Linux, Apache, MySQL und PHP

Herwig Reidlinger

ftp://pcnews.at/pcn/67/reidlinger/linux/

Die Kenntnisse, die für ein Verständnis der Beispiele dieses Artikels gebraucht werden, sind sehr vielfältig. Es wird die Konfiguration des Programms MySQL behandelt, dann werden SQL-Befehle verwendet, die einigen bei Datenbanken schon begegnet sein werden. Weiters wird die sehr funktionsreiche Skriptsprache PHP vorgestellt. (Statt PHP wird als Skriptsprache auch gerne Perl verwendet.)

Für jedes Teilgebiet müsste man ein ganzes Buch und nicht nur einen kurzen Artikel schreiben. Dennoch hoffe ich, dass die Beispiele für Leser(innen) mit Programmiererfahrung nachvollziehbar sind und Lust auf ein intensiveres Studium machen.

Die Werkzeuge für eine Nutzung des Internets durch selbstgeschriebene Programme sind vorhanden und bei Linux noch dazu kostenlos.

Linux mit dem Apache Webserver, MySQL und PHP können Werkzeuge zum Verlieben werden. (**LAMP** wird als Abkürzung für dieses Kombination verwendet.) Bei den meisten wird es vielleicht keine Liebe auf den ersten Blick sein.

Die Skripts und HTML-Dateien, die im Artikel erwähnt werden, müssen nicht abgetippt werden, sondern können über das WWW unter der Adresse *http://www.pinoe-hl.ac.at/material/mysql* heruntergeladen werden. Außerdem werden dort auch eventuelle Verbesserungen und Erweiterungen zu finden sein.

Der SQL-Server MySQL

Wenn Daten in einem lokalen oder globalen Netzwerk bearbeitet werden sollen, bietet sich ein SQL-Server als Lösung an. Auch das Gestalten von dynamischen WWW-Seiten, deren Inhalte in Datenbanken gespeichert sind, kann mit einem SQL-Server und Skripts in der Sprache Perl oder PHP realisiert werden.

Als Beispiel soll hier das Programm MySQL vorgestellt werden, das es sowohl für Unix/Linux als auch für Windows NT und Windows 95/98 gibt. Auch die Skriptsprachen Perl und PHP sind sowohl für Unix/Linux als auch für Windows-Plattformen verfügbar.

Das Programm MySQL kann direkt aus dem Internet von der Adresse http://www.mysql.com heruntergeladen werden. Es ist auch in den meisten Linux-Distributionen enthalten. Die hier beschriebene Version trägt die Nummer 3.22.21, die z. B. bei der SuSE-Distribution 6.1 zu finden ist. Da gebenüber den früheren Versionen von MySQL einige Erweiterungen eingebaut wurden, sollte unbedingt ein Update erfolgen, bevor die hier beschriebenen Beispiele getestet werden.

MySQL installieren und konfigurieren

Beim Standardinstallationsprogramm von SuSE wird MySQL nicht mitinstalliert. Dies muss man als Benutzer **root** mit dem Programm **yast** nachholen. Man findet das Programm in der Serie **ap** (bei Version 6.2 in der Serie **pay**). Der dadurch installierte SQL-Server läuft als Damon im Hintergrund. Damit der Start automatisch erfolgt, wird im Programm **yast** der Punkt

Administration des Systems | Konfigurationsdatei verändern

ausgewählt und der Wert

```
START_MYSQL=true
```

gesetzt. Beim nächsten Linux-Start müsste das Programm automatisch starten und die nötigen Initialisierungen vornehmen.

Bei der SuSE-Installation befindet sich im Verzeichnis **/usr/doc/packages/mysql** die Dokumentation zu MySQL. (Das Unterverzeichnis **html** enthält eine Dokumentation im HTML-Format, die mehr als 1 MB groß ist.) Hier können nur die wichtigsten Eigenschaften erwähnt werden.

Die Datenbanken von MySQL werden im Verzeichnis **/var/mysql** abgespeichert. Für jede Datenbank wird dort ein eigenes Unterverzeichnis angelegt. In diesen Unterverzeichnissen werden für jede Tabelle drei Dateien mit den Ergänzungen **.isd**, **.ism** und **.frm** angelegt.

Die wichtigste Datenbank ist **mysql** (im Unterverzeichnis **mysql**) mit den Tabellen **db**, **user**, **host**, **tables_priv**, **columns_priv** und **func**. In diesen Tabellen werden die Zugriffsrechte und verschlüsselten Passwörter abgespeichert.

MySQL hat sehr viele verschiedene Privilegien:

```
Select_priv, Insert_priv, Update_priv, Delete_priv,
Index_priv, Alter_priv, Create_priv, Drop_priv, Grant_priv,
Reload_priv, Shutdown_priv, Process_priv, File_priv.
```

In der Hilfedatei von MySQL ist das Privilegiensystem genau erklärt. Es kann festgelegt werden, welcher Benutzer eine Datenbank nur lesen oder auch verändern kann. Außerdem kann angegeben werden, ob dieser Benutzer diese Rechte nur vom lokalen Computer aus oder auch über das Internet hat. Rechte können dem Benutzer auch nur für eine Tabelle oder eine Spalte einer Tabelle gewährt werden.

Die Verwaltung kann mit den Programmen **mysqladmin** und **mysqlshow** erfolgen.

