Allgemeines HTML5, PHP, MySQL

HTML dient zur Formatierung einer Webseite. Hiermit kann ein Text formatiert, Bilder, Tabelle oder auch Hyperlinks eingefügt werden. Außerdem kann man mit HTML ein Formular für die Eingabe von Daten (z.B. Namen oder Adressen) erstellen. Für die Eingabe der beschriebenen HTML-Kommandos können beliebige HTML-Editoren verwendet werden. Diese kann man auch als Freeware im Internet herunterladen (z.B. www.brackets.io, www.aptana.com), wobei es auch kommerzielle Editoren wie **Dreamweaver** gibt. Wir können aber für die Eingabe der HTML-und PHP-Kommandos auch einfach einen Editor, wie z.B. den Windows-Editor, verwenden.

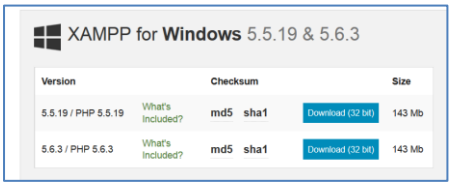
PHP ist eine serverseitige Programmiersprache. Damit läuft ein PHP-Programm auf einem Server ab und nicht bei dem, der sich die entsprechende Seite anschaut. Im Gegensatz zu **JavaScript**, dass beim Anwender abläuft. D.h. mit PHP kann man beispielsweise keine Grafik verändern, sobald sich die Maus über der Grafik bewegt. Sollen aber Daten in einer Datenbank gespeichert werden, auf die andere zugreifen können, oder sollen Daten zwischen zwei Personen - die sich an verschiedenen PCs eine Seite ansehen - ausgetauscht werden (z.B. bei einem Forum, Chat, oder auch Browser-Games mit mehreren Spielern), dann benötigt man eine serverseitige Programmiersprache, wie PHP.

Grundsätzlich funktioniert es so, dass eine Eingabe (z.B. über ein HTML-Formular) von einem Anwender an einen Server geschickt wird, auf dem sich **PHP und MySQL** befinden. Dabei wird beispielsweise eine Seite, die sich auf dem Server befindet, durch das Abschicken eines HTML-Formulars aufgerufen. Auf dem Server wird dann das PHP-Programm, welches in diese Seite eingebunden ist, abgearbeitet. Hier können dann auch gesendete Daten beispielsweise in einer MySQL-Datenbank gespeichert werden. Die Seite, die der Anwender aber in seinem Browser sieht, enthält nur HTML-Anweisungen. Somit merkt der Anwender auch nichts von den dahinter steckenden Programmen.

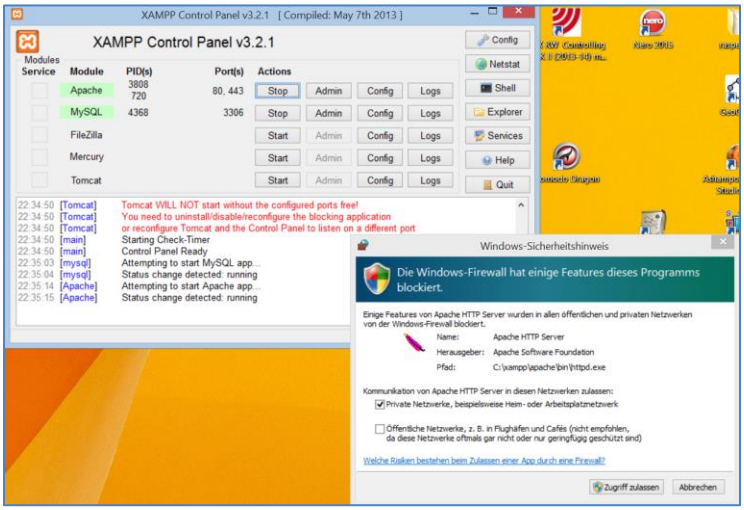
http://www.galaxieswar.de/Buch/

Xampp www.apachefriends.org

Nur 32 Bit möglich

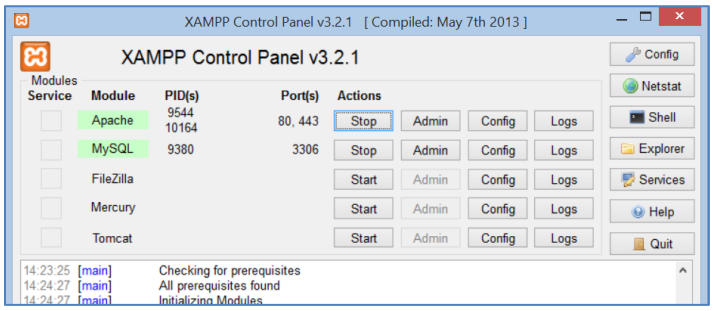


eventuell muss man nach dem Klick auf „Start“ den „Zugriff zulassen“:



Daten verwalten mit MySQL

Öffne XAMPP und starte den MySQL-Sever



MySQL ist eines der am häufigsten in Verbindung mit PHP eingesetzten Datenbankmanagement-systeme. Ein weiterer Vorteil ist, dass MySQL bei den meisten Hostingangeboten mit PHP schon dazugehört.

MySQL ist dual lizensiert. Zum einen ist es eine freie Software, die unter der General Public License (GPL) steht. Es kann aber auch mit einer kommerziellen Lizenz eingesetzt werden.

MySQL basiert auf der Datenbanksprache SQL (Standard Query Language).

Datenbanken:

Am häufigsten werden relationale Datenbanksysteme eingesetzt, in denen die einzelnen Daten in Tabellen gespeichert und die Tabellen in Relationen zueinanderstehen.

Tabellen bestehen aus Zeilen (rows) und Spalten (columns). Eine Zeile einer Tabelle nennt man einen Datensatz (record). Für Spalten verwendet man den Begriff Felder (fields).

phpMyAdmin

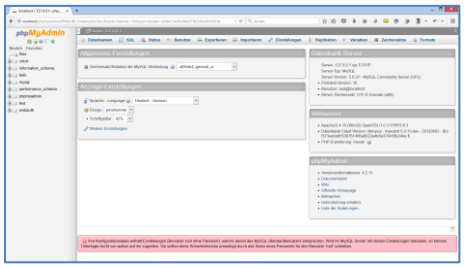
Ist ein komfortables Clientprogramm, das eine Arbeit in einer grafischen Oberfläche ermöglicht. Es ist ein in PHP geschriebenes Tool zur Verwaltung von MySQL-Datenbanken. Es ist bei den meisten Providern vorinstalliert und auch bei XAMPP mit dabei.

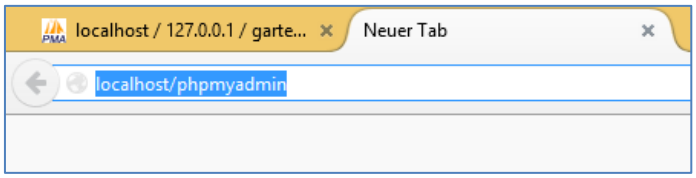
Im Normalfall werden folgende Schritte gesetzt:

1. Man erstellt eine Datenbank und die dazugehörigen Tabellen in phpMyAdmin
2. Um die Daten auszulesen und auf einer Website auszugeben, programmiert man ein PHP-Skript mit den Befehlen zum Auslesen.

Bei XAMPP erreicht man phpMyAdmin in der Adresszeile des Browsers mit der Eingabe von

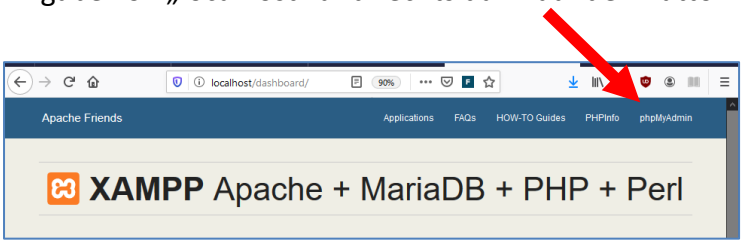
- 2 Varianten
  http://localhost/phpmyadmin





Oder

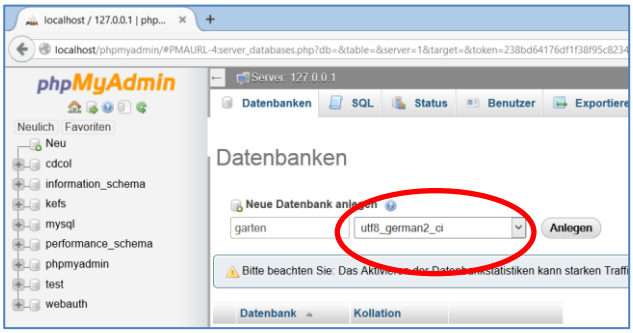
- Eingabe von „localhost“ und rechts dann auf den Button



Datenbank anlegen

Rufe die Startseite von phpMyAdmin auf.

Klicke den Reiter „Datenbank“ an und vergib einen Namen z.B. „garten“. Lege dann die Kollation „utf8_general_ci“ fest. Klicke dann auf „Anlegen“.



