

**Anleitung**

zu

# **BERNHIST**

**Version 4**

Einführung, Konzept und Installation

**Serge Meyer**

[serge.meyer@hist.unibe.ch](mailto:serge.meyer@hist.unibe.ch)

## Inhaltsverzeichnis

BERNHIST in Strom der Modernisierung .....	3
Die Werkzeuge .....	5
Linux .....	5
Apache .....	6
MySQL .....	6
PHP .....	7
Das Prinzip einer dynamischen Webseite .....	8
Das Konzept von BERNHIST .....	9
Die Datenbank .....	9
Die Webanwendung .....	12
Der Ablauf einer Abfrage .....	13
Zugriffskontrolle mit Sessions.....	14
Installation .....	15
Der Server.....	15
Dienste starten .....	19
Installation testen .....	19
Einrichten der Datenbank .....	20
Wohin mit den Skripts? .....	21
Datenblatt BERNHIST .....	23
Literaturverzeichnis .....	24
Anhang.....	25
A Datenbankschema .....	25
B Codefragmente .....	26

## **BERNHIST in Strom der Modernisierung**

Die Geschichte und der Werdegang von BERNHIST verliefen parallel mit der Entwicklung der Informatik. Das Projekt wurde im Verlauf der letzten zwanzig Jahre immer wieder an die neuen Möglichkeiten der EDV angepasst um sich vor allem die Entwicklung der elektronischen Datenspeichermedien zu Nutze zu machen.

Begonnen wurde 1984 mit einem Pilotprojekt zur Erfassung von Taufen und Sterbefällen in Kirchenbüchern. Die Daten wurden in ein EDV-System der BEDAG eingelesen und mittels des Statistikpakets SPSSX ausgewertet. Was zu Beginn noch gut funktionierte, wurde mit zunehmendem Umfang des Datenbestandes schnell zu einem wilden Haufen unübersichtlich gegliederter Daten. Innet kurzer Zeit hatte niemand der Forschungsgruppe mehr die Übersicht. Als, möglicherweise einzigen, Ausweg aus der damaligen Situation drängte sich eine komplette Reorganisation und Umstrukturierung des gesamten Datenbestandes auf. So wurde im Jahr 1991 der Datenbestand in die Form eines relationalen Datenmodells gebracht. Dies war eine sowohl richtige wie auch wichtige Entscheidung. Die relationalen Datenbanksysteme haben sich mittlerweile durchgesetzt und sind zum Standard geworden, um grosse Datenmengen zu verwalten. Heute läuft BERNHIST immer noch auf einer relationalen Datenbank. Die Idee war 1991 die historisch – statistischen Daten einer grösseren Zahl Benutzer zugänglich zu machen. So sollte der Zugriff via Netzwerk von jedem IBM – PC mit standardisierter Software aus über eine einfach zu bedienende Oberfläche ermöglicht werden. Zudem sollten mehrere Personen gleichzeitig mit BERNHIST arbeiten können. Um diese Vorgaben zu realisieren, wurde BERNHIST mittels dem Betriebssystem VMS auf einem Rechner der Informatikdienste der Uni Bern installiert und mit einer komplett in C geschriebenen Abfrageoberfläche den interessierten Nutzer zugänglich gemacht. Um mit der enorm schnellen Entwicklung im Informatikbereich Schritt halten zu können, wurde Mitte der Neunzigerjahre ein neues Konzept zu einer verbesserten Nutzung von BERNHIST entworfen. BERNHIST sollte fortan sowohl wie bisher als Netzwerk- und immer stärker als Internetversion wie auch als CD-ROM Version verfügbar sein. In einem ersten Schritt wurde der Datenbestand in das modernere und einfacher zu bedienende Datenbanksystem MS Access portiert. Von dieser Ausgangslage aus wurde die Datenbank mittels einem Java Applet und einem Delphi – Programm auf einem institutseigenen Server zum laufen gebracht. Diese Internetversion war bis zum Hackerangriff auf den Institutsserver im Frühling 2003 frei verfügbar. Auf MS Access basierend entstand eine CD-ROM, die durch eine Runtimeversion auf Windows Betriebssystemen installiert werden konnte.

Im neuen Jahrtausend drängte sich wiederum eine modernere, den stetig wachsenden Bedürfnissen der Benutzer gerecht werdende Version von BERNHIST auf. Nun wurde sowohl für die CD-ROM Version wie für die Internetversion eine Machbarkeitsstudie erstellt. Bald einmal zeichnete sich immer deutlicher ab, dass die Realisation einer CD-ROM eher schlecht als recht möglich ist. Der Standard der CD-Programme ist heutzutage dermassen hoch, dass ein konkurrenzfähiges Produkt nur mit einem hohen Einsatz an Zeit, Geld und technischem Know - How zu haben ist. Es wäre zwar technisch nicht unmöglich, eine Nachfolgeversion der bestehenden Access CD-ROM zu erstellen, aber die schlechte Dokumentation der Runtimeversion, der ungenügende Support des Herstellers und die Unmöglichkeit, den nötigen Support zu leisten, zwangen bald einmal zum Abbruch des CD-ROM Projekts.

Um möglichst viele Probleme auf Anwenderseite schon von Beginn weg zu eliminieren, wurde einer serverseitigen Lösung an Stelle der clientbasierten CD-ROM der Vorzug gegeben. So kann der Anwender, egal welches Betriebssystem er einsetzt, mit dem ihm vertrauten Browser auf den Datenbestand zugreifen. Diese Lösung ist durch die billigen Surfтарife, und der sich allmählich auch in Haushalten durchsetzenden Standleitungen (ADSL, Cablecom) vertretbar. Zudem hat mit dem Projekt der Swisscom „Schulen ans Netz“ eine wichtige Benutzergruppe freien Zugang zum Internet.

Dies eröffnet bei der Planung einer neuen Internetanwendung völlig neue Perspektiven und Möglichkeiten. Die Anforderungen, die an eine zeitgemässe Internetanwendung gestellt werden sind teilweise immer noch ähnlich wie vor zehn Jahren:

- Die Anwendung muss für alle Interessierten Nutzer uneingeschränkt verfügbar sein.
- Die Benutzeroberfläche soll den heutigen Webstandards entsprechen und einfach zu bedienen sein.
- Der Zugriff kann eingeschränkt und nur einer bestimmten Benutzergruppe gewährt werden.
- Die Administration der Hard- und der Software soll möglichst einfach organisiert und mit wenig Aufwand und ohne spezifisches Fachwissen zu bewerkstelligen sein.
- Die gesamte Lösung soll möglichst kostengünstig realisiert werden.

## Die Werkzeuge

Um diesen Vorgaben gerecht zu werden, wurde der Entschluss gefasst, das ganze Projekt mittels freier Software, so genannter Open Source Software, zu realisieren. Da das Herzstück von BERNHIST eine Datenbank ist, die möglichst schlank und schnell sein sollte, bot sich eine auf Linux basierende Lösung an. Am weitesten verbreitet um eine Datenbank unter Linux ans Internet anzubinden, ist die vielfach auch für grosse kommerzielle Anwendungen (z.B. Yahoo, Amazon) eingesetzte Lösung LAMP (**L**inux (Betriebssystem), **A**pache (Webserver), **M**ySQL (Datenbank) und **P**HP (Skriptsprache)).

Für die Leser, denen die oben genannten Begriffe fremd sind, folgt an dieser Stelle eine kurze Erläuterung der einzelnen Komponenten und ihrer Funktion.

### Linux<sup>1</sup>

Linux ist ein vollständiges Betriebssystem, das nicht von einer renommierten Softwarefirma entwickelt wurde, sondern innert wenigen Jahren aus der Zusammenarbeit einiger Computerfreaks entstanden ist. Allen voran war Linus Torvalds, der nicht länger von kommerzieller Software abhängig sein wollte und sein eigenes Betriebssystem zu entwickeln begann. Dieser Gedanke begeisterte viele Leute rund um den Globus, die sich schon bald tatkräftig an der Entwicklung von Linux beteiligten. Der Grundgedanke war ein freies Betriebssystem zu schaffen. Das heisst, dass Linux gratis einsetzbar und dass der Quellcode offen (eben Open Source) und frei verfügbar ist. Wenn man mit Linux nicht zufrieden ist, kann der Quellcode, sofern man über die nötigen Kenntnisse verfügt, gleich selbst den eigenen Bedürfnissen angepasst werden.