Durch den Befehl

```
mysql
```

wird der MySQL-Monitor gestartet. Händisch können SQL-Befehle eingegeben werden. Wenn MySQL als SQL-Server in einem Intranet oder Internet verwendet wird, sollte die erste Aktion das Löschen der Datenbank **test** mit dem Befehl

```
mysql > drop database test;
```

sein, da für diese Datenbank alle Benutzer volle Schreibrechte besitzen. Um die Eingaben im MySQL-Monitor von den Linux-Befehlen zu unterscheiden, wird jedem Befehl

```
mysql >
```

vorangestellt. Diese Eingabeaufforderung ist natürlich nicht einzugeben! Mit dem Befehl

```
mysql > quit
```

verlässt man den MySQL-Monitor.

In der Grundkonfiguration hat man als Benutzer **root** alle Rechte für alle Datenbanken des Programms MySQL. Allerdings kennt MySQL andere Benutzer als das Linux-System, auf dem das Programm installiert wurde. Standardmäßig ist bei MySQL ein Benutzer **root** bereits definiert. Wenn man in Linux als Benutzer **root** eingeloggt ist und mit **mysql** den MySQL-Monitor startet, dann ist man auch in diesem Datenbankprogramm der Benutzer **root**. Man kann mit

```
mysql -u user_name
```

seine Identität wechseln und sich in MySQL als Benutzer **user_name** einloggen.

Zu Beginn ist für den Benutzer **root** in MySQL kein Passwort definiert! Mit dem Befehl

```
mysql -u root
```

könnte sich jeder Benutzer des Linuxrechners alle Rechte für das Programm MySQL verschaffen. Dies könnte auf einem Linuxrechner, der an das Internet angeschlossen ist, über ein Perl- oder PHP-Skript durch alle Benutzer erfolgen, die z. B. über das WWW derartige Skripts ausführen dürfen. Daher sollte dem Benutzer **root** unbedingt ein Passwort zugewiesen werden. Durch den Befehl

```
mysql -u root mysql
```

startet man den MySQL-Monitor als Benutzer **root** und öffnet die Datenbank **mysql**. Jetzt gibt man mit

```
mysql > update user set
Password=password('neues_passwort ')
where user='root';
mysql > flush privileges;
mysql > quit
```

dem Benutzer **root** ein Passwort.

Um sich die ständige Eingabe des Passwortes als Benutzer **root** zu ersparen, kann man im Verzeichnis **/root** eine Datei mit dem Namen **.my.cnf** anlegen, die aus folgenden zwei Zeilen besteht:

```
[client]
password=neues_passwort
```

Zu beachten ist dabei, dass Perl- oder PHP-Skripts, die über das WWW aufgerufen werden, sich so verhalten, als ob sie ein lokaler Benutzer auf diesem Rechner ausgeführt hätte.

Anlegen einer Schülerdatenbank

Es soll nun eine Datenbank mit Schülernamen angelegt werden. Zusätzlich sollen Klasse, Login-Name, Adresse der Homepage und die E-Mail Adresse gespeichert werden. Wenn eine Schule einen eigenen Web-Server unter Linux betreibt, kann über das WWW der Inhalt dieser Datenbank in Form von Klassenlisten ausgegeben werden. Außerdem soll auch eine alphabetische Ausgabe der Schüler möglich sein. (Einige Felder sind für Erweiterungen wie Sprechstunden der Lehrer, automatisches Anlegen von Benutzern unter Linux usw. vorgesehen.)

Wichtige SQL-Befehle sind:

```
ALTER TABLE _name alter_spec [, alterspec
...]
CREATE DATABASE db name
CREATE TABLE tbl name (col name type, ...)
DROP DATABASE db name
DROP TABLE tbl name
DELETE FROM  tbl_name [WHERE
where_definition]
```

```
<html>
<head>
<title>Daten importieren</title>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
</head>

<body bgcolor="#ffffff">
<?php

$database = "SchulDB";
$table    = "SchulTab";
$admin    = "SchulAdmin";
$txtdatei = "SchulDB.txt";

// Domainnamen für E-Mail Adresse abändern
$domain   = "@schule.ac.at";

function allesklein ($s) {
  $s = strtolower ($s);
  $s = str_replace ("Ä","ae",$s);
  $s = str replace ("ä","ae",$s);
  $s = str_replace ("Ö","oe",$s);
  $s = str replace ("ö","oe",$s);
  $s = str_replace ("Ü","ue",$s);
  $s = str replace ("ü","ue",$s);
  $s = str_replace ("ß","ss",$s);
  return $s;
}

// ****************** zurueck *************************
function zurueck() {
global  $PHP SELF;
?>
<table border=0 width="100%" cellspacing=0 cellpadding=2>
<tr align="center" bgcolor="#80e0e0">

<?php
print "<td><a href=\"$PHP SELF\"> Zurück (Textdatei importieren)</a></td>\n";
?>
<td><a href="schueler.htm">Schüler unserer Schule</a></td>
<td><a href="SchulDB.php3">Daten ändern</a></td></tr>
</table>
<?php
} // *** zurueck ***

function datenimportieren ($Dateiname) {
global $PHP SELF, $txtdatei, $database, $table, $admin, $domain, $Passwort, $Automatik;

 print "<h3 align=\"center\">Import der Daten in die Tabelle $table</h3>\n";
 if (file exists ($Dateiname)) {
     $fp = fopen ($Dateiname, "r"); // Textdatei öffnen
     $db = @mysql_connect ("localhost",$admin,$Passwort); // Verbindung zur Datenbank
herstellen
     if (!db) {
         print "<p>Keine Verbindung zur Datenbank oder Passwortfehler!</p>\n";
         zurueck ();
         return 0; }
     if (!@mysql select db($database,$db)) {
         print "<p>Keine Verbindung zur Datenbank $database möglich! Ist das Passwort
richtig?</p>\n";
         zurueck ();
         return 0; }
     $nr=0;
     while (!feof($fp)) {
        $zeile = fgets($fp,1024); // Zeile einlesen
        $zeile = chop ($zeile);
        list ($zuname,$vorname,$klasse,$login,$homepage,$email, $kommentar,$funktion,$titel,
              $aktion,$rest) = explode (";",$zeile); // Felder aufspalten
        if ($zuname) { // Leerzeilen unterdrücken
           $nr++;
           if ($Automatik) {
              if (!$email) {$email = "$vorname.$zuname";
                            $email = allesklein ($email);
                            $email .= $domain; }
              if (!$homepage) {$homepage = "/~$vorname.$zuname";
                               $homepage = allesklein ($homepage); }
              if (!$login) {$hilfe = allesklein ($vorname);
                            $login = substr ($hilfe,0,1);
                            $hilfe = allesklein ($zuname);
                            $login .= substr ($hilfe,0,7); }
           } // if Automatik
           if (!$aktion) {$aktion = 1;}
           print "$nr: $zuname $vorname $klasse $login $homepage $email $aktion<br>\n";
           $befehl = "INSERT INTO $table (Vorname,Zuname,Klasse,Login,"
                ."Homepage,EMail,Kommentar,Funktion,Titel,Aktion) VALUES "
                ."('$vorname','$zuname','$klasse','$login',"
                ."'$homepage','$email','$kommentar','$funktion',"
                ."'$titel',$aktion)";
```

