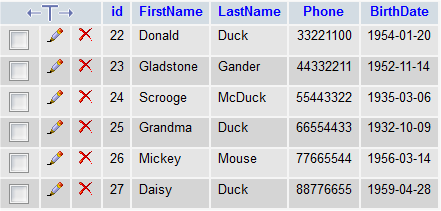
**Lektion 17: Datenbanken**

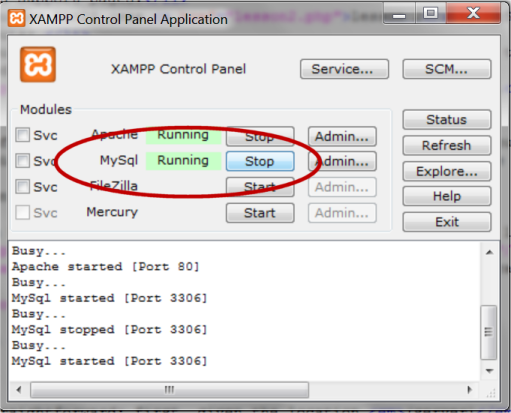
Eine Datenbank ist eine Sammlung von Informationen/Daten, die so organisiert sind, dass sie leicht wiedergefunden, administriert und aktualisiert werden können. Datenbanken bieten außerdem die Möglichkeit, dynamische Webseiten mit einer großen Menge an Informationen zu erstellen. Z.B. alle Mitgliedsdaten von HTML.net und alle Beiträge in den Foren sind in Datenbanken gespeichert.

Eine Datenbank besteht normalerweise aus einer oder mehr Tabellen. Wenn Sie gewohnt sind, mit Tabellenkalkulationsprogrammen zu arbeiten oder bereits mit Datenbanken zu tun hatten, sollten Ihnen Tabellen mit ihren Spalten und Zeilen bekannt vorkommen:



Es gibt viele verschiedene Datenbanken (Datenbanksysteme): MySQL, MS Access, MS SQL Server, Oracle SQL Server, um nur einige zu nennen. In diesem Tutorial werden wir mit MySQL arbeiten. MySQL ist ein guter Startpunkt, wenn Sie Datenbanken mit PHP nutzen wollen.

Sie brauchen Zugang zu einem MySQL-Server, um diese und die nächsten Lektionen bearbeiten zu können:

* Wenn Sie Ihre PHP-Seite bei einem Webhoster haben, ist MySQL vielleicht sogar bereits auf dem Server installiert. Bitte informieren Sie sich auf den Support-Seiten Ihres Hosters.
* Wenn Sie PHP selbst auf Ihrem Computer installiert haben und Sie den Mut haben, dies auch mit MySQL zu probieren, können Sie es in einer kostenlosen Version (MySQL Community Edition) auf der [MySQL Website](http://www.mysql.com/downloads/mysql/) herunterladen.
* Wenn Sie XAMPP (siehe [Lektion 2](http://de.html.net/tutorials/php/lesson2.php)) benutzen, ist MySQL bereits auf Ihrem Rechner installiert und fertig zur Benutzung. Stellen Sie sicher, dass MySQL aktiviert ist (Kontrollbildschirm):  
    
  

Im Rest dieser Lektion, werden wir uns genauer anschauen, wie man sich mit einer Datenbank auf dem Datenbankserver verbinden, bevor wir uns in den folgenden Lektionen mit dem Erstellen von Datenbanken und dem aktualisieren von Daten befassen.

**Verbindung zu einem Datenbankserver**

Zu allererst brauchen Sie einen Zugang zu dem Server, auf dem sich Ihre MySQL-Datenbank befindet. Dies wird mit der Function [[Dokumentation](http://php.net/manual/en/function.mysql-connect.php)mysql\_connect](http://php.net/manual/en/function.mysql-connect.php) mit der folgenden Syntax bewerkstelligt:

mysql\_connect*(Server, Benutzername, Passwort)*

Ziemlich überschaubar: Zuerst die Ortsangabe der Datenbank (die Adresse des *Server*s), dann der *Benutzername* und zuletzt das *Passwort*.

Wenn Sie Ihre eigene Webseite haben, sollten Sie sich auf den Supportseiten Ihres Hosters über die Adresse des MySQL-Servers informieren. Oftmals sind Benutzername und Passwort die gleichen, wie für den FTP-Zugang. Falls nicht, fragen Sie bitte bei Ihrem Anbieter nach.

Beispiel für eine MySQL-Verbindung auf einer gehosteten Webseite:

mysql\_connect("mysql.meinserver.de", "benutzer001", "m31np455w0rt") or die(mysql\_error());

Beispiel für eine MySQL-Verbindung mit XAMPP (Standardeinstellung):

mysql\_connect("localhost", "root", "") or die (mysql\_error());

Den Beispielen wurde der folgende Code hinzugefügt: or die(mysql\_error()). Dieser bricht, einfach gesagt, das Skript ab und gibt einen Fehler aus, wenn die Verbindung zum Server fehlschlägt.

Jetzt haben wir eine Verbindung zu einem MySQL-Server hergestellt und können jetzt beginnen, Datenbanken zu erstellen und Daten abfragen oder einfügen. Genau das werden wir in der nächsten Lektion machen.

Übrigens, bitte merken Sie sich, dass es (zwar nicht zwingend notwendig, aber dennoch) sinnvoll ist, die Verbindung zur Datenbank, bzw. zum Datenbankserver, zu trennen, wenn Sie Ihre Daten abgefragt oder aktualisiert haben. Dies macht man mit der Function [[Dokumentation](http://php.net/manual/en/function.mysql-close.php)mysql\_close](http://php.net/manual/en/function.mysql-close.php).