Obwohl im Gegensatz zu kommerzieller Software keine Gebrauchsanweisung mitgeliefert wird, kann man auf unzählige Onlinedokumentationen zurückgreifen. Wenn man hier keine Lösung findet, so wird einem mit Sicherheit in einer Linux Newsgroup geholfen.

Ein weiterer grosser Vorteil von Linux ist der modulare Aufbau der Software. Je nach Einsatzzweck werden nur die benötigten Module installiert. Dies ermöglicht ein sehr kompaktes, optimal auf die eigenen Bedürfnisse zugeschnittenes Betriebssystem, das keinen unnötigen Ballast mit sich führt und so eine bestmögliche Performance erreicht. Im Prinzip kann jedes Modul einzeln aus dem Internet heruntergeladen und installiert werden. Dies sei aber nur sehr erfahrenen Anwendern empfohlen, die auch vor einem hohen Zeitaufwand nicht

---

<sup>1</sup> <http://www.linux.org/>

zurückschrecken. Diese Arbeit nehmen ein verschiedene Distributionen (z.B. SUSE<sup>2</sup>, RedHat<sup>3</sup>) ab, die ein komplett einsatzfähiges Linux fixfertig auf CD oder DVD anbieten. Mit den letzten Distributionsausgaben wurde die Installation und Konfiguration stetig vereinfacht, so dass heute eine Linuxinstallation in etwa einer Windowsinstallation entspricht.

In den letzten Jahren hat sich Linux vom Spielzeug einiger Computerfreaks zum ernstzunehmenden Konkurrenten von Marktleader Microsoft gemausert. Immer mehr Universitäten, Organisationen und Firmen greifen zu Linux, wenn sie eine ebenso kostengünstige wie auch stabile Alternative zu den Windowssystemen suchen. Dass Linux auch von der Konkurrenz ernst genommen wird zeigt das Beispiel der Stadt München, die vor kurzem beschlossen hat, sämtliche Server und Desktopcomputer mit Linux statt mit Windows zu bestücken. Dies hat den Vize CEO von Microsoft dazu bewogen, extra nach München zu reisen, um der Stadt ein besonderes Windowsangebot zu unterbreiten. Seine Reise war vergebens, die Münchner liessen sich auch von einem billigeren Angebot nicht umstimmen.

## **Apache<sup>4</sup>**

Wenn man sich einmal für Linux als Serverbetriebssystem entschlossen hat, und einen Webserver betreiben will, kommt eigentlich nur Apache in Frage. Der Apache - Webserver ist derzeit der am weitesten verbreitete Webserver im Internet. Momentan wird Apache auf knapp 70% aller Webserver im Internet eingesetzt, Tendenz steigend (zum Vergleich: Microsoft 20%).<sup>5</sup> Es gibt mehrere Gründe, sich für den Apache zu entscheiden. Er ist ebenfalls kostenlos und inklusive Sourcecode verfügbar und zudem bietet er eine sehr hohe Performance bei vergleichsweise geringer Ressourcenauslastung, er bietet sehr viele, flexibel einsetzbare Funktionsmöglichkeiten, und er ist relativ einfach zu konfigurieren. Je nach Linuxdistribution wird er schon einsatzbereit mit dem Betriebssystem mitinstalliert.

## **MySQL<sup>6</sup>**

MySQL schickt sich an, den Erfolg von Linux im Betriebssystemsektor auf dem Datenbankmarkt zu wiederholen. Das Rezept ist eigentlich einfach: Man nehme eine Datenbank, die zwar noch nicht die gleiche Funktionsfülle wie kommerzielle

---

<sup>2</sup> <http://www.suse.de/de/index.html>

<sup>3</sup> <http://www.redhat.com/>

<sup>4</sup> <http://www.apache.org/>

<sup>5</sup> [http://news.netcraft.com/archives/web\\_server\\_survey.html](http://news.netcraft.com/archives/web_server_survey.html)

<sup>6</sup> <http://www.mysql.com/>

Datenbanksysteme anbieten kann, die dafür aber zuverlässig und schnell funktioniert, ohne allzu grosse Anforderungen an die Hardware zu stellen. Dann gebe man dieser Datenbank eine Opensource – Lizenz, was die kostenlose Nutzung ermöglicht.

MySQL hat vor allem in den letzten Jahren als Datenbank für dynamische Webseiten weite Verbreitung gefunden und stellt schon beinahe einen De – Facto - Standard dar. In Kombination mit PHP oder Pearl stellt MySQL für immer mehr Webseiten die Datenbasis zur Verfügung.

## PHP<sup>7</sup>

PHP ist ein rekursives Akronym, bedeutet PHP Hypertext Processor und gehört zu den serverseitigen Skriptsprachen. Eine Skriptsprache ermöglicht es, kleine Programme in HTML einzubetten und auszuführen. Im Gegensatz zu clientseitigen Skriptsprachen wie z.B. JavaScript, bei denen der Code erst durch den Browser ausgeführt wird, werden serverseitige Skriptsprachen wie PHP schon vom Webserver interpretiert, bevor die Seite an den Browser geschickt wird. Nachdem der PHP – Code interpretiert wurde, sendet der Webserver eine reine Standard – HTML – Datei an den Browser. Das Skript wird komplett auf dem Server verarbeitet. Dies hat mehrere Vorteile:

- Keine Probleme mit der Browser – Kompatibilität: Da die PHP – Skripte ausschliesslich vom Webserver verarbeitet werden, braucht man sich um die Browsereinstellungen der User nicht mehr zu kümmern. Es ist ja nicht mehr ersichtlich und nachvollziehbar, welche Codefragmente von PHP generiert wurden, und welche direkt in HTML codiert wurden.
- Zugriff auf serverseitige Ressourcen: Viele Informationen sind nur für ein auf dem Webserver laufendes Skript verfügbar. Dazu gehören einfache Informationen wie Datum und Zeitangaben (wäre auch mit JavaScript realisierbar, würde aber das Datum und die Zeit des Clients anzeigen), oder aber kompliziertere Informationen wie etwa die Daten, die in einer MySQL – Datenbank gespeichert sind.
- Lastverringern auf dem Client: Bei clientseitigen Skriptanwendungen kann es bei langsamen Computern zu beträchtlichen Verzögerungen der Anzeige kommen, weil der Browser zuerst das Skript interpretieren muss, bevor die Webseite angezeigt wird.

---

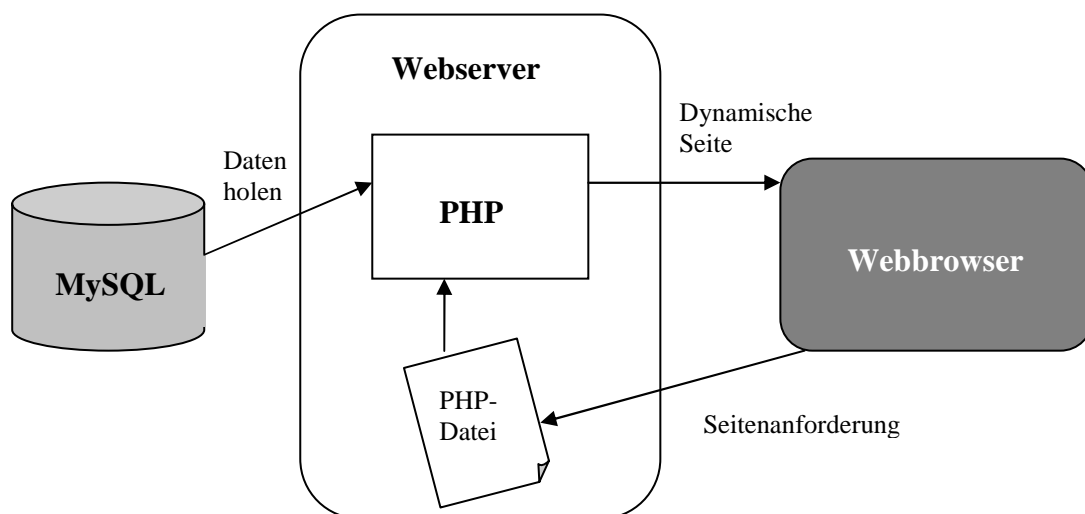
<sup>7</sup> <http://www.php.net/>

Beim serverseitigen Skripting verlagert sich diese Last auf die Maschine, auf der der Webserver läuft.