SchulDB_import.php3

```
FLUSH PRIVILEGES
INSERT [INTO] tbl_name [(col_name, ...)]
VALUES (expression, ...)
INSERT [INTO] tbl_name SET col_name =
expression, ...
REPLACE [INTO] tbl_name [(col_name, ...)]
VALUES (expression, ...)
REPLACE [INTO] tbl_name SET col_name =
expression, ...
SELECT select_expr, ... [FROM tbl_name
[WHERE where definition]]
SHOW COLUMS FROM tbl name
UPDATE tbl_name SET col_name =
expression, ...  [WHERE where_definition]
```

Da das Eintippen von SQL-Befehlen recht mühsam ist, empfiehlt sich folgende Vorgangsweise. Man speichert die folgenden SQL-Befehle in einer Datei mit dem Namen **SchulDB.create** ab:

```
CREATE DATABASE SchulDB;
CONNECT SchulDB;
CREATE TABLE SchulTab (Zuname VARCHAR (30),

  Vorname VARCHAR(25),
  Klasse VARCHAR (5), Login VARCHAR (15),
  Homepage VARCHAR (70), EMail VARCHAR (70),
  Kommentar VARCHAR(130), Funktion VARCHAR
(15),
  Titel VARCHAR(15), Aktion INTEGER,
  ID INTEGER NOT NULL AUTO INCREMENT,
  PRIMARY KEY(ID));
CONNECT mysql;
INSERT INTO user (Host,User,Password)
   VALUES
('localhost','SchulAdmin',PASSWORD('neues_pa
sswort'));
INSERT INTO user (Host,User) VALUES
('localhost','SchulGast',);
INSERT INTO db
(Host,Db,User,Select_priv,Insert_priv,Update
_priv,Delete_priv)
   VALUES
('localhost','SchulDB','SchulAdmin','Y','Y',
'Y','Y');
INSERT INTO db (Host,Db,User,Select_priv)
   VALUES
('localhost','SchulDB','SchulGast','Y');
FLUSH PRIVILEGES;
```

(Statt **neues_passwort** geben Sie bitte jenes Passwort ein, das der Administrator dieser Schülerdatenbank erhalten soll.)

Als Benutzer **root** wird danach

```
mysql < SchulDB.create
```

eingegeben. Wenn keine Fehlermeldung am Bildschirm erscheint, wurden Datenbank und Tabelle erfolgreich angelegt.

Was geschieht durch diese SQL-Befehle? Es wird eine Datenbank mit dem Namen **SchulDB** erzeugt, die eine Tabelle mit dem Namen **SchulTab** enthält. Für diese Datenbank werden zwei Benutzer definiert. Der Benutzer mit dem Namen **SchulGast** darf die Datenbank mit allen ihren Tabellen vom lokalen Computer (**localhost**) nur lesen. Dafür benötigt er kein Passwort. Der Benutzer mit dem Namen **SchulAdmin** darf in der Tabelle **SchulTab** der Datenbank **SchulDB** die Datensätze auch ändern, einfügen und löschen. Er kann aber z. B. die Tabelle nicht aus der Datenbank entfernen. Dies könnte nur der Benutzer **root** machen.