Die Kollation bestimmt die Sortierung von sprachspezifischen Sonderzeichen, die das deutsche Ü, Ä usw.

utf8_general_ci: ist die schnellste, allerdings wird ß als s einsortiert, Umlaute Ä ist A… wird bei www.bplaced.net auch angeboten.

utf8_german2_ci: leider nicht bei www.bplaced.net möglich, daher bitte hier nicht wählen! Hier würde die Sortierung der Umlaute besser funktionieren (z.B. Ä wäre AE)

oder: utf8_unicode_ci

Exkurs: utf8

Der Standardzeichensatz von MySQL ist Latin 1, der dem Zeichensatz ISO-8859-1 entspricht. Wenn man in MySQL eine Tabelle anlegt, dann wird diese automatisch mit dem Zeichensatz Latin 1 angelegt; genauer: mit der schwedischen Variante von Latin 1 – schließlich stammt MySQL aus Schweden.

Mit UTF-8 sind die Zeichen wie das Euro-Symbol und die Anführungszeichen und alle anderen Zeichen kein Problem, weshalb es sinnvoll ist, bei der Entwicklung konsequent und durchgängig mit diesem Zeichensatz zu arbeiten. Das ist ein ganz klein wenig mühseliger, aber es löst sehr viele Probleme: Die kleine Mühe bringt reichlich Ertrag.

Möchte man durchgehend mit UTF-8 arbeiten, dann muss man nicht nur darauf achten, dass alle Skripte in diesem Format gespeichert werden und der Zeichensatz im HTML-Header explizit angegeben wird, sondern man muss bei MySQL-Verbindungen zusätzlich UTF-8 wählen.

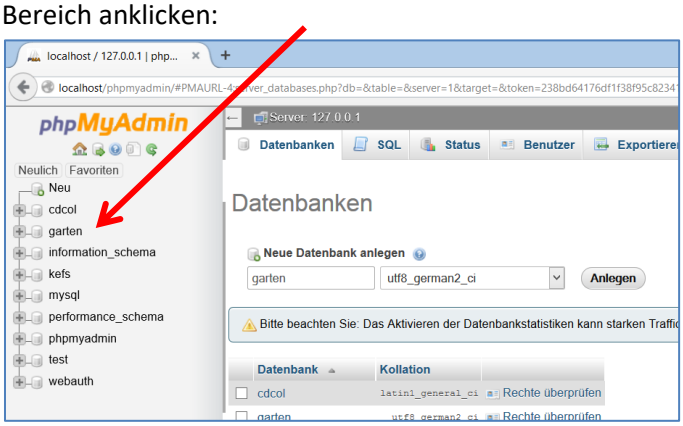
Damit die Verbindung zum Datenbankserver über UTF-8 läuft, muss man nach der Kontaktaufnahme noch die Methode set_charset() mit dem entsprechenden Zeichensatz aufrufen, also set_charset(”UTF8”) (bei MySQL wird der Zeichensatz ohne Bindestrich angegeben).

Warnung

set_charset(”UTF8”) muss man bei allen Skripten einfügen, in denen mit dem Datenbankserver kommuniziert wird, also auch bei *create_tables.php* usw.

In jedem <head>-Bereich aller Dateien muss man die Angabe <meta charset=”utf-8”> benutzen.

Dadurch wird die Datenbank angelegt. Um damit arbeiten zu können, muss man diese im linken Bereich anklicken:

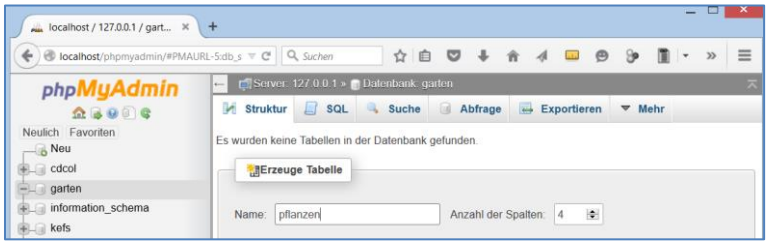


Die Datenbanken die hier schon vorhanden sind, sind teilweise welche, die von MySQL selbst für die eigene Verwaltung benötigt werden.

Tabellen erstellen

Nachdem die richtige Datenbank links ausgewählt wurde, hier „garten“, sieht man einen Bereich zum Anlegen einer Tabelle „Erzeuge Tabelle“. Zuerst muss man bestimmen, wie die Tabelle heißen soll und dann wie viele Spalten (Felder) sie enthalten soll.

Hier „pflanzen“ mit 4 Spalten. Klicke dann auf OK.



Es erscheint ein Fenster, indem man festlegt, wie die einzelnen Felder heißen sollen und welcher Inhalt gespeichert werden soll.

- Name bestimmt den Namen der Spalte
- Typ ist der Inhaltstyp der Daten. Standartmäßig können positive und negative Werte aufgenommen werden. Sollen nur positive gespeichert werden, gib „unsigned“ an. In Klammer steht „M“. Dadurch lässt sich die Anzeigenbreite definieren, ist ein Wert kleiner wird dieser dann mit Leerstellen aufgefüllt. Beispiele:
    - SMALLINT (M): Werte von -32768 bis 32767. Bei „unsigned“ Werte von 0 bis 65535.
    - INT (M): -2.147.483.648 bis 2.147.483.647
    - INTEGER (M): Synonym zu INT (M)
    - FLOAT (M;D): kleine Fließkommazahl
- Länge/Werte legt bei Zeichenketten die mögliche Anzahl der Zeichen in einem Feld fest
- Null legt fest, ob die Felder auch NULL enthalten dürfen. Wenn in einem Feld nichts drinsteht, wird dieses dann von MySQL mit NULL gefüllt.
- Index die fünf Möglichkeiten schließen sich aus. Am häufigsten „primary“

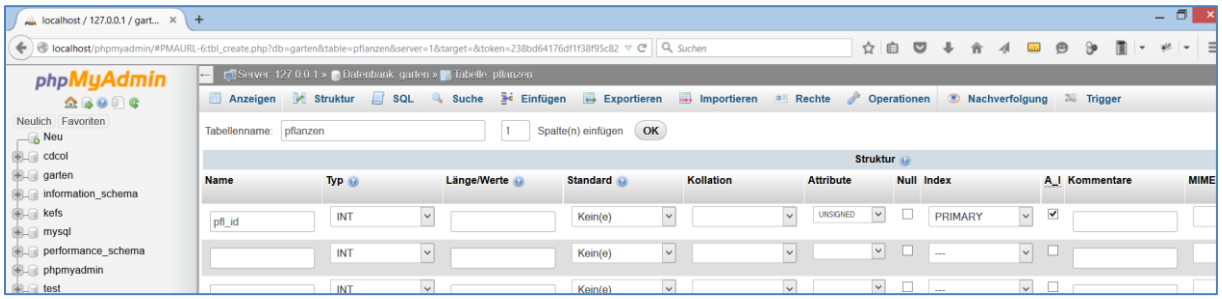
- A_I steht für „auto_increment“. Wenn man hier einen Haken setzt, wird der Wert automatisch bei jedem neuen Datensatz hochgezählt. Das ist praktisch für das Feld mit dem Primärschlüssel.
- Die letzten drei Optionen betreffen nur die interne Darstellung. Falls Bilder in der Datenbank gespeichert werden, kann man diese in phpMyAdmin als kleine Vorschaubilder anzeigen lassen.

Beispieltabelle:

plf_id

1. plf_id ist eine eindeutige ID für jede Pflanze. Diese soll eine fortlaufende Nummer sein, eine ganze Zahl, deswegen wir
2. INT als Typ gewählt. Das
3. Attribut „unsigned“ legt fest, dass nur positive Zahlen möglich sind.
4. Null kreuze NICHT an. Dadurch muss hier immer ein Wert eingetragen sein. Bei
5. Index wähle „PRIMARY“ und setze einen Haken bei
6. A_I.