Nachdem nun die einzelnen Komponenten vorgestellt wurden, zeigt der nächste Abschnitt, wie diese mächtigen Werkzeuge miteinander funktionieren.

## Das Prinzip einer dynamischen Webseite

Die ganze Idee einer datenbankgestützten Webseite besteht darin, dass der Inhalt der Seite in einer Datenbank liegt und dynamisch daraus geholt wird um Webseiten zu erzeugen, die Leute mit einem normalen Webbrowser ansehen können. Auf der einen Seite des Systems hat man also einen Benutzer, der eine Webseite in den Browser lädt und davon ausgeht, eine Standard – HTML – Seite zu sehen. Am anderen Ende ist der Inhalt der Webseite, der sich in einer oder mehreren Tabellen einer MySQL – Datenbank befindet, und diese Datenbank weiss



**Abbildung1:** PHP vermittelt zwischen dem Webserver und der Datenbank.

lediglich, wie sie auf Anfragen oder Befehle antworten muss. Wie in der Grafik zu sehen ist, vermittelt PHP zwischen beiden Anforderungen, denn es spricht beide Sprachen. PHP verarbeitet die Seitenanforderung, holt die gewünschten Daten aus der MySQL – Datenbank und sendet sie dann als formatierte HTML – Seite an den Browser. In den meisten Fällen wird PHP in ein HTML Grundgerüst, welches für das Layout der Seite zuständig ist, eingebettet und zeigt den gewünschten Inhalt an der gewünschten Stelle in der Webseite an. Webdesign



mit Hilfe von PHP ist also auch ein Konzept, um Inhalt und Form voneinander zu trennen. Weiterführende Möglichkeiten dieser Technik und deren Vorteile werden zu einem späteren Zeitpunkt noch diskutiert.

## **Das Konzept von BERNHIST**

Was bedeutet nun die Theorie der letzten Seiten konkret in Bezug auf BERNHIST? Der erste Schritt zur neuen Version war die Portierung der Access – Datenbank auf einen MySQL – Server. Die Datenbankstruktur wurde zuerst 1:1 übernommen und anschliessend vereinfacht. Die aktuelle Version von BERNHIST besteht aus verschiedenen PHP – Skripts, die auf Anfrage eines Benutzers vom PHP – Interpreter eines Apache – Webservers ausgeführt werden, die verlangten Daten aus der Datenbank abfragen und das Resultat an den Browser des Benutzers senden. So ist es möglich von jedem Internetanschluss aus auf die gewünschten Daten, die in BERNHIST gespeichert sind, zuzugreifen.

## **Die Datenbank**

Das Herzstück von BERNHIST ist die Datenbank, in der die Daten strukturiert gespeichert sind. Hier entscheidet sich, wie die Daten verwaltet werden, ob das Datenbankschema optimal ist und ob dieses eine sinnvolle Abfrage ermöglicht. Wenn dieses Fundament nicht besteht, können mit keiner auch noch so ausgeklügelten Anwendung Daten aus BERNHIST abgefragt werden. Aus diesen Gründen ist auf die Administration der Datenbank grossen Wert zu legen und gehört zu den wichtigsten Aufgaben des Administrators von BERNHIST. In der ursprünglichen Konfiguration ist ein MySQL – Server der Version 3.23.56 aufgesetzt. Wie bei OpenSource – Software üblich, werden ständig neue und verbesserte Versionen veröffentlicht. Da MySQL noch nicht über alle Funktionen eines kompletten Datenbanksystems verfügt, sind die Updates bei dieser Software besonders häufig. Dies ist aber kein Grund zur Beunruhigung, die Funktionen, die bei der momentanen Version implementiert sind, reichen für den Betrieb von BERNHIST vollständig. Solange keine bahnbrechenden Neuerungen wie z.B. ein GUI (Graphical User Interface) anstehen, gibt es keinen Grund für einen Versionswechsel. Je nach eingesetzter Linuxversion kann diese sowieso variieren.

Anders als bei Access, verfügt MySQL nicht über eine grafische Benutzerschnittstelle, mit der an der Datenbank manipuliert werden kann. Um die Datenbank zu warten oder um

Änderungen vorzunehmen stehen zwei Möglichkeiten zur Verfügung. Je nach Einsatzzweck ist eher der einen oder der anderen der Vorzug zu geben. Einerseits ist es möglich direkt via Konsole auf den Datenbankserver zuzugreifen. Dies geschieht direkt am Server oder remote von einem Terminal<sup>8</sup> aus. Dies setzt aber Grundkenntnisse von Linux und von MySQL voraus, da alle Manipulationen via Kommandozeile erfolgen müssen. Als Referenz sei an dieser Stelle das Buch MySQL<sup>9</sup> von Michael Kofler empfohlen. Weitere Bücher zum Thema sind in der Literaturangabe aufgeführt. Die etwas benutzerfreundlichere Lösung ist phpMyAdmin<sup>10</sup>. Dies ist eine Ansammlung von komplett in PHP geschriebenen Skripts, die direkt auf dem Webserver in einem eigenen Verzeichnis liegen. Mit Hilfe dieser Skripts kann mit einem Browser auf die Datenbank zugegriffen werden. Mit diesem Werkzeug steht einem der volle Umfang der Datenbankfunktionen (erstellen, ändern, löschen, etc) zur Verfügung. Um phpMyAdmin fehlerfrei zum laufen zu bringen, muss lediglich die Datei config.inc.php dem System angepasst werden. Dabei muss drei Variablen der richtige Wert zugewiesen werden:

Die Variable `$cfg['PmaAbsoluteUri']` gibt den absoluten Ort von phpMyAdmin an. Beim Webserver von BERNHIST ist das:

```
$cfg['PmaAbsoluteUri'] = 'http://hist41.unibe.ch/phpMyAdmin';
```

Zudem muss phpMyAdmin über die Zugriffsrechte des Datenbankservers verfügen. Das geschieht über eine Variable für den Benutzernamen und über eine Variable für das Passwort. Zum Thema Sicherheit: Dieses Passwort steht in Klartext in einer Datei auf dem Server! Aus diesem Grund wird ein Benutzer erstellt, der nur Zugriff auf die Datenbank bernhist und nicht auf den ganzen Server hat. Es sollte ein Passwort verwendet werden, das nicht auch anderweitig in Gebrauch ist!

```
$cfg['Servers'][$i]['user']           = 'bernhist';  
$cfg['Servers'][$i]['password']      = '123456';
```

Wenn man sich mit Hilfe eines dieser Werkzeuge Zugang zum Datenbankserver verschafft hat, sieht man, dass die Datenbank BERNHIST aus sechs Tabellen besteht:

---

<sup>8</sup> Um von einem entfernten Windowsrechner auf einen Linuxserver zuzugreifen eignet sich das Programm SSH-Client (<http://hist41.unibe.ch/downloads/index.html>).

<sup>9</sup> Kofler, Michael: MySQL Einführung .Programmierung, Referenz. Addison-Wesley 2001.

<sup>10</sup> <http://www.phpmyadmin.net>

feedback_db	Speichert die Feedbacks der Feedbackseite, keine relevanten Daten.
obs_base	Enthält alle Werte. Grösste und wichtigste Tabelle.
ort1	Enthält alle Orte.
quellenthema	Enthält alle Quellenbegriffe.
thema	Enthält alle standardisierten Quellenbegriffe.
thema_in	Enthält die Verknüpfungen von Thema und Quellenthema.

Wahrscheinlich stellt man sich als erstes die Frage, weshalb sowohl Quellenbegriffe wie auch standardisierte Quellenbegriffe je eine Tabelle in der Datenbank für sich in Anspruch nehmen. Diese Einteilung wurde von Peter Häberli und Klaus Imfeld bei der Erstellung von BERNHIST V2.0<sup>11</sup> vorgenommen. Um einen möglichst übersichtlich organisierten Themenkatalog zur Auswahl stellen zu können, mussten die in der Datenbanktabelle quellenthema unstrukturiert abgelegten Themen mit einem standardisierten Oberbegriff versehen werden. Diese standardisierten Begriffe befinden sich separiert von den Quellenthemen in der Tabelle thema. Die Tabelle thema\_in speichert nun die Verknüpfung von Quellenthemen mit standardisierten Themen.

