

## Übung 1: Ein Website News-System mit MySQL

In der Vorübung haben wir bereits mit Hilfe eines ERM's den Datenbankentwurf erstellt und daraus die folgenden Tabellen abgeleitet:

**Tabelle User**

ID_user	Name	Passwort	Email
---------	------	----------	-------

**Tabelle News**

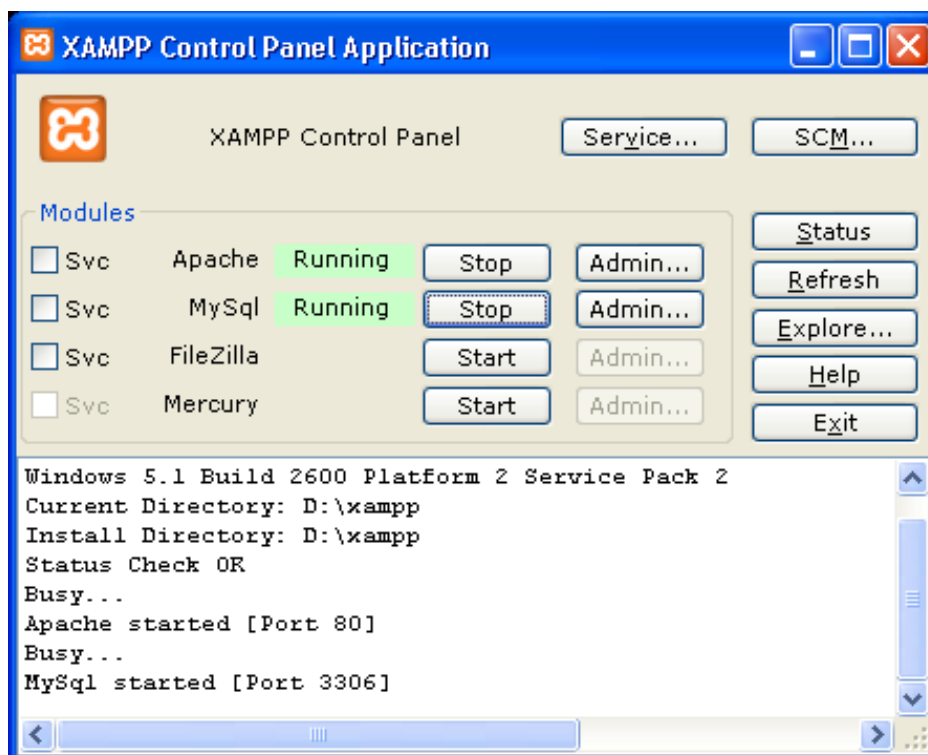
ID_news	Titel	Newstext	Datum	ID_user
---------	-------	----------	-------	---------

Nun muss diese Datenbank in MySQL erstellt werden, die beiden Tabellen müssen angelegt werden und wir werden einige Beispiel-Datensätze einfügen, auslesen, bearbeiten und löschen.