Für die Verwaltung von MySQL gibt es PHP3-Skripts mit dem Namen phpMyAdmin, mit denen diese Datenbank über das WWW gewartet werden kann. Die neueste Version kann im Internet bei der Adresse *http://www.htmlwizard.net/phpMyAdmin/* heruntergeladen werden. Auch das Importieren von Daten aus einer Textdatei, wie dies im folgenden Kapitel beschrieben wird, ist mit diesem Programm möglich.

```
            $result = mysql query ($befehl,$db);
            } // if zuname
          } // while
      fclose ($fp);
      print "<p>$nr Datensätze wurden in die Tabelle $table importiert!</p>\n";
      mysql_close();
      zurueck ();
      } // if file_exists
      else {print "<p>Kann Datei $Dateiname nicht öffnen!</p>\n";
            zurueck ();
            return 0;}
}

// ****************** startseite *************************
function startseite () {
global $PHP SELF, $txtdatei, $database, $table;
?>

<table WIDTH="100%" BORDER="0" bgcolor="#80e0e0">
  <tr>
    <th><font SIZE="+3">Schülerdaten aus Textdatei importieren</font></th>
  </tr>
</table>

<p>Datensätze in der Textdatei müssen durch ; getrennt sein.
<?php
print "Sie werden in die Datenbank <b>$database</b>, Tabelle <b>$table</b> importiert.</p>\n";
print "<form method=\"POST\" action=\"$PHP_SELF\">\n";
print "<table border=0>\n<tr><td>Name der Textdatei</td>";
print "<td><input type=\"text\" name=\"Dateiname\" value=\"$txtdatei\" SIZE=20></td></tr>\n";
print "<td>Automatik</td><td><input type=\"checkbox\" name=\"Automatik\" value=\"Ja\"
CHECKED>";
print " Homepage, E-Mail Adresse und Login-Name automatisch berechnen</td></tr>\n";
print "<tr><td>Passwort</td><td><input type=\"password\" name=\"Passwort\" value=\"\"
SIZE=10></td></tr>\n";
print "<tr><td></td><td><input type=\"submit\" name=\"importieren\" value=\" Daten importieren
\"></td></tr>\n";
print "</table>\n</form>\n";
} // Ende der Funktion startseite ()

// ****************** Hauptprogramm *************************
if ($importieren) {datenimportieren ($Dateiname);}
  else            {startseite ();}
?>
</body>
</html>
```

SchulDB_import.php3

Einfügen der Daten in die Schülerdatenbank

Dieser Vorgang lässt sich weitgehend automatisieren. Wenn die Schülerdaten bereits in einer Datenbank gespeichert sind, dann exportiert man Zuname, Vorname und Klasse als Textdatei im ANSI-Format mit dem Namen **SchulDB.txt**, wobei die einzelnen Datensätze durch das Zeichen ; getrennt sind (Daten nicht in " o. ä. einschließen!):

```
Meier;Franz;1A;
Schuster;Maria;1B;
```

Haben nur einige Schüler einen Login-Namen, eine Homepage bzw. eine persönliche E-Mail Adresse können diese Daten in die Datei **SchulDB.txt** eingefügt werden. Die Reihenfolge Zuname, Vorname, Klasse, Login-Name, Homepage, E-Mail Adresse muss eingehalten werden. Zuname, Vorname und Klasse müssen eingetragen sein. Die anderen Felder können leer bleiben.

```
Meier;Franz;1A;fmeier;www.meier.at;f.meier@provider.at;
Schuster;Maria;1B;;www.maria.schuster.co.at;;
```

Danach startet man den MySQL-Monitor mit

```
mysql SchulDB
```

und gibt folgenden Befehl ein:

```
mysql > LOAD DATA LOCAL INFILE './SchulDB.txt' INTO TABLE
SchulTab FIELDS TERMINATED BY ';' LINES TERMINATED BY '\n';
```

Hat ein Großteil der Schüler bereits einen eigenen Login-Name, eine eigene Homepage oder E-Mail Adresse dann kann für den Import der Daten ein PHP-Skript verwendet werden, das diese Daten automatisch aus Vor- und Zuname erzeugt.

PHP3

PHP wurde im Herbst 1994 von Rasmus Lerdorf entwickelt. Zu Beginn des Jahres 1995 wurde sie unter dem Namen „**P**ersonal **H**ome **P**age Tools“ veröffentlicht. Durch Kombination mit dem „Form Interpreter“ und Unterstützung von mSQL entstand PHP/FI. 1997 wurde der Parser von Zeev Suraski und Andi Gutmans neu programmiert. Diese neue Skriptsprache wird meist als PHP3 bezeichnet.

Im Verzeichnis */usr/doc/packages/apache/php3/html* findet man bei der SuSE-Distribution eine Dokumentation im HTML-Format. Bei dieser Sprachbeschreibung sind etliche Details nicht vollständig dokumentiert.

Damit die folgenden Beispiele getestet werden können, benötigt man einen Linux-Rechner, auf dem der Apache Webserver mit PHP-Unterstützung läuft. (Bei der SuSE-Distribution ist dies standardmäßig der Fall.) Auch auf dem eigene Linux-Rechner können die PHP-Skritps mit einem Browser getestet werden. Ein Vorteil von Linux, der das Testen von PHP-Skripts sehr erleichtert.

Bei der Standardinstallation des Apache Webservers unter SuSE werden Seiten, die PHP-Befehle enthalten durch die Dateiergänzung **.php3** oder **.phtml** gekennzeichnet. Diese Seiten werden vor ihrer Übertragung an den Client vom Webserver auf PHP-Befehle untersucht. Das Ergebnis wird an den Client geschickt, der die PHP-Befehle nicht zu Gesicht bekommt.

Folgende Möglichkeiten gibt es, PHP-Code in HTML-Seiten einzufügen:

```
<? print "Das ist die einfachste Variante\n";
?>
<?php print "Ähnlich wie bei
XML-Dokumenten\n"; ?>
<script language="php">
        print "Diese Variante wollen manche
HTML-Editoren\n";
</script>
<% print "Ähnlich wie bei ASP-Befehlen\n"; %>
```

Für die Erstellung von PHP-Skripts reicht ein einfacher Editor aus. Ein HTML-Editor ist nur bedingt geeignet, da die Hauptarbeit im Einfügen von PHP-Code besteht.

Mit dem oben abgedruckten Skript **SchulDB_import.php3** können Daten importiert werden.