Ergebnis:



Nächste Zeile:

name

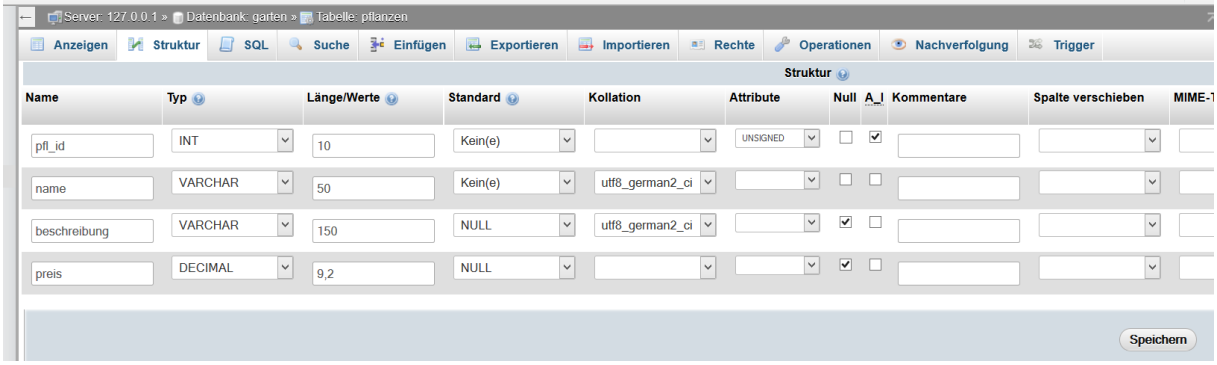
1. name enthält den späteren Namen der Pflanze. Hier wird der
2. Type „VARCHAR“ gewählt. Dies bedeutet, dass Zeichenketten eingegeben werden können. Die
3. Länge wird mit 50 beschränkt.

beschreibung

1. Enthält kurze Stichproben zur Beschreibung der Pflanz.
2. Typ ist ebenfalls VARCHAR, die Zeichenkette wird auf 150 festgelegt. Die Beschreibung muss nicht unbedingt eingegeben werden, daher setze in der
3. Spalte NULL einen Haken.

preis

1. Nimmt auf, was das Produkt kostet. Hier wird als
2. Typ DECIMAL gewählt. Bei der Länge wird 9,2 eingegeben. D.h. dass der Betrag nicht länger als 9 Zeichen (der Dezimalpunkt nicht mitgerechnet) sein kann und dass es zwei Stellen hinter dem Dezimalpunkt gibt. Da es sein kann, dass wir eine Pflanze aufnehmen, deren Preis noch nicht feststeht, mache ein Häkchen bei NULL.



Klicke auf Speichern.

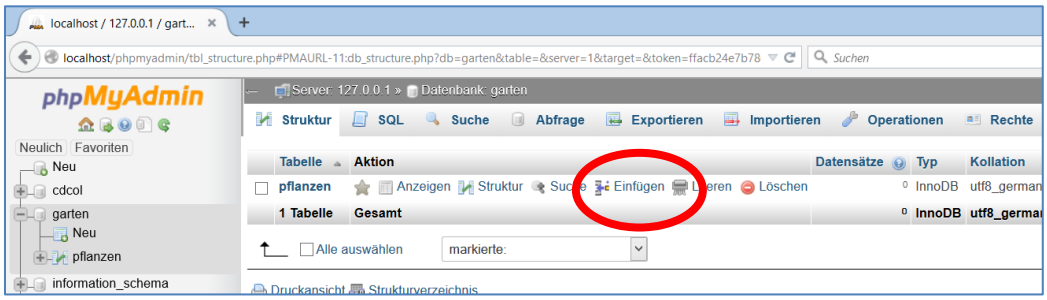
Daten einfügen

Daten können aus verschiedenen Quellen stammen:

1. Handelt es sich um eine überschaubare Anzahl kann man diese per Hand in phpMyAdmin eingeben.
2. Daten können auch aus Formularen stammen, die die Benutzer ausfüllen. Hierfür benötigt man den MySQL-Insert-Befehl.
3. Wenn Daten bereits in einer Excel-Tabelle oder ähnlichen vorliegen, kann man diese über phpMyAdmin importieren. Hierfür muss die Datei als CSV (comma separated value) abgespeichert werden.

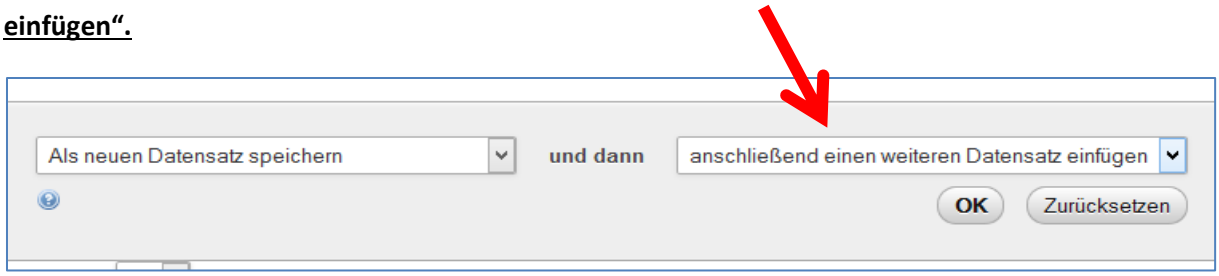
Daten in phpMyAdmin einfügen

Klicke neben der Tabelle auf „Einfügen“. Eventuell muss man vorher links auf den Datenbanknamen „garten“ klicken, um zu dieser Ansicht zu gelangen:



Man erhält ein Formular zum Eintragen der Daten. Füge die Daten unterhalb von „Wert“ ein. Da MySQL den Wert für die „pfl_id“ selbst vergeben soll, lasse dieses Feld leer. (Primärschlüssel).

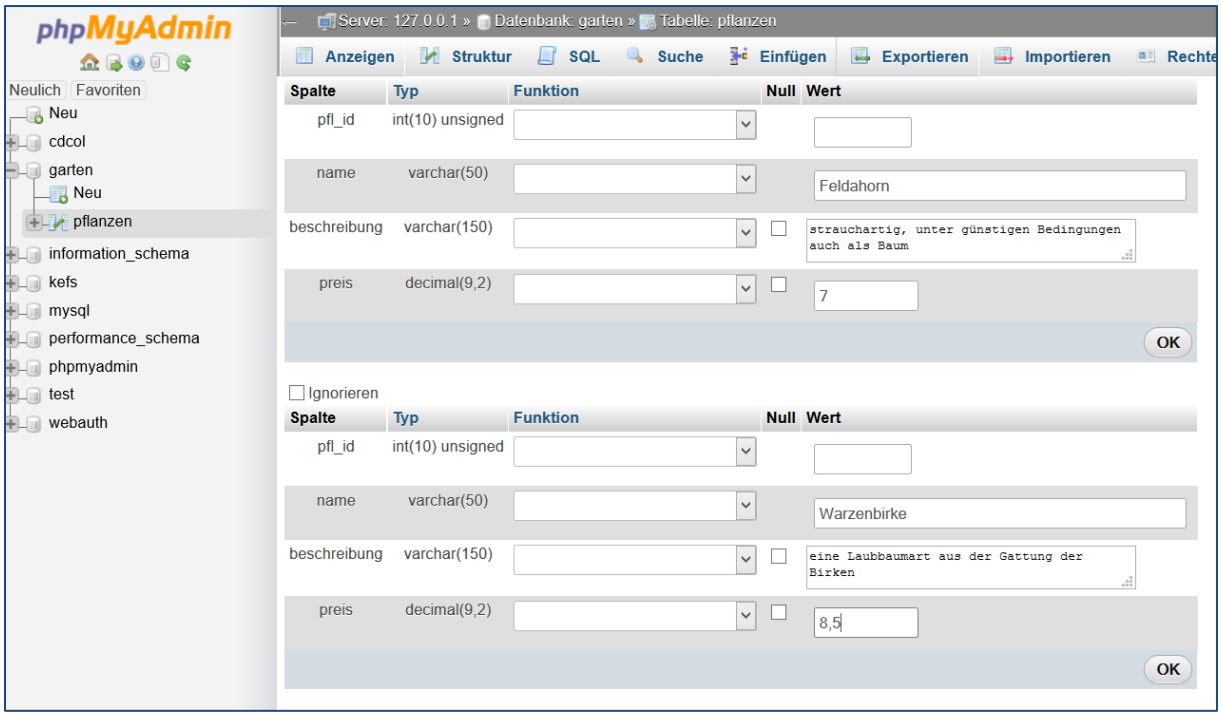
Standardmäßig werden zwei Formulare zum Eintragen von zwei Datensätzen angezeigt. Man kann auch nur den oberen ausfüllen und dann auf OK klicken. **Will man gleich weitere Datensätze eingeben, wähle unten anstelle von „zurück“ die Option „anschließend einen weiteren Datensatz einfügen“.**



Übung: Füge folgende Daten ein:

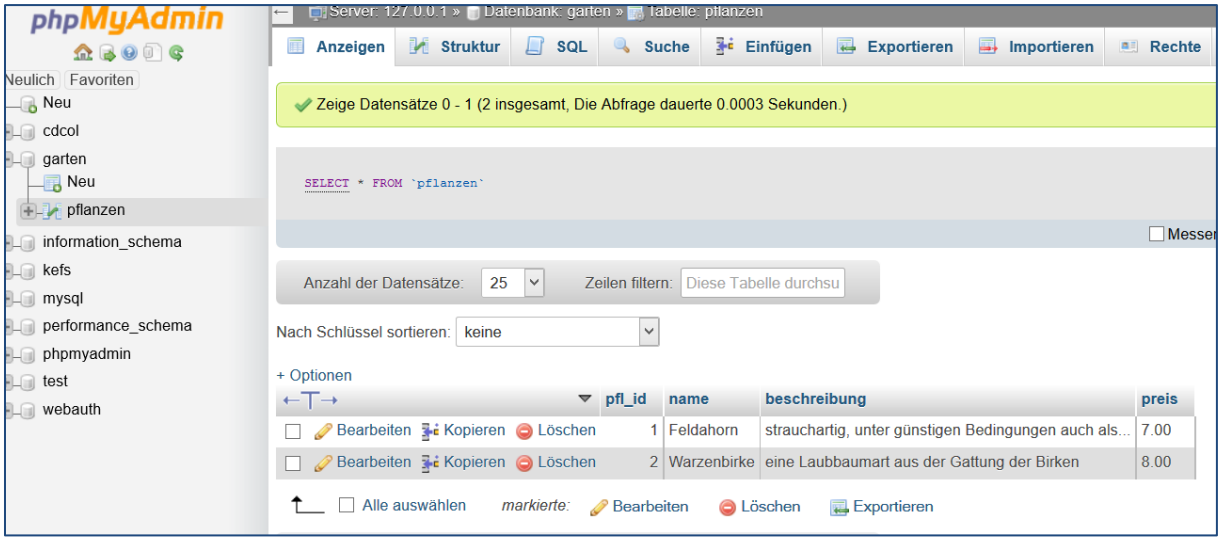
- Feldahorn; strauchartig, unter günstigen Bedingungen auch als Baum; Preis: 7
- Warzenbirke; eine Laubbaumart aus der Gattung der Birken; Preis: 8,5

Eingabe mit PUNKT statt Beistrich, sonst Fehlermeldung und er nimmt die Kommawerte nicht!!!



Datensätze verändern

Bestehende Datensätze lassen sich auch ändern. Wenn man die passende Datenbank ausgewählt hat (pflanzen), sieht man darunter die angelegten Tabellen und kann auf einen Tabellennamen klicken. Man sieht die bisher eingegebenen Daten.



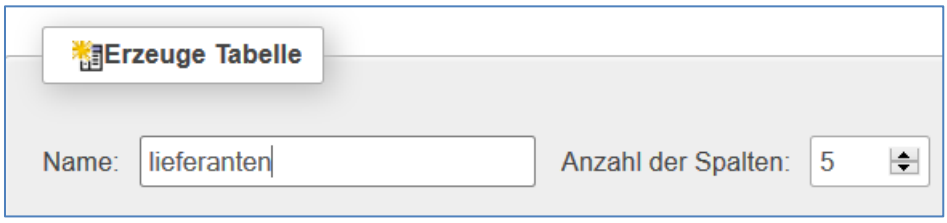
Ein Klick auf das Stiftsymbol vor dem Datensatz öffnet ein neues Formular, in dem man die bestehenden Inhalte modifizieren kann.

Verändere den Preis des Feldahorns auf 7,20.

Garten erweitern inkl. Verknüpfung von 2 Tabellen:

Erstelle eine neue Tabelle namens „lieferanten“.

Anzahl der Spalten: 5



Folgende Angaben:

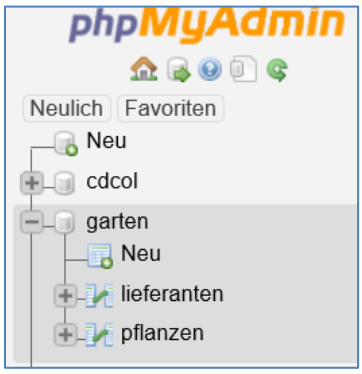
- liefer_id ist vom Typ INT, Attribut UNSIGNED, Index Primary und soll vollautomatisch hochzählen (A_I)
- firma ist vom Typ VARCHAR und hat die Länge 50
- strasse ist vom Typ VARCHAR mit der Länge 50, und NULL ist erlaubt
- plz ist ebenfalls vom Typ VARCHAR, aber mit einer Länge von 5 (auch Deutschland, nicht nur Österreich), und NULL ist erlaubt
- ort ist auch vom Typ VARCHAR mit einer Länge von 50, und NULL ist erlaubt

Daten eintragen:

1. Gärtnerbedarf Müller, Dorfstraße 8, 12345 Dorfen
2. Grünes Allerlei, Stadtstraße 9, 54321 Stadt
3. Grüner Finger, Am Waldrand, 22333 Wald

Das Feld liefer_id wird nicht ausgefüllt.

Jetzt gibt es zwei Tabellen, die „lieferanten“ und die „pflanzen“.

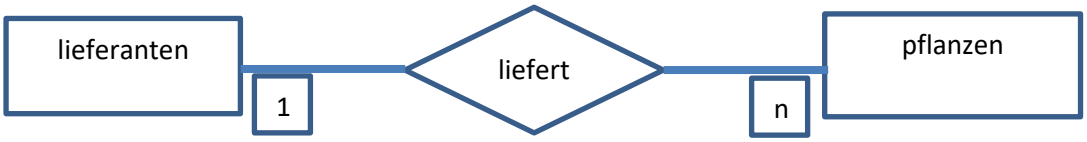


Jedoch fehlt die Verknüpfung zwischen diesen.

Hierfür benötigen wir eine weitere Spalte in der Tabelle „pflanzen“, in die wir die ID des Lieferanten eintragen, der die Pflanzen anbietet.

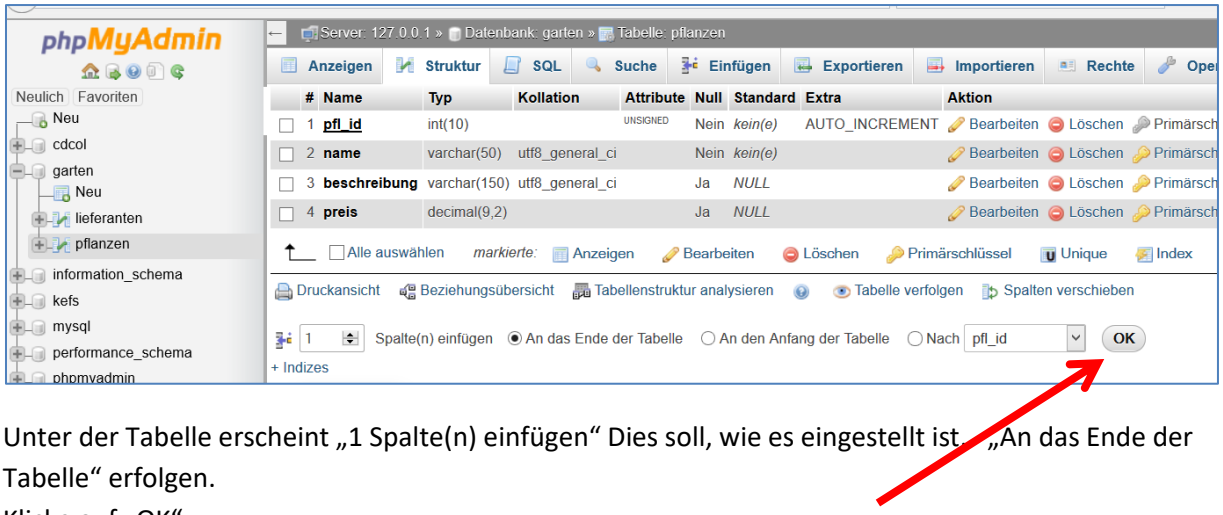
Erinnere dich an das Entity-Relationship-Model:

Eine Pflanze wird immer von genau bei einem Lieferanten bestellt, aber ein Lieferant bringt sehr wohl mehrere Pflanzen. Beziehung daher 1:n



Daher muss man die Einser-Beziehung in die n-Beziehung „einbauen“.

Rufe die Tabelle „pflanzen“ auf und lasse die Struktur anzeigen. Erstelle daher in der Pflanzentabelle eine neue Spalte.

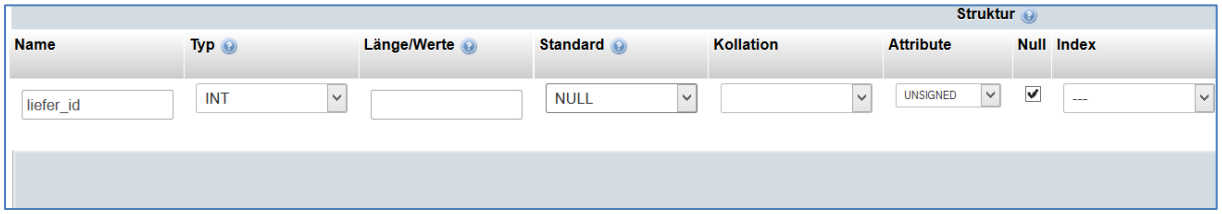


Unter der Tabelle erscheint „1 Spalte(n) einfügen“ Dies soll, wie es eingestellt ist, „An das Ende der Tabelle“ erfolgen.
Klicke auf „OK“.

Eingabe:

Nenne das Feld „liefer_id“. Typ INT, UNSIGNED, und als Standard NULL.