In der Tabelle ort1 sind die Orte aller Hierarchieebenen gespeichert. Hier wurde die Struktur der Tabelle den neuen Anforderungen angepasst. Um die Abfrage der Orte in einer hierarchischen Baumstruktur zu ermöglichen, wurde bisher ein rekursiver Algorithmus verwendet. Um die Geschwindigkeit der Abfrage zu erhöhen und gleichzeitig den Arbeitsspeicher des Servers zu entlasten, wurde die Datenbanktabelle um fünf Spalten erweitert, die jeweils die Hierarchiestufe anzeigen. So ist nun ersichtlich, auf welchem Level (Kanton, Landesteil, Amtsbezirk, Kirchgemeinde, Einwohnergemeinde) sich ein Ort befindet und welche Orte ihm über- und welche untergeordnet sind.

In der Tabelle obs\_base sind alle in BERNHIST verfügbaren Werte gespeichert. Zu jedem Wert hat es neben dem Primärschlüssel je eine Spalte für das Start- und das Endjahr, den Fremdschlüssel des Ortes und den Fremdschlüssel des Themas. So kann mit Angabe von Start- und Endjahr, Ort- und Themenschlüssel der zugehörige Wert abgefragt werden.

Auf der Startseite von BERNHIST besteht die Möglichkeit ein Feedback zu schreiben. Um dies zu ermöglichen und die Feedbacks via Webseite anzuzeigen, müssen sie in einer Datenbank gespeichert werden. Diese Aufgabe übernimmt die feedback\_db, die ansonsten mit der Datenabfrage nichts zu tun hat.

Das detaillierte Datenbankschema befindet sich im Anhang.

---

<sup>11</sup> Häberli Peter, Imfeld Klaus: HGIS BERNHIST V2.0. Seminararbeit Uni Bern 1995.

Ein weiterer Grund die bestehende Datenbank zu erweitern, wäre die Implementation eines Benutzermanagements. Falls es möglich sein sollte, eine grössere Anzahl von Benutzern zu verwalten, kommt man nicht um zusätzliche Datenbanktabellen, die diese Arbeit abnehmen, herum. Wie eine einfache Zugangskontrolle ohne Datenbank realisiert werden kann, wird im Abschnitt zur PHP – Programmierung und im Anhang gezeigt.

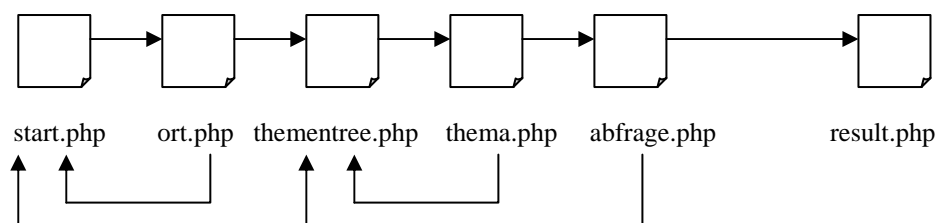
## Die Webanwendung

Anstatt sich auf ein blosses Abfragen von historisch – statistischen Daten zu beschränken, wurde BERNHIST als Portal konzipiert, das dem Benutzer die Möglichkeit bietet mehr über BERNHIST und seine Entstehung zu erfahren, sich mit anderen Benutzern via Feedback auszutauschen und zeigt wo weiterführende Informationen erhältlich sind. Wie bei jedem Portal im Internet, wird sich nach einiger Zeit weisen, welche Dienste auf Interesse stossen und welche eher nicht. Es ist auch denkbar die Webseite den Bedürfnissen der Benutzer entsprechend auszubauen.

Da BERNHIST auf einem Drei – Achsen – Modell basiert muss auch eine für den Benutzer logische Abfrage nach den drei Achsen Raum, Thema und Zeit geschaffen werden. Jeder Wert kann somit durch die folgenden drei Achsen festgehalten werden:

- Raum: geografischer Bezugsort, auf welchen sich der Wert bezieht.
- Thema: inhaltliche Aussage, die dem Wert zugeordnet ist.
- Zeit: Zeitpunkt oder –intervall, welches mit dem Wert korrespondiert.

In dieser Reihenfolge ist die Abfrage, durch die der Benutzer geführt wird auch organisiert:



**Abbildung 2:** Ablauf der Abfrage.

Damit die Abfrage funktioniert, müssen in jedem Skript zwei Dateien eingebunden werden. Zum einen ist die Verbindung zum Datenbankserver in der Datei conn.php im Verzeichnis conn ausgelagert. Dies hat den Vorteil, dass wenn sich die Zugangsdaten einmal ändern sollten, nur eine Datei modifiziert werden muss. In den Skripten mit den Abfragebäumen, muss auf die Datei TreeMenu.php verwiesen werden. Bei einer Neuinstallation ist darauf zu achten, dass diese Zusatzdateien vorhanden sind und dass die Pfadangaben in den Skripten stimmen. Die Datei header.php, die für den Kopf jeder Seite verantwortlich ist, ist ebenfalls mit dem include – Befehl jeder Webseite beigefügt, trägt aber nichts zum Gelingen einer Abfrage bei.

## **Der Ablauf einer Abfrage**

- start.php: Auf dieser Seite werden alle Orte aus der Tabelle ort1 der Datenbank bernhist ausgelesen und ihrer Hierarchiestufe entsprechend im Baum angeordnet. Alle Orte erscheinen als Link, der bei einem Mausklick den Key, den Namen und den Ortstyp des angeklickten Ortes an die Seite ort.php weitergibt.
- ort.php: Dient dem Benutzer nur zur Überprüfung des gewählten Ortes und führt ihn weiter zur Themenauswahl. Im Falle dass der Benutzer unabsichtlich auf einen falschen Ort geklickt hat, kann er von hier aus zur Ortsauswahl zurückkehren.
- thementree.php: Auf dieser Seite werden alle standardisierten Themen und alle Quellenthemen aus der Datenbank geholt und nach Themenbereich gegliedert. Alle wählbaren Themen erscheinen als Link, der bei einem Mausklick den Key, das Thema und den Thementyp an die Seite abfrage.php weitergibt.
- abfrage.php: Dies ist die eigentliche Hauptseite der Abfrage. Bis hier hat der Benutzer seinen gewünschten Ort und sein gewünschtes Thema ausgewählt. Mit diesen beiden Keys wird nun in der Datenbank überprüft, zu welchen Jahren Werte zu den gewünschten Variablen verfügbar sind. Sind zu einer Ort – Thema – Kombination keine Werte vorhanden, wird dies dem Benutzer an dieser Stelle angezeigt. Falls Daten vorhanden sind, ist ersichtlich zu welchen Jahren dies der Fall ist und wie viele Datensätze gefunden wurden. Die verfügbaren Jahrzahlen werden als Start- und als Endjahr in einer Menüliste angezeigt, was dem Benutzer eine Eingrenzung des Zeitraumes erlaubt. Nachdem alle drei Achsen bestimmt sind, kann die Abfrage gestartet werden.
- result.php: Auf dieser Seite werden alle die vom Benutzer gewünschten Daten angezeigt. Um den Umgang und die weitere Nutzung zu erleichtern, sind noch einige

Komfortfunktionen implementiert. Es besteht die Möglichkeit sich die Daten per Mausklick als Liniendiagramm anzeigen zu lassen. In einem Popup – Fenster erscheint eine dynamisch generierte Grafik. Sie hat die Funktion, dem Benutzer eine praktische Übersicht über die gewählten Daten zu liefern um schneller über deren weitere Verwendbarkeit zu entscheiden. Um die Daten professioneller grafisch aufzubereiten, sollte auf ein Statistikprogramm zurückgegriffen werden. Um dies zu vereinfachen wird ein Export nach Microsoft Excel angeboten. Der Export nach dem universell einsetzbaren Dokumentformat PDF rundet die Möglichkeiten der Datenweiterverwendung ab.