Das Lesen derartiger Seiten ist am Anfang etwas gewöhnungsbedürftig. Die Datei besteht aus einer Mischung von HTML-Code und PHP-Code. Das Grundgerüst ist der HTML-Code

```
<html>
<head>
<title>Daten importieren</title>
<meta http-equiv="Content-Type"
content="text/html; charset=ISO-8859-1">
</head>
```

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<title>Schüler der Schule</title>
</head>

<body bgcolor="#FFFFFF">
<table border="1" cellpadding="5" width="100%">
  <tr><td align="center" colspan="3"><$[Fixed im Text>Klassen</b></td>
  </tr>
  <tr><td width="33%"><a href="zeigeschueler.php3?klasse=1A">1A Klasse</a></td>
      <td width="33%"><a href="zeigeschueler.php3?klasse=1B">1B Klasse</a></td>
      <td><a href="zeigeschueler.php3?klasse=8B">8B Klasse</a></td>
  </tr>
     <!– Hier weitere Klassen einfügen –>
</table>

<table border="1" cellpadding="5" width="100%">
    <tr><td align="center" colspan="9"><$[Fixed im Text>Alphabetische Schülerliste</b></td>
    </tr>
    <tr><td align="center"><a href="zeigeschueler.php3?anfang=A">A</a></td>
        <td align="center"><a href="zeigeschueler.php3?anfang=B">B</a></td>
       <!– usw. für die restlichen Buchstaben –>
        <td align="center"><a href="zeigeschueler.php3?anfang=Y">Y</a></td>
        <td align="center"><a href="zeigeschueler.php3?anfang=Z">Z</a></td>
        <td align="center"> </td>
    </tr>
</table>
</body>
</html>
```

schueler.htm

```
<html>
<head>
<title>Schule in Hollabrunn</title>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
</head>
<body bgcolor="#ffffff">

<?php
// Diese Werte bitte anpassen (Klassenvorstände):
$kv       = array ("1A" => "Bauer Walter",
                   "1B" => "Müller Theresia",
                   "8B" => "Schneider Andrea");
// Domainname für E-Mail Adresse
$domain = "@schule.ac.at";
$database = "SchulDB";
$table    = "SchulTab";
$user     = "SchulGast";
// ****************** zurueck *************************
function zurueck () {
?>
<hr size="1">
<table border="0" width="100%" cellspacing="0" cellpadding="5">
 <tr align="center" bgcolor="#ffff00">
  <td><a href="schueler.htm"> Schüler unserer Schule</a></td>
 </tr></table>
<?php
}

// ****************** erg email ***************************
function erg_email ($email) {
 global $domain;
  $pos = strpos ($email,"@");
  if ($pos > 1) { return $email; }
     else { return "$email$domain";}
}

// ****************** fdbverbindung ***********************
function fdbverbindung ($befehl) {
global $database, $user;
$db = mysql_connect("localhost",$user);
mysql select db($database,$db);
$result = mysql_query($befehl,$db);
return $result;
}

// ****************** klassesuchen ************************
function klassesuchen () {
 global $PHP SELF, $table, $klasse, $kv;
?>
 <table border="0" width="100%" cellpadding="5"><tr>
 <td align="center" bgcolor="#0000ff"><font size="+4" color="#ffffff">
<?php
 print "$klasse Klasse</font></td></tr>\n</table>\n";
?>
<table border="0" cellpadding="5" cellspacing="0" width="100%">
  <tr bgcolor="#ffff00"><th width="50%"><$[Fixed im Text>Klassenvorstand</th>
```

zeigeschueler.php3

http://www.pinoe-hl.ac.at/material/mysql/

```
<body bgcolor="#ffffff">
<?php
// Hier steht PHP-Code
?>
</body>
</html>
```

Der PHP-Code besteht in der Definition mehrer Funktionen mit dem Namen `zurueck()`, `allesklein($s)`, `datenimportieren($Dateiname)` und `startseite()`, die immer in der Datei zuerst definiert werden müssen, bevor sie aufgerufen werden können. Beim ersten Aufruf wird die Funktion `startseite()` ausgeführt, die den Benutzer auffordert, Dateinamen und Passwort einzugegen. Wird dieses Formular abgesendet, dann wird die Seite **`SchulDB_import.php3`** nochmals aufgerufen. (Den Name des Skripts erhält man über die globale Variable `$PHP_SELF`.) Diesmal hat die Variable `$importieren` einen Wert, der an dieses Skript übergeben wird. Im Hauptprogramm wird mit

```
if ($importieren) {datenimportieren
($Dateiname);}
```

zur Funktion `datenimportieren($Dateiname)` verzweigt. Es wird versucht, die Datei **`SchulDB.txt`** zu öffnen. Jede eingelesene Zeile wird in die Datenfelder zerlegt. Wenn die Daten für E-Mail Adresse, Homepage und Login-Namen nicht vorhanden sind, können sie automatisch berechnet werden. Es wird dann für jeden Benutzer eine E-Mail Adresse der Form

`Vorname.Zuname@Domainname`

erzeugt, wenn in der Datei **`SchulDB.txt`** nicht bereits eine E-Mail Adresse angegeben wurde. Ähnlich geschieht dies in den folgenden Zeilen für die Homepage bzw. für den Login-Namen.

Schließlich werden die Daten in der MySQL-Datenbank abgespeichert. Die wesentlichen PHP-Befehle sind:

```
$db = mysql_connect
("localhost",$admin,$Passwort);
mysql select db($database,$db);
$result = mysql query ($befehl,$db);
mysql_close();
```

Die Funktion `allesklein()` wandelt in einer Zeichenkette alle Großbuchstaben in Kleinbuchstaben und die Umlaute ä in ae usw. um.

Zum Testen werden die Dateien **`SchulDB.txt`** und **`SchulDB_import.php3`** in ein beliebiges Verzeichnis kopiert, in dem sich die WWW-Seiten für den Apache-Server befinden. (Bei SuSE ist es das Verzeichnis **`/usr/local/httpd/htdocs`**.) Diese Übertragung kann natürlich auch mit einem FTP-Programm erfolgen. Dann kann das PHP-Skript getestet werden, indem man es mit einem Web-Browser aufruft. Die importierten Daten werden als HTML-Seite ausgegeben.