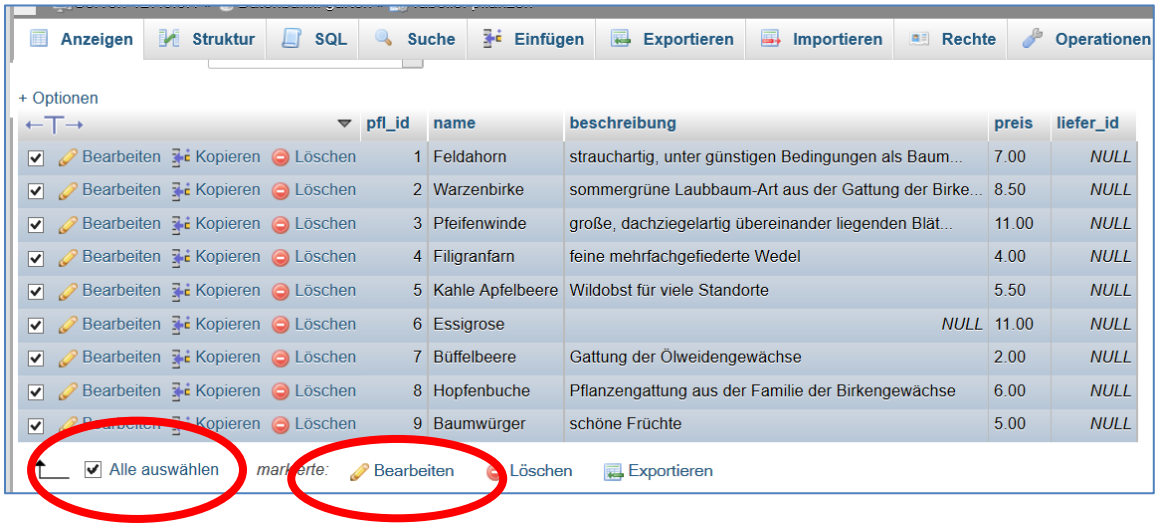
Klicke „Speichern“.



Zuordnung der Lieferanten

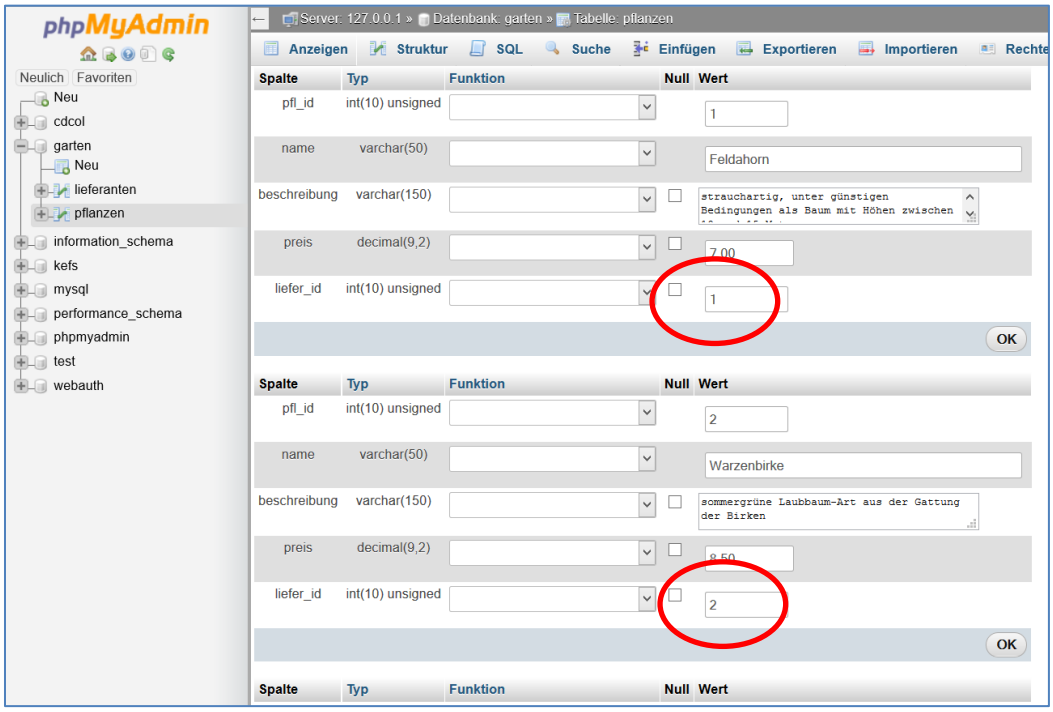
Nun muss in der Tabelle „pflanzen“ noch dieses eben neu erstellte Feld ausgefüllt werden. Jede Pflanze benötigt nun einen Lieferanten, der mithilfe der ID zugeordnet wird. Daher klicke auf den **Reiter „Anzeigen“** und dann sieht man, dass in der „liefer_id“ NULL steht und noch keine Zuordnung vorhanden ist. Das ändert man am besten genau jetzt.

Klicke unten auf das Kästchen bei „**Alle auswählen**“ und dann auf „**Bearbeiten**“ rechts daneben:



Nun können auf einmal alle Lieferanten zugeordnet werden:

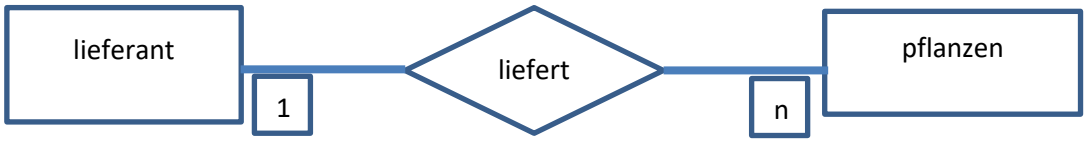
pfl:id:	1	2	3	4	5	6	7	8	9 (Baumwürger)
Liefer_id:	1	2	2	3	3	1	1	3	3



Ganz unten dann „OK“.

Wenn man mit mehreren Tabellen arbeitet, so stehen diese in bestimmten Relationen zueinander. Es gibt drei Typen von Relationen:

- 1:1 bedeutet, dass jeder Datensatz der einen Tabelle genau mit einem Datensatz der anderen Tabelle verknüpft ist. Das ist eher selten.
- 1:n bedeutet, dass ein Datensatz der einen Tabelle mit mehreren Datensätzen der zweiten Tabelle verknüpft ist. Das ist hier oben der Fall. Ein Lieferant kann mehrere Pflanzen liefern, aber jede Pflanze wird nur von genau einem Lieferanten geliefert.
- n:m bedeutet, dass ein Datensatz der einen Tabelle mit mehreren Datensätzen der zweiten Tabelle verknüpft ist um umgekehrt.



Das Feld liefer_id ist der Primärschlüssel in der Tabelle lieferanten und der Fremdschlüssel in der Tabelle pflanzen.

MySQL

Die Abkürzung **SQL steht für Structured Query Language**, was auf Deutsch übersetzt strukturierte Abfragesprache bedeutet. **SQL ist die Standardsprache**, um mit den meisten Datenbanken zu kommunizieren. Somit kann man damit jede Datenbank abfragen, egal ob sie MySQL, Access oder Oracle heißt.

Der große Erfolg von MySQL ist unter anderem auf die Verwendung im Web zurückzuführen. Gebündelt mit PHP wurde MySQL bereits vor vielen Jahren zum Standard für die Programmierung von Web-Applikationen.

MySQL ist eine Open Source Software, was bedeutet, dass jeder diese Software benutzen, aber auch verändern kann und das, ohne etwas dafür zu bezahlen. Mit MySQL-Befehlen kann man beispielsweise

- eine Datenbank anlegen (CREATE DATABASE)
- eine Tabelle mit verschiedenen Feldern in einer Datenbank erstellen (CREATE TABLE)
- Datensätze, die in einer Tabelle stehen anzeigen (SELECT)
- Datensätze einfügen (INSERT), löschen (DELETE) oder ändern (UPDATE)

Datenbanken spielen im täglichen Leben eine große Rolle. Eine MySQL-Datenbank besteht aus einer oder mehreren Tabellen. Diese enthalten wiederum Felder bzw. Spalten, wie z.B. ein Feld mit dem Namen PLZ, in dem die PLZ einer Person gespeichert wird. Somit können alle möglichen Informationen gespeichert werden.

Um Datenbanken administrieren zu können, stellt **XAMPP den phpMyAdmin** zur Verfügung.

- Dabei kann man durch Klicken Datenbanken und Tabellen erstellen kann und Datensätze manipulieren.
- Aber man kann das auch mit entsprechenden SQL-Befehlen tun.

Verwendet man nur das Menü, so wird bei jeder Änderung, die man per Mausklick durchführt, auch der entsprechende SQL-Befehl angezeigt.

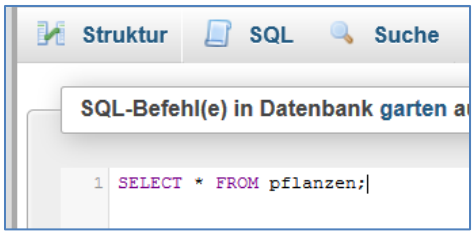
phpMyAdmin ist eine Sammlung von PHP-Skripten, um eine MySQL-Datenbank über das Internet zu administrieren. phpMyAdmin soll das Erstellen und Arbeiten mit MySQL-Datenbanken erleichtern.

