Komponenten die nötig sind um einen dynamischen Webshop realisieren zu können:

- Scriptsprache zur Erstellung dynamischer Websites
- Webserver
- Datenbank

Da wir uns für eine Veröffentlichung des Webshops unter der GPL (siehe Anhang) entschieden haben, war es uns auch wichtig Technologien zu verwenden die selbst Open Source sind. Deshalb wählten wir PHP4 (Personal Home Page Tools Version 4). Unter anderem weil PHP fürs Web erstellt wurde und nicht wie z.B. ASP, das mit VBScript daher kommt welches nur 'schwer' an die Bedürfnisse eines Webserver angepasst werden kann und weiter nur für die Microsoft Plattform verfügbar ist. Perl hingegen bietet sehr mächtige Zeichenketten-Funktionen, hat aber den Nachteil einen eher unleserlichen Code zu eigen zu haben. PHP nimmt sich der Nachteile der oben genannten Technologien an und verbindet diese mit ihren Vorteilen. PHP bietet die von ASP bekannte, einfache Kombination mit HTML, die leistungsstarke und weit verbreitete C-Syntax, die vielen schon von JAVA und JavaScript her bekannt ist. PHP besticht ausserdem durch einen fast schon grandiosen Funktionsumfang, hier kann keine andere Sprache mithalten. Das sieht man vor allem bei der Datenbank Unterstützung, was uns auch sehr wichtig war.

Die Entscheidung welchen Webserver wir verwenden war einfach zu beantworten: Apache ist der heute im Internet am meisten eingesetzte Webserver, zuverlässig, Open Source und für die wichtigsten Plattformen verfügbar. Er hat auch grosse Zukunftsaussichten. Wir wollten den Einsatz des Shops aber nicht grundsätzlich an eine Webserver-Technologie binden. Es braucht nur minimalste Änderungen, um den Shop auf einem anderen Webserver laufen zu lassen (Passwortschutz für Administration und Pfade bei Wechsel von UNIX- zur Windows-Plattform).

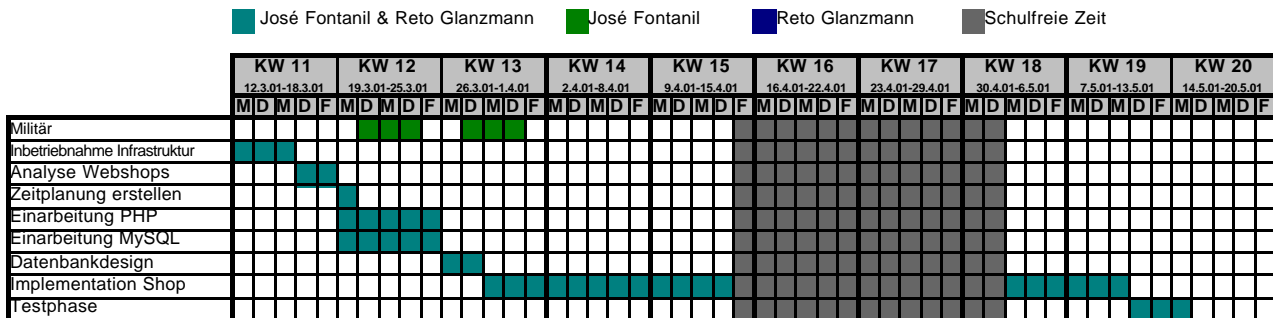
Bei der Datenbank konnten wir eine gute Idee unseres Dozenten und Praktikumbetreuers

umsetzen. Unser Webshop wickelt alle Datenbankzugriffe zentral über einen Datenbank-Wrapper ab. Das heisst, der Shop-Betreiber kann wählen, was für eine Datenbank er benutzen will. Dies ist sicher von Vorteil, wenn in einem bestehenden Unternehmen schon eine Datenbank läuft. Der Shop ist momentan zu 100% auf MySQL einsetzbar und getestet. Die Programmierung der Sybase Unterstützung ist so gut wie abgeschlossen, für alle weiteren Datenbanken muss man nur noch zwei Klassen neu schreiben.

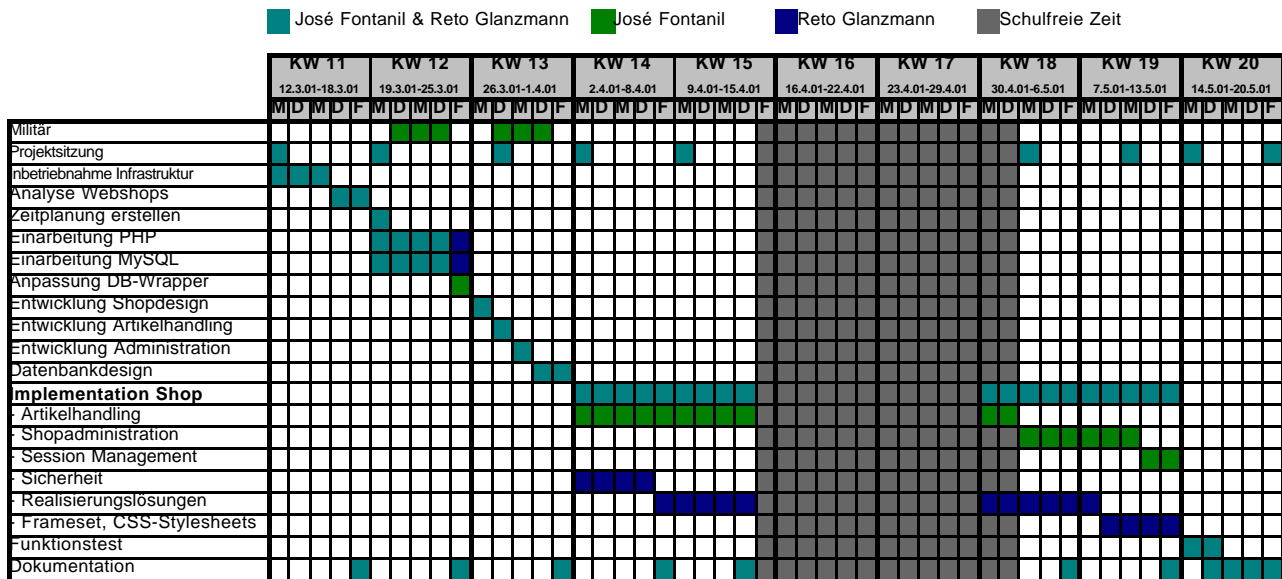
MySQL war für uns erste Wahl, weil diese Datenbank für jedermann gratis zur Verfügung steht und bei den Internet Service Provider vielfach schon vorinstalliert ist. Ihr Leistungsumfang kommt zwar nicht an den grosser Datenbanken heran, wir haben uns aber mit dem DB-Wrapper diese Türen offen gehalten.

## 6. Zeitplanung

In einer ersten Zeitplanung listeten wir die von uns erwarteten Arbeitsvorgänge auf, wobei wir eine grobe Einschätzung der von uns erwarteten Zeitaufwendungen vornahmen und die Arbeitsgänge in einer Zeitachse darstellten. Da wir noch nicht wussten, wer welchen Teil der Arbeiten übernimmt, ordneten wir alle Aktivitäten vorerst einmal beiden zu.



Die Zeitplanung passten wir dann fortlaufend dem aktuellen Stand des Projektes an, woraus sich dann schlussendlich folgende Zeiteinteilung ergab. An vielen Problemen arbeiteten wir gemeinsam, deshalb konnten diese Aktivitäten nicht spezifisch einer Person zugeordnet werden.



Abschliessend ist zur Zeitplanung zu sagen, dass es sich lohnt, jede Woche einen Review zu machen und für die nächste Woche zu planen. Meilensteine, bei einem solch kleinen Projekt, wie wir es hier hatten, zu setzen erachten wir als nicht notwendig.

## 8. Shopaufbau

### 8.1 Security

Wir haben uns viele Gedanken betreffend der Security des Shops gemacht und Sicherheitsmassnahmen auf vielen Ebenen eingebunden. Im Folgenden gehen wir genauer auf die einzelnen Teile der Shop-Security ein.

#### 8.1.1 PHP

Da ein potentieller Hacker/Cracker den Sourcecode der HTML-Seiten anschauen, sogar Header-Informationen begutachten kann, besteht die Gefahr, dass er sich so ein Bild über den Shop-Aufbau machen kann und so zu wichtigen Hintergrundinformationen kommen könnte.

Um dem vorzubeugen haben wir uns schon beim Design dafür entschieden, die PHP-Scripte welche Administrations-Aufgaben wahrnehmen, von den USER-Level-Scripts so gut es geht zu entkoppeln.

Dies fängt schon bei den SQL-Scripts an: Alle SQL-Scripts, die von USER-Funktionen aus benutzt werden, kommen aus einer Datei, alle Administrator-Scripts sind in einer anderen Datei, die von keinem USER-Modul included wird. Auf diese Weise ist sichergestellt, dass von einer USER-Funktion niemals ein Administrations-Script aufgerufen werden wird.

PHP bietet einen sogenannten `SAFE MODE` an, durch ihn versucht PHP einige Sicherheitslücken zu schliessen. Diesem Modus liegt ein Schema zugrunde, dass die Zugriffsberechtigungen auf Dateien regelt. Ein Zugriff auf eine Datei wird dann nur noch gewährt, wenn die Datei, das Verzeichnis und das Script dem gleichen Benutzer gehören. Dieses Kriterium gilt weiter für alle Funktionen, die ein Risiko für die Systemsicherheit darstellen könnten. Wir haben den `SAFE MODE` selbstverständlich eingeschaltet.

Damit ein Benutzer nicht eine Bestellung mit manipulierten Preisen aufgeben kann,

werden die Preise immer aus der Datenbank geholt und an das jeweils anfordernde Script übergeben.

**8.1.2 htaccess**

Der Administrations-Teil ist weiter per htaccess geschützt, alle Administrations-Scripte sind in einem Unterverzeichnis abgelegt. Wer einen anderen Server bsp. IIS einsetzt, muss lediglich einen Link abändern, um den dort vorhanden Verzeichnis-Passwort-Schutz zu aktivieren.

**8.1.3 MySQL**

Um dieses Konzept auch auf der Datenbank Ebene fortzusetzen arbeitet der Shop grundsätzlich mit zwei Datenbank Benutzern. Einerseits mit dem `shopuser` und um die Administrator-Scripte auszuführen als `shopadmin`. Dem `shopuser` wurden nur die nötigsten Rechte gewährt. Er darf die Datenbank lesen und er darf Bestellungen löschen / updaten, mehr nicht.

MySQL selbst wird auch nicht als `root` gestartet sondern mittels der Anweisung:

```
../mysql/bin/safe_mysqld --user mysql&
```

auf diese Weise wird MySQL von einem User gestartet, für den es gar kein UNIX-Pendant gibt und der keinerlei Rechte ausserhalb des MySQL-Verzeichnisses hat.

Zum Schluss stellten wir noch sicher, dass die beiden User `shopuser` und `shopadmin` kein GRANT-Recht erhielten.

**8.2 Module und Funktionen**

Im folgenden wird nun die Implementation des Shops aus struktureller Sicht genauer erläutert. Die Übersicht soll helfen den Aufbau des Shops zu besser verstehen zu können. Die Modul- und Funktionsbeschreibungen geben einen Überblick auf die Abbildung der Shop-Funktionalitäten auf die PHP-Scripte.

**8.2.1 Übersicht**

Hier wird der Shop schematisch im Aufbau dargestellt. Man sieht die einzelnen Ebenen und die USER- / ADMINISTRATOR-Aufteilung. Im nächsten Kapitel sieht man dann die Module und ihre Funktionen.

	<b>USER</b> <i>./shop/</i>	<b>ADMIN</b> <i>./shop/Admin/</i>
<b>HTML-Darstellung</b> (Eingangsseite, Frameset, CSS-Konfigurations-Datei)	../index.html ../index.php ./Frameset/top.php ./Frameset/content.html shopstyles.css (Bilder und Buttons werden jeweils in <i>./Bilder/*</i> und <i>./Buttons/*</i> gespeichert)	
<b>Administrations Menü</b>	Shop_Einstellungen_Menu_1.php Shop_Einstellungen_Menu_Artikel.php Shop_Einstellungen_Menu_Kategorien.php	

	<b>USER</b> <i>./shop/</i>		<b>ADMIN</b> <i>./shop/Admin/</i>	
<b>Darstellungs-Module</b> (hier werden komplexe Strukturen in HTML / plain-Text gehandelt)	USER_BESTELLUNG_DARSTELLUNG.php		SHOP_ADMINISTRATION_ARTIKEL.php	
<b>Aufruf-Module</b> (rufen Funktionen auf und generieren HTML)	Artikel-Handling	Bestellungswesen		Shop Administration
	USER_ARTIKEL_HANDLING_AUFRUF.php	USER_BESTELLUNG_AUFRUF.php	USER_BESTELLUNG_1.php	SHOP_ADMINISTRATION_AUFRUF.php
<b>PHP-Funktionen</b>	Artikel-Handling	User seitiges Bestellungswesen	Einzelne Shop-Settings auslesen	Shop Administrations-Funktionen
	USER_ARTIKEL_HANDLING.php	USER_BESTELLUNG.php	USER_ARTIKEL_HANDLING.php	SHOP_ADMINISTRATION.php bild_up.php
<b>Hilfe aus DB auslesen</b>	USER_ADMIN_HILFE.php			
<b>Bild aus DB auslesen</b>	bild_view.php			
<b>Datenbankanbindung</b> (DB-Objekt erstellen)	initialize.php		ADMIN_initialize.php	
<b>SQL-Statements</b> (Variablen zugeordnet)	USER_SQL_BEFEHLE.php		ADMIN_SQL_BEFEHLE.php	
<b>Datenbank Wrapper</b> (DB-spezifische Zugriffsfunktionen)	database.php			
<b>Definitionen</b> (Komplexe Datentypen als Klassen)	artikel_def.php bestellung_def.php			

### 8.2.2 Module und ihre Funktionen

<b>Modul: Definitions</b>	
<b>Datei</b>	<b>Klassenname</b>
artikel_def.php	Artikel
<b>Beschreibung</b>	Definiert eine Klasse Artikel, um mit den vielen Attributen eines Artikels umgehen zu können. Ein Artikel-Objekt kann auch alle Optionen und Variationen (beliebige Menge) eines Artikels fassen. Auf die Variablen der Klasse wird direkt, ohne set- / get-Funktionen zugegriffen. Für die Optionen- und Variationen-Arrays gibt es entsprechende Funktionen.
bestellung_def.php	Bestellung
<b>Beschreibung</b>	Definiert eine Klasse Bestellung, die der Forderung gerecht wird eine ganze Bestellung inklusive all ihren Attribute und Artikel-Zuordnungen aufnehmen zu können. Die Artikel werden dabei in einem Array (für den es Zugriffsfunktionen gibt) verwaltet. Dieser Array enthält Artikel_info-Objekte (siehe weiter unten).
bestellung_def.php	Artikel_info

<b>Modul: Definitions</b>	
<b>Beschreibung</b>	Diese Klasse wird nur im Zusammenhang mit Bestellungen verwendet. Sie enthält Informationen bezüglich eines Artikels in einer Bestellung (Also auch Anzahl u.s.w.).

<b>Modul: Datenbank Wrapper</b>	
<b>Datei</b>	<b>Klassenname</b>
database.php	TRecordSet / TSybaseRecordSet / TMySQLRecordSet
<b>Beschreibung</b>	TRecordSet ist eine abstrakte Klasse, welche dann für die jeweilige Datenbank-Implementation vererbt wird. Ein RecordSet bietet Funktionen, um komfortabel und Datenbank unabhängig die einzelnen Felder auszulesen, die nächste Zeile einzulesen und die Anzahl erhaltener Zeilen auszugeben.
database.php	TDatabase / TSybaseDatabase / TMySQLDatabase
<b>Beschreibung</b>	TDatabase ist ebenfalls eine abstrakte Klasse, welche dann für die jeweilige Datenbank-Implementation weiter vererbt wird. Sie kümmert sich darum Datenbankunabhängige Connections zu bieten. Dies beinhaltet ein Connection-Objekt, Zugriffsfunktionen für Queries und eine vereinfachte Zugriffsfunktion ohne Rückgabewerte (INSERT, UPDATE).

<b>Modul: SQL-Statements</b>	
<b>Datei</b>	<b>Klassenname</b>
USER_SQL_BEFEHLE.php	Artikel
<b>Beschreibung</b>	Hier werden Variablen SQL-Queries zugewiesen (siehe Kapitel 8.5).
ADMIN_SQL_BEFEHLE.php	Bestellung
<b>Beschreibung</b>	Hier werden wie bei USER_SQL_BEFEHLE.php die Query-Teile verschiedenen Variablen zugeordnet. Bloss handelt es sich hier um Queries von den Administrator-Funktionen. Diese Datei wird deshalb von USER-Modulen nicht included.