```
<?php
 print "<th>$kv[$klasse]</th>\n</tr>\n</table>\n";

 $befehl = "SELECT Vorname,Zuname,Homepage,EMail from $table "
          . "WHERE Klasse='$klasse' ORDER BY Zuname, Vorname";
 $result = fdbverbindung ($befehl);

 print "<table border=\"1\" width=\"100%\" cellpadding=\"5\">\n";
 print "<tr><th> </th><th>Name</th><th>E-Mail</th></tr>\n";
 $nr=0;
 while (list ($vorname,$zuname,$homepage,$email) = mysql fetch row($result)) {
        $nr++;
        print "<tr><td>$nr</td><td>";
        if ($homepage) {print "<a href=\"$homepage\">$zuname $vorname</a>";}
           else {print "$zuname $vorname";}
        print "</td><td>";
        if ($email) {$email = erg email ($email);
                     print "<a href=\"mailto:$email\">$email</a>";}
           else {print " ";}
        print "</td></tr>\n";
        } // while
 print "</table>\n";
 mysql close();

 print "<hr size=\"1\">";
 print "<table border=\"0\" width=\"100%\">\n<tr align=\"center\">";
 print "<td>Weiter zur Klasse</td>";
 while (list ($kl, $val) = each ($kv)) {
       print "<td><a href=\"$PHP SELF?klasse=$kl\">$kl</a></td>\n";}
 print "</tr></table>\n";

 zurueck ();
} // Ende der Funktion klassesuchen ()

// ****************** namesuchen *************************
function namesuchen () {
 global $PHP_SELF, $table, $anfang;
?>
 <table border="0" width="100%" cellpadding="5"><tr>
  <td align="center" bgcolor="#0000ff"><font size="+4" color="#ffffff">
<?php
 print "Schüler(innen) mit dem Anfangsbuchstaben $anfang</font></td></tr>\n</table>\n";

 $befehl = "SELECT Vorname,Zuname,Klasse,Homepage,EMail from $table "
           . "WHERE (Zuname LIKE '$anfang%') AND (Klasse  'L') "
           ."ORDER BY Zuname, Vorname";
 $result = fdbverbindung ($befehl);

 print "<table border=\"1\" width=\"100%\" cellpadding=\"5\">\n";
 $nr = 0;
 while (list ($vorname,$zuname,$klasse,$homepage,$email) = mysql_fetch_row($result)) {
       $nr++;
       if ($nr==1) {
       print "<tr><th> </th><th>Name</th><th>Klasse</th><th>E-Mail</th></tr>\n";}
       print "<tr><td>$nr</td><td>";
       if ($homepage) {print "<a href=\"$homepage\">$zuname $vorname</a>";}
           else {print "$zuname $vorname";}
       print "</td><td>$klasse</td><td>";
       if ($email) {$email = erg_email ($email);
                    print "<a href=\"mailto:$email\">$email</a>";}
          else {print " ";}
       print "</td></tr>\n";
       } // while
 print "</table>\n";
 if (!$nr) {print "<p><$[Fixed im Text>Keine Schüler(innen) gefunden!</b></p>";}
 mysql close();

 print "<p>Neue Suche mit Anfangsbuchstaben: ";
 for ($i = Ord("A"); $i <= Ord("Z"); $i++) {
     $c = Chr($i);
     print "<a href=\"$PHP_SELF?anfang=$c\">$c</a>\n";
     }
 print "</p>\n";
 zurueck ();
} // Ende der Funktion namesuchen ()

// ****************** Hauptprogramm *************************
if ($klasse)          {klassesuchen ();}
if ($anfang)          {namesuchen ();}
?>
</body>
</html>
```

zeigeschueler.php3

Eine erste Kontrolle, ob die Daten importiert werden, kann dann auf dem Linux-Rechner mit dem Befehl

mysql SchulDB

erfolgen. Dort erhält man mit

mysql> SELECT * FROM SchulTab;

eine unformatierte Liste aller Datensätze. Im MySQL-Monitor können durch Eingabe von

mysql> DELETE FROM SchulTab;

alle Datensätze der Tabelle `SchulTab` wieder gelöscht werden.

Daten als HTML-Seite anzeigen

Der erste Schritt besteht in der Erstellung einer HTML-Seite mit dem Namen **schueler.htm**, die mit einem einfachen Text- oder HTML-Editor geschrieben werden kann. (siehe oben)

In dieser HTML-Seite sind Links eingefügt, die ein PHP-Skript mit dem Namen **zeigeschueler.php3** aufruft. An dieses Skript werden nach dem Zeichen ? Parameter übergeben:

```
<a href="zeigeschueler.php3?klasse=1A">1A Klasse</a>
<a href="zeigeschueler.php3?anfang=A">A</a>
```

die von dem Skript ausgewertet werden sollen. Diese Skriptdatei mit dem Namen **zeigeschueler.php3** ist oben abgedruckt.

Zu Begin dieses Skripts sind noch einige Werte von Variablen an die eigene Schulsitutation anzupassen. Es sind dies:

```
// Klassenvorstände:
$kv       = array ("1A" => "Bauer Walter",
                   "1B" => "Müller Theresia",
                   "8B" => "Schneider Andrea");
// Domainname für E-Mail Adresse
$domain = "@schule.ac.at";
```

Der PHP-Code besteht in der Definition mehrer Funktionen mit dem Namen `zurueck ()`, `erg_email ($email)`, `fdbverbindung ($befehl)`, `klassesuchen ()` und `namesuchen()`. Von der Seite **schueler.htm** wird die Variable `$klasse` oder `$name` an dieses Skript übergeben. Im Hauptprogramm wird mit