Groß- und Kleinschreibung:

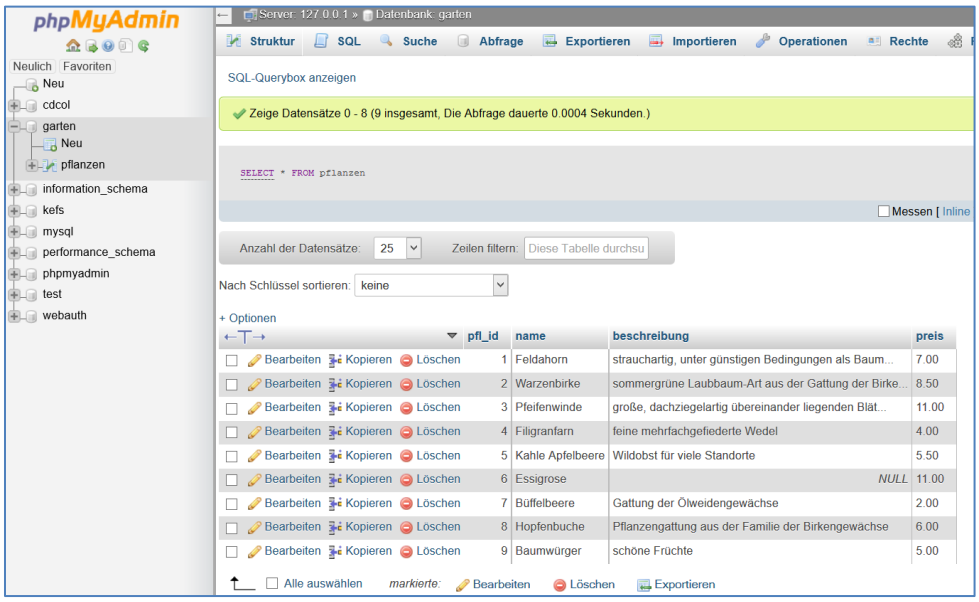
- Datenbanken, Tabellennamen und Variablen kleinschreiben
- MySQL-Befehle in Großbuchstaben – eigentlich könnte man sie auch klein schreiben, aber die Großschreibung dient zur besseren Unterscheidung!

a)Befehle zum Auslesen in SQL

Verwende den Menüpunkt „SQL“. Um alle Daten der Tabelle „pflanzen“ auszulesen, schreibe:

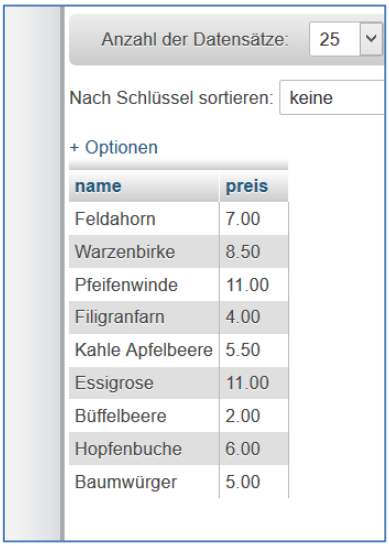


Klicke auf „OK“; Ergebnis:



Wenn man aber nur ausgewählte Spalten sehen will, notiere anstelle des Sternchens z.B. „name, preis“

Klicke auf „OK“. Ergebnis:



ORDER BY

Mit diesem Befehl bestimmt man die Sortierung: aufsteigend

```
1 SELECT name, preis FROM pflanzen ORDER BY preis;
```

Ergebnis:

+ Optionen

name	preis ▲ 1
Büffelbeere	2.00
Filigranfarn	4.00
Baumwürger	5.00
Kahle Apfelbeere	5.50
Hopfenbuche	6.00
Feldahorn	7.00
Warzenbirke	8.50
Pfeifenwinde	11.00
Essigrose	11.00

- Soll gleichzeitig nach einem weiteren Sortierkriterium gesucht werden, so schreibe dieses nach einem Komma:

```
1 SELECT name, preis FROM pflanzen ORDER BY preis, name;
```

- Um die Sortierreihenfolge umzudrehen, schreibe DESC
  LIMIT begrenzt die Anzahl der angezeigten Datensätze, wenn dies gewünscht ist:

```
1 SELECT name, preis FROM pflanzen ORDER BY preis DESC LIMIT 3;
2
```

  Aufsteigend: ASC

- DISTINCT
  Mit SELECT DISTINCT entfernt man Duplikate. Wenn man z.B. in einer Tabelle lieferanten die Adressen gespeichert hat, kann man sich durch folgende Abfrage eine Liste der Orte ausgeben lassen, in denen es Lieferanten gibt. Durch DISTINCT werden Mehrfachnennungen unterdrückt.
  SELECT DISTINCT ort
  FROM lieferanten
  ORDER BY ort;

b)Datensätze auswählen und filtern

- **WHERE**

Die Bedingungen werden hinter WHERE notiert.
Beispiel: Hier erhält man nur Pflanzen, die mehr als 9 kosten:

```
SELECT name, preis FROM pflanzen WHERE preis > 9;
```

Mit dem „**Istgleichzeichen**“ überprüft man auf Übereinstimmung:

```
SELECT * FROM pflanzen WHERE name = "Büffelbeere"
```

Ergebnis:

+ Optionen

	pfl_id	name	beschreibung	preis
Bearbeiten Kopieren Löschen	7	Büffelbeere	Gattung der Ölweidengewächse	2.00

Alle auswählen *markierte:* Bearbeiten Löschen Exportieren

Bei der Übereinstimmung ist die Groß-/Kleinschreibung nicht relevant. Soll etwas nicht übereinstimmen, verwende **<>**.

- **IS NULL**

Eine Besonderheit gibt es bei der Überprüfung, ob in einer Spalte nichts steht. Das ist bei der Essigrose der Fall, denn hier gibt es keine Beschreibung

```
SELECT * FROM pflanzen WHERE beschreibung IS NULL;
```

Im Gegenteil erhält man mit IS NOT NULL alle Datensätze, in denen etwas bei der Beschreibung steht.

Jedoch: Eine Überprüfung mit „beschreibung = NULL“ oder „beschreibung <> NULL“ führt hingegen nicht zum gewünschten Ergebnis.

- **Platzhalter und LIKE**

LIKE: ist sehr nützlich beim Suchen nach Zeichenketten oder Teilen von Zeichenketten. Dabei können auch Platzhalter (Wildcards) eingesetzt werden.

- Ein % (Prozentzeichen) steht für eine beliebige Anzahl unbekannter Zeichen,
- ein _ (Unterstrich) für genau ein unbekanntes Zeichen. Die untersuchte Zeichenkette muss dabei weiterhin in einfache Hochkommata gesetzt werden.

Übung: alle Pflanzen, die im Namen „beere“ haben

SELECT *
FROM pflanzen
WHERE name LIKE “%beere%“;

```
1 SELECT *
2 FROM pflanzen
3 WHERE name LIKE "%beere%";
```

Ergebnis:

+ Optionen

	pfl_id	name	beschreibung	preis
Bearbeiten Kopieren Löschen	5	Kahle Apfelbeere	Wildobst für viele Standorte	5.50
Bearbeiten Kopieren Löschen	7	Büffelbeere	Gattung der Ölweidengewächse	2.00

Alle auswählen *markierte:* Bearbeiten Löschen Exportieren

- **AND oder OR**

Mehrere Bedingungen können durch AND oder OR verknüpft werden.

SELECT *
FROM pflanzen
WHERE name LIKE “%beere%“
OR name LIKE „%ahorn%“;

Ergebnis: Alle Pflanzen mit beere oder ahorn im Namen.

Bei AND müssen beide Bedingungen zutreffen.

SELECT *
FROM pflanzen
WHERE name LIKE “%beere%“
AND preis < 4;

Ergebnis: Name muss beere beinhalten und der Preis muss kleiner als 4 sein. (Lösung: Büffelbeere).

- **IN**

Um Vergleiche mit mehreren Werte durchzuführen, kann man auch IN benutzen. Anstelle von

SELECT *
FROM pflanzen
WHERE pfl_id = 1 OR pfl_id = 2 OR pfl_id = 3;

Kürzer:

```
SELECT *
FROM pflanzen
WHERE pfl_id IN (1,2,3);
```

- **COUNT**

Um zu ermitteln, wie viele Datensätze es sind, benutze COUNT(*)

```
SELECT COUNT(*)
FROM pflanzen;
```

Die Spaltenüberschrift kann man mit AS gefolgt vom neuen Namen ändern

```
SELECT COUNT(*) AS gesamt
FROM pflanzen;
```

Ergebnis:

+ Optionen

gesamt
9

AS zur Angabe eines anderen Spaltennamens funktioniert auch bei allen Spaltennamen. Man kann es auch benutzen, um einen Alias bei einer Tabelle anzugeben.

- **MAX() MIN() AVG()**

Mit MAX() kann man z.B. den höchsten Preis einer Pflanze ermitteln. Mit MIN() den niedrigsten und mit AVG() den durchschnittlichen Preis einer Pflanze.