## Zugriffskontrolle mit Sessions

Damit nicht alle Surfer in Internet vollen Zugang zu den Daten haben, ist der Einstieg in die Abfrage mit Benutzername und Passwort gesichert. Um Manipulationen an allen geschützten Seiten auszuschliessen, bietet sich in PHP der Einsatz einer Session an. Mit dem Start einer Session wird automatisch eine Sessionidentifikation generiert und als Cookie auf dem Rechner des Benutzers hinterlegt. Eine Session bietet nun die Möglichkeit, verschiedene Variablen zu speichern, auf die solange zurückgegriffen werden kann, wie die Session aktiv ist. Um Daten aus BERNHIST abzufragen wird nun nach der Variabel `$authorized` in der aktuellen Session verlangt. Diese wird aber nur registriert, wenn zuvor der richtige Benutzername und das richtige Passwort eingegeben und verifiziert wurden. Erst wenn beide Angaben stimmen, wird mit dem Befehl `session_register('authorized');` die Variabel in der Session registriert und dem Benutzer Zugang zu den Daten gewährt. Der Klick auf Logout, führt auf eine Seite auf der nichts anderes gemacht wird als mit dem Befehl `session_unregister('authorized');` die Zugangsvariabel wieder aus der Session zu löschen.

Der Sourcecode aller HTML – Seiten und aller PHP – Skripte ist auf beigelegter CD verfügbar. Weiter Kopien befinden sich im Historischen Institut sowohl auf CD wie auf dem Institutsserver.

## Installation

Sollte dem Server einmal etwas zustossen oder will man die Webanwendung samt Datenbank auf einem anderen Server laufen lassen, so kommt man um eine Neuinstallation des kompletten Systems nicht herum. Alle Software und sämtlicher Code der benötigt wird, liegt diesem Manual bei. Die Anleitung beschreibt zuerst die Installation von Linux samt dem Apache Webserver und dem Datenbankserver MySQL. Es wird ein RedHat Linux der Version 9 (shrike) verwendet. Diese oder aktuellere Versionen sind entweder frei verfügbar und können vom Internet heruntergeladen<sup>12</sup> werden oder es besteht die Möglichkeit die CD's samt Handbuch im Handel zu kaufen.

## Der Server

Bevor die Anleitung der Installation von Linux beginnt, noch ein Wort zur Hardware. Im Gegensatz zu früheren Distributionen spielen die einzelnen Hardwarekomponenten keine grosse Rolle mehr, respektive stellen sie kein Hindernis bei der Installation mehr dar. Wenn keine exotischen Grafikkarten oder ähnliches verwendet wird, erkennt ein modernes Linux alle wichtigen Bestandteile selbst und installiert analog zu Windows die passenden Treiber gleich mit. Es ist empfohlen für die Installation und den Betrieb von BERNHIST einen Rechner zu verwenden, der mindestens mit einem Pentium III Prozessor und mit 512 MB Arbeitsspeicher ausgestattet ist. Die Grösse der Festplatte spielt nur eine untergeordnete Rolle.

Um die Installation zu beginnen, muss der Computer mit eingelegter CD1 gestartet werden. Es erscheint ein Menu, das durch die verschiedenen Installationsschritte führt und mehr oder weniger selbsterklärend ist. Die wichtigen Punkte, die es zu beachten gilt sind hier aufgeführt.

Zuerst muss die Installationssprache (German) und die Tastatur (Swiss German) gewählt werden. Bei der Mouseerkennung sollte trotz einer umfangreichen Auswahl an Möglichkeiten auf Experimente verzichtet werden. Am sichersten ist die vom System gewählte Voreinstellung. Da keine der vordefinierten Installationen den Anforderungen von BERNHIST entspricht, muss die Option Benutzerdefiniert gewählt werden. Das ermöglicht eine gezielte Auswahl der benötigten Module und man hat die Möglichkeit Softwarepakete die hier nicht von Nutzen sind (Grafik, Sound, Film, etc...) auszuschliessen. Diese Auswahl erfolgt erst später.

---

<sup>12</sup> <http://www.redhat.com/download/products.html>  
<http://www.linuxiso.org/>



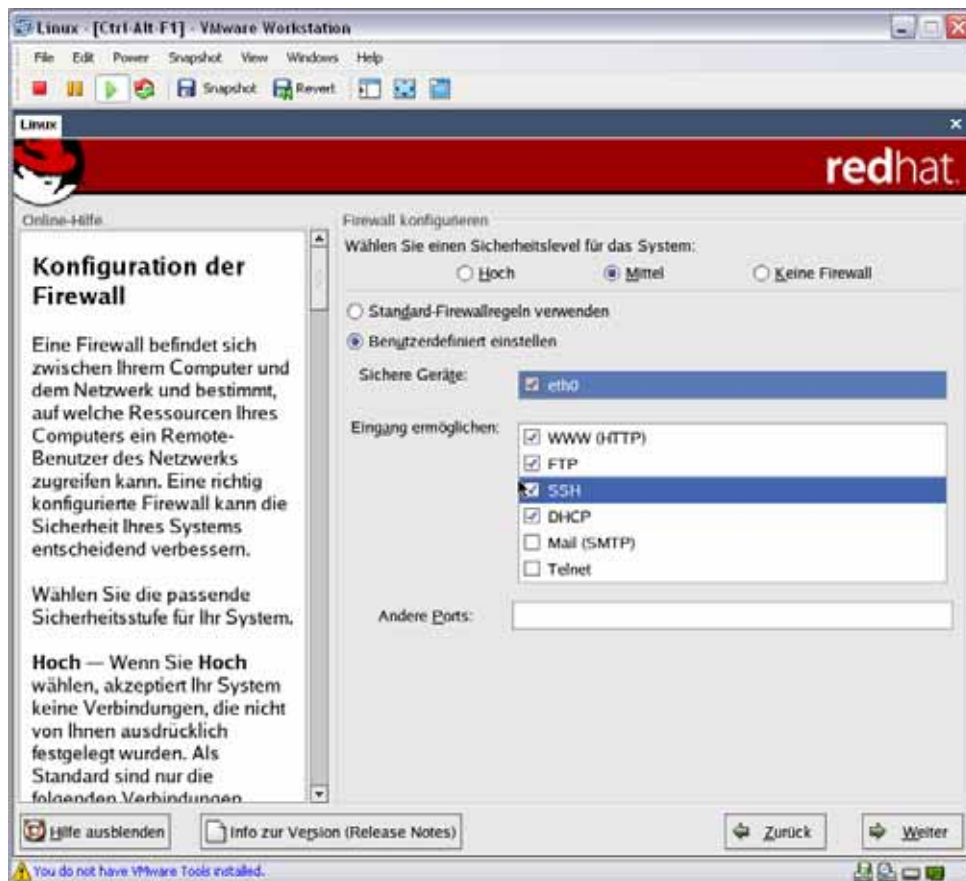
Bevor irgendetwas auf die Festplatte geschrieben werden kann, wird diese neu formatiert. Es ist empfohlen, die automatische Partitionierung zu verwenden, die noch zweimal nachfragt, ob man auch wirklich sicher sei alles zu löschen und alle Linuxpartitionen in diesem System zu entfernen. Dies ist immer mit Ja zu beantworten.

Im nächsten Fenster wird der Bootloader konfiguriert. Hier sind alle Einstellungen wie vom System vorgeschlagen zu belassen und kein Passwort zu setzen. Sonst muss bei jedem Startvorgang dieses Passwort eingegeben werden, was ein Neustart von einem entfernten Rechner verhindern würde.

Die Netzwerkkarte (eth0 / DHCP) sollte erkannt worden sein und die Einstellung des Hostnamens erfolgt automatisch über DHCP (Dynamic Host Configuration Protocol).

Die nächsten Benutzerdefinierten Änderungen erfolgen wie in unten stehender Abbildung gezeigt bei den Einstellungen der Firewall. Hier muss das Sicherheitslevel des Systems auf Mittel gestellt und die Firewallregeln müssen benutzerdefiniert festgelegt werden. Das geschieht mit Häkchen, die bei eth0, WWW (HTTP), FTP, SSH und DHCP gemacht werden müssen.





Bei der Sprachunterstützung wählt man German (Switzerland) und mit ein wenig scrollen und suchen findet man bei den Ländereinstellungen Europa/Zürich.