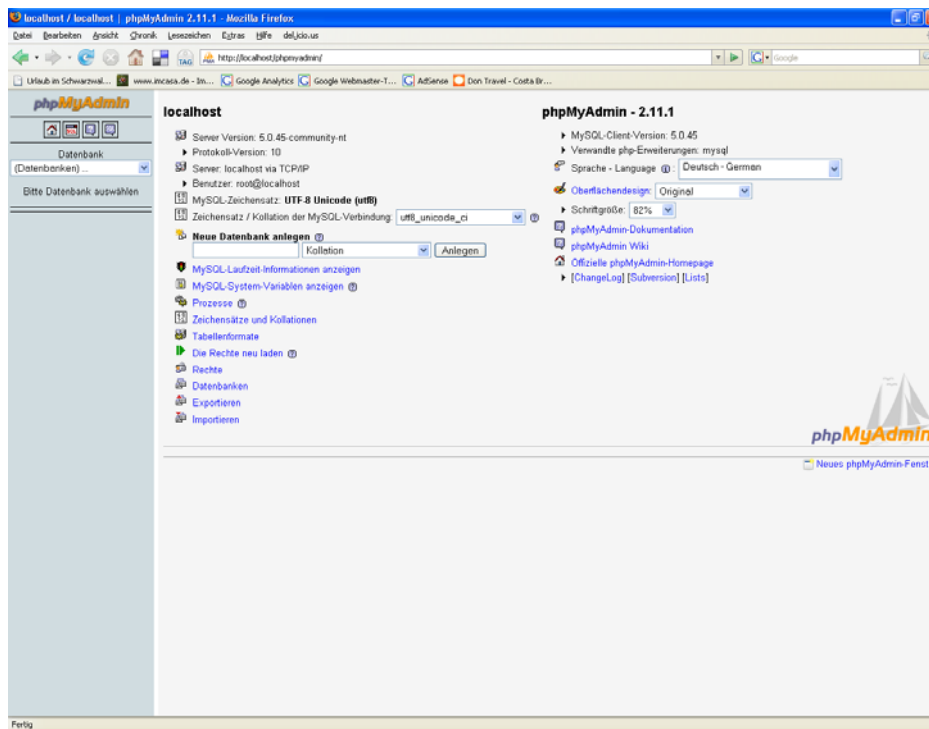
Als Entwicklungsumgebung verwenden wir XAMPP.

### Schritt-für-Schritt-Anleitung:

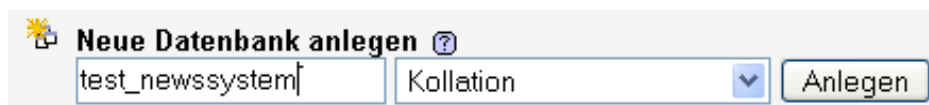
1. Damit alles funktioniert, müssen der **Apache-Server** und **MySQL** laufen. Starten Sie dazu das **XAMPP Control Panel** und klicken Sie auf die entsprechenden Schaltflächen zum starten.



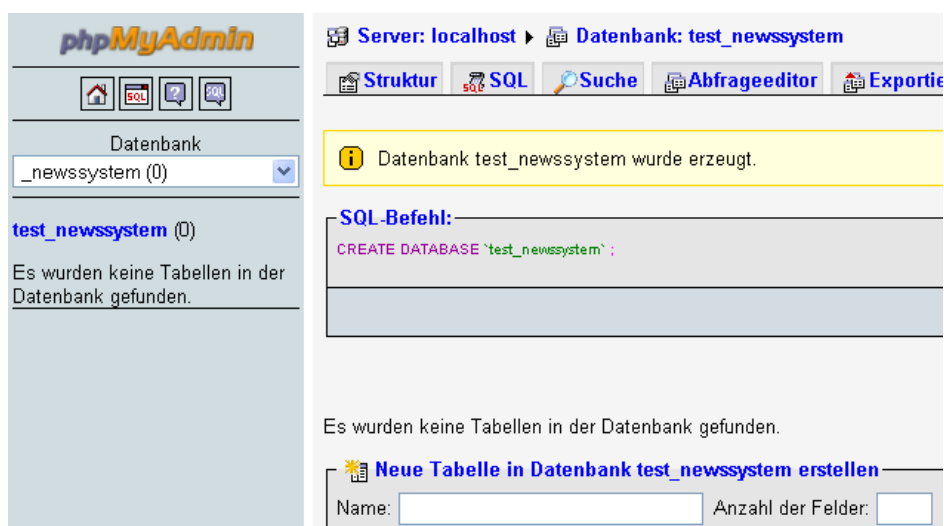
2. Öffnen Sie nun einen Webbrowser, bspw. Firefox, und geben Sie als Adresse **localhost** ein. Sie sollten nun die Begrüßungsseite von XAMPP sehen.
3. Wählen Sie auf der linken Seite den Link *phpMyAdmin* in der Kategorie *Tools* aus. Sie sollten sich nun auf der Startseite von phpMyAdmin befinden. Dieses Programm stellt eine komfortable Weboberfläche zur Verwaltung von MySQL-Datenbanken bereit.