```
if ($klasse)          {klassesuchen ();}
if ($anfang)          {namesuchen ();}
```

abgefragt, ob diese Variablen einen Wert besitzen. Wenn in **schueler.htm** auf eine Klasse geklickt wurde, dann hat `$klasse` einen Wert, wenn auf einen Anfangsbuchstaben geklickt wurde, besitzt `$anfang` einen Wert. (Eine Variable, die keinen Wert, einen Leerstring oder den Wert 0 hat, wird in Perl und PHP als logisch falsch interpretiert.) Es wird daher entweder die Funktion `klassesuchen ()` oder `namesuchen()` aufgerufen. Diese Funktionen richten eine Abfrage an MySQL und geben das Ergebnis als HTML-Code zurück.

Die Grundstruktur der Datenbankabfrage besteht aus den folgenden Befehlen:

```
$db = mysql_connect("localhost",$user);
mysql select db($database,$db);
$result = mysql_query($befehl,$db);
while (list ($vorname,$zuname,$klasse,$homepage,$email) =
mysql_fetch_row($result)) {
       // hier wird das Ergebnis ausgegeben
       }
mysql_close();
```

Die Abfrage nach der Klasse geschieht mit dem SQL-Befehl

```
$befehl = "SELECT Vorname,Zuname,Homepage,EMail from $table "
          . "WHERE Klasse='$klasse' ORDER BY Zuname, Vorname";
```

die Abfrage von Schülern mit einem bestimmten Anfangsbuchstaben durch