<b>Modul: Datenbankbindung</b>	
<b>Datei</b>	
initialize.php	
<b>Beschreibung</b>	Wird von User-Modulen included und stellt ein Objekt \$Database zur Verfügung. Dieses Objekt eröffnet eine Verbindung als shopuser. Die Verbindung ist nicht persistent und wird bei Script-Ende automatisch wieder geschlossen.
ADMIN_initialize.php	
<b>Beschreibung</b>	Diese Datei wird von Administrator-Modulen eingebunden (included) und bietet eine nicht persistente Datenbank-Connection als shopadmin an.

<b>Modul: Bild aus DB auslesen</b>	
<b>Datei</b>	<b>Funktionsname</b>
bild_view.php	bild_view
<b>Beschreibung</b>	Diese Datei wird nur innerhalb des HTML-Tags <IMG src="bild_view.php?..."> verwendet. Es liest aus der Datenbank das entsprechende Bild aus und stellt es dann bereit. Die Datei ruft ihre einzige Funktion selbst auf.

<b>Modul: Hilfe aus DB auslesen</b>	
<b>Datei</b>	<b>Funktionsname</b>
USER_ADMIN_HILFE.php	get_Hilfe
<b>Beschreibung</b>	Diese Funktion liefert einen String (inkl. HTML- s) der den entsprechenden Hilfetext der angeforderten Seite beinhaltet. Die zu übergebende Hilfe_ID setzt sich aus dem Namen der anfordernden Datei zusammen und hat angehängt noch einen Suffix von _1, _2...

<b>Modul: PHP-Funktionen - USER_ARTIKEL_HANDLING</b>	
<b>Datei</b>	<b>Funktionsname</b>
USER_ARTIKEL_HANDLING.php	get_Artikel
<b>Beschreibung</b>	Liefert aufgrund einer übergebenen Artikel_ID ein Artikel-Objekt mit dem entsprechenden Artikel.
USER_ARTIKEL_HANDLING.php	getArtikeleinerKategorie
<b>Beschreibung</b>	Liefert aufgrund eines übergebenen Kategorienamens alle Artikel als Artikel-Objekte in einem Array die in dieser Kategorie eingeordnet sind.
USER_ARTIKEL_HANDLING.php	getgesuchterArtikel
<b>Beschreibung</b>	Hier wird ebenfalls ein Array mit Artikel-Objekten zurückgegeben. Darin enthalten sind alle Artikel, in deren Namen oder Beschreibung der im Suchstring übergebenen Text gefunden wurde.
USER_ARTIKEL_HANDLING.php	getallKategorien

<b>Modul: PHP-Funktionen - USER_ARTIKEL_HANDLING</b>	
<b>Beschreibung</b>	Liefert ein Array mit allen Kategorienamen sortiert nach Positions-Nummern.
USER_ARTIKEL_HANDLING.php	getKategorie_eines_Artikels
<b>Beschreibung</b>	Liefert den Kategorienamen eines Artikels.
USER_ARTIKEL_HANDLING.php	getvar_opt_preise
<b>Beschreibung</b>	Liefert die Texte und Preise aller Optionen und Variationen eines Artikels. Diese Funktion wird bei einem Bestellvorgang benötigt, da aus Sicherheitsgründen keine Preise von Script zu Script übertragen werden.

<b>Modul: Userseitiges Bestellweszen</b>	
<b>Datei</b>	<b>Funktionsname</b>
USER_BESTELLUNG.php	getBestellung
<b>Beschreibung</b>	Gibt aufgrund einer Session-ID (falls existierend) eine Bestellung als Bestellungs-Objekt zurück (inkl. allen zugeordneten Artikel).
USER_BESTELLUNG.php	test_create_Bestellung
<b>Beschreibung</b>	Diese Funktion gehört einerseits zum Bestellweszen, andererseits aber auch zum Session-Management. Sie überprüft zuerst, ob in der Datenbank eine zur aktuellen Session-ID passende Bestellung existiert. Wenn ja, ist alles gut, wenn nein, so wird eine Neue erstellt. Auf diese Weise ist sichergestellt, dass jeder Benutzer jederzeit ohne Fehlermeldung in den Bestellvorgang einsteigen kann.
USER_BESTELLUNG.php	addArtikel
<b>Beschreibung</b>	Diese Funktion wird benötigt, wenn der Kunde auf den "In den Warenkorb"-Button klickt. Sie fügt der Bestellung des Kunden den Artikel hinzu (Zuordnung).
USER_BESTELLUNG.php	del_B_Artikel
<b>Beschreibung</b>	Löscht aus einer Bestellung den angewählten Artikel
USER_BESTELLUNG.php	delSession
<b>Beschreibung</b>	Löscht eine Session und damit auch die Bestellung
USER_BESTELLUNG.php	delallexpiredSessions
<b>Beschreibung</b>	Löscht alle abgelaufenen Bestellungen (und damit auch Sessions). Diese Funktion ist sehr zeitaufwändig und wird nur dann ausgeführt, wenn ein Benutzer die Allgemeinen Geschäftsbedingungen des Shops akzeptiert und damit die Bestellung auslöst.
USER_BESTELLUNG.php	setKundendaten
<b>Beschreibung</b>	Über diese Funktion gelangen die Kundendaten (Name, Adresse,...) in die Datenbank und werden persistent in der jeweiligen Bestellung abgespeichert.
USER_BESTELLUNG.php	schliessenBestellung
<b>Beschreibung</b>	Diese Funktion wird ebenfalls erst aufgerufen, nachdem ein Kunde die Bestellung ausgelöst hat. Die in der Datenbank gespeicherte Bestellung wird als abgeschlossen markiert (Attribut Bestellung_abgeschlossen) und die Session_ID wird gelöscht.

<b>Modul: PHP-Funktionen - Einzelne Shop-Settings auslesen</b>	
<b>Datei</b>	<b>Funktionsname</b>
USER_ARTIKEL_HANDLING.php	getWaehrung
<b>Beschreibung</b>	Liefert als einen String die in Shop-Settings gespeicherte Wahrung (z.B: SFr., DM, s,...)
USER_ARTIKEL_HANDLING.php	getGewichts_Masseinheit
<b>Beschreibung</b>	Liefert als einen String die in Shop-Settings gespeicherte Gewichts-Masseinheit (z.B. kg, ml, g, t,... )
USER_ARTIKEL_HANDLING.php	getThumbnail_Breite
<b>Beschreibung</b>	Da in bild_up.php ein Thumbnail eines Artikel-Bildes erzeugt wird, kann ber dieses shop_settings-Attribut die Breite des Thumbnails festgelegt werden. Diese Funktion liefert nun den Integer-Wert in Pixel.
USER_ARTIKEL_HANDLING.php	getshopname
<b>Beschreibung</b>	Liefert in einem String, den in den Shop-Settings gespeicherten Namen des Shops.
USER_ARTIKEL_HANDLING.php	getmax_session_time
<b>Beschreibung</b>	In den Shop-Settings lasst sich ein fixer Wert in Sekunden einstellen, der als die maximal gltige Dauer einer Session zahlt. So ist man etwas freier bei der Wahl der Session-Dauer. (Nach oben gibt es die Limite in der php.ini) Diese Funktion liefert nun diesen Wert als Integer zurck.
USER_ARTIKEL_HANDLING.php	getBezahlungsart
<b>Beschreibung</b>	Liefert einen Array, wo drin steht welche Zahlungsarten gewahlt werden knnen. Der jeweilige Name ist 'Y' wenn er in den Shop-Settings aktiviert wurde, sonst 'N'.
USER_ARTIKEL_HANDLING.php	getAGB
<b>Beschreibung</b>	Gibt in einem String die allgemeinen Geschaftsbedingungen wieder.
USER_ARTIKEL_HANDLING.php	getEmail
<b>Beschreibung</b>	Liefert die E-Mail Adresse des Kunden (aktuelle Bestellung)
USER_ARTIKEL_HANDLING.php	getShopEmail
<b>Beschreibung</b>	Liefert die E-Mail Adresse des Shop Betreibers (oder des Verantwortlichen, in Shop-Settings eingestellt)

<b>Modul: PHP-Funktionen - SHOP_ADMINISTRATION</b>	
<b>Datei</b>	<b>Funktionsname</b>
SHOP_ADMINISTRATION.php	newArtikel
<b>Beschreibung</b>	Einen neuen Artikel in die Datenbank speichern (Betrifft Tabellen artikel, artikel_optionen, artikel_variationen, artikel_kategorien !)
SHOP_ADMINISTRATION.php	delArtikel
<b>Beschreibung</b>	Einen Artikel aus der shopdb-Datenbank entfernen (Betrifft mehrere Tabellen).

<b>Modul: PHP-Funktionen – SHOP_ADMINISTRATION</b>	
SHOP_ADMINISTRATION.php	updArtikel
<b>Beschreibung</b>	Einen Artikel updaten. Diese Funktion ruft eigentlich nur diverse weitere auf. Ein Artikel-Update zieht sich durch die Funktionen updArtikel, getArtikel, darstellenArtikel, updbestellungen, updkategorien, und durch die Datei bild_up.php. Diese "Funktion" löscht einen schon bestehenden Artikel, trägt ihn mit den neuen Werten wieder in die Datenbank ein und aktualisiert alle betroffenen Tabellen. Es wäre gut diese Funktion mit der UPDATE-SQL Anweisung neu zu schreiben.
SHOP_ADMINISTRATION.php	bild_kopieren
<b>Beschreibung</b>	Wird beim Update eines Artikels verwendet. Wenn der zu updatende Artikel ein Bild hat, wird mit dieser Funktion das Bild zum aktualisierten Artikel kopiert.
SHOP_ADMINISTRATION.php	getshopsettings
<b>Beschreibung</b>	Diese Funktion liefert in einem Array alle Shop-Settings zurück. (Key = Name-, Value = Wert des Attributs).
SHOP_ADMINISTRATION.php	setshopsettings
<b>Beschreibung</b>	Macht ein Update auf alle Attribute der Tabelle shop_settings und aktualisiert somit alle Shop-Settings.
SHOP_ADMINISTRATION.php	updbestellungen
<b>Beschreibung</b>	Diese Funktion aktualisiert nach einem Artikel-Update die artikel_bestellungen Tabelle indem sie überall die alte, jetzt ungültige Artikel_ID durch die Neue ersetzt.
SHOP_ADMINISTRATION.php	updkategorien
<b>Beschreibung</b>	Analog updbestellungen aktualisiert diese Funktion nach einem Artikel-Update die artikel_kategorie Tabelle, indem sie überall die Alte, jetzt ungültige Artikel_ID durch die Neue ersetzt.
SHOP_ADMINISTRATION.php	delBild
<b>Beschreibung</b>	Löscht das Bild des aktuellen Artikels.
SHOP_ADMINISTRATION.php	newKategorie
<b>Beschreibung</b>	Erstellt eine neue Kategorie mit dem übergebenen Namen und der übergebenen Positions-Nr.
SHOP_ADMINISTRATION.php	delKategorie
<b>Beschreibung</b>	Löscht eine Kategorie. Die Überprüfung, ob eine solche überhaupt existiert, wurde an den Aufruf-Level delegiert.
SHOP_ADMINISTRATION.php	setallKategorien
<b>Beschreibung</b>	Schreibt alle Kategorien mit neuen Werten in die Datenbank zurück (UPDATE auf die Tabelle kategorien)
bild_up.php	Variable \$\$speichern = false, \$loeschen_Bild = 0

<b>Modul: PHP-Funktionen - SHOP_ADMINISTRATION</b>	
<b>Beschreibung</b>	Dies ist der Zustand, wenn man im Administrations-Menü entweder einen neuen Artikel einfügen will, oder einen bestehenden Artikel updaten will und schon alle Eingaben gemacht hat. Jetzt wird eine Seite geöffnet, wo man entweder ein Bild zum Artikel in die Datenbank speichern kann, das schon vorhandene Bild weiterverwenden, oder das schon bestehende Bild löschen.
bild_up.php	Variable \$Speichern = true
<b>Beschreibung</b>	Das bedeutet, der Shop-Administrator will ein Bild zu diesem Artikel in der Datenbank speichern. Nun wird ein neues Thumbnail vom eingelesenen Bild angefertigt und alle Bilddaten (insgesamt 4 Attribute der Tabelle artikel) in der Datenbank gespeichert. Diese Funktionalität verlangt, dass diese Datei dem UNIX-User root gehört (dies ist nicht gleichzusetzen mit einer Datei die Root-Rechte hat!).
bild_up.php	Variable \$loeschen_Bild = 1, \$Speichern = false
<b>Beschreibung</b>	Der Shop-Administrator hat sich in diesem Fall entschieden das vorhandene Bild zu löschen, dies wird nun ausgeführt.

<b>Modul: Aufruf-Module - USER_ARTIKEL_HANDLING</b>	
<b>Datei</b>	<b>Variable \$darstellen =</b>
USER_ARTIKEL_HANDLING_AUFRUF.php	1
<b>Beschreibung</b>	Alle Artikel einer Kategorie im Content-Frame darstellen. Wird das Artikel-Thumbnail angeklickt, geht ein Pop-Up Fenster auf, worin das Artikelbild in Originalgrösse angezeigt wird.
USER_ARTIKEL_HANDLING_AUFRUF.php	30
<b>Beschreibung</b>	Artikel Suchen – Teil 1 von 2: Eingabefenster
USER_ARTIKEL_HANDLING_AUFRUF.php	3
<b>Beschreibung</b>	Artikel Suchen – Teil 2 von 2: Verarbeitung der Eingabe und Ausgabe des Resultats
USER_ARTIKEL_HANDLING_AUFRUF.php	4
<b>Beschreibung</b>	Dieser Aufruf wird vom index.php aus gebraucht um im linken Frame die Kategorien-Auswahlliste darstellen zu können.

<b>Modul: Aufruf-Module - USER_BESTELLUNG</b>	
<b>Datei</b>	<b>Variable \$darstellen =</b>
USER_BESTELLUNG_AUFRUF.php	1
<b>Beschreibung</b>	Anzeige des Warenkorbs des Kunden im Content-Frame.
USER_BESTELLUNG_AUFRUF.php	2
<b>Beschreibung</b>	Wird aufgerufen, wenn der Kunde einen Artikel in seinen Warenkorb gelegt hat. (Weiche: Zurück zu Artikel, in Warenkorb- oder zur Kasse gehen)
USER_BESTELLUNG_AUFRUF.php	3
<b>Beschreibung</b>	Wenn ein Kunde in seinem Warenkorb bei einem Artikel auf löschen klickt, so wird dieser Meldungsbildschirm angezeigt

<b>Modul: Aufruf-Module - SHOP_ADMINISTRATION</b>	
<b>Datei</b>	<b>Variable \$darstellen =</b>
SHOP_ADMINISTRATION_AUFRUF.php	1
<b>Beschreibung</b>	Ein neuer, leerer Artikel wird erzeugt und an darstellen Artikel weitergegeben. Auf diese Weise kann man neue Artikel in die Datenbank aufnehmen.
SHOP_ADMINISTRATION_AUFRUF.php	2

<b>Modul: Aufruf-Module - SHOP_ADMINISTRATION</b>	
<b>Beschreibung</b>	Auswahlmenü 1 (Kategorienwahl), wenn man einen Artikel zum Bearbeiten / Löschen auswählen will.
SHOP_ADMINISTRATION_AUFRUF.php	4
<b>Beschreibung</b>	Auswahlmenü 2 (Artikelwahl), wenn man einen Artikel zum Bearbeiten / Löschen auswählen will. (Ahm. Sdarstellen = 3 wurde früher für eine Testfunktion verwendet und ist jetzt nicht mehr in Gebrauch)
SHOP_ADMINISTRATION_AUFRUF.php	5
<b>Beschreibung</b>	Nachdem der Artikel bearbeitet wurde, wird hier entweder ein Update des Artikels oder das Löschen des Artikels eingeleitet
SHOP_ADMINISTRATION_AUFRUF.php	6
<b>Beschreibung</b>	Eine Kategorie einfügen 1 / 2: Eingabe der Daten der neuen Kategorie (Diese Funktion ist noch "Beta" – da sie erst die Eingabe von Name und Positions-Nr erlaubt. Beschreibung und Bilddaten blieben bei dieser Shop-Version noch aussen vor).
SHOP_ADMINISTRATION_AUFRUF.php	7
<b>Beschreibung</b>	Eine Kategorie einfügen 2 / 2: Einfügen der neuen Kategorie
SHOP_ADMINISTRATION_AUFRUF.php	9
<b>Beschreibung</b>	Eingabeformular, um eine Kategorie zu löschen
SHOP_ADMINISTRATION_AUFRUF.php	10
<b>Beschreibung</b>	Eingabe überprüfen (existiert überhaupt eine Kategorie wie es eingegeben wurde, wenn ja...) Die gewählte Kategorie löschen.
SHOP_ADMINISTRATION_AUFRUF.php	11
<b>Beschreibung</b>	Bestehende Kategorien bearbeiten, Schritt 1: Darstellen der vorhandenen Kategorien (Name und Positions-Nr.)
SHOP_ADMINISTRATION_AUFRUF.php	12
<b>Beschreibung</b>	Update aller bestehenden Kategorien.
SHOP_SETTINGS.php	
<b>Beschreibung</b>	Ein Eingabeformular für alle Attribute der Tabelle shop_settings wird eingeblendet und mit den derzeit aktuellen Einstellungen als value geladen.
SHOP_SETTINGS.php	10
<b>Beschreibung</b>	Wenn der Shop-Administrator auf Speichern geklickt hat, werden alle Shop-Einstellungen zurück in die Datenbank gespeichert.