4. Zuerst werden wir eine neue Datenbank für unser News-System anlegen. Tragen Sie dazu den Namen der Datenbank in das Feld *Neue Datenbank anlegen* ein und klicken Sie auf die Schaltfläche *Anlegen*:



5. Wenn alles geklappt hat verändert sich einiges in phpMyAdmin:



Auf der linken Seite ist unter Datenbank nun bereits unsere neu angelegte Datenbank *test\_newssystem* ausgewählt. Dies erkennt man auch an der Pfadangabe rechts oben. Im mittleren Bereich der Seite weist eine Meldung, erkennbar am Ausrufezeichen, auf den soeben durchgeführten Befehl hin. Darunter sieht man auch den verwendeten SQL-Befehl:

```
CREATE DATABASE `test_newssystem` ;
```

6. An mehreren Stellen erscheint auch die Meldung, dass die Datenbank noch keine Tabellen enthält. Die sollen jetzt angelegt werden. Für die erste Tabelle können wir auch wieder die grafische Benutzeroberfläche von phpMyAdmin verwenden. Die zweite Tabelle legen wir dann direkt über einen SQL-Befehl an, da dies für unsere Übung sinnvoller ist. Tragen Sie den Namen der ersten Tabelle und die Anzahl der Felder (Spalten) in dieser Tabelle in die entsprechenden Felder ein und klicken Sie dann auf *OK*:

Neue Tabelle in Datenbank test\_newssystem erstellen

Name:  Anzahl der Felder:

Darauf hin erscheint ein Formular, mit dessen Hilfe die neue Tabelle genau definiert werden kann:

Server: localhost ▶ Datenbank: test\_newssystem ▶ Tabelle: user

Feld	Typ	Länge/Set <sup>1</sup>	Kollation	Attribute	Null	Standard <sup>2</sup>	Extra
<input type="text"/>	VARCHAR	<input type="text"/>	<input type="text"/>	<input type="text"/>	not null	<input type="text"/>	<input type="text"/>
<input type="text"/>	VARCHAR	<input type="text"/>	<input type="text"/>	<input type="text"/>	not null	<input type="text"/>	<input type="text"/>
<input type="text"/>	VARCHAR	<input type="text"/>	<input type="text"/>	<input type="text"/>	not null	<input type="text"/>	<input type="text"/>
<input type="text"/>	VARCHAR	<input type="text"/>	<input type="text"/>	<input type="text"/>	not null	<input type="text"/>	<input type="text"/>

Tabellenkommentar:  Tabellenformat: MyISAM Kollation:

oder

<sup>1</sup> Wenn das Feld vom Typ 'ENUM' oder 'SET' ist, benutzen Sie bitte das Format: 'a','b','c',... Wann immer Sie ein Backslash ("\") oder ein einfaches Anführungszeichen (") verwenden, setzen Sie bitte ein Backslash vor das Zeichen. (z. B.: 'xyz' oder 'a\b').

<sup>2</sup> Bitte geben Sie jeweils nur einen Standardwert ohne Escape- oder Anführungszeichen an.

