

Datenbanken für Online Untersuchungen

Im vorliegenden Text wird die Verwendung einer MySQL Datenbank für Online Untersuchungen beschrieben. Es wird davon ausgegangen, dass die Untersuchung aus mehreren Seiten besteht, die Werte mittels Post übergeben werden und PHP auf dem Server installiert ist. Dieses Setting ist an der Uni Wien auf dem Server gerda.univie.ac.at gegeben. Im Folgenden werden die Beispiele darauf zugeschnitten sein.

VERWENDUNG VON DATENBANKEN ALLGEMEIN

In (relationalen) Datenbanken werden Informationen in Tabellenformat gespeichert. Die Spalten stellen üblicherweise die zu speichernden Variablen dar. Deren Anzahl ist begrenzt (auf 255 bzw. 1024 Spalten). Für jede Person wird eine Zeile angelegt.

ID	IP	Alter	Einkommen
1	196.123.22.2	23	1000
2	160.123.22.3	25	

Es ist sinnvoll jeder Person eine eindeutige fortlaufende Nummer zu geben. Im Folgenden wird diese Nummer ID genannt.

VORGEHEN BEIM SPEICHERN IN EINE TABELLE

Für jede Versuchsperson wird zu Beginn der Untersuchung ein neuer Satz (=Zeile in der Tabelle) angelegt. Dieser Satz enthält eine neue ID und wenn gewünscht die IP Adresse und andere Informationen (z.B. Browser etc.). Alle anderen Werte (wie in unserem Beispiel Alter und Einkommen) bleiben frei. Erst im Laufe der Untersuchung werden diese Werte gezielt zu der jeweiligen Person geschrieben. Im Konkreten sieht es so aus, dass auf der ersten Seite ein neuer Satz mit ID in die Tabelle geschrieben wird. Auf den folgenden Seiten werden die Werte der vorherigen Seite gespeichert. Angenommen ein Fragebogen besteht aus 4 Seiten: einer Begrüßungsseite, einer Seite mit demographischen Daten, einer mit Informationen über die finanzielle Umwelt der Person und einer Bedankungsseite. So werden auf der Begrüßungsseite die Befehle zum Schreiben eines neuen Satzes in die Tabelle codiert. Auf der Seite mit den Infos zur finanziellen Umwelt werden die demographischen Daten gespeichert (die auf der Vorseite eingegeben wurden) und auf der Bedankungsseite wiederum werden die Daten zur finanziellen Umwelt gespeichert. Der Speichervorgang erfolgt also immer um eine Seite versetzt. Für die Implementierung eines MySQL Zugriffs sind daher folgende Schritte notwendig:

- 1) Erstellen einer Tabelle mit allen nötigen Spalten (=Variablen)
- 2) Anlegen eines neuen Satzes mit einer neuen ID auf der ersten Seite
- 3) Hinzufügen der neuen Variablen zu dem jeweiligen Personensatz

ERSTELLEN EINER NEUEN TABELLE

Unter dem Link <http://gerda.univie.ac.at/phpmyadmin/> befindet sich die graphische Benutzeroberfläche der MySQL Datenbank auf dem Server gerda.univie.ac.at. Ist keine solche zur Verfügung muss der SQL (Structured Query Language) Befehl „CREATE TABLE ...“ abgesetzt werden. Wenn gewünscht wird dieser separat besprochen.

Folgt man dem oben angeführten Link so wird man nach Benutzername und Passwort gefragt. Hier ist der gleiche Benutzername und das gleiche Passwort zu verwenden wie beim SFTP-Transfer. Links oben kann man die gewünschte Datenbank auswählen. Zum Testen bitte die Datenbank „TEST“ verwenden indem man Sie aus der Liste auswählt.

Im rechten Fenster erscheint eine Liste der derzeit angelegten Tabellen. Scrollt man bis an das Ende sieht man folgende Information:

The screenshot shows the phpMyAdmin interface for the 'test3' database. At the top, there are buttons for 'Anzeigen', 'Teilw. anzeigen', and 'Einfügen E'. Below this, it indicates '16 Tabellen' and 'Gesam'. There are links for 'Alle auswählen / Auswahl entfernen' and a 'markierte:' input field. A menu is visible with options: 'Druckansicht', 'Strukturverzeichnis', 'Neue Tabelle in Datenbank onlineresearch erstellen', and 'Fehler'. The 'Neue Tabelle in Datenbank onlineresearch erstellen' dialog is open, showing a 'Name:' input field and a 'Felder:' input field with an 'OK' button.