<b>Modul: Darstellungs-Module - USER_BESTELLUNG</b>	
<b>Datei</b>	<b>Funktionsname</b>
USER_BESTELLUNG_DARSTELLUNG.php	darstellenBestellung
<b>Beschreibung</b>	Diese Funktion stellt die im Argument gelieferte Bestellung in HTML dar, das zweite Argument loeschen dient dazu um optional noch einen Löschen-Button zu jedem Artikel im Warenkorb anzuzeigen. (Dies ist aber nicht überall erwünscht)
USER_BESTELLUNG_DARSTELLUNG.php	darstellenStringBestellung
<b>Beschreibung</b>	Diese Funktion schreibt den Warenkorbinhalt formatiert in eine String der später per E-Mail an Kunde und Shop-Betreiber versandt wird.

<b>Modul: Darstellungs-Module - SHOP_ADMINISTRATION</b>	
<b>Datei</b>	<b>Funktionsname</b>
SHOP_ADMINISTRATION_ARTIKEL.php	darstellenArtikel
<b>Beschreibung</b>	Diese Funktion stellt den im Argument gelieferten Artikel (als Artikel-Objekt) dar. Weiter bietet es ein Eingabeformular, wo der Shop-Administrator den Artikel mutieren kann. Am Ende des Eingabeformulars werden noch Buttons (Reset, Submit, Abbrechen, Hilfe) zur Navigation dargestellt.

<b>Modul: Administrations Menu - SHOP_ADMINISTRATION</b>	
<b>Datei</b>	<b>Funktionsname</b>
Shop_Einstellungen_Menu_1.php	
<b>Beschreibung</b>	Dies ist das Hauptmenü der Shop-Administration. Von hier aus kann man in der momentan aktuellen Shop-Version Artikel bearbeiten, Kategorien bearbeiten und die meisten Shop-Settings ändern. Auch die Hilfe-Funktion ist eingeschaltet, wenn auch der Text noch etwas angepasst werden muss
Shop_Einstellungen_Menu_Artikel.php	
<b>Beschreibung</b>	Von hier aus kann man einen Artikel neu einfügen, bearbeiten oder löschen.
Shop_Einstellungen_Menu_Artikel.php	
<b>Beschreibung</b>	Von hier aus kann man Kategorien neu einfügen, bearbeiten oder löschen.

<b>Modul: HTML-Darstellung</b>	
<b>Datei</b>	<b>Funktionsname</b>
index.html / index.php	

<b>Modul: HTML-Darstellung</b>	
<b>Beschreibung</b>	Das <code>index.html</code> leitet direkt auf <code>index.php</code> weiter. Dort wird dem Surfer bereits eine Session und somit eine leere Bestellung zugeordnet. Nun kann er den Shop vollumfänglich benutzen. Die Datei <code>index.php</code> baut weiter noch das Frameset.
<code>top.php / content.html</code>	
<b>Beschreibung</b>	Bilden das Frameset ( <code>FRAME='left'</code> wird von einer Aufruf-Funktion mit Inhalt gefüllt). Das File <code>content.html</code> hält einen Willkommens-Bildschirm bereit, <code>top.php</code> zeigt den Shop-Namen an und bietet dem Kunden Navigationselemente wie Warenkorb, Kasse und Hilfe. Ausserdem versteckt sich dort hinter dem kleinen Stern der Passwort geschützte Eingang zur Shop-Administration.

### 8.3 Session-Management

Das Session-Management wird über die PHP4-interne Session-Verwaltung abgewickelt. PHP4 ist grundsätzlich in der Lage, die Verknüpfung der Zugriffe über URL oder Cookies vorzunehmen. Die Session Konfiguration läuft weitgehend über die Datei `php.ini` ab (bei uns im Verzeichnis `/etc/httpd/`). Dort belassen wir alle Sessionrelevanten Einstellungen auf den Default-Werten. Auf diese Weise konfiguriert, arbeitet unser Session-Management mit Cookies, deren Lebenszeit auf die Laufzeit des Browsers beschränkt ist.

#### 8.3.1 Maximale Session Dauer

Weiter wird in der `php.ini` die maximale Zeit einer Session auf 1440 Sek. also rund 24Min. beschränkt. In den Shop-Settings kann der Shop-Administrator aber die Zeit einer Session verändern. Was immer er in diesem Feld eingibt, wird nur zum Tragen kommen, solange die dort angegebene maximale Session-Zeit kürzer oder gleich lang ist als die in der `php.ini`. Weiter ist zu beachten, dass die Zeit die in den Shop-Settings eingestellt ist, sich nicht "direkt" auf die Session-Variable auswirkt:

Wenn ein Kunde auf den Shop geht, so erhält er nicht nur eine Session-ID, sondern auch gleich eine leere Bestellung. Dort drin gibt es ein Feld namens `expired`, darin wird die Summe aus der aktuellen Zeit und der in den Shop-Settings definierten maximalen Session-Dauer abgespeichert. Bei jeder neuen Seite die den Session-Management-Header (siehe weiter unten) beinhaltet, wird geprüft ob diese Zeit nicht schon abgelaufen ist. Ist dies der Fall, so wird die Bestellung in der Datenbank gelöscht, der Kunde erhält eine neue, leere Bestellung.

Rekapitulierend kann man also sagen: Er erhält zwar keine neue Session-ID, er erhält aber eine neue Bestellung, was für ihn zum gleichen Resultat führt. Er wird vom Shop aus als "neuer" Kunde gesehen.

#### 8.3.2 Session-Management-Header

Hier der Code, den wir Session-Management-Header nennen. Er ist überall dort in die Files integriert worden, wo der Kunde vom Shop einen HTML-Header (also eine neue Seite) zugeschickt bekommt.

Dieser Header schaut, ob die Session-ID, bei uns \$mySession\_ID genannt, schon vorhanden ist, dann soll die Session-ID Ausgabefunktion diese Variable auslesen. Falls die Variable noch nicht existiert, soll unsere Session\_ID \$mySession\_ID heissen und diese Variable an die aktuelle Session gebunden werden und eine Session eröffnet werden.

```
// -----
// Session_Management: (Ueberpruefen ob eine Session_ID uebergeben wurde,
// sonst eine Neue erzeugen
if(empty($mySession_ID)){
    session_name("mySession_ID");
    if (!session_register("mySession_ID")) {
        die("<HTML><BODY><P><H1>USER_BESTELLUNG_AUFRUF.php: ACHTUNG session_register()
            = false, konnte keine Session_ID zuteilen</H1></P><BR><BR></BODY></HTML>");
    }
}
else {
    session_id($mySession_ID);
}
```

## 8.4 Datenbank - shopdb

Das Datenbankdesign ist eine sehr zentrale Komponente bei der Entwicklung eines Web-Shop-Frameworks, dem entsprechend investierten wir viel Zeit und Energie in die fortlaufende Entwicklung dieses Designs. *Security-Aspekte* zum Datenbankdesign wurden schon im Kapitel 8.3 behandelt.

### 8.4.1 Idee und Konzept

Damit man einen möglichst dynamischen Shop, also einen Shop mit vielen Freiheiten und grossem Einsatzgebiet erhält, haben wir uns für ein etwas komplexeres Design entschieden.

Weiter war es eine Grundidee, dass man den Shop *komplett* in der Datenbank hat. Dies ist vom Backup her ein Vorteil (einfach Backup der Datenbank anfertigen, fertig) und vom Zugriff, so dachten wir, auch. Leider haben wir uns vor dem Design wenig mit BLOBs in Datenbanken beschäftigt. Es erwies sich später als hinderlich, wenn man BinaryLargeObjects in der gleichen Tabelle hat wie weitere Daten. Es ist uns bewusst, dass man saubererweise die Bilder der Artikel in einer gesonderten Tabelle hätte ablegen müssen. Leider war die Zeit schon zu knapp, als dass wir dieses Redesign noch hätten umsetzen können, Jetzt sind die Bilder halt in der selben Tabelle (analoges gilt natürlich auch für die Bilder der Kategorien).

### 8.4.2 Implementation

Wir haben darauf geachtet, allen Tabellen klein geschriebene Namen zu geben, da z.B. MySQL die Tabellen als Dateien im Verzeichnis mit dem Namen der Datenbank speichert. Sicher ist sicher.

Ein Artikel besteht aus einer zentralen Tabelle in der alle seine Kern-Attribute gespeichert werden. Ein Artikel kann aber zusätzlich noch beliebig viele Optionen und Variationen enthalten. Diese Fähigkeit implementierten wir via zwei Tabellen, die mit einer 1 zu n – Verbindung mit der Artikel-Kern-Tabelle verbunden sind. (Tabellen:

artikel\_optionen und artikel\_variationen) Bei den beiden zusätzlichen Tabellen haben wir auf einen expliziten Key verzichtet. Ihr Schlüssel ist der Fremdschlüssel FK\_Artikel\_ID, der dem jeweiligen Artikel\_ID in der Kern-Tabelle entspricht. Wir haben ihm absichtlich einen anderen Namen gegeben, so dass man bei Queries keine unerwarteten Namenskonflikte hat.

Weiter kann ein Artikel in mehreren Kategorien eingeteilt sein. Die Datenbank ist dafür ausgelegt. Die derzeitige Shop-Implementation (PHP-Scripte und HTML-Teil) ist aber erst dafür ausgelegt mit *einer* Kategorie pro Artikel zu arbeiten. Weiter arbeitet der Shop im Moment auch nur mit den Kategorienamen und deren Positions-Nummer, Beschreibung und Bilder blieben aus Zeitgründen vorerst aussen vor. Die Datenbank ist aber für zukünftige Erweiterungen in diese Richtung vorbereitet und bietet mit einer m zu n – Beziehung zwischen der Artikel-Kern-Tabelle 'artikel' und der Kategorien-Tabelle 'kategorien', sowie allen nötigen Attributen die Möglichkeit, hier einfach weiter zu entwickeln.

Zu guter letzt muss ein Artikel auch einer Bestellung zugeordnet werden können. Dies erledigt die Verbindungstabelle 'artikel\_bestellung' mit einer m zu n – Verbindung zwischen 'artikel' und 'bestellung'. Damit man bei einer Bestellung auch mehrere gleiche Artikel in einer Bestellung ordern kann, wird in der Verbindungstabelle zusätzlich noch die Anzahl bestellter Artikel und optional die gewählte Variation und Optionen "kompakt" gespeichert. Ziel war es in dieser Tabelle pro Artikel und Bestellung nur *eine* Zeile zu verwenden, deshalb werden dort alle Optionen nacheinander als String gespeichert.

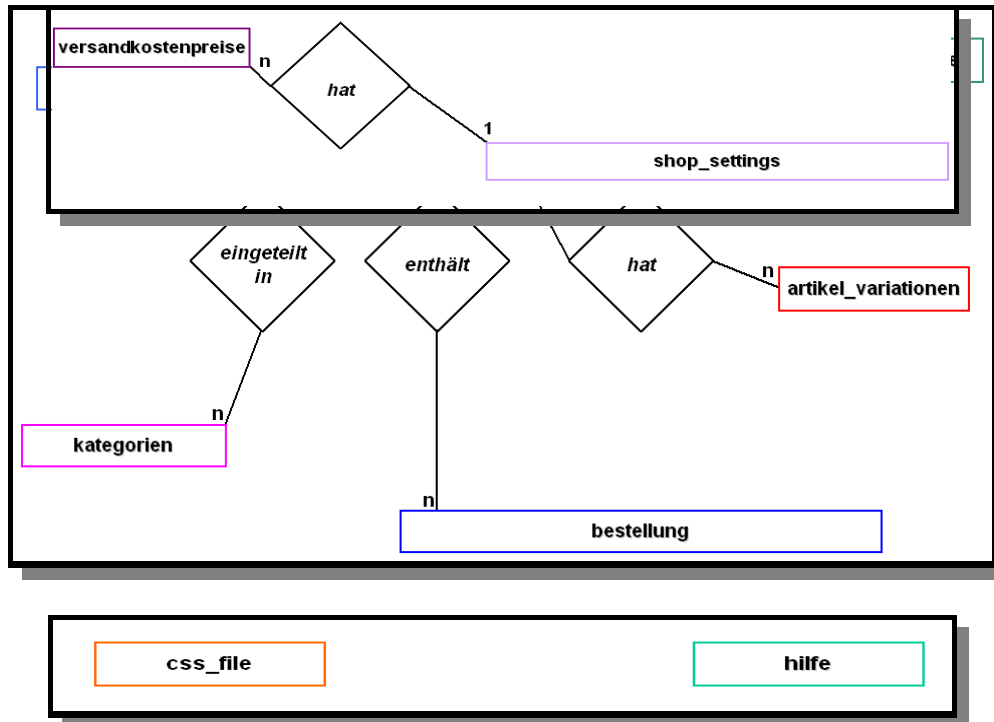
Die Tabelle 'bestellung' nimmt, wie ihrem Namen sicher leicht entnommen werden kann, die jeweiligen Bestelldaten auf. Die Bestellungstabelle ist auch ein zentrales Element in unserer Session-Verwaltung, da eigentlich alles über die Bestellung eines potentiellen Kunden abläuft (siehe Kapitel 8.5 Session-Management).

Von den oben genannten Tabellen unabhängig, haben wir zwei weitere zentrale Tabellen: 'shop\_settings' und 'versandkosten', die erste enthält in ihren Attributen alle Einstellungen, um den Shop konfigurieren zu können, die Zweite enthält pro Zeile je einen Preis, der für eine definierte Preis- oder Gewichtsspanne gültig ist. Auf diese Weise wird dem Shop-Betreiber ermöglicht, seine Versandkosten den Preisen oder Gewichten seiner Waren anzupassen. Leider konnte aus Zeitgründen auch diese sehr interessante Möglichkeit nicht mehr implementiert werden.

Zu guter letzt existieren noch zwei Tabellen, die in keinerlei Beziehung zu anderen Tabellen dieser Datenbank stehen. Zuerst wäre da die Tabelle 'hilfe'. Sie kann jeweils einer Hilfe-ID einen Hilfetext zuweisen. Wir dachten uns, auf diese Weise die Hilfe-Settings persistent und dynamisch für jede Seite sauber speichern zu können.

Die zweite Page heisst 'css\_file' und ist in der jetzigen Shop-Implementation noch nicht berücksichtigt. Da wir geplant haben, einen möglichst dynamischen und komplett in der Datenbank speicherbaren Shop zu kreieren, wäre diese Tabelle der Platz, wo die Einstellung zu jeder CascadingStyleSheet-Definition, die im Shop verwendet wird, gespeichert werden könnte. Im Moment sind die CSS-Optionen noch in einem File und statisch.