SELECT MIN(preis) AS Mindestpreis, MAX(preis) AS TeuesterPreis, AVG(preis) AS Mittelwert FROM pflanzen;

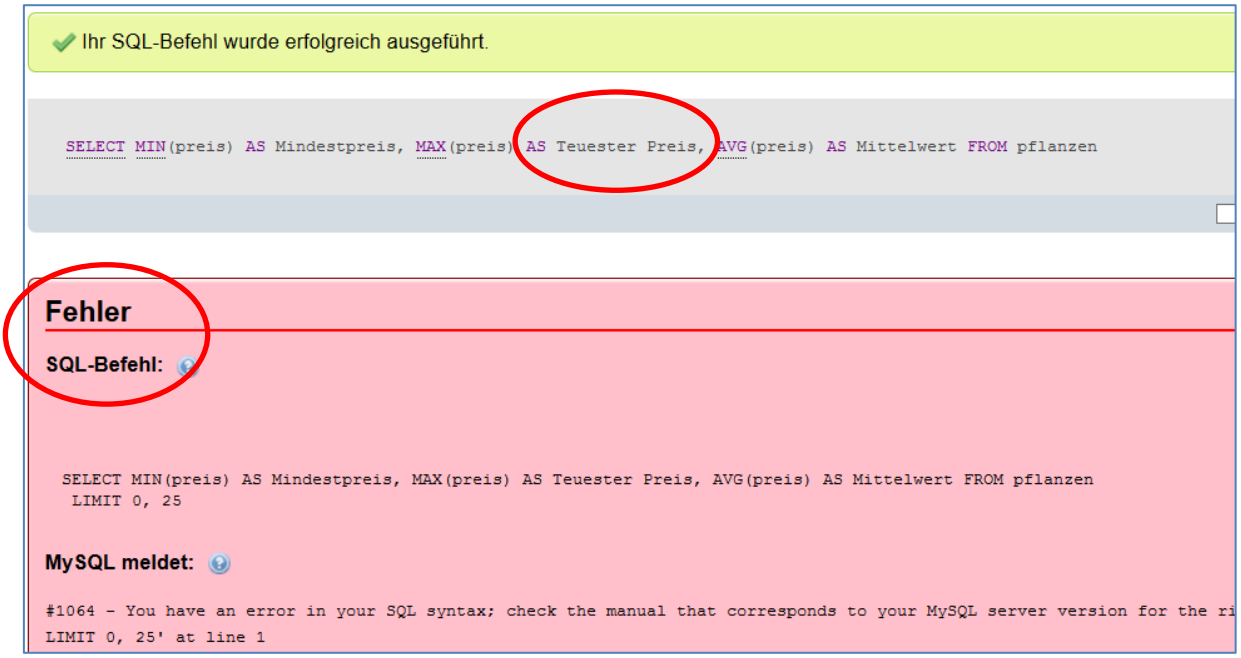
Ergebnis:

+ Optionen

Mindestpreis	TeuesterPreis	Mittelwert
2.00	11.00	6.666667

Beachte bei AS: folgt eine Bezeichnung, muss dies ein Wort sein, ohne Leerzeichen!!!

Fehlermeldung: wenn Teuester Preis (mit Leerzeichen) statt TeuesterPreis geschrieben wurde:



- **Taschenrechner**

Man kann MySQL auch als Taschenrechner benutzen.

SELECT 20 * 5;

Gibt 100 aus.

Übungen:

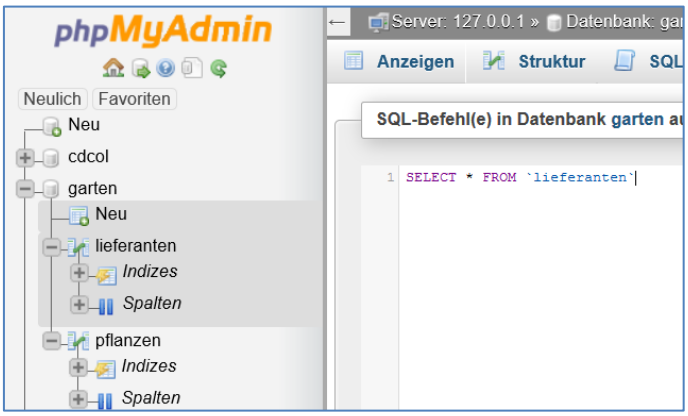
1. Gib alle Namen der Pflanzen aus, die günstiger als 6 sind.

2. Gib alle Pflanzennamen aus, die im Namen ein „rose“ haben.

Übungen mit der Tabelle „lieferanten“:

1. Lass dir alle Daten der Tabelle zeigen: Klicke auf die Tabelle „lieferanten“ und gib den SQL Befehl ein:

   SELECT * FROM lieferanten;



   Ergebnis:

2. Lass dir alle Lieferanten anzeigen, die in „Wald“ wohnen.

Datentypen

Bei der Definition eines Feldes können verschiedene Datentypen angegeben oder im PHPMyAdmin gewählt werden.

PHP selbst erkennt automatisch, ob es sich um eine Ganzzahl, eine Kommazahl oder um Text handelt. Da braucht man sich um Datentypen **NICHT zu kümmern**.
Ganz anders ist dies bei **Datenbanken**! Hier spielen die Datentypen eine enorm wichtige Rolle. Bei Datenbanken mit mehreren tausend Einträgen geht es schließlich um die **optimale Ausnutzung des Speicherplatzes.**

a)Zeichenketten:

Zu den Zeichentypen gehören CHAR, VARCHAR und TEXT.

- **CHAR(n)** ist eine Zeichenkette mit fester Länge, wobei n für die Anzahl der Zeichen steht. Die Zahl n darf maximal **255** Zeichen beinhalten. Es wird dann genau der Speicherplatz für n Zeichen reserviert, egal wie viel wirklich benötigt wird. VARCHAR(n) reserviert hier einen Variablen Speicherplatz mit bis zu n Zeichen. Die maximal erlaubte Länge wird durch „n“ gekennzeichnet, „n“ darf nicht weggelassen werden.

- **TEXT**
  Für längere Texte der Typ **TEXT** zur Verfügung. Diese kann eine Länge von 2^16-1 Zeichen speichern. Man kann auch TINYTEXT (255 Zeichen), den MEDIUMTEXT (2^24-1 Zeichen) oder LONGTEXT (2^32-1 Zeichen) verwenden. Zu den binären Zeichentypen gehören BLOB, TINYBLOB, MEDIUMBLOB und LONGBLOB. Die maximale Länge entspricht der des Typs TEXT. In Blob-Feldern können auch Dateien wie Bilder gespeichert werden. Im Gegensatz zu **VARCHAR**, die führende Leerzeichen löscht, werden bei TEXT – und BLOB – Datentypen führende Leerzeichen gespeichert.

- **VARCHAR(n)**
  Löscht führende Nullen. Zeichenkette mit variabler Länge. Die maximal erlaubte Länge wird durch „n“ gekennzeichnet, „n“ darf nicht weggelassen werden.

Numerische Datentypen:

MySQL unterstützt alle numerischen Datentypen mit fester Länge wie NUMERIC, DECIMAL, INTEGER (INT) und SMALLINT sowie die Fließkommastellen FLOAT, REAL und DOUBLE.

Darüber hinaus sind die Typen TINYINT (1 Byte), MEDIUMINT (3 Byte) und BIGINT (8 Byte) möglich.

- **INTEGER**
  Werte sind Ganzzahlen mit einer Länge von 32 Bit (4 Byte). Die zulässigen Werte für den Datentyp INTEGER sind von -2147483648 bis +2147483648. INTEGER UNSIGNED ist für nur positive ganze Zahlen möglich.

- **Kommazahlen:**
  Für die Zahlen mit Kommastellen stehen uns in MySQL zwei Datentypen zu Verfügung. Nämlich **FLOAT** und **DOUBLE**. Der Unterschied zwischen diesen beiden Datentypen ist ihre Größe. Während

  - FLOAT eine einfache Genauigkeit bei einer Größe von 32 Bit (4 Byte) aufweist, kann

  - DOUBLE eine doppelte Genauigkeit bei einer Größe von 64 Bit (8 Byte) aufweisen.
    Bei der Definition gibt man mit DOUBLE(v.n) durch v die Vorkommastelen und durch n die Nachkommastellen an.