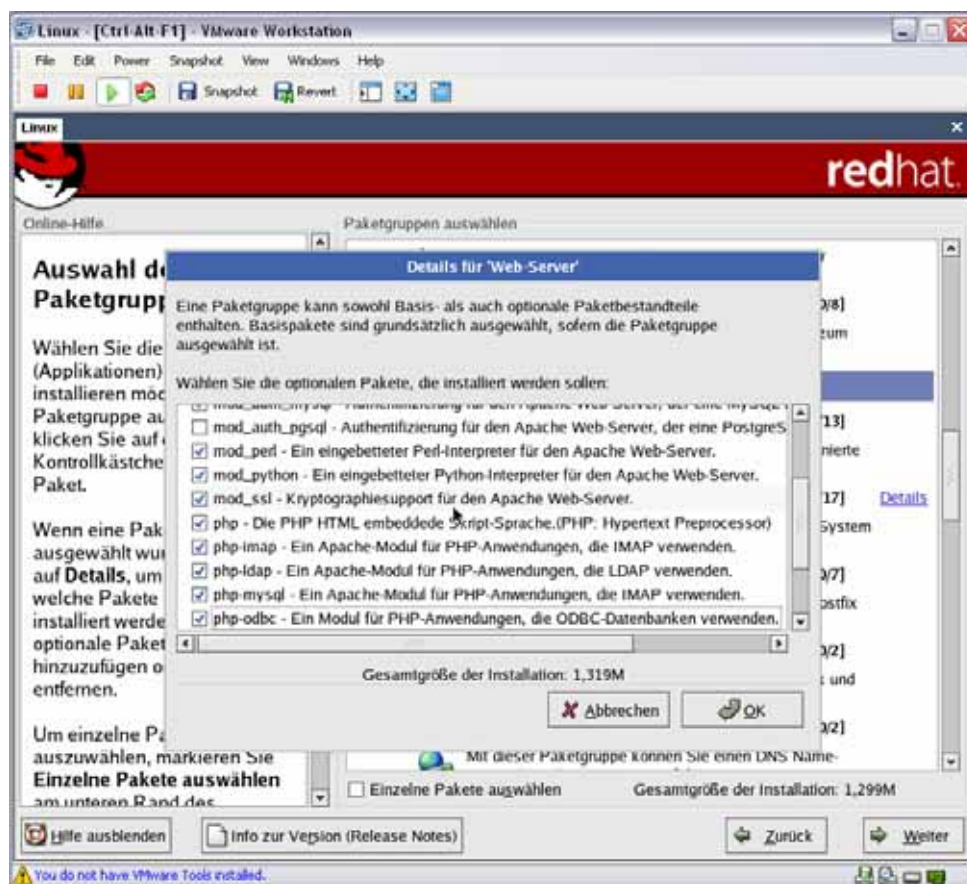
Das nächste Fenster ist das Wichtigste der ganzen Installation. Hier muss das Passwort für den root festgelegt werden. Root ist in Unix / Linux – Umgebungen so etwas wie der Administrator bei Windows. Wenn man sich als root anmeldet, verfügt man über alle Rechte des Systems. Jede Manipulation am Betriebssystem verlangt nach dem Rootpasswort. Das heisst, ohne dieses Passwort kann der Computer neu installiert werden. Hoffentlich reicht diese kurze Ausführung, um die Bedeutung dieses Passwortes zu unterstreichen.

Im nächsten Schritt muss die Authentifizierung konfiguriert werden. Es können die Standardwerte übernommen werden. Sowohl die MDS– wie auch die Shadow – Passwörter müssen aktiviert sein.

Bevor nun das Programm mit der Installation beginnt, besteht jetzt die Möglichkeit das Betriebssystem benutzerdefiniert mit den einzelne Komponenten zusammenzustellen. Die meisten Einstellungen wie die Wahl des Desktops, sind Geschmackssache und nur für den Administrator relevant. Wichtig ist das Server Konfigurationstool, der Webserver und der SQL – Datenbankserver. Sowohl beim Web- wie beim Datenbankserver müssen mit einem Klick auf die Details weitere Änderungen vorgenommen werden. Wie in der nächsten

Abbildung ersichtlich ist, öffnet sich mit einem Mausklick auf den Link Details ein kleines Fenster in dem ersichtlich ist, was standardmässig installiert wird und was nicht. Wie auf Seite 7 in diesem Manual beschrieben, wird der PHP – Code vom Webserver interpretiert und dieser generiert anschliessend die HTML – Seiten, die der Surfer zu sehen bekommt. Damit der Webserver weiss, mit welchen Seiten er was zu tun hat, wird ein PHP – Interpreter als Modul in den Webserver integriert. Genau diese Möglichkeit hat man nun in dieser Detailansicht des Webserver. Damit die PHP – Skripte überhaupt interpretiert werden können, müssen alle PHP – Module gemäss der Abbildung mitinstalliert werden. Mit einem Klick auf Ok schliesst sich das Detailfenster und es bleibt noch die Auswahl des richtigen Datenbankservers.

Wie zuvor muss auch bei der SQL – Datenbank das Detailfenster geöffnet und mit einem Häkchen der für BERNHIST relevante Datenbankserver mysql – server ausgewählt werden.



Nach dem Schliessen des Fensters mit OK, wird durch einen Mausklick auf Weiter der Installationsprozess in Gang gesetzt. Unabhängig davon welche Softwarepakete zuvor ausgewählt wurden, verlangt der Installationsprozess nach allen drei CD's.

Ist die Installation fertig, startet man den Computer neu und meldet sich mit dem Benutzernamen root und dem Rootpasswort an.

## Dienste starten

Um einen funktionierenden Webserver mit einem Datenbankserver betreiben zu können, müssen beide Dienste noch gestartet werden. Damit diese beiden Server nach jedem Systemstart zur Verfügung stehen, empfiehlt es sich diese Startvorgänge zu automatisieren. Dies erreicht man unter dem Hauptmenu > Systemeinstellungen > Servereinstellungen > Dienste. Von da aus lassen dich alle Dienste starten, stoppen oder automatisieren. Wichtig ist nun, dass sowohl bei httpd (Webserver) und bei mysqld (Datenbankserver) ein Häkchen gemacht wird. Mit je einem Klick auf den Startknopf erreicht man, dass die Dienste unmittelbar gestartet werden.

## Installation testen

Um die frische Installation und die eben gestarteten Dienste zu testen, öffnet man eine leere Textdatei und schreibt folgenden Code hinein:

```
<?php  
phpinfo();  
?>
```

und benennt sie mit phpinfo.php. Diese Testdatei, bei der die Endung .php wichtig ist, wird nun im Webroot mit dem Pfad /var/www/html gespeichert. Webroot ist das Verzeichnis, das durch Eingabe des Rechnernamens oder von Localhost im Browser angezeigt wird. Um die Testdatei im Browser zu betrachten muss folgende Adresse eingegeben werden:

localhost/phpinfo.php oder wenn die Netzwerkeinstellungen bereits gemacht sind:

<http://hist41.unibe.ch/phpinfo.php>

Wenn die PHP – Angaben im Browser erscheinen, funktioniert das Zusammenspiel des Webservers mit dem PHP – Interpreter.



Der nächste und letzte Schritt, damit BERNHIST über das Internet verfügbar ist, ist die Installation der HTML – Seiten und der PHP – Skripte und die Anbindung der Datenbank.

## Einrichten der Datenbank

Bevor die Datenbank von Nutzen ist, muss sie zuerst mit Daten gefüttert werden. Als erster Schritt muss eine Datenbank mit dem Namen bernhist erstellt werden. Am einfachsten ist dies via phpMyAdmin (Seite 9), wo gleich von der Hauptseite aus neue Datenbanken erstellt werden können. Selbstverständlich kann auch von der Konsole aus mit SQL eine neue Datenbank erstellt werden. Die restliche Struktur wird mit SQL – Dateien, die auf der CD im Verzeichnis sql liegen, automatisch erstellt. In diesen SQL – Dateien, es hat pro Tabelle eine gleichnamige Datei (z.B. obs\_base.sql), sind sowohl die Struktur mit den Spalten und ihren Bezeichnungen wie auch alle zugehörigen Daten, also der eigentliche Inhalt der Datenbank, gespeichert. Wenn die Datenbank erstellt ist, können die SQL – Dateien relativ einfach eingelesen werden. Der einfachere Weg führt wiederum über phpMyAdmin, wo der Pfad einer Batchdatei angegeben werden kann, und mit einem Klick auf Ok wird die entsprechende Tabelle erstellt und mit den Inhalten gefüllt. Dieses Vorgehen funktioniert aber nur bis zu

einer gewissen Grösse der SQL – Dateien. Da aber die Datei obs\_base.sql die 100 MB - Grenze deutlich überschreitet, muss zum Einlesen auf die Konsole zurückgegriffen werden. Dieser Vorgang sei hier kurz beschrieben:

Als erstes muss sichergestellt sein, dass man über root - Rechte verfügt. Falls nicht, wird im Terminal der Befehl su eingegeben, anschliessen wird nach dem root – Passwort verlangt.