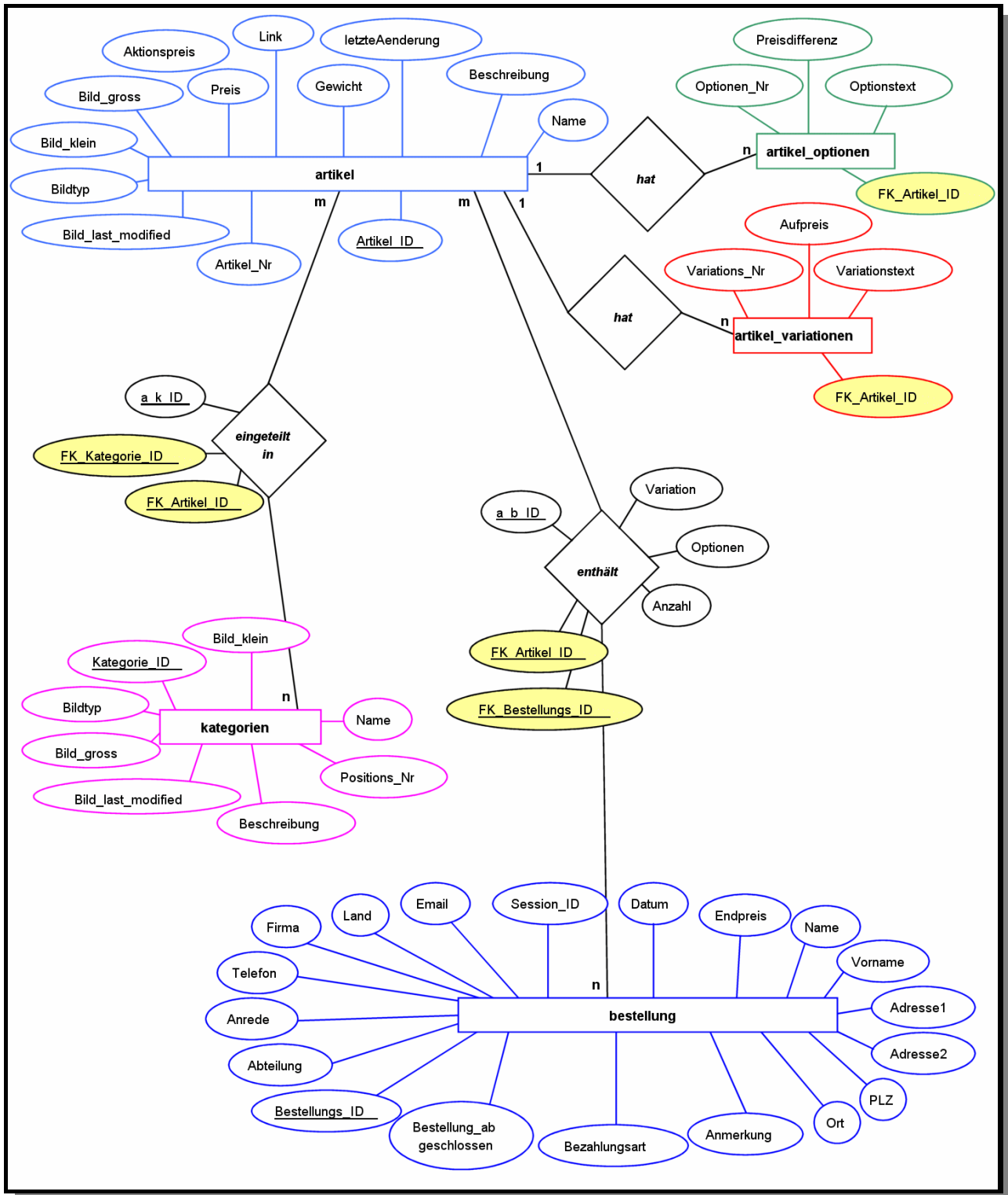
### 8.4.3 shopdb – Relationen



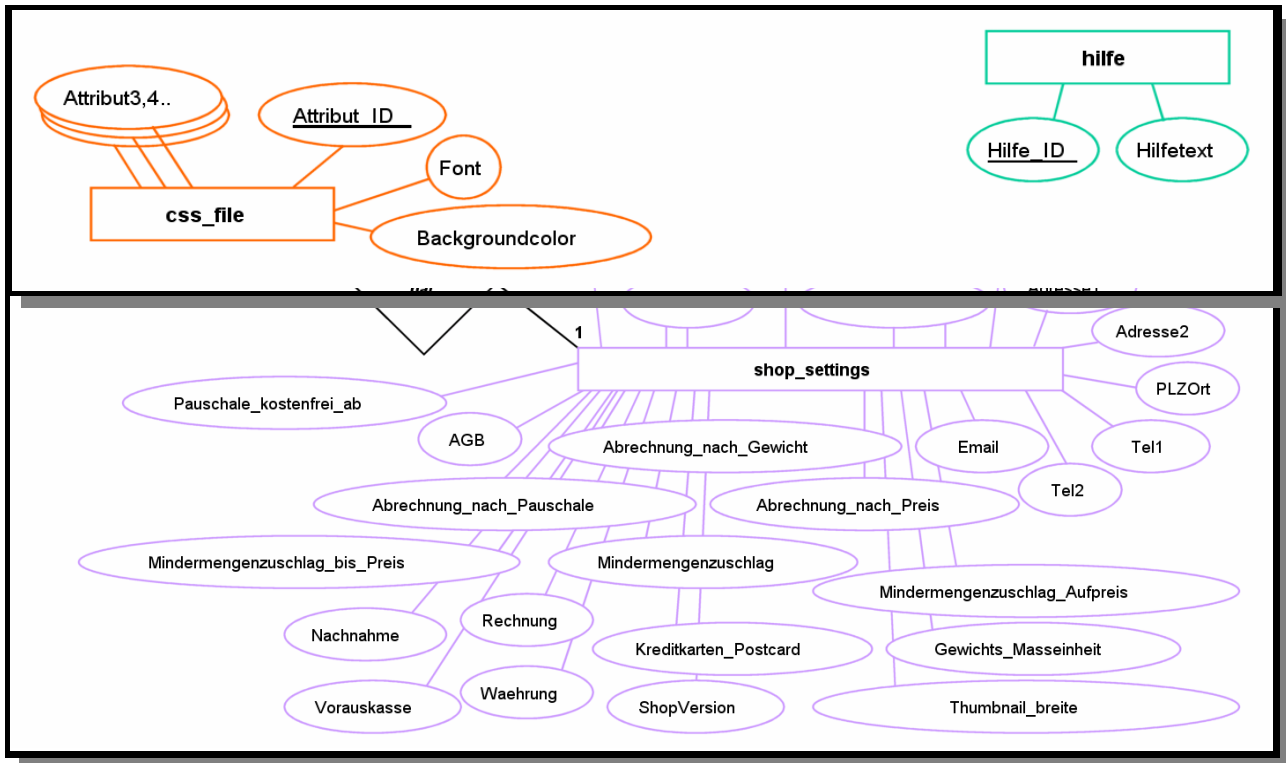
### 8.4.4 Entity Relationship Diagramm der Shop Datenbank

Da die ganze Tabelle inklusive aller Attribute nicht mehr sauber auf einer A4-Seite dargestellt werden kann, haben wir das ER-Diagramm in zwei Teile geteilt:

#### 8.4.4.1 ER-Diagramm Teil 1



8.4.4.2 ER-Diagramm Teil 2



### 8.4.5 Tabellen und ihre Attribute im Detail

Es folgen nun die Tabellen im Detail. Hier sieht man auch welche Datentypen für die jeweiligen Attribute verwendet wurden:

<b>artikel</b>	<b>kategorien</b>
<pre> Artikel_ID INT NOT NULL DEFAULT 0 AUTO_INCREMENT, Artikel_Nr CHAR(32) NOT NULL, Name CHAR(32) NOT NULL, Beschreibung VARCHAR(127) NOT NULL, Preis DOUBLE NOT NULL DEFAULT 0.0, Aktionspreis DOUBLE NOT NULL DEFAULT 0.0, Gewicht DOUBLE NOT NULL DEFAULT 0.0, Link VARCHAR(64), Bild_gross MEDIUMBLOB, /* Achtung: Bei anderer DB, Typ ev. = BLOB */ Bild_klein BLOB, Bildtyp VARCHAR(15), Bild_last_modified TIMESTAMP(14), letzteAenderung DATE, PRIMARY KEY (Artikel_ID), UNIQUE UC_Artikel_ID (Artikel_ID)                     </pre>	<pre> Kategorie_ID INT NOT NULL DEFAULT 0 AUTO_INCREMENT, Positions_Nr INT NOT NULL DEFAULT 0, Name CHAR(32) NOT NULL, Beschreibung VARCHAR(127) NOT NULL, Bild_gross MEDIUMBLOB, /* Achtung: Bei anderer DB, Typ ev. = BLOB */ Bild_klein BLOB, Bildtyp VARCHAR(15), Bild_last_modified TIMESTAMP(14), PRIMARY KEY (Kategorie_ID), UNIQUE UC_Kategorie_ID (Kategorie_ID), INDEX IDX_kategorien_1 (Name)                     </pre>

<b>bestellung</b>	<b>kategorien</b>
<pre> Bestellungs_ID INT NOT NULL DEFAULT 0                     AUTO_INCREMENT, Session_ID VARCHAR(32) NOT NULL DEFAULT 0, expired BIGINT NOT NULL DEFAULT 0, <i>/*In andern DBs ev. anderer Typ (10 Stellen) */ /*Unsere getesteten Session_IDs waren immer CHAR(32)... aber man weiss ja nie */ Datum Date, Endpreis DOUBLE NOT NULL DEFAULT 0.0, Firma CHAR(40) NOT NULL, Anrede CHAR(24) NOT NULL, Abteilung CHAR(40) NOT NULL, Vorname CHAR(32) NOT NULL, Name CHAR(32) NOT NULL, Adressel CHAR(64) NOT NULL, Adresse2 CHAR(64) NOT NULL, Telefon CHAR(16) NOT NULL, Email CHAR(128) NOT NULL, PLZ INT NOT NULL, Ort CHAR(64) NOT NULL, Land CHAR(40) NOT NULL, Anmerkung CHAR(255), Bezahlungsart CHAR(32) NOT NULL, Bestellung_abgeschlossen enum('N','Y') NOT NULL                                 DEFAULT                                 'N', PRIMARY KEY (Bestellungs_ID), UNIQUE UC_Bestellungs_ID (Bestellungs_ID) </i></pre>	<pre> Setting_Nr INT NOT NULL DEFAULT 0 AUTO_INCREMENT, MwStsatz DOUBLE NOT NULL DEFAULT 7.6, MwStpflichtig enum('N','Y') NOT NULL                     DEFAULT 'Y', MwStNummer INT NOT NULL DEFAULT 0, Name CHAR(48) NOT NULL DEFAULT 'Shop in PHP', Adressel CHAR(48) NOT NULL DEFAULT 'Adresse 1', Adresse2 CHAR(48) NOT NULL DEFAULT 'Adresse 2', PLZOrt CHAR(48) NOT NULL DEFAULT 'PLZ und Ort', Tell CHAR(16) NOT NULL DEFAULT 'Telefon Nummer1', Tel2 CHAR(16) NOT NULL DEFAULT 'Telefon Nummer2', Email CHAR(128) NOT NULL DEFAULT 'you@shop.com', Admin_pwd CHAR(16) NOT NULL DEFAULT 'machshop', Abrechnung_nach_Preis enum('N','Y') NOT NULL                     DEFAULT                     'N', Abrechnung_nach_Gewicht enum('N','Y') NOT NULL                                 DEFAULT                                 'N', Abrechnung_nach_Pauschale enum('N','Y') NOT NULL                                 DEFAULT                                 'Y', Pauschale_kostenfrei_ab DOUBLE NOT NULL                     DEFAULT 0.0, Mindermengenzuschlag enum('N','Y') NOT NULL                     DEFAULT                     'N', Mindermengenzuschlag_bis_Preis DOUBLE NOT NULL                                 DEFAULT                                 0.0, Mindermengenzuschlag_Aufpreis DOUBLE NOT NULL                                 DEFAULT                                 0.0, Rechnung enum('N','Y') NOT NULL DEFAULT 'Y', Nachnahme enum('N','Y') NOT NULL DEFAULT 'Y', Vorauskasse enum('N','Y') NOT NULL DEFAULT 'N', Kreditkarten_Postcard enum('N','Y') NOT NULL                     DEFAULT                     'N', Waehrung CHAR(4) NOT NULL DEFAULT 'CHF', ShopVersion CHAR(48) NOT NULL DEFAULT 'Mai 2001, DB Version v.1.03', Gewichts_Masseinheit CHAR(16) NOT NULL                     DEFAULT 'kg', Thumbnail_Breite INT NOT NULL DEFAULT 100, AGB TEXT, <i>/* Max. 65535 Bytes gross in MySQL */</i> </pre>

```
max_session_time BIGINT, #/* In anderen DBs
ev.
                               ein anderer Typ (10
Stellen) */
PRIMARY KEY (Setting_Nr),
UNIQUE UC_Setting_Nr (Setting_Nr)
```

**Z:W**

<b>artikel_kategorie</b>	<b>artikel_bestellung</b>
<pre>a_k_ID INT NOT NULL DEFAULT 0 AUTO_INCREMENT, FK_Artikel_ID INT NOT NULL, FK_Kategorie_ID INT NOT NULL, FOREIGN KEY (FK_Artikel_ID) REFERENCES                     artikel (Artikel_ID), FOREIGN KEY (FK_Kategorie_ID) REFERENCES                     kategorien (Kategorie_ID), PRIMARY KEY (a_k_ID,FK_Artikel_ID,FK_Kategorie_ID)</pre>	<pre>a_b_ID INT NOT NULL DEFAULT 0 AUTO_INCREMENT, FK_Artikel_ID INT NOT NULL, FK_Bestellungs_ID INT NOT NULL, Anzahl INT NOT NULL DEFAULT 1, Variation VARCHAR(128) NOT NULL                     DEFAULT 'Keine Variation gewaehlt', /* Aufbau einer Variation: Variationstext p Aufpreis */ Optionen VARCHAR(128) NOT NULL                     DEFAULT 'Keine Optionen gewuenscht', /* Aufbau einer Option: Optionstext p Preisdifferenz */ FOREIGN KEY (FK_Artikel_ID) REFERENCES                     artikel (Artikel_ID), FOREIGN KEY (FK_Bestellungs_ID) REFERENCES bestellung (Bestellungs_ID), PRIMARY KEY (a_b_ID,FK_Artikel_ID,FK_Bestellungs_ID) /* Anm. Das Trennzeichen ist ALT+0254 */</pre>

<b>artikel_optionen</b>	<b>artikel_variationen</b>
<pre>Optionen_Nr INT NOT NULL DEFAULT 1, Optionstext CHAR(64), Preisdifferenz DOUBLE NOT NULL DEFAULT 0.0, FK_Artikel_ID INT NOT NULL, FOREIGN KEY (FK_Artikel_ID) REFERENCES                     artikel (Artikel_ID)</pre>	<pre>Variations_Nr INT NOT NULL DEFAULT 1, Variationstext CHAR(64), Aufpreis DOUBLE NOT NULL DEFAULT 0.0, FK_Artikel_ID INT NOT NULL, FOREIGN KEY (FK_Artikel_ID) REFERENCES                     artikel (Artikel_ID)</pre>

<b>hilfe</b>	<b>css_file</b>
<pre>Hilfe_ID CHAR(50) NOT NULL, Hilfetext TEXT, PRIMARY KEY (Hilfe_ID), UNIQUE UC_Hilfe_ID (Hilfe_ID)</pre>	<pre>Attribut_ID INT NOT NULL DEFAULT 0                     AUTO_INCREMENT, Font CHAR(32) NOT NULL DEFAULT 'Helvetica', Backgroundcolor CHAR(5) NULL DEFAULT 'FFF', PRIMARY KEY (Attribut_ID), UNIQUE UC_Attribut_ID (Attribut_ID)</pre>

<b>versandkosten</b>
<pre>Von_Bis_ID INT DEFAULT 0 NOT NULL AUTO_INCREMENT, Von DOUBLE DEFAULT 0.0, Bis DOUBLE DEFAULT 100.0, Betrag DOUBLE DEFAULT 0.0, FK_Setting_Nr INT NOT NULL, FOREIGN KEY (FK_Setting_Nr) REFERENCES shop_settings (Setting_Nr), PRIMARY KEY (Von_Bis_ID), UNIQUE UC_Von_Bis_ID (Von_Bis_ID)</pre>

## 8.5 SQL-Query Handling

Diese Kapitel hängt eigentlich stark mit der Security (Kapitel 8.3) zusammen. Neben diesen Aspekten haben wir uns nach den ersten Gehversuchen mit PHP und MySQL dafür entschieden, anstatt die Query selbst in die Funktion einzubinden, dort nur Variablen zu benutzen. Dieser Ansatz hat verschiedene Vorteile: Erstens werden so die SQL-Queries von den Funktionen entkoppelt, was entscheidend ist, sobald man gedenkt eine andere Datenbank einzusetzen (und sich somit die SQL-Syntax leicht ändert). Weiter hat man so alle Queries zentral abgelegt. Bei uns in zwei Dateien und das für den ganzen Shop. Zu guter Letzt lässt sich so unsere Security gut implementieren (siehe Kapitel 8.3).

Als Anmerkung sei hier noch erwähnt, dass sich die Konvertierung der Queries am einfachsten und schnellsten mit einem Tool erledigen lässt (Bsp. siehe Kapitel 8.8).

## 8.6 DB-Wrapper (*database.php*)

Von unserem Dozenten und Praktikumsbetreuer erhielten wir einen Datenbank Wrapper für Sybase. Dieser Wrapper stellt auf einer Seite eine PHP Datenbank-Connection zur Verfügung, bietet die zum Datenbank-Handling nötigen Funktionen an (Query ausführen, Resultset auswerten, Resultzeilen zählen u.s.w.) und hat auf der anderen Seite für jede Datenbank, auf die zugegriffen werden soll, verschiedene Klassen. Wie schon erwähnt, war dieser Wrapper für Sybase schon fertig programmiert und getestet. Wir hatten wirklich leichtes Spiel und konnten den DB-Wrapper um MySQL-Zugriffe erweitern. Er ist also jetzt für Sybase und MySQL einsetzbar und getestet.

Leider hat auch hier die knappe Zeit ihre Spuren hinterlassen. Im Laufe der Shop-Entwicklung haben wir herausgefunden, dass wir beim Einfügen eines neuen Artikels seine `Artikel_ID` haben sollten (Funktion `newArtikel` zum Einfügen der Optionen und Variationen). Dies lässt sich per MySQL sehr leicht und effizient erledigen. Sobald bei einem Insert ein neuer Autowert generiert wurde, kann dieser mit dem Kommando `mysql_insert_id()` ausgelesen werden. Wir haben nun die Exec-Funktion der `TMySQLDatabase`-Klasse in `database.php` so erweitert, dass sie nicht nur `true` oder `false` zurückgibt sondern:

- `false`, wenn Exec nicht korrekt ausgeführt werden konnte
- `true`, wenn kein Insert mit Auto-Increment Funktion ausgeführt wurde
- Den zuletzt erzeugten Auto-Increment Wert, wenn es bei einem Insert so einen gegeben hat.