7. Ganz links, in der Spalte Feld, gibt man den Namen des Feldes an. In der Spalte Typ muss der Datentyp ausgewählt werden, von dem dann auch die Länge/Set abhängig ist. Diese 3 Einstellungen müssen mindestens gesetzt werden, damit die Tabelle angelegt werden kann. Geben Sie in das Formular die folgenden Daten ein und klicken Sie auf *Speichern*:

Server: localhost ▶ Datenbank: test\_newssystem ▶ Tabelle: user

Feld	Typ	Länge/Set <sup>1</sup>	Kollation
ID_user	INT	<input type="text"/>	<input type="text"/>
Name	VARCHAR	255	<input type="text"/>
Pw	VARCHAR	255	<input type="text"/>
Email	VARCHAR	255	<input type="text"/>

8. Das Resultat ist wieder eine Erfolgsmeldung von phpMyAdmin, dass die neue Tabelle *user* angelegt wurde. Außerdem wurde diese Tabelle auch gleich ausgewählt, d.h. wir als Benutzer von phpMyAdmin befinden uns jetzt in der Verwaltungsansicht für diese Tabelle.

Server: localhost ▶ Datenbank: test\_newssystem ▶ Tabelle: user





























Anzeigen Struktur SQL Suche Einfügen Exportieren Importieren Operationen Leeren Löschen








! Tabelle 'test\_newssystem`.`user` wurde erzeugt.

SQL-Befehl:

```
CREATE TABLE `test_newssystem`.`user` (
  `ID_user` INT NOT NULL ,
  `Name` VARCHAR( 255 ) NOT NULL ,
  `PW` VARCHAR( 255 ) NOT NULL ,
  `Email` VARCHAR( 255 ) NOT NULL
) ENGINE = MYISAM
```


Profiling [ Bearbeiten ] [ PHP-Code erzeugen ]

Field	Typ	Kollation	Attribute	Null	Standard	Extra	Aktion
<input type="checkbox"/> ID_user	int(11)			Nein			      
<input type="checkbox"/> Name	varchar(255)	latin1_general_ci		Nein			      
<input type="checkbox"/> PW	varchar(255)	latin1_general_ci		Nein			      
<input type="checkbox"/> Email	varchar(255)	latin1_general_ci		Nein			      

Alle auswählen / Auswahl entfernen markierte:       

Druckansicht Tabellenstruktur analysieren

1 Felder hinzufügen  An das Ende der Tabelle  An den Anfang der Tabelle  Nach ID\_user


Indizes:  Kein Index definiert!


Speicherplatzverbrauch		Zeilenstatistik	
Typ	Verbrauch	Angaben	Wert
Daten	0 Bytes		
Index	0 Bytes		
Insgesamt	0 Bytes		

Index über 1 Spalten anlegen

Hier sieht man u.a. auch wieder den SQL-Befehl, der verwendet wurde um die Tabelle user anzulegen:

```
CREATE TABLE `test_newssystem`.`user` (
  `ID_user` INT NOT NULL ,
  `Name` VARCHAR( 255 ) NOT NULL ,
  `PW` VARCHAR( 255 ) NOT NULL ,
  `Email` VARCHAR( 255 ) NOT NULL
) ENGINE = MYISAM;
```

Unter der Überschrift Indizes befindet sich auch eine Warnmeldung, die darauf hinweist, dass noch kein Index definiert wurde. Dieses Problem lässt sich leicht beheben, indem man einen Primärschlüssel für diese Tabelle definiert. Sobald man mit einem Klick auf das Schlüsselsymbol  für das Feld *ID\_user* den Primärschlüssel setzt, verschwindet diese Fehlermeldung.

Nun lässt sich diese Tabelle noch etwas verbessern, indem man festlegt, dass für das Feld ID\_user automatische Werte vergeben werden. Sobald ein neuer User in die Tabelle eingetragen wird, erhält dieser User die nächste freie ID. Dazu muss das Feld ID\_user nochmals bearbeitet werden, mit einem Klick auf das Bleistiftsymbol . Wählen Sie auf der folgenden Seite unter *Extra* die Eigenschaft *auto\_increment* aus und klicken Sie auf *Speichern*.