```
$befehl = "SELECT Vorname,Zuname,Klasse,Homepage,EMail from $table "
          . "WHERE (Zuname LIKE '$anfang%') AND (Klasse  'L') "
          ."ORDER BY Zuname, Vorname";
```

Für die Lehrer ist als Klasse L vorgesehen. Daher wird diese Klasse bei der Ausgabe ausgeschlossen.

Zusätzliche Abfragen lassen sich leicht durch Abändern dieser SQL-Befehle einbauen. Vorteil dieser Datenbanklösung ist eine stets aktuelle Klassen- und Schülerliste, wenn der Inhalt der Tabelle **SchulTab** am letzten Stand ist.

Wie bereits oben bei der Beschreibung von MySQL erwähnt wurde, greift dieses PHP-Skript auf MySQL als lokaler Benutzer auch dann zu, wenn dieses Skript über das WWW von einem beliebigen Rechner im Internet aufgerufen wurde. Daher muss der Benutzer **SchulGast** auch nur Leserechte als lokaler Benutzer haben.

Zum Testen werden die Dateien **schueler.htm** und **zeigeschueler.php3** in ein beliebiges Verzeichnis kopiert, in dem sich die WWW-Seiten für den Apache-Server befinden.

Im WWW ist dieses Skript unter der Adresse *http://www.ebgymhollabrunn.ac.at/schueler/* bereits im Einsatz.

Wartung der Tabelle SchulTab über das WWW

Der Benutzer **SchulAdmin** hat, wenn er das Passwort weiß, auch das Recht, Daten in der Tabelle **SchulTab** zu verändern. Daher kann auch für die Wartung ein PHP-Skript geschrieben werden. Da mit diesem Skript die Daten sowohl ein Ändern, Einfügen als auch Löschen möglich sein soll, ist es etwas umfangreicher. Dateiname ist **SchulDB.php3**. *(Das Listing dazu finden Sie in der Webversion dieses Beitrags unter der Adresse http://www.pinoe-hl.ac.at/material/mysql/)*

Dieses PHP-Skript benötigt keine HTML-Seite zum Aufruf. Beim erstmaligen Start ist keine der Variablen `$einfuegen`, `$aendern` usw. gesetzt. Es wird daher die Funktion `startseite()` ausgeführt. Diese Funktion gibt eine HTML-Seite mit mehreren Formularen aus. Durch Absenden dieser Formulare ruft sich das Skript selbst auf. (Den Name des Skripts erhält man über die globale Variable `$PHP_SELF`.)

Die Schaltflächen vom Typ **submit** haben verschiedene Namen (**suchen** bzw. **einfuegen**). Klickt man die Schaltflächen an, wird der Wert mit der Variablen **$suchen** bzw. **$einfuegen** an das PHP-Skript übergeben. Der Wert interessiert uns nicht sondern nur die Tatsache, dass in diesen Variablen eine Zeichenkette gespeichert ist. Bei Abfragen ergeben sie den Wahrheitswert TRUE. Dadurch kann im „Hauptprogramm“ zu den Funktionen **datensuchen()** bzw. **dateneinfuegen()** verzweigt werden. Bei diesen Funktionen wiederholt sich das Spiel. Es wird wieder eine HTML-Seite mit Formularen aufgebaut. Die Schaltflächen tragen verschiedene Variablennamen und das „Hauptprogramm“ kann bei einem neuerlichen Aufruf entscheiden, welche Funktion ausgeführt werden soll. In der Funktion **datenspeichern()** findet dann mit einem SQL-Befehl die Speicherung oder Löschung eines Datensatzes statt.

Mit dieser Grundstruktur für das Einfügen, Suchen, Ändern und Löschen von Datensätzen lassen sich auch eigene Datenbanklösungen aufbauen.

Weltweiter Datenaustausch - eine Vision

MySQL kann nicht nur lokal Daten abfragen. Über das Internet kann mit dem Befehl

mysql -h hostname -u username

sogar eine MySQL-Datenbank auf einem anderen Server abgefragt werden. In diesem Zusammenhang träume ich von einem weltweiten Datenbanknetz, das untereinander Daten austauscht und weitergibt. Derzeit wird im Internet sehr oft dieselbe Information (Schulen im WWW, interessante WWW-Adressen) von verschiedenen Personen oder Stellen gesammelt ohne dass Änderungen an andere „Sammler“ weitergegeben werden. Mein Vorschlag ist, diese Information gratis allen zur Verfügung zu stellen

Weitere Stellenangebote siehe ***Seite 21***

UNIX SystemmanagerIn	HTL EDV o.ä.	ÖAMTC, Schubertring01-71199-1378 Frau Mag. Stadlmair
Wartung InternetSeiten EDV-TechnikerIn	fundierte Kenntnisse, technische Ausbildung	Bit by Bit Hirtschner & Faschingoffice@bitbybit.at, 01-3308836-0
Untertützung für technischen Sicherheitbeauftragten	HTL Medizintechnik	Amt für Wehrtechnik, Wien, Dr. Lembacher, 01-5200-30200
EDV-Techniker		Microcash Lindenbauer & Sagmüller, Wiener Neudorf, Frau Mayer, 02236-4252513
C++ oder Java Anwendungsentwickler Systemadministratoren	WinNT, Win95, Lotus Notes	BOC, Wipl.-Wi.-Inf. Harald Kühn, 01-5132736-0, boc@boc-eu.com
HTL-Techniker	HTL, NT, EL, BIO	Institut für Sportwissenschaft, Frau Fritz, 04277-48881
Internet Routing Specialist NCC & Installation Technician	Router, Netzwerkkonzeption, Support	Colt, Take It - Personalberatung, 01-58826-42, juergen.jakobi@takeit.co.at
EDV-Assistent	Windows NT, MS-Office, Datenbank. SQL-Server	Stadtgemeinde Klosterneuburg, Dr. Fronz, 02243-444-205
Netzwerk Management High End Networking	Netzwerktechnik	Comnet, Thomas Ochsenbauer, 01-8768844-0, ochsenbauer@comnet.at
NetzwerkadministratorIn	Novell, X-Base, TCP/IP, PC, Clipper, Führerschein B	Syscom, Frau Gregor, 01-73229-74
Elektrotachniker	Digitalschaltungen, Mikroprozessoren, MS-Access, Visual Basic	Follow Me, Patrick Fally, 01-4786266 p.fally@chello.at
Informatiker	NT-Server, SQL-Server, Datenbankdesign, MS-Access, Visual Basic	Follow Me, Patrick Fally, 01-4786266 p.fally@chello.at
Software-EntwicklerInnen	C, Assembler	RSE,01-6025482, officeqradner.co.at
Techniker	Zutritskontrolle, Elektronik	EVVA,Herr Haslinger, 01-8116510

und mit einem Netz von SQL-Datenbanken anderen weiterzugeben (Projekt FDB = Freie Datenbank).

Auf dem Server des Pädagogischen Instituts für Niederösterreich in Hollabrunn wurde unter der Adresse *www.pinoe-hl.ac.at* eine MySQL-Datenbank **FDB** für alle Benutzer des Internets zum Lesen freigegeben. Diese Datenbank enthält eine Tabelle **SCHULEN**, in über 6400 österreichische Schulen gespeichert sind. Von mehr als 1180 Schulen sind auch die WWW-Adressen gespeichert. Dies werden täglich automatisch mit einem Perl-Skript auf ihre Richtigkeit überprüft.

Für die Tabelle **SCHULEN** existieren bereits PHP-Skripts für Linux, mit denen über WWW-Seiten die Datensätze verändert werden können. Diese Software und auch der Inhalt der Tabelle **SCHULEN** darf von allen heruntergeladen und genutzt werden.

Außerdem gibt es bereits Perl-Skripts, mit denen die Daten zwischen mehreren MySQL-Server aktualisiert werden können. Damit können z. B. die geänderten Daten vom Server des Pädagogischen Instituts für Niederösterreich heruntergeladen und in die eigene Datenbank gespeichert werden.

Dieser Austausch könnte mit beliebigen Daten stattfinden, die nicht dem Datenschutz unterliegen und sich in Datenbanken speichern lassen. Der Aufbau einer Datenbank mit WWW-Adressen für den Unterricht ist geplant. Mitarbeit an diesem Projekt ist erwünscht auch wenn sie nur darin besteht, die Daten regelmäßig vom PI-Server herunterzuladen und damit die eigene Datenbank zu aktualisieren.

Copyright

Literatur

Egon Schmid u. a., php - dynamische webauftritte professionell realisieren. Markt&Technik, München 1999. (Dieses Buch ist eher für Fortgeschrittene geeignet.)

WWW-Adressen

Skripts für diesen Artikel herunterladen
http://www.pinoe-hl.ac.at/material/mysql/

MySQL-Homepage
http://www.mysql.com/

Projekt FDB
http://www.pinoe-hl.ac.at/material/fbd/

Perl-Homepage
http://www.perl.com/

Perl-Skripts
http://www.pinoe-hl.ac.at/material/perl/

PHP Code Exchange (Sammlung von PHP-Skripts)
http://px.sklar.com/

PHP-Homepage
http://www.php.net/

PHP-Skripts
http://www.pinoe-hl.ac.at/material/php/

phpMyAdmin zur Verwaltung von MySQL-Datenbanken
http://www.htmlwizard.net/phpMyAdmin/

Schülerverzeichnis mit PHP-Skript
http://www.ebgymhollabrunn.ac.at/schueler/

Im Web

ftp://pcnews.at/pcn/67/reidlinger/schulDB.php3