Das **Dezimaltrennzeichen ist im Übrigen immer ein Punkt.**

Datums – und Zeit – Datentypen:

In MySQL gibt es zwei Datumsformate

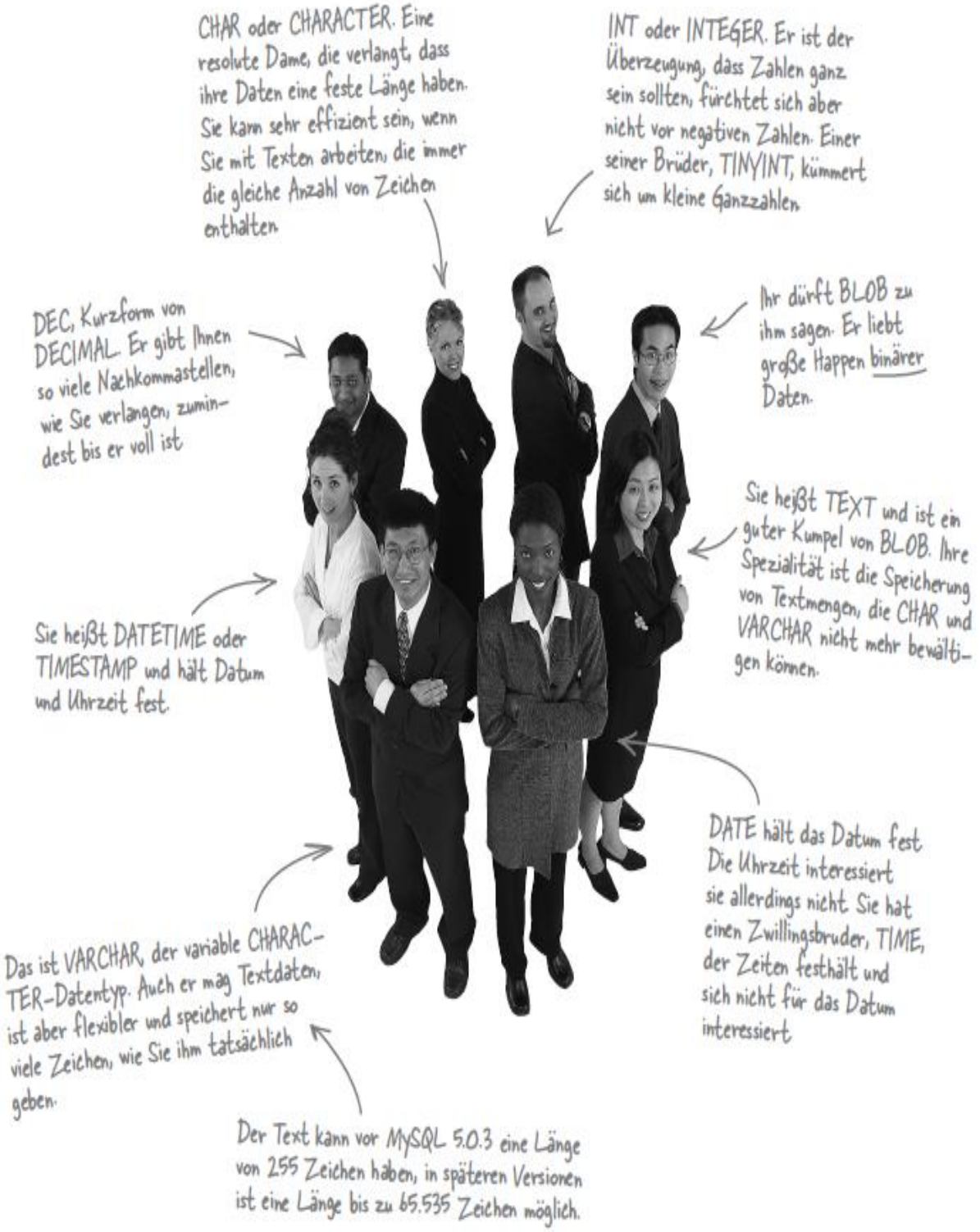
- DATETIME und
- DATE.

Im Datentyp DATETIME werden die Werte für das Datum im Format JJJJ-MM-DD und die Zeit im Format SS:MM:SS gespeichert. Der Datentyp DATE wird nur für das Datum benutzt.

Außerdem steht noch **TIMESTAMP** zur Verfügung. Wird ein neuer Datensatz eingefügt, oder ein alter geändert, dann wird in dieses Feld automatisch das aktuelle Datum mit Uhrzeit geschrieben, in einer 14-stellig in der Form JJJJMMDDHHMMSS. Z.B. 20041208125511 wäre dann der 8. Dezember 2004, 12 Uhr 55 Minuten und 11 Sekunden.

Für Zeitangaben steht noch der Type TME zur Verfügung. Hier wird das Datum in der Form HH:MM:SS gespeichert.

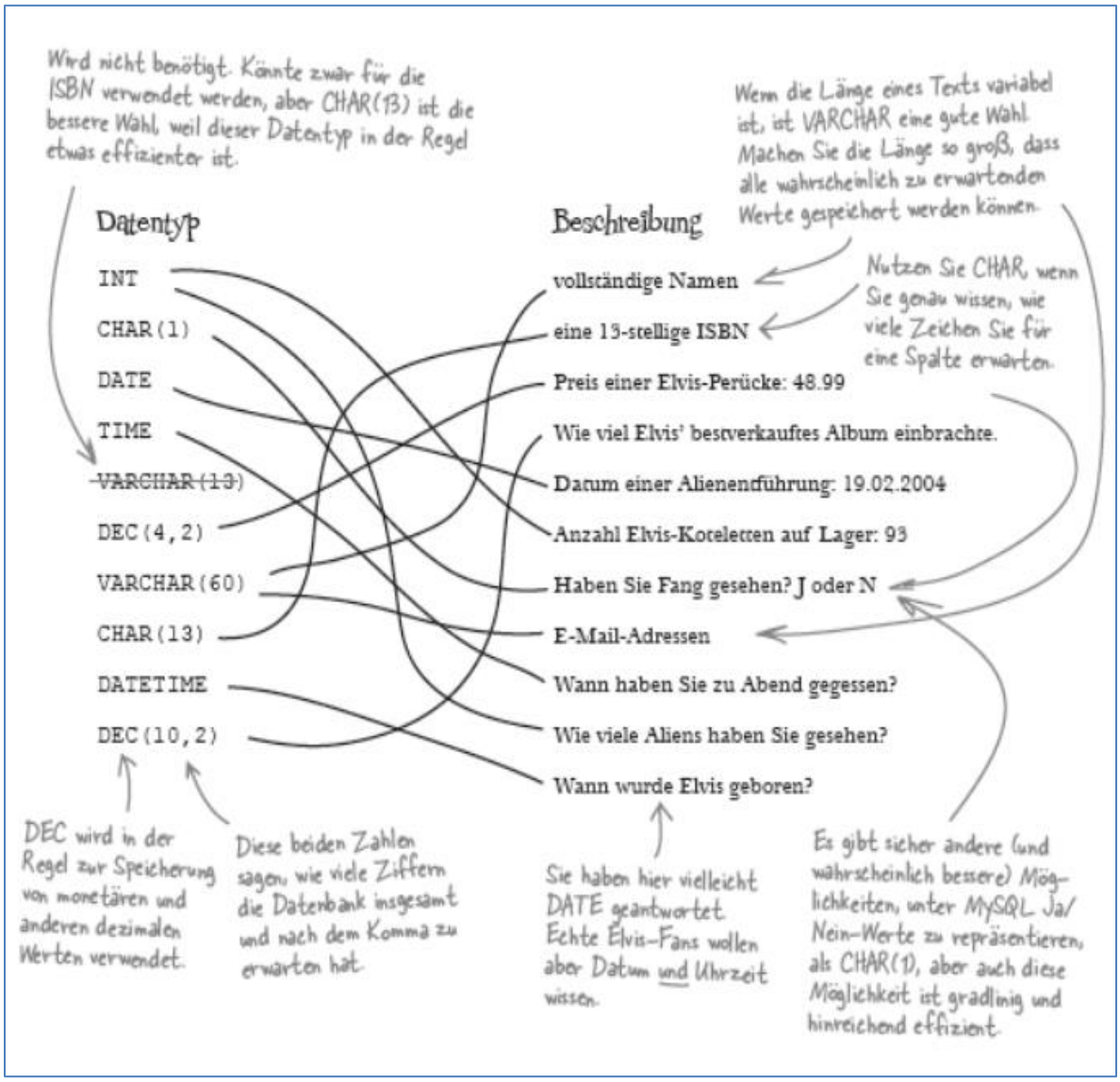
Übersicht Datentypen:



Übung:
Ordne die folgenden Datentypen der passenden Beschreibung zu, die in einer Tabellenspalte gespeichert werden soll.

Datentyp	Beschreibung
INT	vollständige Namen
CHAR(1)	eine 13-stellige ISBN
DATE	Preis einer Elvis-Perücke: 48.99
TIME	Wie viel Elvis' bestverkauftes Album einbrachte.
VARCHAR(13)	Datum einer Alienentführung: 19.02.2004
DEC(4,2)	Anzahl Elvis-Koteletten auf Lager: 93
VARCHAR(60)	Haben Sie Fang gesehen? J oder N
CHAR(13)	E-Mail-Adressen
DATETIME	Wann haben Sie zu Abend gegessen?
DEC(10,2)	Wie viele Aliens haben Sie gesehen?
	Wann wurde Elvis geboren?

Lösung:



Quelle:

Lynn Beighley und Michael Morrison in: PHP & MySQL von Kopf bis Fuß;
Verlag =O´Reilly, 2009