Server: localhost ▶ Datenbank: test\_newssystem ▶ Tabelle: user

Feld: ID\_user

Typ: INT

Länge/Set<sup>1</sup>: 11

Kollation:

Attribute:

Null: not null

Standard<sup>2</sup>:

Extra: auto\_increment

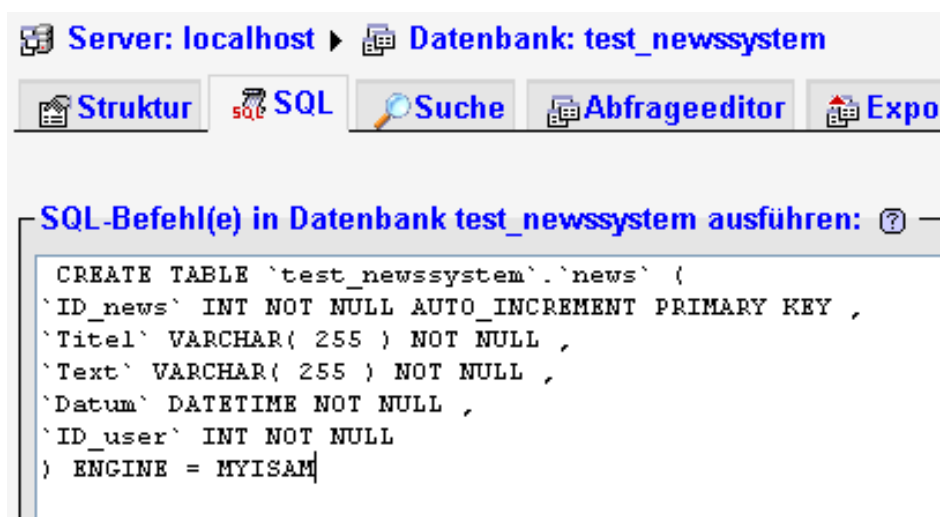
Kommentare: auto\_increment

9. Nun folgt die zweite Tabelle, der wir den Namen **news** geben werden. Diese Tabelle wollen wir nun direkt über einen SQL-Befehl anlegen, den wir in phpMyAdmin eingeben werden. Außerdem soll diese Tabelle nun komplett richtig angelegt werden, damit wir nicht wieder im Nachhinein den Primärschlüssel vergeben müssen oder für das Feld ID\_news die Eigenschaft auto\_increment von Hand einstellen müssen.

Der folgende SQL-Befehl erfüllt alle diese Wünsche:

```
CREATE TABLE `test_newssystem`.`news` (
  `ID_news` INT NOT NULL AUTO_INCREMENT PRIMARY KEY ,
  `Titel` VARCHAR( 255 ) NOT NULL ,
  `Newstext` VARCHAR( 255 ) NOT NULL ,
  `Datum` DATETIME NOT NULL ,
  `ID_user` INT NOT NULL
)
```

Diesen Befehl können Sie in phpMyAdmin eingeben, wenn Sie auf die Registerkarte SQL klicken:



10. Nun da beide Tabellen erstellt worden sind, ist die Datenbank bereit für die ersten Dateneingaben. Wir werden erst einen Benutzer in die Tabelle **user** über die grafische Benutzeroberfläche von phpMyAdmin eintragen und anschließend einen Newsbeitrag verfassen. Die News werden wir jedoch wieder direkt über einen SQL-Befehl in die Tabelle **news** eintragen.

Um einen Benutzer über die Weboberfläche von phpMyAdmin einzugeben, muss zuerst die Tabelle **user** mit einem Klick auf den Tabellennamen im Navigationsbereich auf der linken Seite ausgewählt werden. An der Pfadangabe im oberen Bereich des Bildschirms erkennt man ganz genau, wo man sich gerade innerhalb der Datenbank befindet.



11. Nach einem Klick auf die Registerkarte **Einfügen** erscheint ein oder mehrere Formulare zur Eingabe von Daten. Tragen Sie den folgenden Benutzer in die Tabelle **user** ein: Name = tester, PW = test, Email = test@test.com.