In das Feld mit der Beschriftung Name: gibt man den Tabellennamen ein (z.B. Ergebnisse oder Rohdaten). Dieser wird dann für die weiteren Befehle benötigt. In das Feld mit der Beschriftung Felder: wird die Anzahl der Variablen bzw. Spalten angegeben. Im Anschluss klickt man auf OK. Für jede Spalte kann sodann folgendes definiert werden:

- **Feld:** Name des Feldes (sollte den Inhalt beschreiben, möglichst gleich wie im SPSS und im "Name" Attribut des <Input...> Tags.)
- **Typ:**
 - **VARCHAR** bezeichnet Felder die sowohl Zahlen als auch Buchstaben und Sonderzeichen beinhalten und max. 255 Zeichen fassen.
 - **TEXT** bezeichnet Felder die länger als 255 Zeichen fassen
 - **INT** ist nur für Zahlen geeignet.

Für die Zwecke der Fragebogen möchte ich weder genauer auf die Unterscheidung eingehen noch auf die anderen Datentypen. Als Regel bitte für alle Felder VARCAHR verwenden außer für die ID und Felder die als Textarea definiert wurden. ID Felder erhalten den Typ INT und Textarea Felder den Typ TEXT.

- **Länge/Set:** Hier wird die maximale Länge des Feldes definiert. Dieser Wert ist gleich dem Wert bei maxlength des INPUT Tag. Werden längere Texte eingegeben, werden diese abgeschnitten.
- **NULL:** Hier wählt man „NULL“ aus damit auch keine Werte in den Feldern gespeichert sein können (außer für die ID, diese muss immer einen Wert haben).
- **Extra:** Hier wird für das Feld ID der Wert auto increment eingetragen. Dadurch wird die ID jeweils um 1 erhöht wenn eine neue Person angelegt wird. ACHTUNG bei ID auch Primary Key anklicken!!!!

Feld	Typ [Dokumentation]	Länge/Set*	Attribute	Null	Standard	Extra	Primärschlüssel
id	INT			not null		auto_increment	<input checked="" type="radio"/>
alt	VARCHAR	2		null			<input type="radio"/>
geschl	VARCHAR	2		null			<input type="radio"/>
ausbild	VARCHAR	2		null			<input type="radio"/>
freunde	VARCHAR	4		null			<input type="radio"/>

Nachdem die Tabelle erfolgreich erstellt wurde gehen wir im nächsten Schritt dazu über, Werte aus unserem PHP/HTML File in diese Tabelle zu speichern.

ANLEGEN EINES NEUEN SATZES

Auf der ersten Seite wird für jede Person ein Satz angelegt. Dieses „anlegen“ erfolgt mittels PHP. Darum müssen wir im ersten Schritt unsere HTML Datei in eine PHP umwandeln indem wir die Erweiterung (das ist jener Teil im Dateinamen, der hinter dem Punkt steht) auf .php ändern. Im HTML File selbst muss folgende Befehlsfolge zu Beginn notiert werden:

```
<?php
// Mit diesem Befehl erfolgt die Verbindung zu Datenbank.
// statt user und passwort bitte jene Werte eintragen die zum
// Anmelden an die Datenbank verwendet werden.
$db = mysql_connect("localhost", "user", "passwort")
    or die ("Konnte keine Verbindung zur Datenbank
herstellen bitte informieren Sie mich!!!!");

// Statt test wird hier der Name der Datenbank bestimmt
// in der die Tabellen liegen (bei uns test).
mysql_select_db("test", $db);

// Mit diesem SQL Befehl werden die Daten in die
// Datenbank geschrieben. Er wird später noch näher erklärt
$$sql = "INSERT INTO ergebnisse (alt, geschl, ausbild)
values ('$alter', '$geschl', '$ausbild') " ;

// Hier wird der obige Befehl durchgeführt
$result = mysql_query($sql);

// Hier trennt man die Verbindung zu Datenbank
```

```
$db_close = mysql_close($db);
```

```
?>
```

rot markierte Werte müssen angepasst werden, grün markierte können.