Zum einlesen der Datei obs\_base.sql muss folgendes Kommando eingetippt werden:

```
>mysql -u root -p bernhist < obs_base.sql
```

```
Enter password: xxx
```

An Stelle von xxx muss das root – Passwort eingegeben werden.

## Wohin mit den Skripts?

Wo und in welchem Verzeichnis die Webseiten und Skripts genau liegen spielt nur eine untergeordnete Rolle. Wichtig ist, dass alle Pfadangaben stimmen und aus diesem Grund ist es am einfachsten, die bestehende Struktur zu übernehmen.

Der Pfad zum Verzeichnis, wo die Webseiten liegen müssen (der so genannte Webroot ) ist:

```
/var/www/html
```

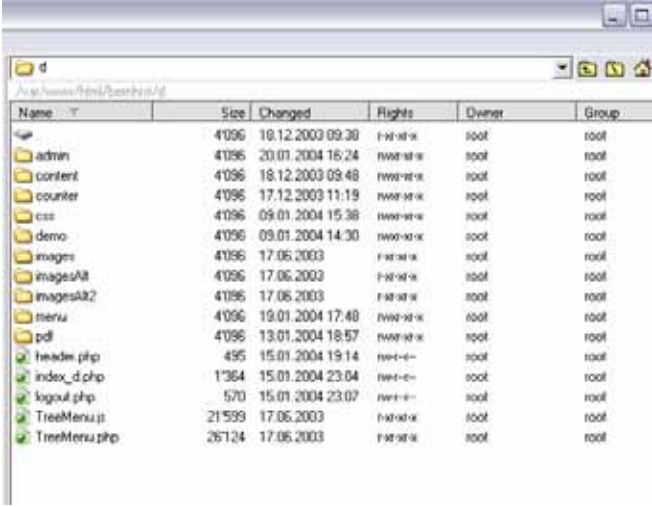
Um Ordnung auf dem Server zu haben, wird hier ein Verzeichnis bernhist angelegt, wo alle weiteren Dateien abgelegt werden. Die wichtigsten Verzeichnisse hier sind:

- conn mit der Datei conn.php, die für die Verbindung zur Datenbank von allen Skripts benötigt wird,
- png das alle Bilddateien enthält und
- d (für Deutsch, andere Sprachversionen könnten an dieser Stelle mit einem eigenen Verzeichnis verzweigt werden) das alle weiteren Dateien zur Abfrage enthält.

Die weitere Aufteilung der Dateien ist folgendermassen organisiert:

- admin enthält alle Skripte die zur Administration von BERNHIST benötigt werden (Einlesetool).
- content ist das Verzeichnis, das alle Skripte der Abfrage (Seite 12) enthält.

- css enthält alle Cascading Style Sheets.
- demo hat den identischen Inhalt wie das Verzeichnis content, ohne aber die Möglichkeit zur Datenabfrage.
- menu ist das Verzeichnis der eigentlichen Webseiten, die erscheinen wenn ein Surfer die BERNHIST – Seite besucht.
- Treemenu.js und Treemenu.php sind die beiden Dateien, die für den Aufbau des Auswahlbaumes von Ort und Thema verantwortlich sind und sollten weder geändert noch an einem anderen Ort abgespeichert werden.



Name	Size	Changed	Rights	Owner	Group
	4096	18.12.2003 09:30	rwxr-xr-x	root	root
admin	4096	20.01.2004 16:24	rwxr-xr-x	root	root
content	4096	18.12.2003 09:48	rwxr-xr-x	root	root
counter	4096	17.12.2003 11:19	rwxr-xr-x	root	root
css	4096	09.01.2004 15:36	rwxr-xr-x	root	root
demo	4096	09.01.2004 14:30	rwxr-xr-x	root	root
images	4096	17.06.2003	rwxr-xr-x	root	root
imagesAll	4096	17.06.2003	rwxr-xr-x	root	root
imagesAll2	4096	17.06.2003	rwxr-xr-x	root	root
menu	4096	19.01.2004 17:40	rwxr-xr-x	root	root
pdf	4096	13.01.2004 18:57	rwxr-xr-x	root	root
header.php	495	15.01.2004 19:14	rwxr-xr-x	root	root
index_d.php	1364	15.01.2004 23:04	rwxr-xr-x	root	root
logout.php	570	15.01.2004 23:07	rwxr-xr-x	root	root
TreeMenu.js	21599	17.06.2003	rwxr-xr-x	root	root
TreeMenu.php	26124	17.06.2003	rwxr-xr-x	root	root

Nun ist der Webserver bereit die ersten Surfer zu empfangen. Wird in einen Browser die URL <http://hist41.unibe.ch/bernhist/d/menu/index.php> eingetippt, erscheint nun die Startseite von BERNHIST. Um Dateien von einem entfernten Computer auf den Server zu kopieren eignet sich das Programm WinSCP<sup>13</sup>, das ebenfalls auf der CD verfügbar ist.

Vielleicht mag dem Leser dieser Anleitung die ganze Installationsprozedur etwas mühsam und kompliziert vorkommen. Es ist aber möglich, innert weniger Stunden ein komplett neues System aufzusetzen und BERNHIST in kurzer Zeit der Öffentlichkeit zugänglich zu machen.

<sup>13</sup> <http://winscp.vse.cz>

## Datenblatt BERNHIST

Rechnername:	hist41
URL:	<a href="http://hist41.unibe.ch">http://hist41.unibe.ch</a>
IP – Adresse:	130.92.61.41
Subnet:	255.255.255.0
Gateway:	130.92.61.1
DNS 1:	130.92.9.52
DNS 2:	130.92.9.51
Root – Passwort:	7wU45%A

Die Internetadresse <http://www.bernhist.ch> wird von Namecenter verwaltet. Auf deren Webseite besteht die Möglichkeit die Zielverzeichnisse der Hauptdomain und der Subdomains zu managen.

URL:	<a href="http://www.namecenter.ch">http://www.namecenter.ch</a>
Id:	dmt@bluewin.ch
Passwort:	Urs314159
<a href="http://www.bernhist.ch">www.bernhist.ch</a>	<a href="http://hist41.unibe.ch/bernhist/d/menu/index.php">http://hist41.unibe.ch/bernhist/d/menu/index.php</a>
<a href="http://statistik.bernhist.ch">statistik.bernhist.ch</a>	<a href="http://extremetracking.com/open?login=bernhist">http://extremetracking.com/open?login=bernhist</a>
<a href="http://admin.bernhist.ch">admin.bernhist.ch</a>	<a href="http://hist41.unibe.ch/bernhist/d/admin/index.php">http://hist41.unibe.ch/bernhist/d/admin/index.php</a>

Im weiteren kann eine eMail – Weiterleitung eingerichtet werden. Wenn keine speziellen Adressen definiert sind, werden alle eMails @bernhist.ch, egal was vor dem @ - Zeichen steht, an die festgelegte Weiterleitungsadresse (meist die des Administrators von BERNHIST) gesendet.

MySQL – Version:	3.23.54
MySQL – Benutzername:	bernhistadmin
MySQL – Passwort:	topsecret
URL von phpMyAdmin:	<a href="http://hist41.unibe.ch/phpMyAdmin/index.php">http://hist41.unibe.ch/phpMyAdmin/index.php</a>