Server: localhost ▶ Datenbank: test\_newssystem ▶ Tabelle: user

Anzeigen Struktur SQL Suche Einfügen Exportieren Importieren Opt

Feld	Typ	Funktion	Null	Wert
ID_user	int(11)			
Name	varchar(255)			tester
PW	varchar(255)			test
Email	varchar(255)			test@test.com

OK

Ignorieren

Feld	Typ	Funktion	Null	Wert
ID_user	int(11)			
Name	varchar(255)			
PW	varchar(255)			
Email	varchar(255)			

OK



Als neuen Datensatz speichern und dann zurück



OK Zurücksetzen

Restart insertion with 2 rows

Das Feld **ID\_user** lassen wir leer, da wir diesem Feld ja die Eigenschaft `auto_increment` zugewiesen haben, d.h. MySQL wird dort automatisch einen Wert eintragen.

12. Wenn man nach dem Speichern auf die Registerkarte **Anzeigen** klickt, sieht man den ersten Datensatz in der Tabelle **user**. Als **ID\_user** sollte diesem Datensatz automatisch die Ziffer 1 zugewiesen worden sein.

	ID_user	Name	PW	Email
<input type="checkbox"/>  	1	tester	test	test@test.com

Alle auswählen / Auswahl entfernen markierte:   

13. Nun verfassen wir, als der Benutzer **tester**, unseren ersten News-Beitrag. Um den Beitrag in die Tabelle **news** einzutragen, werden wir wieder den SQL-Befehl dafür benutzen. Wählen Sie zuerst die Tabelle **news** aus und klicken Sie dann wieder auf die Registerkarte **SQL**. Geben Sie dort den folgenden SQL-Befehl ein:

```
INSERT INTO `test_newssystem`.`news` (
  `Titel` ,
  `Newstext` ,
  `Datum` ,
  `ID_user`
)
VALUES (
  'Test-News', 'Das ist nur ein Test der Datenbank.', '2007-12-03 00:00:00' , '1'
);
```

14. Nun wurde ein vollständiger Datensatz in der Datenbank test\_newssystem erstellt. Diese Daten sollen nun ausgelesen werden, so wie es bspw. das Skript einer Website machen würde, um die News anzuzeigen. Schauen wir uns erst einmal an, wie phpMyAdmin das macht.

Wählen Sie wieder die Tabelle **news** aus und klicken Sie dann auf die Registerkarte **Anzeigen**. Zu sehen sind einerseits alle Zeilen dieser Tabelle, aber auch der verwendete SQL-Befehl ist hier sichtbar:

```
SELECT *
FROM `news`
LIMIT 0 , 30;
```

Das Platzhalterzeichen **\*** wählt alle Felder der angegebenen Tabelle aus und der Befehl **LIMIT** begrenzt die angezeigten Datensätze auf 30 pro Seite. In dieser Abfrage werden noch keine Filter verwendet, um die Resultate auf bestimmte Datensätze zu begrenzen. Diese Abfrage liefert stets alle Datensätze aus der entsprechenden Tabelle.

15. Um eine Abfrage auf bestimmte Datensätze zu beschränken verwendet man eine oder mehrere **WHERE**-Bedingungen in der **SELECT**-Anweisung. Um bspw. nur die Beiträge zu sehen, die der Benutzer tester verfasst hat, könnte man folgenden SQL-Befehl verwenden:

```
SELECT *
FROM `news`
WHERE `ID_user` = 1
LIMIT 0 , 30;
```

An dieser Abfrage wird bereits deutlich, welches Problem in relationalen Datenbanken mit einfachen **SELECT**-Anweisungen besteht, die nur auf eine Tabelle zugreifen. Wenn die Daten aus der Tabelle **news** ausgegeben werden, erscheint im Feld **ID\_user** nur eine Zahl. Um diese Zahl einem Benutzer zuzuordnen, bedarf es eigentlich noch einer zweiten **SELECT**-Anweisung, die den Namen des Benutzers aus der Tabelle **user** ausliest und auch den richtigen Namen anzeigt, also den Inhalt des Feldes **Name** aus der Tabelle **user**. Mit Hilfe der Anweisung **JOIN** kann man sich diese zweite **SELECT**-Abfrage jedoch sparen und gleich beim ersten Mal sämtliche Daten auslesen.