Diese Erweiterung haben wir bei der `TSybaseDatabase`-Klasse nicht vorgenommen weil wir keine analoge Funktion finden konnten. Man muss also diesen neuen Auto-Increment Wert zuerst per Select-Anweisung anfordern und dann zurückgeben. Will man den Shop also unter Sybase laufen lassen, *muss* diese Funktion zuerst nachprogrammiert werden.

## 8.7 Datenbank Installations- / Deinstallations-SQL-Skripts

Damit man den Shop installieren kann, muss man zuerst die Datenbank aufsetzen. Dies geschieht mit zwei Installations-Skripts (SQL-Skripts). Diese sind momentan nur für MySQL erstellt worden. Will man den Shop nun z.B. mit Sybase als Datenbank laufen lassen, so müssen auch diese Installations-Skripts und auch das Deinstallations-Skript auf die Sybase-SQL-Syntax angepasst werden (Strichpunkt wird zu `go` und ähnliche kleine Änderungen). Wer hier nicht genügend Kenntnisse hat, oder die Zeit zu aufwändig,

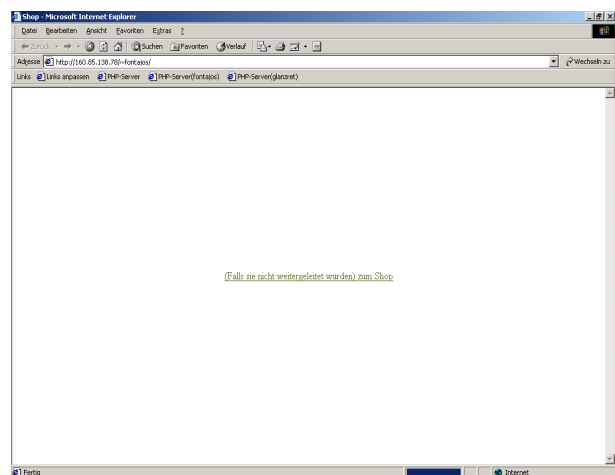
respektive zu teuer ist, kann am einfachsten ein Tool verwenden, das diese Konvertierung automatisiert. Bewährt hat sich hier die Kombination aus DATANAMICs DeZign for databases mit DATANAMICs ImportER Scripts für Windows ([www.datanamic.com](http://www.datanamic.com)). Diese Tools vermögen weit mehr als das einfache Konvertieren von SQLs zwischen Datenbanken-Syntax und kosten zusammen nur \$180.00.

Was einem aber leider kein Tool abnimmt, ist der erste Teil des `shopdb_create.sql`-Skripts, es müssen zwei Datenbank-Benutzer angelegt werden die je unterschiedliche Rechte haben. Diese Aktion ist in jeder Datenbank anders und muss entsprechend angepasst werden.

## 8.8 Shoplout (HTML / CSS / Javascript)

### 8.8.1 HTML Shopbenutzer

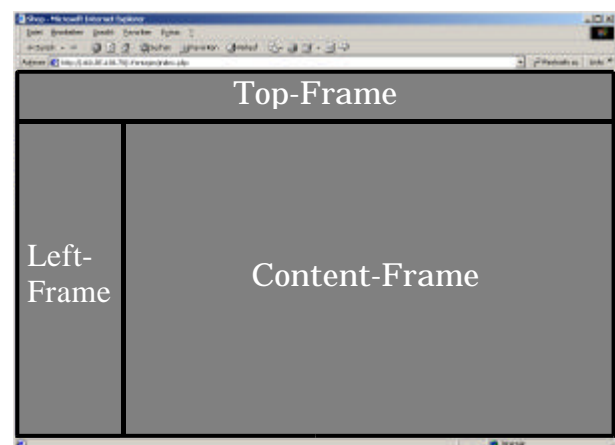
Zuerst wird die Datei `index.html` vom Webserver geladen und an den Browser ausgegeben. Diese Datei hat die Funktion, den Shopuser sofort auf `index.php` weiterzuleiten, welches sich im Shopverzeichnis befindet. Unterstützt der Browser diese Funktion nicht, kann sich der Surfer durch anklicken eines Links zum Shop navigieren. Ausserdem kann diese Datei noch dazu verwendet werden, Informationen für Suchrobots bereitzustellen. Wird der Shop in eine bestehende Homepage eingebettet, kann `index.html` weggelassen und der Shop direkt mit einem Link auf `index.php` aufgerufen werden.



Danach wird von `index.php` das Frameset aufgebaut und die Shopstartseite dargestellt. Der gesamte Shop (Usersicht) wird nun in diesem Dreierframeset dargestellt. Das oberste Frame „Top-Frame“ wird verwendet, um Funktionen anzubieten, die jederzeit ausgeführt werden können (Warenkorb anzeigen, Zur Kasse, Hilfe, Suchfunktion) und um den Shopnamen permanent anzuzeigen.

Im linken „Left-Frame“ werden die Artikelkategorien zur Auswahl angezeigt.

Dieses Frame wird einmal am Anfang gefüllt, danach werden keine Änderungen mehr vorgenommen. Das grösste Frame „Content-Frame“ wird zur Userinteraktion verwendet. Es ist das einzige Frame, dessen Inhalt während der Navigation im Shop verändert wird. Folgende Ausgaben werden im Content-Frame dargestellt:

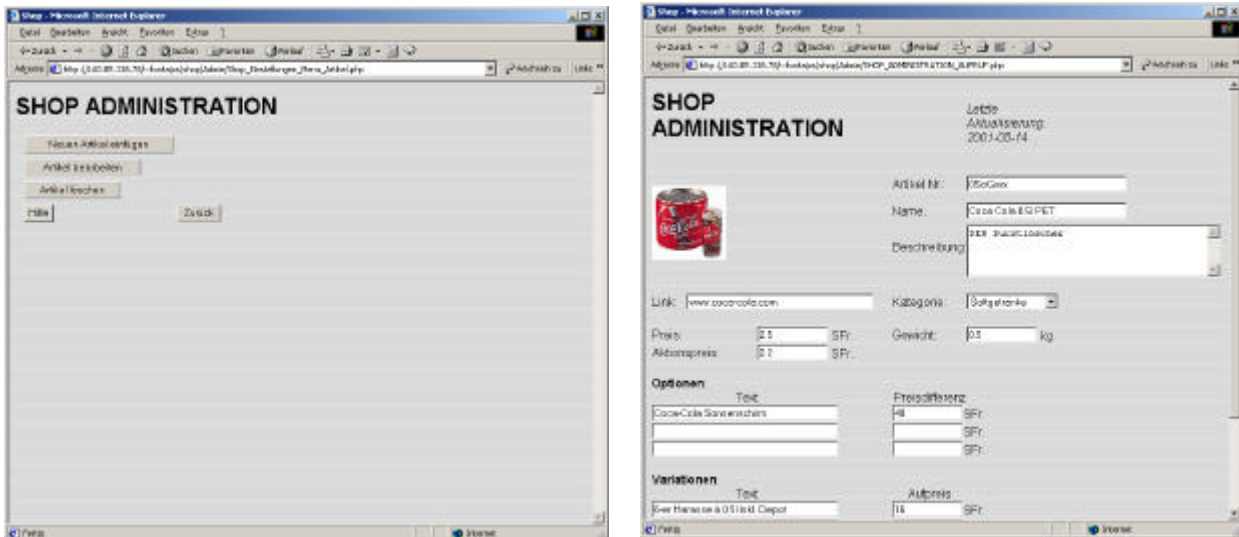


- Willkommensbildschirm
- Warenkorb
- Kategorienansicht (Artikeldetails)
- Adresseingabeformulare
- Suchmaske
- Suchergebnisse
- Hinweise bei Falscheingaben
- Bestätigungen

Beim Anklicken eines Artikelthumbnails und des Hilfe-Buttons, wird ein neues Fenster geöffnet. Dieses POP-UP ist mittels JavaScript realisiert. Die entsprechenden Java-Script Funktionen befinden sich direkt im HTML-Quelltext der jeweiligen Sites. Die POP-UP-Fenster können entweder per eingeblendetem Link oder mittels Schliessen-Button wieder geschlossen werden.

### 8.8.2 HTML Shopadministrator

Das Benutzerinterface für den Shopadministrator ist gestalterisch sehr einfach gehalten. Alle Interaktions- und Informationsausgaben werden in einem Frame dargestellt. Der Administrator hat die Möglichkeit mittels eingeblendeten Buttons (Links) komplett im Administrationstool zu navigieren.



### 8.8.3 CSS

Alle CSS-Darstellungsinformationen sind in dem File `shopstyles.css` abgelegt. Jedes Frame hat eine eigene Darstellungsklasse, in welchem das Hintergrundbild, die Hintergrundfarbe, spezielle Linkdarstellungen und Textfarben definiert sind. Globale Definitionen wie Fontset, Fontsize und Title-Sizes sind im Body-Tag definiert und nicht spezifisch einer Frameklasse zugeordnet.

Das CSS-File kann mit einem beliebigen Editor (auch vi) verändert werden. Alle Einstellungen sind als Kommentar erläutert. Somit ist es möglich, mit ein paar wenigen Veränderungen den Shop komplett auf das firmeneigene Corporate-Design abzustimmen.

### 8.10 Test

Da der Administrationsmodus noch nicht fertig ausprogrammiert ist, haben wir auf einen detaillierten, dokumentierten Funktionstest dieser Funktionen verzichtet.



<b>Testfall</b>	<b>Erwartet</b>	<b>Ergebnis</b>
<b>Warenkorb anzeigen</b>	Alle Artikel im Warenkorb Gewählte Optionen und Variationen angezeigt Einzelpreis angezeigt Preis angezeigt Gesamtpreis angezeigt <Zur Kasse> anzeigen <löschen> für jeden Artikel anzeigen Top-Frame und Left-Frame werden nicht verändert	ok ok ok ok ok ok ok ok
<b>Artikel aus Warenkorb entfernen</b>	Meldung, dass Artikel aus Warenkorb entfernt wurde Artikel auch wirklich entfernt <Warenkorb> anzeigen Top-Frame und Left-Frame werden nicht verändert	ok ok ok ok
<b>Zur Kasse</b>	Eingabeformular für Lieferadresse angezeigt Falls Informationen schon eingegeben wurden, diese anzeigen. Erlaubte Zahlungsarten werden angezeigt. <weiter> anzeigen Top-Frame und Left-Frame werden nicht verändert	ok ok ok ok ok
<b>Bestellung anzeigen</b>	Warnung, falls geforderte Adressangaben fehlen oder E-Mail-Adresse nicht korrekt ist Bestellung druckbar anzeigen ..mit Adressinformationen ..mit Warenkorb ..ohne "Artikel aus Warenkorb entfernen" Möglichkeit <Bestellung absenden> anzeigen Top-Frame und Left-Frame werden nicht verändert	ok ok ok ok ok ok ok ok
<b>Bestellung absenden</b>	AGB anzeigen <AGB akzeptieren> anzeigen Top-Frame und Left-Frame werden nicht verändert	ok ok ok
<b>Bestellung abgesendet</b>	Dankesmeldung ausgeben E-Mail Adresse des Shopbetreibers anzeigen Bestätigungs-E-Mail an Kunden Bestellungs-E-Mail an Shopbetreiber Warenkorb geleert Top-Frame und Left-Frame werden nicht verändert	ok ok ok ok ok ok

<b>Testfall</b>	<b>Erwartet</b>	<b>Ergebnis</b>
<b>Rücksprung auf Eingabeformular für Lieferadresse</b>	Alle zuvor eingegebenen Adress- und Zahlungsangaben werden in das Formular eingefüllt Top-Frame und Left-Frame werden nicht verändert	<b>ok</b>  <b>ok</b>

Die umseitig gedruckte Grafik zeigt alle Navigationsmöglichkeiten des Shopbenutzers. Ein Pfeil bedeutet, dass wir diese Aktion erfolgreich getestet haben. Die grünen Pfeile sind Zustandsspezifische Aktionen, die Roten sind Aktionen, welche von jeder Stelle innerhalb des Shops jederzeit ausgeführt werden können. Natürlich haben wir diese auch überall getestet. Um die Übersichtlichkeit zu wahren, zeichneten wir nur jeweils einen Stellvertreter dieser Aktionen ein (ein roter Pfeil steht also als Stellvertreter für viele andere, die alle an das gleiche Ziel / die gleichen Ziele zeigen würden).

Für die Richtigkeit der Testergebnisse,

*R. Glanzmann*

R. Glanzmann

Winterthur 15.05.2001





## 9. Bedienung

### 9.1 Bedienung Shop

#### 9.1.1 Bedienung Shop

Maximale Usability war eines unserer Hauptziele. Da jede Person, die einen Internetanschluss hat, zum Benutzer unserer Software werden kann, musste der Shop selbsterklärend aufgebaut werden. Wir hätten unser Ziel weit verfehlt, wenn wir hier eine Bedienungsanleitung abdrucken müssten, denn wer kauft schon an einem Ort ein, wo er zuerst eine Bedienungsanleitung lesen muss.

#### 9.1.2 Bedienung Administrationstool

Da das Administrationstool noch nicht komplett ausprogrammiert ist, wird hier noch auf eine detaillierte Bedienungsanleitung verzichtet. Wir möchten aber gerne die Funktionen aufzeigen, die im momentanen Entwicklungsstand des Shops möglich sind.

In den Administrationsmodus gelangt man, indem man auf das Sternchen im Top-Frame (befindet sich links vom Knopf „Warenkorb anzeigen“) klickt. Danach wird man bei ordnungsgemäßer Installation des Shops aufgefordert, sein Loginname und das Passwort einzugeben.

Das Administrationstool ist nach dem Step-by-Step Verfahren aufgebaut. Dem Administrator wird jederzeit die Möglichkeit geboten, eine Aktion abzubrechen, oder zum vorangegangenen Menüpunkt zurückzukehren.

<b>Schon implementierte Funktionen</b>	
<b>Menu</b>	<b>Einstellungen / Funktionen</b>
Shop Einstellungen	Kontaktadresse Shopbetreiber Shopname Versandoptionen Bezahlungsarten Shop Konfiguration Allgemeine Geschäftsbedingungen
Kategorien	Neue Kategorie Kategorien bearbeiten Kategorie löschen
Artikel	Neuen Artikel einfügen Artikel bearbeiten Artikel löschen Artikelbild hinzufügen Artikelbild löschen

<b>Noch nicht implementierte Funktionen</b>	
<b>Menu</b>	<b>Einstellungen / Funktionen</b>

<b>Noch nicht implementierte Funktionen</b>	
Versandsettings	Einstellung der Versanddaten
Bestellungen	Eingegangene Bestellungen ansehen Eingegangene Bestellungen verändern Eingegangene Bestellungen löschen

**Screenshots Administrationstool:**

**SHOP ADMINISTRATION**

Shopname:   
 Adresse 1:  E-Mail Adresse des Shops:   
 Adresse 2:  Tel. 1:   
 PLZ und Ort:  Tel. 2:

MwSt pflichtig  MwSt Nummer:  MwSt Prozentsatz:  %

Abrechnung nach Postfach:  Pauschale kostenlos ab:  SFr.  
 Abrechnung nach Preis:  Mindestmengenzuschlag:  SFr.  
 Abrechnung nach Gewicht:  Mindestmengenzuschlag bis Preis:  SFr.  
 Gewicht/Masseinheit:  Aufpreis:  SFr.

**Bezahlungsarten**  
 Per Rechnung:  Währung:   
 Per Kreditkarte / Postcard:

**Shop-Konfiguration**  
 Administrator Passwort:   
 Breite der Bilder in Pixel:  ACHTUNG: Erst aktiv für Bilder die nach dem Ändern aufgefüllt werden!  
 maximale Session Zeit:  Sek. ACHTUNG: Eine weitere Funktion nach oben in der Menü-Leiste!

**Allgemeine Geschäftsbedingungen**  
 Allgemeine Geschäftsbedingungen:

Zurücksetzen | Speichern | Abbrechen | Hilfe

**SHOP ADMINISTRATION** Letzte Aktualisierung: 2001-05-14

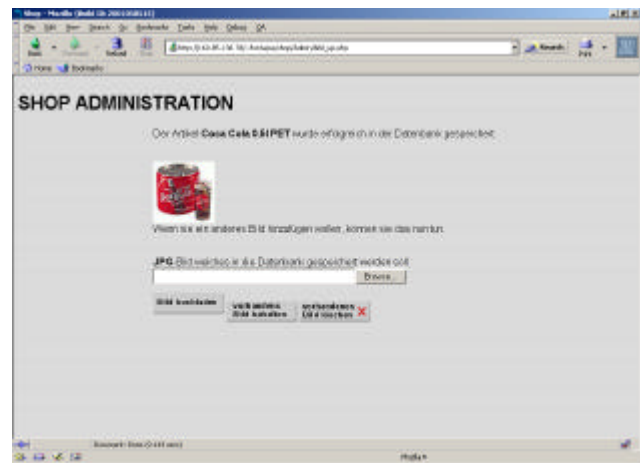
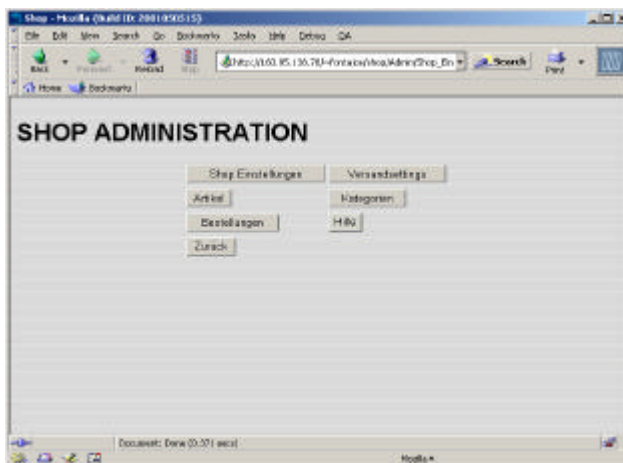
 Artikel Nr.:   
 Name:   
 Beschreibung:

Link:  Kategorie:    
 Preis:  SFr. Gewicht:  kg  
 Aktionspreis:  SFr.

