
MySQL, phpmyadmin und SQL

MySQL (sprich: „Mai Ess Ku Ell“)

ist eine kostenlose relationale Datenbank (relational heißt, es können zwischen Tabellen Beziehungen erstellt werden). Der Name ist nicht restlos geklärt, eine schöne Version lautet so: „My“ heißt die Tochter des Entwicklers Michael Widenius, ein Norweger. „SQL“ ist die Abkürzung für „Structured Query Language“ – eine Sprache, mit welcher Datenbanken angesprochen werden.

phpmyadmin

ist eine komfortable Software, die serverseitig läuft und über die MySQL prima administriert, d.h. eingerichtet und gewartet werden kann.

Wozu dann noch HTML und PHP???

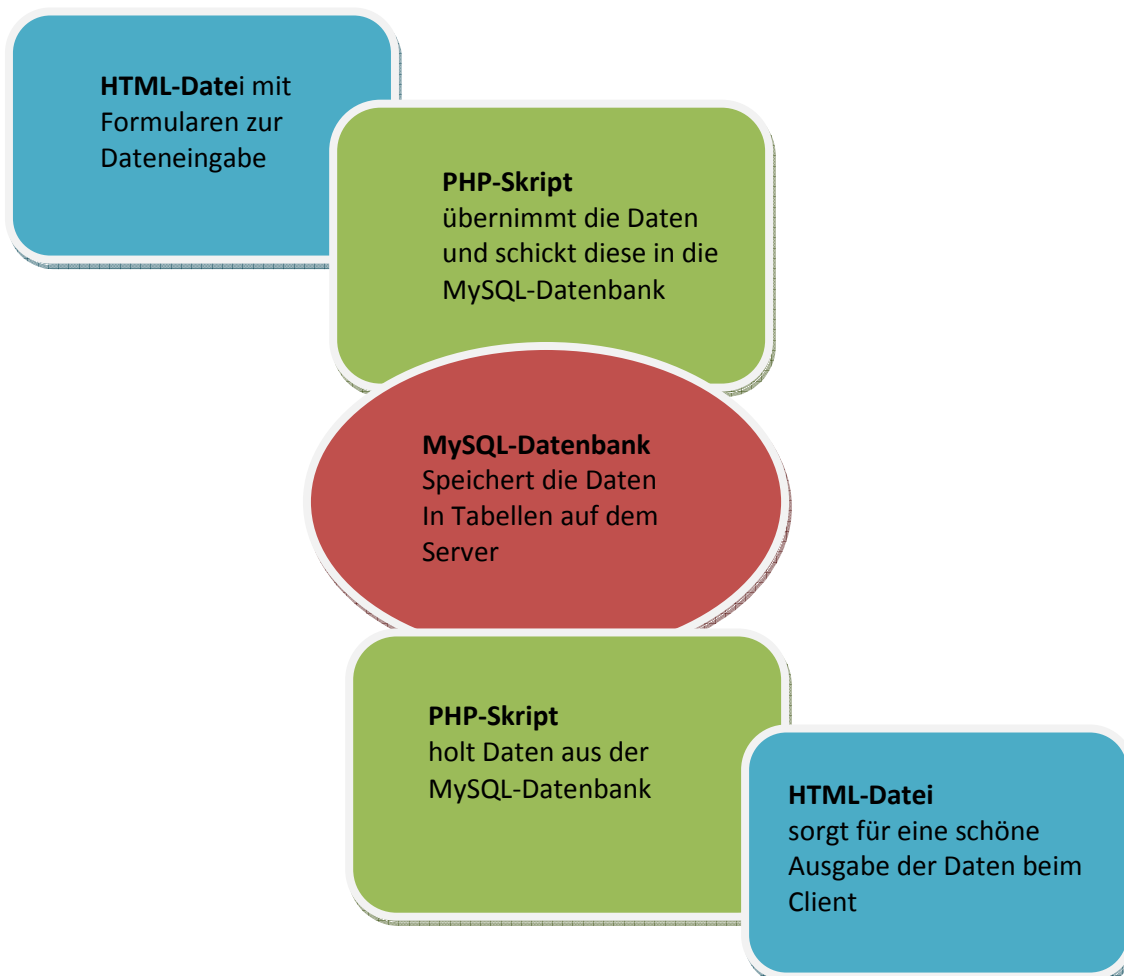
Diese beiden benötigen wir, um unseren Besuchern unserer Websites die Gelegenheit zu geben, Daten einzugeben und auch abzufragen, ohne dass sie direkt auf unsere Datenbank draufmüssen. Da hat nämlich außer uns niemand etwas drauf verloren!!!

Grafisch dargestellt schaut das ungefähr so aus:

clientseitig:

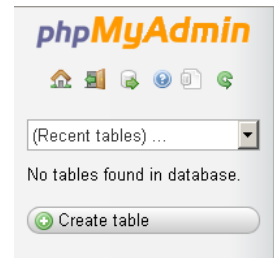
serverseitig:

clientseitig:

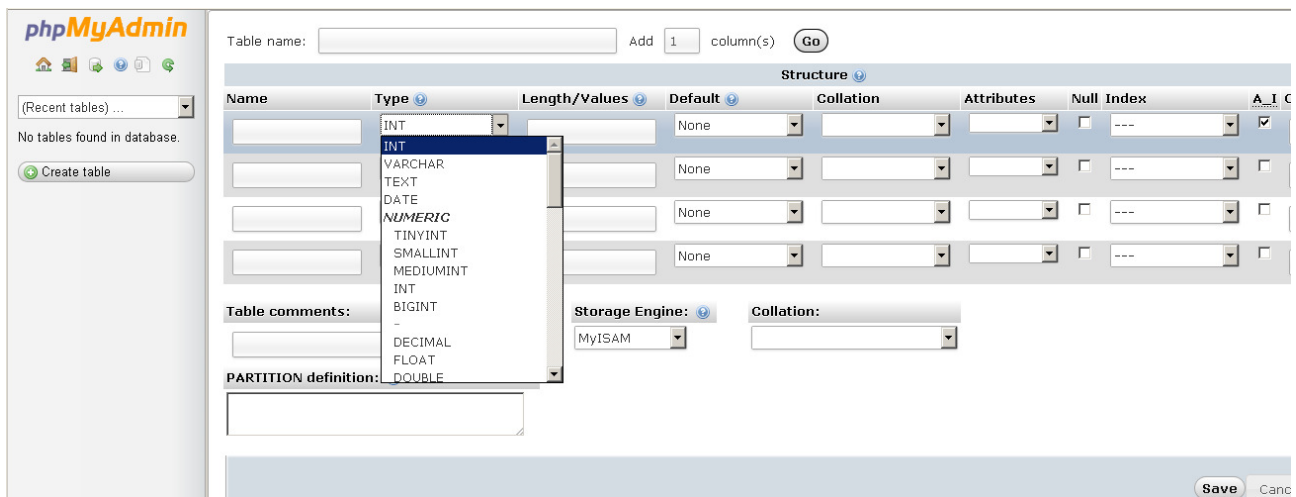


phpmyadmin

phpmyadmin wird im Kontrollpanel aufgerufen:



Zuerst wird mit  eine Tabelle angelegt:



Um eine Tabelle in einer Datenbank anzulegen sind folgende Punkte wichtig:

1. Die Tabelle braucht einen eindeutigen Namen
2. Die Felder der Tabelle brauchen eindeutige Namen
3. Den Feldern muss ein Datentyp zugewiesen werden
4. Das erste Feld soll die ID enthalten, den Datentyp INT bekommen und einen Haken bei A.I. (das steht für „Auto Increment“ und heißt, dass für jeden neuen Eintrag die ID um „1“ erhöht wird und so eine eindeutige Identifikationsnummer (ID) erstellt wird, über die ein Datensatz später gefunden werden kann.
5. Wenn alle Angaben gemacht sind muss [Save] gedrückt werden.

Zu den „Datentypen“:

Diese bestimmen, wieviel Speicherplatz für Variablen reserviert wird und das sollte möglichst sparsam geschehen (ist zwar für unsere Zwecke eigentlich egal, aber stellen wir uns vor, wir wollen Millionen von Datensätzen speichern, z.B. als Telefongesellschaft, als amazon.de, zalando.de oder eine Fluggesellschaft, dann kommt es auf jedes einzelne Byte an . . .)

INT (Integer) – gibt eine ganze Zahl an – also ideal für unsere ID, aber z.B. auch für PLZ

TEXT – speichert Zeichen aller Art, brauchen wir für Namen, Texte, Orte etc.

DATE – speichert Datumswerte und zwar immer in der Form JJJJ-MM-TT

Alle anderen sind für unsere Zwecke im Moment nicht von Bedeutung und können unter <http://dev.mysql.com/doc/refman/5.1/de/data-types.html> nachgelesen werden.

SQL

Um Daten in eine Datenbank schreiben und von dieser wieder auf vielfältigste Art und Weise auslesen zu können, benötigen wir einige SQL-Befehle, die in das PHP-Skript integriert werden.

Zunächst muss eine Verbindung zur Datenbank hergestellt werden:

```
<?php
mysql_connect ("Datenbankserver", "Username", "Passwort")
or die ("Keine Verbindung möglich!");
mysql_select_db ("Datenbankname");
?>
```

Tip: Am besten speichert man diesen Teil des Programms in einer sogenannten „Include-Datei“ (Endung .inc) ab. Man braucht später, wenn man mehrere Skripten hat, die die Datenbankverbindung benötigen, nur mit dem include-Befehl die Datei angeben und nicht immer die gesamte Prozedur angeben.

Daten in die MySQL-Datenbank einfügen

```
mysql_query(„insert into tabellename (feldname1,
feldname2, feldname...) values (`Wert1`, `Wert2`, `Wert3`)“);
```

mysql_query stellt eine PHP-Anweisung dar.
Insert into . . . ist die SQL-Anweisung.

Daten aus der Datenbank auslesen

```
$ergebnis=mysql_query(„select * from tabellename“);
```

Mit diesem Kommando werden alle Daten der Tabelle ausgelesen und in der Variablen \$ergebnis gespeichert. Allerdings können wir uns die Daten jetzt noch nicht anschauen.

Daten ausgeben

Nachdem wir nicht wissen können, wie viele Datensätze in einer Tabelle enthalten sind und wir nicht jedesmal vor einer Datenbankabfrage über phpmyadmin nachschauen wollen, lassen wir dies den Computer für uns erledigen. Er macht das, indem er solange Daten aus der Tabelle holt, wie er welche findet, d.h. bis er ans Ende der Tabelle gelangt ist.

```
while ($result=mysql_fetch_object($ergebnis))
{
echo $result->feldname1, „ „ , $result->feldname2;
}
```

Bitte genau auf die Kommas und Anführungszeichen achten. Außerdem steht am Ende der Zeile, in der die while-Funktion steht, **kein** Strichpunkt am Ende!!!