Der folgende Befehl sucht alle Datensätze aus der Tabelle **news**, die vom Benutzer mit der **ID\_user = 1** verfasst worden sind. Zusätzlich werden aus der Tabelle **user** auch sämtliche Daten ausgelesen, die mit den ausgewählten Datensätzen in Beziehung stehen, d.h. die beiden Tabellen **news** und **user** werden über diese Abfrage verknüpft:

```
SELECT *
FROM news
INNER JOIN user ON news.ID_user = user.ID_user
WHERE news.ID_user = 1
LIMIT 0 , 30;
```

Das Ergebnis einer Auswahlabfrage mit **SELECT** nennt sich **Sicht** oder **view**. Es entsteht eine neue, virtuelle Tabelle, die sämtliche abgefragten Daten enthält. Diese virtuelle Tabelle wird in dem Moment erstellt, in dem die Abfrage ausgeführt wird. Wenn sich die Daten in der Datenbank ändern, muss die Abfrage neu ausgeführt werden, um den aktuellen Stand der Daten anzuzeigen.

Sichten oder views kann man sich sehr gut als Momentaufnahmen des Datenbestandes vorstellen. Abgefragt und angezeigt werden immer die aktuell zur Verfügung

stehenden Daten.

In unserem Fall sollte das Ergebnis der SELECT-Abfrage mit JOIN-Anweisung und WHERE-Bedingung in phpMyAdmin in etwa so aussehen:

**SQL-Befehl:**

```
SELECT *
FROM news
INNER JOIN user ON news.ID_user = user.ID_user
WHERE news.ID_user = 1
LIMIT 0, 30
```

Profiling | [Bearbeiten](#) | [SQL erklären](#) | [...](#)

Zeige:  Datensätze, beginnend ab

untereinander  angeordnet und wiederhole die Kopfzeilen nach  Datensätzen.

ID_news	Titel	Text	Datum	ID_user	ID_user	Name	PW	Email
1	Test-News	Das ist nur ein Test der Datenbank.	2007-12-02 00:00:00	1	1	tester	test	test@test.com

16. Die am häufigsten verwendeten SQL-Befehle sind mit CREATE, INSERT und SELECT eigentlich bereits abgedeckt, jedoch braucht man häufig auch die Möglichkeit, Datensätze zu ändern (UPDATE) oder zu löschen (DELETE). Über phpMyAdmin sind diese Funktionen relativ einfach und intuitiv zu benutzen, daher wollen wir den Umgang mit diesen Befehlen gleich in SQL üben.

Fügen Sie als erstes einen neuen Datensatz in die Tabelle *news* ein. Verwenden Sie dazu den SQL-Befehl, mit dem die erste News eingetragen wurde. In der Tabelle *news* sollten sich jetzt mindestens 2 Datensätze befinden.

17. Nun werden wir den neuen Datensatz in den Feldern *Titel* und *Text* aktualisieren, d.h. für *Titel* und *Text* sollen neue Inhalte eingetragen werden. Dies geschieht mit folgendem SQL-Befehl:

```
UPDATE `news`
SET `Titel` = 'Neuer Titel', `Newstext` = 'Neuer Text'
WHERE `ID_news` = 2;
```

18. Jetzt werden wir den eben aktualisierten Datensatz löschen. Hier ist der SQL-Befehl dafür:

```
DELETE FROM `news`
WHERE `ID_news` = 2;
```

19. Bei den Kursmaterialien finden Sie eine **Übersicht über die häufigsten MySQL-Befehle** zusammen mit Anwendungsbeispielen. Sie sollten alle diese Beispiele einmal mit unserer Datenbank *test\_newssystem* ausprobieren.