**Optionen**  
 Text:  Preisdifferenz:  SFr.  
 SFr.  
 SFr.

**Variationen:**

Zurücksetzen | Weiter | Zurück | Hilfe



**9.2 Installation des Shops**

Was muss vor der Installation des Webshops gemacht werden:

- Webserver aufsetzen
- PHP4 installieren (entweder als Webserver-Modul oder als CGI-Version)

- MySQL installieren

Nun geht es an die eigentliche Installation des Webshops:

Zuerst müssen alle benötigten Daten von der Installations-CD in die richtigen Verzeichnisse kopiert werden (wir gehen hier von einem UNIX-System aus):

- Das Verzeichnis Datenbank kann in das User-Verzeichnis kopiert werden
- Das Dokumentations-Verzeichnis wird nicht zwingend benötigt, vielleicht ist es aber manchmal noch ganz nützlich eine Shop-Dokumentation gleich beim Shop zu haben.
- Den Inhalt des Verzeichnisses Webshop muss in das entsprechende Web-Verzeichnis des Webservers kopiert werden (Unix-Apache-Standard: ~/htdocs/)

Als ein fiktives Beispiel könnte nach dem Kopieren die Verzeichnisstruktur des Users der den Shop betreiben will wie folgt aussehen:

```
/home/hans_muster/Datenbank/
/home/hans_muster/Dokumentation/
/home/hans_muster/htdocs/index.html
/home/hans_muster/htdocs/index.php
/home/hans_muster/htdocs/shop/
```

Bevor wir die Datenbank einrichten, müssen wir noch die Datenbank-Zugriffspasswörter ändern, sonst kann jeder der diese Dokumentation liest, gleich als Administrator auf den entsprechenden Shop zugreifen.

Im Datenbankverzeichnis in der Datei shopdb\_create.sql in den Zeilen 31 und 33 muss das Passwort geändert werden:

```
Zeile 31      VALUES("localhost","shopadmin",PASSWORD('machshop'));
Zeile 33      VALUES("localhost","shopuser",PASSWORD('gibshop'));
```

Weiter müssen dieselben Passwörter jeweils noch im Shop geändert werden:

1.)in der Datei htaccess/shop/initialize.php auf de Zeile 39:

```
Zeile 39  $Database = new TMySQLDatabase("localhost", "shopdb", "shopuser", "gibshop");
```

2.)in der Datei htaccess/shop/Admin/ADMIN\_initialize.php auf der Zeile 42:

```
Zeile 42  $Admin_Database = new TMySQLDatabase("localhost", "shopdb", "shopadmin", "machshop");
```

Nun muss die Datenbank des Webshops 'shopdb' eingerichtet werden. Dazu geht man ins Datenbank-Verzeichnis auf dem Server. Bevor man nun die zwei Installations-Skripte zur Erstellung der Shop-Datenbank startet muss man das root-Passwort der MySQL-Datenbank kennen, resp. das Passwort eines Users, der in der Datenbank die Erlaubnis hat neue Datenbanken zu erstellen. Im weiteren nennen wir diesen User root.

Man startet nun zwei Skripte wie folgt:

```
mysql -u root -p <./shopdb_create.sql
mysql -u root -p <./shopdb_insert.sql
```

Nach jeder Eingabe muss man mit dem root-Passwort bestätigen, dass man die entsprechende Berechtigung hat, dieses Skript ausführen zu dürfen.

Das erste Skript legt nun die shopdb an und fügt die beiden Shop-Benutzer shopuser,

shopadmin mit ihren Rechten an. Danach werden alle Tabellen mit ihren Attributen erstellt.

Nun kommt das zweite Script und fügt einen Beispiel-Shop in die shopdb ein. So sieht ein potentieller Shop-Betreiber gleich mal das Aussehen und kann sich in diesem Demo-Shop frei bewegen. Einzig Bilder sind noch keine drin. Dieser Shop ist so klein, dass er ihn auch schnell mit seinen Daten überschrieben hat, so dass man nicht zuerst noch Stunden lang einen nicht gewünschten Shop löschen muss.

Bevor man jetzt den Shop ausprobieren will, muss man noch etwas tun was auch uns etwas widerstrebt: Man muss alle Shop-Dateien im htaccess-Verzeichnis als dem UNIX-root gehörend markieren.

Wieso dieser umständliche Schritt? Da wir den PHP-Safe-Mode aktiviert haben, darf nur derjenige User in ein Verzeichnis schreiben und lesen, welchem das entsprechende Verzeichnis gehört. /tmp/ gehört halt nun mal root und somit darf laut dem Safe-Mode nur root dort hinein schreiben! Beim Bild in die Datenbank laden wird dieser 'Dateisystem-Zwischenschritt' aber benötigt, somit muss die Datei bild\_up.php root gehören! Der 'Wurm' zieht sich aber weiter. Da laut dem Safe-Mode Scripte eines x-beliebigen Users keine Scripte die root (UID=0) gehören includen darf, muss man wohl oder übel:

- entweder den Safe-Mode ausschalten (in der php.ini)
- oder alle Shop-Dateien per (chown -R root \*) die UID 0 zuweisen.

Das ganze ist ein Tribut an die Idee 'alles in der Datenbank'. Gegen Ende der Praktikumsarbeit haben wir beim Session-Management noch eine Möglichkeit gefunden, wie man dieses Zugriffsrechte-Problem vielleicht über den session.save\_handler. Die Zeit hat aber dafür nicht mehr gereicht.

Nun kann man den Shop nach belieben ausprobieren und wenn man in konfigurieren will (Im Administrations-Teil des Shops), so kann man auf den kleinen Stern links neben dem Warenkorb-Symbol klicken.

Wer seinem Shop noch andere Farben oder einen anderen Font gönnen will, muss das vorderhand noch in der statischen, vom Backup nicht berücksichtigten shopstyles.css im Shop-Verzeichnis tun.

### 9.3 Backup

Hier zeichnet sich nun der Vorteil unseres Konzepts aus: Der *ganze* Shop ist in der Datenbank abgebildet, inkl. Konfiguration, Daten und Bilder. Wenn man also ein Backup der Datenbank erstellt, hat man zugleich auch ein Backup des kompletten Shops.

Ein Backup einer MySQL Datenbank lässt sich am einfachsten über die MySQL-Tools mysqldump (sichern) und mysqlimport (einlesen) erledigen. Sie befinden sich im Unterverzeichnis bin der MySQL-Datenbank.

Wir hatten ursprünglich geplant für das Backup einen dritten Benutzer einzurichten, der die Rechte hat die Datenbank herunter- und hochzufahren. Dieser hätte dann in einem Shell-Script die Datenbank gestoppt (oder zumindest ein READ LOCK ausgelöst), die Daten kopiert und die Datenbank wieder freigegeben / -gestartet. Leider hat auch hierzu die Zeit nicht mehr gereicht.

## 10. Erfahrungen

### 10.1 Datenbanken allgemein

Auch im Datenbankbereich haben wir viel profitiert. Die zum Teil doch komplexen Queries (mit verschachteltem `LEFT JOIN` und so) hatten es in sich. Heute können wir aber mit Fug und Recht behaupten, dass wir wissen, wie man ein ER-Diagramm inkl. den verschiedenen Beziehungstypen verstehen und implementieren muss. Uns fielen sogar Grenzen von MySQL auf an die wir sonst wohl nicht so schnell herangekommen wären (z.B. kein `INSERT` mit `SELECT` im `VALUE` Wert und `WHERE` Klausel). Wie man eine Datenbank anbindet und sie benutzt, ist uns dank der vielen Praxis damit auch ziemlich vertraut. Sogar Security Aspekte einer Datenbank, im speziellen jetzt bei MySQL, haben wir kennen gelernt.

### 10.2 Bilder in der Datenbank speichern

Am Anfang stellten wir uns die Frage, ob wir die Artikelbilder und Thumbnails direkt in die Datenbank, oder im Filesystem ablegen sollten. Wir entschlossen uns, die Bilder direkt in der Datenbank zu speichern, da bei einem Datenbankbackup gleich auch die Bilder mitgesichert werden können. So kann der ganze Shop (momentan noch abgesehen von der CSS-Datei) nach einem Hardwaredefekt wieder als ein Datenbankrecovery zurückgeladen werden.

Die Implementation dieser Lösung gestaltete sich aufwändiger, als wir bei der Planung angenommen hatten. So müssen die Bildinformationen (z.B. Bildtyp) getrennt von den eigentlichen Bilddaten in einer eigenen Spalte in der Datenbank gespeichert werden. Da PHP keine Funktionen bietet, die einem das Abspeicherprozedere in die Datenbank abnimmt, mussten wir die Bildinformationen selbst für die Speicherung in der Datenbank aufbereiten. Dazu kommt, dass die Funktion, welche wir zur Verkleinerung des Artikelbildes (Thumbnailerzeugung) brauchen, das Bild nicht als String zurückliefern kann (nur Ausgabe an Browser oder an Filesystem), Daruch mussten wir einen Umweg über das Filesystem gehen, um die Bilder überhaupt in die Datenbank zu bekommen, was wiederum mit Zugriffsrechtproblemen verknüpft war.

Schliesslich haben wir es jedoch geschafft, dass der Shopadministrator ein beliebig grosses Bild hochladen kann (nur JPEG), welches dann auf eine voreingestellte Thumbnailbreite verkleinert wird. Das Produktbild und das Thumbnail werden in der Datenbank abgespeichert, und können aus dieser wieder dargestellt werden.

Jedoch würden wir die Daten bei einer Neuprogrammierung nicht mehr in die Datenbank speichern, da gemäss verschiedenen Quellen daraus ein erheblicher Performanceverlust des Servers resultiert. Es wird sogar empfohlen, bei gut frequentierten Shops die Bilder direkt von einem eigenständigen Bildserver zu beziehen.

In der jetztigen Shopversion speichern wir die Bilder direkt in der Artikeltabelle. Zusammen mit allen anderen Informationen wie Preis, Beschreibung,... Dies ist aus zwei Gründen nicht sinnvoll:

- MySQL wird sehr ineffizient, wenn man grosse BLOBs zusammen mit sehr kleinen Objekten in der gleichen Tabelle speichert
- Kann man mit einem Datenbankbearbeitungstool wie PHPed solche Tabellen sehr schlecht bearbeiten, da durch die grossen BLOBs die Übersichtlichkeit

verloren geht.

Zusammenfassend ist zu sagen, dass es bei nicht so stark frequentierten Shops sicher eine Königslösung ist, die Bilder in die Datenbank zu speichern, da so wirklich alle Shopdaten von einer Quelle kommen. Speichert man jetzt noch die CSS-Daten in einer Tabelle, hat man aus Backupsicht den perfekten Shop. Sobald es jedoch um Performancefragen geht, ist die Speicherung wirklich aller Einstellungen und Daten in der Datenbank nicht mehr empfehlenswert.

### **10.3 Das Projekt**

Da es unsere erste Projektarbeit in diesem Umfang überhaupt war, haben wir wirklich vieles dazugelernt. Weil wir von den verwendeten Technologien noch keinerlei Praxiserfahrungen hatten, konnten wir auch nicht so genau abschätzen wie unser Zeitplan ausfallen würde, was die Planung doch ziemlich schwierig gestaltete (siehe Kapitel 6). Wir haben aber dort schon gemerkt, dass es knapp werden würde und einige Punkte der Aufgabenstellung als 'optional' markiert sowie die Offline-Shop-Administration zugunsten einer Online-Version ausgetauscht.

Rückblickend müssen wir sagen, dass das avisierte Projekt, so wie es jetzt programmiert wurde zu gross war für eine 7½ wöchige Projektarbeit mit 12 Stunden pro Woche (und weiteren 12 erwarteten Stunden...). Es gab zwei Grundsatzentscheide die uns zwar sehr viel Erfahrungen brachten uns aber auch ungemein viel Zeit kosteten:

- Höchstmögliche Dynamisierung des Shops: u.a. dass ein Artikel beliebig viele Optionen / Variationen haben kann.
- Bilder in die Datenbank ablegen, dies vor allem weil PHP4 ganz einfach nicht dafür ausgelegt ist.

Abschliessend stellten wir fest, dass wir mehr Zeit für das Testen des Shops hätten planen sollen.

### **10.4 PHP Programmierung**

Wir kamen eigentlich immer recht gut voran mit unseren Zwischenzielen, bis wir dann mit dem Codieren anfangen. Das Codieren benötigte einfach zu viel Zeit, im Vergleich zu dem erzielten Fortschritt.

Dies lag vor allem daran, dass wir am Anfang nicht genau wussten, wie man mit den Fehlermeldungen von MySQL, PHP und Browser umgehen sollte, respektive, was sie bedeuteten (Beispiel: Man ändert in einer Funktion den Code und vergisst eine Klammer zu schliessen, danach will man das Ergebnis der Änderung im Shop begutachten und siehe da, die Seite wird im Internet Explorer als nicht existierend angezeigt, Netscape meint, die Seite habe keinen Content. Die Verzwickte Lage ist, dass die Funktion in der man die Änderung vornahm in einem Modul liegt, das von einem anderen (das die Seite darstellen sollte) included ist,... bis man da drauf kommt... u.s.w.)

Nach den Fehlermeldungen kam dann ziemlich schnell das 'include'-Problem. Um eine Funktion von einer anderen Datei benutzen zu können, muss man diese Datei per `include` oder `require` einbinden. Dies ist grundsätzlich kein Problem, wenn PHP von selbst verstehen würde, dass eine in dieser Datei schon eingebundene Datei nicht zweimal eingebunden werden muss. Zuerst versuchten wir dem Problem mit einer Art `include`-Liste beizukommen. Wir kreierten eine Abfolge, in der jede Datei nur eine weitere `included` und diese wieder nur eine, auf diese Weise kann nie eine Datei zweimal `included`

werden. Dieser Ansatz scheiterte aber daran, dass man auf der PHP-Funktionen-Modul-Ebene doch auch Funktionen von Aufruf-Modulen benötigte, welche dann natürlich nicht zur Verfügung standen (siehe Modul-Übersicht). Wir änderten den Ansatz und includeten die Dateien fortan nur noch mit einem `if(isset(Dateiname))` davor. Mit diesem Konzept kamen wir sehr weit, bloss stellte sich dann das Problem, dass wenn man eine weitere Verzeichnis-Ebene (z.B. `Admin/`) einführt, alle darin enthaltenen Module die Dateien in einer Verzeichnisstufe höher nicht mehr richtig einbinden können. Hier konnten wir das Problem auf UNIX-Seite per Links lösen. Wir erstellten Links der includeten Dateien im jeweiligen Unterverzeichnis, dann klappte alles wieder.

Erstaunlich positiv waren wir vom Datenbank-Wrapper, der uns gegeben wurde, überrascht. Dieser konnte in nur einem Nachmittag für MySQL erweitert werden und funktionierte auch gleich tadellos. Wir können ihn nur weiter empfehlen.

## 11. Was sollte noch gemacht werden

Leider haben wir unsere sehr hoch gesteckten Ziele bis zum Zeitpunkt des Projektabschlusses nicht alle erreicht. Dies ist insofern schade, da wir den kompletten Shop gerne der Internetgemeinde zum Gebrauch übergeben hätten. Falls wir, oder jemand anders, diesen Shop noch zu Ende entwickelt (Diplomarbeit?), haben wir eine Liste mit Problemen und Funktionen erstellt, die noch korrigiert oder ausprogrammiert werden müssen.

Wir haben eine dreistufige Prioritätsliste erstellt, welche folgendermassen aufgebaut ist

### **11.1 Probleme, die vor einer Shopinbetriebnahme unbedingt behoben werden sollten**

- Momtan besteht das Problem, dass wenn man einen Artikel in den Warenkorb legt überprüft wird, ob dieser schon vorhanden ist, ist dem so, wird nur seine Anzahl erhöht, nicht aber wie erwartet ein 'neuer' Artikel in den Warenkorb gelegt. Momentan ist es also nicht möglich in einer Bestellung vom gleichen Artikel mehrere Varianten zu bestellen.
- Einige Funktionen im Shopadministrationsmodus sind noch nicht ausprogrammiert. Notfalls können jedoch alle Shop- und Kategorieneinstellungen auch mittels eines Datenbankbearbeitungstools (z.B. `phpMyAdmin`) direkt in der Datenbank eingegeben und verändert werden.