DER INSERT BEFEHL

```
$sSql = "INSERT INTO ergebnisse (alt,geschl)
values('$alter','$geschl') " ;
```

Statt `ergebnisse` geben Sie hier den Namen der Tabelle an (jenen den die Tabelle beim Anlegen bekommen hat).

`alt,geschl,ausbild` ist die Liste der Felder in die Werte geschrieben werden sollen. Hier werden die Bezeichnungen der Felder, die bei der Tabellenerstellung festgelegt wurden, verwendet.

'`$alter`', '`$geschl`', '`$ausbild`': Hier greift man auf die Eingabewerte der Versuchsperson zu. Auf jene Eingabefelder, die auf der vorhergehenden Seite mit `<input name=wert >` definiert wurden, können jetzt mit `$wert` zugegriffen werden. Das bedeutet, dass alle Eingabefelder einen Namen bekommen müssen und dass diese auf der nächsten Seite verarbeitet werden können, wenn man vor dem Namen ein `$` setzt. In diesem Fall werden die Werte in einen SQL Befehl verpackt. Statt den roten Feldern müssen also jene Namen eingetragen werden, die im Input Tag vergeben wurden. Die Reihenfolge der Felder gibt die Zuordnung an von Eingabefeld zu Tabellenfeld. Im obigen Beispiel wird das Eingabefeld `alter` (das z.B. den Wert 25 hat) in das Tabellenfeld `alt` geschrieben. `Geschl` wird auf `geschl` geschrieben. Möchte man ein weiteres Feld hinzufügen, so muss sowohl die grüne als auch die rote Liste um das Feld erweitert werden. Soll z.B. noch die Ausbildung mitgespeichert werden so sieht der Befehl folgendermaßen aus:

```
$sSql = "INSERT INTO ergebnisse (alt,geschl,ausbild)
values('$alter','$geschl','ausbild') " ;
```

HINZUFÜGEN NEUER WERTE AUF DEN FOLGSEITEN

Die Eingabewerte werden auf der nachfolgenden Seite gespeichert. Um einen bestehenden Satz zu ändern (und der Personensatz besteht ja weil er auf der ersten Seite eingegeben wurde) wird der UPDATE Befehl verwendet. Der PHP-Code selbst ist sehr ähnlich:

```
<?php
```

```
    $db = mysql_connect("localhost","user","passwort")
        or die ("Konnte keine Verbindung zur Datenbank
herstellen bitte informieren Sie mich!!!!");
```

```
    mysql_select_db("test",$db);
```

```
// Mit diesem SQL Befehl werden die Daten in die
// Datenbank geändert. Er wird später noch näher erklärt
```

```
$sSql = "UPDATE ergebnisse set freunde='$freunde' WHERE  
id=$userid" ;
```

```
$result = mysql_query($sSql);
```

```
$db_close = mysql_close($db);  
?>
```

```
$sSql = "UPDATE ergebnisse set freunde='$freunde' WHERE  
id=$userid" ;
```

Im obigen Befehl wird in der Tabelle ergebnisse in der Spalte freunde der Person userid ein neuer Wert gesetzt. Dieser neue Wert ist \$freunde, also die Eingabe der Versuchsperson. (ACHTUNG auf der vorherigen Seite steht z.B. <input type=text name=freunde ...>)

Bevor der obige Befehl jedoch funktioniert muss die id von einer Seite auf die nächste weitergegeben werden. Dies können Sie sich im Beispiel8.php ansehen.

IDs WEITERGEBEN

Zuerst wird gleich nach dem Insert die ID ermittelt mit den Befehlen:

```
$sSql = "Select max(id) id from ergebnisse ";
```

```
$result = mysql_query($sSql);
```

```
$row = mysql_fetch_array($result);  
$userid=$row['id'];
```

Auf jeder Seite wird im HTML jener Befehl benötigt:

```
<input type=hidden name="userid" value="<?=$userid?>">
```