20. Unsere einfache Datenbank *test\_newssystem* ist nun einsatzbereit und könnte für den eigentlichen Zweck, nämlich ein Website News-System verwendet werden. Dazu bedarf es einer Wirtssprache mit der Fähigkeit, Daten aus MySQL-Datenbanken zu bearbeiten. PHP ist so eine Programmiersprache für das Web, die sehr gut mit MySQL zusammenarbeiten kann.

Ohne jetzt auch noch PHP erlernen zu müssen, wollen wir das Ganze einfach mal anhand von einem vorgefertigten PHP-Skript testen. Bei den **Kursmaterialien** finden Sie eine Beispieldatei für diesen Zweck. Erstellen Sie im Verzeichnis

**/xampp/htdocs/**

ein Unterverzeichnis mit dem Namen „*news*“. Speichern Sie in diesem

Unterverzeichnis die Beispieldatei unter dem Namen „*index.php*“ ab (achten Sie auf die Dateiendung .php!). Wenn Sie nun im Browser zu <http://localhost/news/> surfen, sehen Sie das Resultat.



## Übung 2: Erweiterung des Website News-Systems um eine Kommentarfunktion

In der letzten Übung haben wir ein bereits funktionierendes News-System für eine Website erstellt. Dieses soll nun um eine Kommentarfunktion erweitert werden, d.h. zu jedem Newsbeitrag sollen Benutzer beliebig viele Kommentare abgeben können. Bei den Kommentaren soll es so sein, dass jeder Besucher Kommentare abgeben kann, also nicht nur registrierte Benutzer.

**Ein Kommentar soll aus den folgenden Feldern bestehen:**

- Titel des Kommentars
- Text des Kommentars
- Datum des Kommentars
- Name des Verfassers

Zu dieser Übung gibt es keine Schritt-für-Schritt-Anleitung, sondern nur ein paar Hinweise und Ratschläge:

- Beginnen Sie damit, das ERM anzupassen. Überlegen Sie sich, wie und an welcher Stelle sich die Entität „Kommentare“ in die vorhandene Skizze einbinden lässt. Es ist wichtig, dass Sie die Beziehungen der Tabellen in dieser Phase erkennen und nötigenfalls zusätzliche Felder für Primär- und Fremdschlüssel in die Tabellen mit aufnehmen.
- Erstellen Sie dann die Tabellen über die phpMyAdmin Weboberfläche. Sie müssen nicht unbedingt den SQL-Befehl dazu verwenden, es geht ja auch alles über die Oberfläche. Beachten Sie dabei vor allem die Wahl der richtigen Datentypen.

Die wichtigsten Felddatentypen in MySQL:	
Datentyp	Definition
Varchar	Text, max. 255 Zeichen
Text	Text, max. <216 Zeichen
Int	Zahl, von -2147483648 bis + 2147483648
Date	Datum, Format YYYY-MM-DD
Time	Zeit, Format HH:MM:SS
Timestamp	Datum und Zeit, Format YYYYMMDDHHMMSS, wird bei Einfüge- und Aktualisierungsvorgängen automatisch eingetragen. Von mehreren Timestampfeldern in einer Tabelle wird immer nur das erste automatisch aktualisiert.
Eine vollständige Liste findet sich im SQL-Handbuch, unter <a href="http://dev.mysql.com/doc/refman/5.1/de/data-types.html">http://dev.mysql.com/doc/refman/5.1/de/data-types.html</a>	

- Fügen Sie in die fertige Tabelle einen Kommentar zu einer vorhandenen News ein. Kontrollieren Sie sich selbst über die Registerkarte Anzeigen in phpMyAdmin.
- Erstellen Sie eine SELECT-Abfrage mit JOIN-Anweisung und WHERE-Bedingung die die erste News und sämtliche Kommentare dazu anzeigt.