### **11.2 Funktionen und Probleme die zur optischen Einheitlichkeit und Usability gelöst werden sollten**

- Der Shop wurde auf Browsern der neusten Generation getestet und für diese auch optimiert. Viele Surfer benutzen jedoch noch ältere Browser. Das Problem bei diesen ist die externe Stylesheet-Datei (`shopstyles.css`), die wir zur einheitlichen Darstellung des Shops benutzen. Vor allem Netscape-Browser haben Mühe mit den Farbinformationen für die Schriftarten, sowie „decoration“-Anweisungen. Bei Tests mit den älteren Browsern haben wir festgestellt, dass diese gewisse Stylesheetangaben nur interpretieren, wenn sie sich direkt im HTML-File befinden. Da wir aber alle Darstellungsoptionen momentan nur in der Datei „`shopstyles.css`“ haben, müsste man bei jeder Änderung noch in allen PHP-Files, die HTML-Ausgaben erzeugen, die Stylesheetangaben nachführen. Die einzige Lösung, um Darstellungsparameter zentral

an einem Ort einstellen zu können, ist ein Funktionspaket im Administrationsmodus, welches Änderungen am Layout an allen relevanten Stellen einträgt, oder diese direkt aus der DB bei erzeugung des HTML-Dokuments einfügt.

- Einige Buttons (nur im Administrationsmodus, Usermodus schon bereinigt!) sind Formularbuttons, welche vom Browser selbst erzeugt werden. Diese sehen je nach Browser anders aus. Da aber noch andere Funktionen gewünscht sind, hat es teilweise auch Links, die als Buttons dargestellt werden. Um ein einheitliches Buttonaussehen zu haben, müssen alle Formularbuttons, durch .gif-Bilder (welche Buttons darstellen) ersetzt werden. Ausserdem hat es noch einige .jpg-Buttons, die durch äquivalente .gif-

Buttons ersetzt werden sollten.

- Gibt der Shopbeutzer bei der Bestellung seine Anschrift ein, werden gewisse Felder auf Vollständigkeit überprüft. Das E-Mail Feld wird mit einer einfachen Funktion auf Plausibilität geprüft. Wird keine gültige E-Mailadresse eingegeben, wird der Benutzer angewiesen, eine korrekte einzugeben. Die Bestellung wird nicht abgesendet, solange keine gültige E-Mail-Adresse eingegeben wurde. Hat aber ein Shopuser kein E-Mail-Konto (das soll es wirklich geben!), kann er die Bestellung nicht absenden. Man müsste ihm eine Warnung anzeigen, dass er, wenn er keine E-Mail-Adresse eingibt, auch kein Bestätigungs-E-Mail erhält. Trotzdem soll er aber seine Bestellung absenden können.
- Einige Testuser haben bemängelt, dass der Warenkorb nicht direkt angezeigt wird, wenn man einen Artikel hineinlegt, sondern „nur“ die Meldung „Der gewählte Artikel wurden in ihren Warenkorb gelegt“ ausgegeben wird. Da schon Funktionen zur Ausgabe des Warenkorb Inhalts an beliebiger Stelle eines HTML-Dokuments existieren, ist dies eine kleine Änderung.
- Im Administrationstool kann zu jedem Artikel ein Aktionspreis eingegeben werden, der dann hervorgehoben dargestellt werden soll. Diese Funktionalität ist jedoch noch nicht ausprogrammiert, erfordert jedoch zur Realisierung auch nur einen sehr kleinen Programmieraufwand, da alle Informationen schon am richtigen Ort vorhanden sind.
- Artikel können momentan per Administrationstool nur einer Kategorie zugeordnet werden. Das Datenbankdesign ist jedoch darauf ausgelegt, dass jeder Artikel in mehreren Kategorien sein kann. So hat man die Möglichkeit, einen Artikel zum Beispiel einer Top-Seller- oder Monatshit-Kategorie zuzuordnen, die dann bei Aufruf des Shops ausgegeben werden kann.
- Der im GIF-Grafikformat verwendete Kompressionsalgorithmus LZW ist von der Firma Unisys patentrechtlich geschützt. Darum wurde die GIF-Unterstützung aus der GD-Bibliothek von PHP4 entfernt. Aus diesem Grund ist es mit der Shopsoftware im „Rohzustand“ nicht möglich, aus GIF-Bildern Thumbnails zu erzeugen. Auch ist das PNG-Bildformat momentan noch nicht aktiviert, da ältere Browser es nicht verstehen. Will ein Shopbetreiber andere Bildformate als JPG hochladen, muss der Programmcode geringfügig ergänzt werden.
- Bearbeitet der Shopadministrator einen Artikel erhält er immer zwei leere Felder, wo er weitere Optionen und Variationen einfügen kann. Will er aber vier neue Optionen einfügen, muss er den Artikel speichern und nochmals bearbeiten, damit er wieder zwei neue Eingabefelder hat. Wünschenswert wäre ein Button im Stil von „Weitere leere Felder anzeigen“.
- Befinden sich keine Artikel im Warenkorb, wird er trotzdem angezeigt. Eine Meldung „Ihr Warenkorb ist leer“ wäre hier sicher benutzerfreundlicher. Ausserdem kann man Bestellungen absenden, die keine Artikel beinhalten, auch das müsste noch abgefangen werden.

### **11.3 Sinnvolle Zusatzfunktionalität**

- Momentan werden Bestellungen in der Datenbank solange gespeichert, bis eine weitere Bestellung eingeht. Mit dieser werden alle alten, nicht abgeschlossenen Bestellungen (deren Session-ID nicht mehr gültig, also expired sind) gelöscht. Diese Prozedur geschieht, weil ein Bestellungsmanagementtool im Administrationsmodus geplant ist, mit welchem der Shopbetreiber eingegangene Bestellungen bearbeiten, auslösen und

löschen können soll. Dieses Modul muss noch programmiert werden.

- Da ein grosser Teil des E-Shopping über Kreditkarten bezahlt wird und deren Akzeptanz ständig steigt, wäre es sehr sinnvoll eine sichere Kreditkartenlösung mit einem Finanzinstitut als Abrechnungspartner zu implementieren. Nach unseren Recherchen bietet Yellowworld eine attraktive Lösung an.
- Damit der Shop nicht nur zusammen mit einem einzigen Finanzinstitut gebraucht werden kann, würde sich die Programmierung eines Wrappers (der ähnlich arbeitet, wie der bereits eingesetzte Datenbankwrapper) anbieten, der eine Schnittstelle zwischen Shop und Finanzinstitut darstellt und von beiden Seiten erweitert und angepasst werden kann.
- Useradressmanagement: Nach erfolgter Bestellung fragt man den Kunden, ob man seine Adressdaten speichern soll, damit er bei einem erneuten Einkauf in diesem Shop sich mit einem Usernamen und Passwort anmelden kann. Dies bedingt jedoch ein Kundenadressmanagement, welches auch Funktionen wie „Passwort vergessen“ unterstützt.
- Importmöglichkeit, um von anderen Shops Artikeldaten zu importieren.
- Verschiedene Layouts zur Auswahl bereitstellen (Frameset, Farbe, Fontset), die über das Administrationstool ausgewählt werden können.
- Unterstützung von mehreren Währungen
- Unterstützung von mehreren Sprachen

## **12. Schlusswort**

Hier sitzen wir nun und schreiben unser Schlusswort. Lieber wären wir jetzt noch am Programmieren. Es ist sehr schade, dass wir gezwungen sind jetzt aufzuhören. Zur Marktreife des Produktes fehlt nur noch wenig, obwohl wir die letzten Wochen unsere gesamte Energie für die Entwicklung dieses Shop-Frameworks aufgewendet haben.

Trotz diesem abrupten Ende haben wir noch nie so viel in so kurzer Zeit gelernt. Es war spannend, faszinierend und motivierend zugleich. Wenn auch fordernd und manchmal ganz schön anstrengend.

Bei Herrn Feisthammel möchten wir uns für die stets prompte und kompetente Hilfe bedanken.

Auch ganz herzlich möchten wir unseren Freundinnen für ihre Geduld und Unterstützung während dieser Zeit, in der wir doch des öfteren erst spät Abends nach Hause kamen und bis spät in die Nacht arbeiteten, danken.

Winterthur den 18. Mai 2001

## 13. Literaturverzeichnis

Nachfolgend eine Liste mit allen Quellen und Literaturverweisen aufgliedert nach den einzelnen Themen die durch das Webshop-Programming tangiert werden.

### 13.1 PHP

#### **PHP 4 Grundlagen und Profiwissen**

Carl Hanser Verlag München Wien  
Jörg Krause  
November 2000  
1161 Seiten  
ISBN 3-446-21546-8

Dies war unser 'Hauptbuch'. Es enthält eine Vielfalt an Themen die alle ziemlich ausführlich erklärt werden. Aber auch dieses Buch hat seinen Grenzen (Bilder in Datenbank speichern, grössere Projekte mit PHP)

#### **PHP Grundlagen und Profiwissen**

Carl Hanser Verlag München Wien  
Jörg Krause  
März 2000  
1067 Seiten  
ISBN 3-446-21301-5

Dieses Buch ist die Vorgänger-Version des oberen. Wir haben es selten konsultiert, aber doch ab und wann etwas daraus brauchen können.

#### **PHP kurz & gut**

O'Reilly Verlag Köln  
Rasmus Lerdorf  
1. Auflage 2000  
111 Seiten  
ISBN 3-89721-225-0

Eine praktische kompakte Zusammenfassung, aber leider nur für PHP3.

#### **PHP 4 Das bhv Taschenbuch**

bhv Verlag Landsberg  
Uwe Hess, Günther Karl  
1. Auflage April 2001  
764 Seiten  
ISBN 3-8287-5057-5

Eine gut Einführung in die Grundlagen von PHP4, der Stoff geht aber nicht all zu tief.

<http://www.phparchiv.de/>

Diese Page ist wirklich sehr gut und hilfreich. Hier fanden wir diverse Editoren und viele Beispiel-Scripts. Wir haben auch vor unser Projekt u.a. dort zu publizieren.

### 13.2 Apache Webserver

#### **Apache Web-Server Installation, Konfiguration, Administration**

MITP-Verlag GmbH Bonn  
Lars Eilebrecht  
3. Auflage 2000  
671 Seiten  
ISBN 3-8266-0612-4

Ein sehr tief gehendes Buch. Half uns sehr gut bei der Apache-Konfiguration und beim Einrichten von htaccess.

### 13.3 Datenbanken und im speziellen MySQL

#### MySQL in 21 Tagen

Markt + Technik Verlag München  
Mark Maslakowski  
2001  
580 Seiten  
ISBN 3-8272-5850-2

Dies war unser 'Hauptbuch' im Bereich Datenbanken. Man findet auch knifflige Aspekte von MySQL hier drin. Vor allem für die MySQL Konfiguration sehr zu empfehlen.

#### SQL der Schlüssel zu relationalen Datenbanken

Rowohlt Taschenbuch Verlag GmbH  
Reineck bei Hamburg  
Gregor Kuhlmann, Friedrich Müllmerstadt  
Mai 1999  
318 Seiten  
ISBN 3-499-60063-3

Dieses Buch half uns vor allem bei schwierigen Queries, aber auch bei Fragen wie man gewisse Datenbank-Aktionen überhaupt erst angehen soll. Sehr zu empfehlen.

#### Datenbanken – Vorlesung an der Zürcher Hochschule Winterthur

Dr. Hans-Walter Buff, zweites Jahr IT

Der Stoff half uns beim Entity Relationship Design, aber auch bei den Installations- und Deinstallations-SQL-Scripts.

### 13.4 HTML, JavaScript und Cascading Style Sheets

#### HTML 4 – Die Sprache des Web

dpunkt-Verlag Heidelberg  
Robert Tolksdorf  
3. Auflage 1997  
308 Seiten  
ISBN 3-920993-77-2

Bietet einen tiefgehenden Blick ins HTML, leider schon etwas in die Jahre gekommen.

#### Jetzt lerne ich HTML

Markt + Technik Verlag München  
Harald Taglinger  
1997  
290 Seiten  
ISBN 3-8272-5305-5

Gutes Nachschlagewerk (Farbtabelle, ...) immer wieder praktisch für 'wie ging das jetzt schon wieder?'-Momente. Leider auch schon etwas in die Jahre gekommen.

#### Das grosse Buch JavaScript

DATA BECKER GmbH & Co KG Düsseldorf  
Rainer Kolbeck  
1. Auflage 1997  
422 Seiten  
ISBN 3-8158-1321-2

Diesem Buch konnten wir nicht viel abgewinnen, wir brauchten aber auch nicht so viel JavaScript.

<http://www.teamone.de/selfhtml/>

Selfhtml ist und bleibt das beste Nachschlagewerk in sachen HTML. Sogar für JavaScript konnten wir einige Informationen von dieser Adresse gebrauchen. Sehr zu empfehlen

## 14. Software Versionen

### 14.1 Auf dem Server eingesetzte Software

SuSE Linux 7.1  
Kernel: 2.4.0-4GB (i686)

Apache  
Version 1.3.14 (Unix) (SuSE/Linux)

MySQL  
Version 3.23.35

PHP4  
Version: 4.0.4pl1

phpMyAdmin  
Version 2.1.0

Einzigste Linux-Distribution die zu Projektbeginn schon den Kernel 2.4.0 beinhaltet.

In SuSE Linux Distribution enthalten, genügend aktuell.

Die zu Projektbeginn aktuellste MySQL Datenbank Version aus dem Internet.

Diese damals aktuellste Version von PHP4 war glücklicherweise schon in der SuSE Distribution enthalten.

Datenbank Administrationstool mit Webinterface. Dieses Programm brauchten wir um SQL-Queries auszutesten und Änderungen in der Datenbank manuell vorzunehmen

### 14.2 Auf den Arbeitsstationen eingesetzte Software

Windows 2000 Professionell  
SP1

Apache für Windows  
Version 1.3.19

PHP4  
Version 4.04pl1

PHPed  
Version 2.96.1.2

CUTE FTP  
Version 4.2.3 Build 2.14.1

Putty  
Version 0.5.1

Komfortable Entwicklungsoberfläche.

Zu Testzwecken eingerichtet

Zu Testzwecken eingerichtet

Sehr nützlicher PHP-Editor mit integriertem FTP-Tool welches direktes Arbeiten auf dem Server ermöglicht. Leider noch etwas Beta. FTP-Client

Telnet / SSH Client

## 15. GPL – GNU General Public Licence

Dieses Programm ist freie Software. Sie können es unter den Bedingungen der GNU General Public License, wie von der Free Software Foundation herausgegeben, weitergeben und/oder modifizieren, entweder unter Version 2 der Lizenz oder (wenn Sie es wünschen) jeder späteren Version.

Die Veröffentlichung dieses Programms erfolgt in der Hoffnung, dass es Ihnen von Nutzen sein wird, aber OHNE JEDE GEWÄHRLEISTUNG – sogar ohne die implizite Gewährleistung der MARKTREIFE oder der EIGNUNG FÜR EINEN BESTIMMTEN ZWECK.

Details finden Sie in der GNU General Public License. Sie sollten eine Kopie der GNU General Public License zusammen mit diesem Programm erhalten haben. Falls nicht, schreiben Sie an die Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

### 15.1 Deutsche Fassung der GPL (Auszug)

Übersetzt von Katja Lachmann Übersetzungen ([na194@fim.uni-erlangen.de](mailto:na194@fim.uni-erlangen.de)), modifiziert von Peter Gerwinski ([peter.gerwinski@uni-essen.de](mailto:peter.gerwinski@uni-essen.de)) und Jörg Krause ([joerg@krause.net](mailto:joerg@krause.net)). Es handelt sich nicht um eine offizielle oder im rechtlichen Sinne anerkannte Übersetzung.

Die Free Software Foundation (FSF) ist nicht der Herausgeber dieser Übersetzung, und sie hat diese Übersetzung auch nicht als rechtskräftigen Ersatz für die Original GNU-GPL anerkannt. Da die Übersetzung nicht sorgfältig von Anwälten überprüft wurde, können die Übersetzer nicht garantieren, dass die Übersetzung die rechtlichen Aussagen der GNU-GPL exakt wiedergibt. Wenn Sie sichergehen wollen, dass von Ihnen geplante Aktivitäten im Sinne der GNU-GPL gestattet sind, halten Sie sich bitte an die englischsprachige Originalversion. Die Free Software Foundation möchte Sie darum bitten, diese Übersetzung nicht als offizielle Lizenzbedingungen für von Ihnen geschriebene Programme zu verwenden. Bitte benutzen Sie hierfür stattdessen die von der Free Software Foundation herausgegebene englischsprachige Originalversion. Bedingungen für die Vervielfältigung, Verbreitung und Bearbeitung

**§0** Diese Lizenz gilt für jedes Programm und jedes andere Werk, in dem ein entsprechender Vermerk des Copyright-Inhabers darauf hinweist, dass das Werk unter den Bestimmungen dieser General Public License verbreitet werden darf. Im Folgenden wird jedes derartige Programm oder Werk als »das Programm« bezeichnet; die Formulierung »auf dem Programm basierendes Werk« bezeichnet das Programm sowie jegliche Bearbeitung des Programms im urheberrechtlichen Sinne, also ein Werk, welches das Programm, auch auszugsweise, sei es unverändert oder verändert oder in eine andere Sprache übersetzt, enthält. Andere Handlungen als Vervielfältigung, Verbreitung und Bearbeitung werden von dieser Lizenz nicht berührt; sie fallen nicht in ihren Anwendungsbereich. Der Vorgang der Ausführung des Programms wird nicht eingeschränkt und die Ausgaben des Programms unterliegen dieser Lizenz nur, wenn der Inhalt ein auf dem Programm basierendes Werk darstellt (unabhängig davon, dass die Ausgabe durch die Ausführung des Programmes erfolgte). Ob dies zutrifft, hängt von den Funktionen des Programms ab.