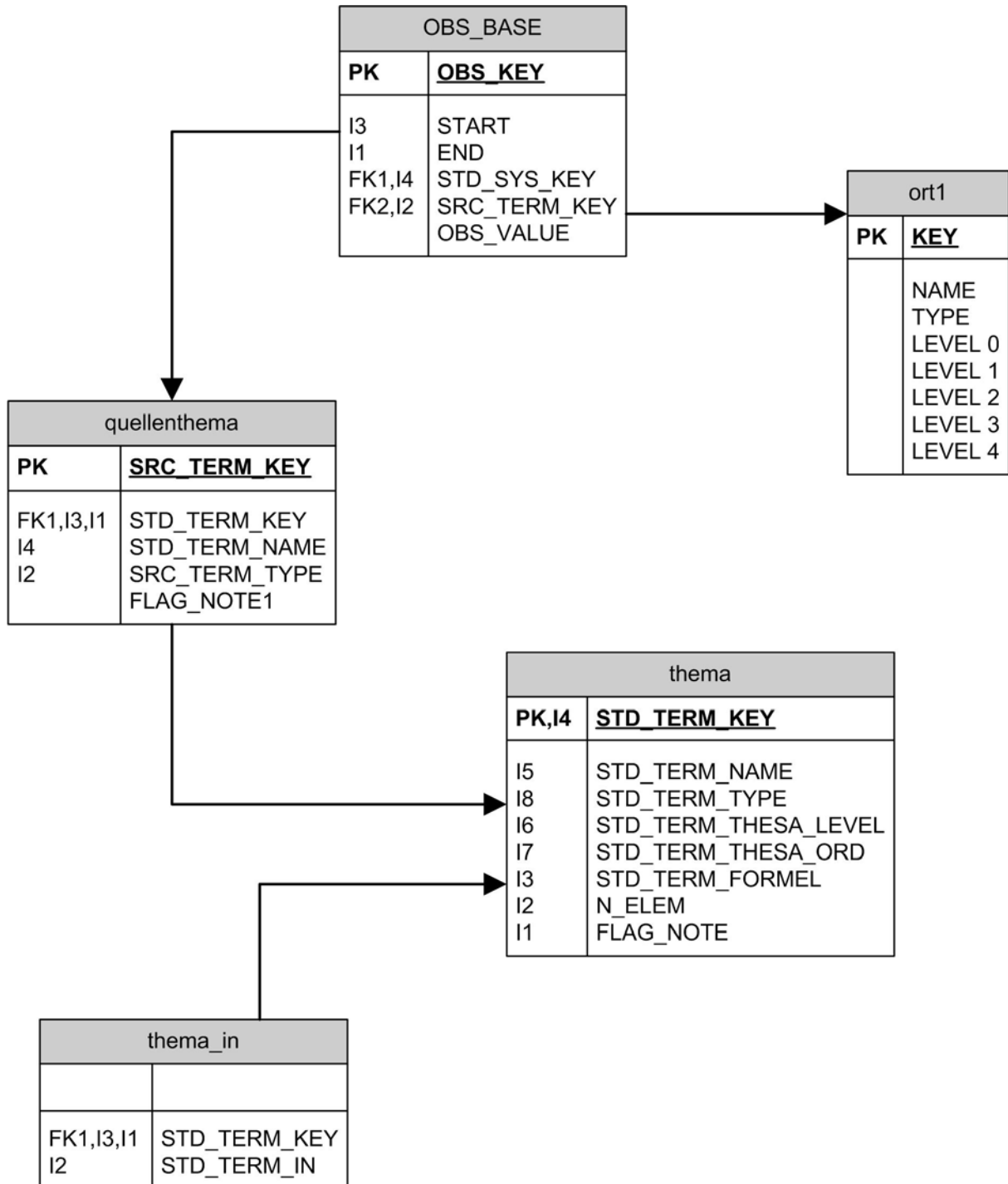
## Literaturverzeichnis

- Eilebrecht Lars, Rath Nikolaus, Rohde Thomas:** *Apache Webserver, Installation, Konfiguration, Administration.* mitp – Verlag 2002.
- Greenspan Jay, Bulger Brad:** *MySQL / PHP – Datenbankanwendungen.* mitp – Verlag 2001.
- Hässler Ulrike:** *Cascading Stylesheets, Stil mit <Stil>.* Springer – Verlag 2003.
- Kemper Alfons, Eickler André:** *Datenbanksysteme, Eine Einführung.* Oldenbourg Verlag 1997.
- Kofler Michael:** *MySQL, Einführung, Programmierung, Referenz.* Addison - Wesley 2001.
- Kuhlmann Gregor, Müllerstadt Friedrich:** *SQL, Der Schlüssel zu relationalen Datenbanken.* Rowohlt – Verlag 2000.
- Meier Andreas:** *Relationale Datenbanken, Eine Einführung für die Praxis.* Springer - Verlag 1998.
- Pfister Christian:** *Geschichte des Kantons Bern seit 1798, Band IV, Im Strom der Modernisierung, Bevölkerung, Wirtschaft und Umwelt, 1700-1914.* Historischer Verein des Kantons Bern, Bern 1995.
- Pfister, Christian; Egli, Hans-Rudolf [Hg.]:** *Historisch-Statistischer Atlas des Kantons Bern 1750-1995, Umwelt, Bevölkerung, Wirtschaft, Politik.* Historischer Verein des Kantons Bern, Bern 1998.
- Samar Richard, Stocker Christian:** *PHP de Luxe, Fortgeschrittene PHP – Programmierung.* mitp – Verlag 2002.
- Shafer Dan, Yank Kevin:** *Cascading Stylesheets, Anspruchsvolle Websites mit CSS gestalten.* Dpunkt.verlag 2004.
- Taggesell Dirk:** *PHP4, Dynamische Webseiten mit Apache, PHP und MySQL.* dtv 2001.
- Yank Kevin:** *PHP und MySQL, Schritt für Schritt zur datenbankgestützten Website.* Dpunkt.verlag 2002.



# Anhang

## A Datenbankschema



## B Codefragmente

Dieser Abschnitt soll dazu dienen, einzelne wichtige Ausschnitte des PHP – Codes zu kommentieren und genauer zu erläutern.

Überprüfung von Benutzernamen und Passwort:

```
session_start();    //Session wird gestartet

if ($_POST['username'] == 'bernhist' and //entsprechen die Variabeln den
    $_POST['password'] == 'topsecret') //verlangten Vorgaben
    $_SESSION['authorized'] = true;    //wenn ja, wird die Variable
                                        //$_authorized auf true gesetzt
session_register('authorized');        //und in der Session registriert

    if (!$SESSION['authorized']):        //prüft ob $_authorized nicht true
                                        //ist
include ("header.php");                //der Header kommt auf jede Seite

include ("error.php");                  //die Errormessage

else:                                    //falls $_authorized true ist
```

Hier kommen nun die supergeheimen Inhalte.

Variablen auslesen:

Zuerst werden alle Variablen, die sich noch im `$result` der SQL – Abfrage befinden mit dem Befehl `mysql_fetch_array()` in den Array `$row` gelesen.

Jetzt werden die einzelnen Werte der Tabellenfelder (KEY, NAME, TYPE) einer Variabel zugewiesen. So kann auf die gewünschten Werte zugegriffen werden. Der ganze Prozess verläuft in einer `while` – Schlaufe, die den Array bis ans Ende durchläuft. Es ist darauf zu achten, wo die `while` – Schlaufe beendet wird.

```
while($row = mysql_fetch_array($result)) {
    $key = $row["KEY"];
    $name = $row["NAME"];
    $type = $row["TYPE"];
    $value = $key."|".$name."|".$type;

    echo .....;                                //hier könnte eine Ausgabe erfolgen

} //Ende der while - Schlaufe
```

Die Variabel `$value` dient dem „Transport“ von mehreren Variablen von einer Webseite auf die Nächste. Beim Zusammenhängen wurde zur Trennung das `|` - Zeichen verwendet. Um diese wieder zu entpacken, wird der String an jeder Stelle mit einem `|` - Zeichen getrennt. Folgender Code splittet die Variable wieder in die ursprünglichen Werte:

```
$sortarray = explode("|", $sortvar);  
  
$sortkey = $sortarray[0];  
$sortname = $sortarray[1];  
$sorttype = $sortarray[2];
```

Zum Schluss noch die SQL – Abfrage der Resultatseite. In PHP ist es möglich, Variablen gleich im SQL – Konstrukt einzusetzen. Bei diesem Beispiel werden die Variablen `$sortkey`, `$themenkey`, `$startjahr` und `$endjahr` benötigt, um die gewünschten Werte aus der Tabelle `OBS_BASE` auszulesen.

```
mysql_select_db("bernhist"); //Wahl der Datenbank  
  
$result = @mysql_query("SELECT START ,  
END , ort1.NAME, OBS_VALUE, thema.STD_TERM_NAME  
FROM `obs_base` , ort1, quellenthema, thema  
WHERE obs_base.src_term_key = quellenthema.src_term_key  
AND quellenthema.std_term_key = thema.std_term_key  
AND obs_base.STD_SYS_KEY = $sortkey  
AND thema.std_term_key = $themakey  
AND START >= $startjahr  
AND END <= $endjahr  
AND obs_base.STD_SYS_KEY = ort1.key  
ORDER BY START ;");
```