**§1** Sie dürfen auf beliebigen Medien unveränderte Kopien des Quelltextes des Programms, wie sie ihn erhalten haben, anfertigen und verbreiten. Voraussetzung hierfür ist, dass Sie mit jeder Kopie einen entsprechenden Copyright-Vermerk sowie einen Haftungsausschluss veröffentlichen, alle Vermerke, die sich auf diese Lizenz und das Fehlen einer Garantie beziehen, unverändert lassen und des Weiteren allen anderen Empfängern des Programms zusammen mit dem Programm eine Kopie dieser Lizenz zukommen lassen. Sie dürfen für den eigentlichen Kopiervorgang eine Gebühr verlangen. Wenn Sie es wünschen, dürfen Sie auch gegen Entgelt eine Garantie für das Programm anbieten.

**§2** Sie dürfen Ihre Kopien des Programms oder eines Teils davon verändern, wodurch ein auf dem Programm basierendes Werk entsteht; Sie dürfen derartige Bearbeitungen unter den Bestimmungen von Paragraph 1 vervielfältigen und verbreiten, vorausgesetzt, dass zusätzlich alle folgenden Bedingungen erfüllt werden:

- (a) Sie müssen die veränderten Dateien mit einem auffälligen Vermerk versehen, der auf die von Ihnen vorgenommene Modifizierung und das Datum jeder Änderung hinweist.
- (b) Sie müssen dafür sorgen, dass jede von Ihnen verbreitete oder veröffentlichte Arbeit, die ganz oder teilweise von dem Programm oder Teilen davon abgeleitet ist, Dritten gegenüber als Ganzes unter den Bedingungen dieser Lizenz ohne Lizenzgebühren zur Verfügung gestellt wird.
- (c) Wenn das veränderte Programm normalerweise bei der Ausführung inter-aktiv Kommandos einliest, müssen Sie dafür

sorgen, dass es, wenn es auf dem üblichsten Wege für solche interaktive Nutzung gestartet wird, eine Meldung ausgibt oder ausdrückt, die einen geeigneten Copyright-Vermerk enthält sowie einen Hinweis, dass es keine Gewährleistung gibt (oder an-derenfalls, dass Sie Garantie leisten), und dass die Benutzer das Programm unter diesen Bedingungen weiter verbreiten dürfen. Auch muss der Benutzer darauf hingewiesen werden, wie er eine Kopie dieser Lizenz ansehen kann. (Ausnahme: Wenn das Programm selbst interaktiv arbeitet, aber normalerweise keine derartige Meldung ausgibt, muss Ihr auf dem Programm basierendes Werk auch keine solche Meldung ausgeben.) Diese Anforderungen betreffen das veränderte Werk als Ganzes. Wenn identifizierbare Abschnitte des Werkes nicht von dem Programm abgeleitet sind und vernünftigerweise selbst als unabhängige und eigenständige Werke betrachtet werden können, dann erstrecken sich diese Lizenz und ihre Bedingungen nicht auf diese Abschnitte, wenn sie als eigenständige Werke verbreitet werden. Wenn Sie jedoch dieselben Abschnitte als Teil eines Ganzen verbreiten, das ein auf dem Programm basierendes Werk darstellt, dann muss die Verbreitung des Ganzen nach den Bedingungen dieser Lizenz erfolgen, deren Bedingungen für weitere Lizenznehmer somit auf die Gesamtheit ausgedehnt werden – und damit auf jeden einzelnen Teil, unabhängig vom jeweiligen Autor. Somit ist es nicht die Absicht dieses Abschnittes, Rechte für Werke in Anspruch zu nehmen oder zu beschneiden, die komplett von Ihnen geschrieben wurden; vielmehr ist es die Absicht, die Rechte zur Kontrolle der Verbreitung von Werken, die auf dem Programm basieren oder unter seiner auszugsweisen Verwendung zusammengestellt worden sind, auszuüben. Ferner bringt ein einfaches Zusammenstellen eines anderen Werkes, das nicht auf dem Programm basiert, zusammen mit dem Programm oder einem auf dem Programm basierenden Werk auf ein- und demselben Speicher- oder Betriebsmedium das andere Werk nicht in den Anwendungsbereich dieser Lizenz.

**§3** Sie dürfen das Programm (oder ein darauf basierendes Werk gemäß Paragraph 2) als Objectcode oder in ausführbarer Form unter den Bedingungen von Paragraph 1 und 2 vervielfältigen und verbreiten – vorausgesetzt, dass Sie außerdem eine der folgenden Leistungen erbringen:

(a) Liefern Sie das Programm zusammen mit dem vollständigen zugehörigen maschinenlesbaren Quelltext auf einem für den Datenaustausch üblichen Medium aus, wobei die Verteilung unter den Bedingungen der Paragraphen 1 und 2 erfolgen muss. Oder:

(b) Liefern Sie das Programm zusammen mit einem mindestens drei Jahre lang gültigen schriftlichen Angebot aus, jedem Dritten eine vollständige maschinenlesbare Kopie des Quelltextes zur Verfügung zu stellen – zu nicht höheren Kosten als denen, die durch den physikalischen Kopiervorgang anfallen –, wobei der Quelltext unter den Bedingungen der Paragraphen 1 und 2 auf einem für den Datenaustausch üblichen Medium weiter-gegeben wird. Oder:

(c) Liefern Sie das Programm zusammen mit dem schriftlichen Angebot der Zurverfügungstellung des Quelltextes aus, das Sie selbst erhalten haben. (Diese Alternative ist nur für nicht-kommerzielle Verbreitung zulässig und nur, wenn Sie das Programm als Objectcode oder in ausführbarer Form mit einem entsprechenden Angebot gemäß Absatz b erhalten haben.) Unter dem Quelltext eines Werkes wird diejenige Form des Werkes verstanden, die für Bearbeitungen vorzugsweise verwendet wird. Für ein ausführbares Programm bedeutet »der komplette Quelltext«: Der Quelltext aller im Programm enthaltenen Module einschließlich aller zugehörigen Modulschnittstellen-Definitionsdateien sowie der zur Compilation und Installation verwendeten Skripte. Als besondere Ausnahme jedoch braucht der verteilte Quelltext nichts von dem zu enthalten, was üblicherweise (entweder als Quelltext oder in binärer Form) zusammen mit den Hauptkomponenten des Betriebssystems (Kernel, Compiler usw.) geliefert wird, unter dem das Programm läuft – es sei denn, diese Komponente selbst gehört zum ausführbaren Programm. Wenn die Verbreitung eines ausführbaren Programms oder des Objectcodes dadurch erfolgt, dass der Kopierzugriff auf eine dafür vorgesehene Stelle gewährt wird, so gilt die Gewährung eines gleichwertigen Zugriffs auf den Quelltext als Verbreitung des Quelltextes, auch wenn Dritte nicht dazu gezwungen sind, den Quelltext zusammen mit dem Objectcode zu kopieren.

**§4** Sie dürfen das Programm nicht vervielfältigen, verändern, weiter lizenzieren oder verbreiten, sofern es nicht durch diese Lizenz ausdrücklich gestattet ist. Jeder anderweitige Versuch der Vervielfältigung, Modifizierung, Weiterlizenzierung und Verbreitung ist nichtig und beendet automatisch Ihre Rechte unter dieser Lizenz. Jedoch werden die Lizenzen Dritter, die von Ihnen Kopien oder Rechte unter dieser Lizenz erhalten haben, nicht beendet, solange diese die Lizenz voll anerkennen und befolgen.

**§5** Sie sind nicht verpflichtet, diese Lizenz anzunehmen, da Sie sie nicht unter-zeichnet haben. Jedoch gibt Ihnen nichts anderes die Erlaubnis, das Programm oder von ihm abgeleitete Werke zu verändern oder zu verbreiten. Diese Handlungen sind gesetzlich verboten, wenn Sie diese Lizenz nicht anerkennen. Indem Sie das Programm (oder ein darauf basierendes Werk) verändern oder verbreiten, erklären Sie Ihr Einverständnis mit dieser Lizenz und mit allen ihren Bedingungen bezüglich der Vervielfältigung, Verbreitung und Veränderung des Programms oder eines darauf basierenden Werkes.

**§6** Jedesmal, wenn Sie das Programm (oder ein auf dem Programm basierendes Werk) weitergeben, erhält der Empfänger automatisch vom ursprünglichen Lizenzgeber die Lizenz, das Programm entsprechend den hier festgelegten Bestimmungen zu vervielfältigen, zu verbreiten und zu verändern. Sie dürfen keine weiteren Einschränkungen der Durchsetzung der hierin zugestandenen Rechte des Empfängers vornehmen. Sie sind nicht dafür verantwortlich, die Einhaltung dieser Lizenz durch Dritte durchzusetzen.

**§7** Sollten Ihnen infolge eines Gerichtsurteils, des Vorwurfs einer Patentverletzung oder aus einem anderen Grunde (nicht auf Patentfragen begrenzt) Be-.Copyright und Lizenzhinweise 9 Bedingungen (durch Gerichtsbeschluss, Vergleich oder anderweitig) auferlegt werden, die den Bedingungen dieser Lizenz widersprechen, so befreien Sie diese Umstände nicht von den Bestimmungen dieser Lizenz. Wenn es Ihnen nicht möglich ist, das Programm unter gleichzeitiger Beachtung der Bedingungen in dieser Lizenz und Ihrer anderweitigen Verpflichtungen zu verbreiten, dann dürfen Sie als Folge das Programm überhaupt nicht verbreiten. Wenn zum Beispiel ein Patent nicht die gebührenfreie Weiterverbreitung des Programms durch diejenigen erlaubt, die das Programm direkt oder indirekt von Ihnen erhalten haben, dann besteht der einzige Weg, sowohl das Patentrecht als auch diese Lizenz zu befolgen, darin, ganz auf die Verbreitung des Programms zu verzichten. Sollte sich ein Teil dieses Paragraphen als ungültig oder unter bestimmten Umständen nicht durchsetzbar erweisen, so soll dieser Paragraph seinem Sinne nach angewandt werden; im Übrigen soll dieser Paragraph als Ganzes gelten. Zweck dieses Paragraphen ist nicht, Sie dazu zu bringen, irgendwelche Patente oder andere Eigentumsansprüche zu verletzen oder die Gültigkeit solcher Ansprüche zu bestreiten; dieser Paragraph hat einzig den Zweck, die Integrität des Verbreitungssystems der freien Software zu schützen, das durch die Praxis öffentlicher Lizenzen verwirklicht wird. Viele Leute haben großzügige Beiträge zu dem großen Angebot der mit diesem System verbreiteten Software im Vertrauen auf die konsistente Anwendung dieses Systems geleistet; es liegt am Autor/Geber, zu entscheiden, ob er die Software mittels irgendeines anderen Systems verbreiten will; ein Lizenznehmer hat auf diese Entscheidung keinen Einfluss. Dieser Paragraph ist dazu gedacht, deutlich klarzustellen, was als Konsequenz aus dem Rest dieser Lizenz betrachtet wird.

**§8** Wenn die Verbreitung oder die Benutzung des Programms in bestimmten Staaten entweder durch Patente oder durch urheberrechtlich geschützte Schnittstellen eingeschränkt ist, kann der Urheberrechtsinhaber, der das Programm unter diese Lizenz gestellt hat, eine explizite geographische Begrenzung der Verbreitung angeben, in der diese Staaten ausgeschlossen werden, so dass die Verbreitung nur innerhalb und zwischen den Staaten erlaubt ist, die nicht ausgeschlossen sind. In einem solchen Fall beinhaltet diese Lizenz die Beschränkung, als wäre sie in diesem Text niedergeschrieben.

**§9** Die Free Software Foundation kann von Zeit zu Zeit überarbeitete und/oder neue Versionen der General Public License veröffentlichen. Solche neuen Versionen werden vom Grundprinzip her der gegenwärtigen entsprechen, können aber im Detail abweichen, um neuen Problemen und Anforderungen gerecht zu werden. Jede Version dieser Lizenz hat eine eindeutige Versionsnummer. Wenn in einem Programm angegeben wird, dass es dieser Lizenz in einer bestimmten Versionsnummer oder »jeder späteren Version« unterliegt, so haben Sie die Wahl, entweder den Bestimmungen der genannten Version zu folgen oder denen jeder beliebigen späteren Version, die von der Free Software Foundation veröffentlicht wurde. Wenn das Programm keine Versionsnummer angibt, können Sie eine beliebige Version wählen, die je von der Free Software Foundation veröffentlicht wurde.

**§10** Wenn Sie den Wunsch haben, Teile des Programms in anderen freien Programmen zu verwenden, deren Bedingungen für die Verbreitung anders sind, schreiben Sie an den Autor, um ihn um die Erlaubnis zu bitten. Für Software, die unter dem Copyright der Free Software Foundation steht, schreiben Sie an die Free Software Foundation; wir machen zu diesem Zweck gelegentlich Ausnahmen. Unsere Entscheidung wird von den beiden Zielen geleitet werden, zum einen den freien Status aller von unserer freien Software abgeleiteten Werke zu erhalten und zum anderen das gemeinschaftliche Nutzen und Wiederverwenden von Software im Allgemeinen zu fördern.

**§11** Da das Programm ohne jegliche Kosten lizenziert wird, besteht keinerlei Gewährleistung für das Programm, soweit dies gesetzlich zulässig ist. Sofern nicht anderweitig schriftlich bestätigt, stellen die Copyright-Inhaber und/oder Dritte das Programm so zur Verfügung, »wie es ist«, ohne irgendeine Gewährleistung, weder ausdrücklich noch implizit, einschließlich – aber nicht begrenzt auf – Marktreife oder Verwendbarkeit für einen bestimmten Zweck. Das volle Risiko bezüglich Qualität und Leistungsfähigkeit des Programms liegt bei Ihnen. Sollte sich das Programm als fehlerhaft herausstellen, liegen die Kosten für notwendigen Service, Reparatur oder Korrektur bei Ihnen.

**§12** In keinem Fall, außer wenn durch geltendes Recht gefordert oder schriftlich zugesichert, ist irgendein Copyright-Inhaber oder irgendein Dritter, der das Programm wie oben erlaubt modifiziert oder verbreitet hat, Ihnen gegenüber für irgendwelche Schäden haftbar, einschließlich jeglicher allgemeiner oder spezieller Schäden, Schäden durch Seiteneffekte (Nebenwirkungen) oder Folgeschäden, die aus der Benutzung des Programms oder der Unbenutzbarkeit des Programms folgen (einschließlich – aber nicht beschränkt auf – Datenverluste, fehlerhafte Verarbeitung von Daten, Verluste, die von Ihnen oder anderen getragen werden müssen, oder dem Unvermögen des Programms, mit irgendeinem anderen Programm zusammenzuarbeiten), selbst wenn ein Copyright-Inhaber oder Dritter über die Möglichkeit solcher Schäden unterrichtet worden war.

## 16. Anhang

### A Source Code

```
htdocs/  
    index.html  
    index.php  
  
htdocs/shop/  
    artikel_def.php  
    bestellung_def.php  
    bild_view.php  
    database.php  
    initialize.php  
    pop_up.php  
    shopstyles.css  
    USER_ADMIN_HILFE.php  
    USER_ARTIKEL_HANDLING.php  
    USER_ARTIKEL_HANDLING_AUFRUF.php  
    USER_BESTELLUNG.php  
    USER_BESTELLUNG_1.php  
    USER_BESTELLUNG_AUFRUF.php  
    USER_BESTELLUNG_DARSTELLUNG.php  
    USER_SQL_BEFEHLE.php  
  
htdocs/Frameset/  
    content.html  
    top.php  
  
htdocs/Admin/  
    .htaccess  
    ADMIN_initialize.php  
    ADMIN_SQL_BEFEHLE.php  
    bild_up.php  
    SHOP_ADMINISTRATION.php  
    SHOP_ADMINISTRATION_ARTIKEL.php  
    SHOP_ADMINISTRATION_AUFRUF.php  
    Shop_Einstellungen_Menu_1.php  
    Shop_Einstellungen_Menu_Artikel.php  
    Shop_Einstellungen_Menu_Kategorien.php  
    SHOP_SETTINGS.php  
  
htdocs/Datenbank/  
    shopdb_create.sql  
    shopdb_del.sql  
    shopdb_insert.sql
```

### B Zusammenfassungen

Am Schluss dieser Dokumentation sind noch die Zusammenfassungen angehängt, die wir