

In diesem Kapitel:

- Der Kommandozeilenclient `mysql`
- `phpMyAdmin`

Mit MySQL arbeiten

Arbeit um der Arbeit willen ist gegen die menschliche Natur.

John Locke

Nach dem praktischen Schnelleinstieg im vorigen Kapitel erfahren Sie in den folgenden Kapiteln ausführlich, wie Sie MySQL-Datenbanken erstellen und bearbeiten können. Zunächst werden in diesem Kapitel zwei wichtige Tools für die Arbeit mit MySQL genauer behandelt: der Kommandozeilenclient `mysql` und der Webclient `phpMyAdmin`.

Der Kommandozeilenclient `mysql`

In den beiden vorangehenden Kapiteln haben Sie bereits intuitiv mit dem Konsolentool `mysql` gearbeitet. An dieser Stelle werden seine wichtigsten Funktionen systematischer vorgestellt.

Wie Sie bereits erfahren haben, wird das Programm üblicherweise mit den Optionen `-u <Benutzername>` und `-p` (ohne Wert) für die anschließende Passwordeingabe gestartet. Zum Beispiel:

```
$ mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1 to server version: 4.1.13-nt
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
```

Die Sternchen für die Passwordeingabe werden nur unter Windows angezeigt; auf Unix-Rechnern erfolgt kein Feedback.

Es gibt zahlreiche weitere Optionen für den Start von `mysql`. Wenn Sie eine Gesamtübersicht erhalten möchten, können Sie Folgendes eingeben:

```
$ mysql --help
```

Da die Ausgabe zu lang ist, sollten Sie sie durch eine Pipe an einen Pager wie *less* (Linux) oder *more* (Windows) weiterleiten:

```
$ mysql --help |less
> mysql --help |more
```

Alternativ können Sie den gesamten Text auf beiden Systemen in eine Datei umleiten:

```
$ mysql --help >myhelp.txt
```

Hier noch zwei weitere wichtige Kommandozeilenoptionen:

`-h=Hostname` oder `--host=Hostname`

Verbindung zum MySQL-Server auf einem anderen Rechner herstellen. Sie können einen Hostnamen wie *dbserver.test.local* oder eine IP-Adresse wie *192.168.0.17* angeben.

`--tee=Dateiname`

Sämtliche Ein- und Ausgaben der *mysql*-Sitzung zusätzlich in die angegebene Datei schreiben. Falls die Datei bereits besteht, werden die neuen Daten angehängt.

Hinter sämtlichen Optionen können Sie den Namen der Datenbank angeben, die als Standard verwendet werden soll. Diese kann innerhalb des Programms mittels *use* (siehe unten) geändert werden.

Das folgende Beispiel stellt eine Verbindung mit dem MySQL-Server auf dem Host *db.test.local* her, kopiert die Ein- und Ausgaben in die Datei *mysql.log* und wählt die Datenbank *gewinnspiel* als Standard aus:

```
$ mysql -u root -p -h=db.test.local --tee=mysql.log gewinnspiel
```

Nach erfolgreichem Login erscheint der Prompt des Clients:

```
mysql>
```

Nun können Sie beliebige SQL-Abfragen sowie interne Befehle des Clients selbst eingeben. Für eine erste Übersicht über Letztere können Sie Folgendes eingeben:

```
mysql> HELP;
```

Ein Problem der Windows-Eingabeaufforderung



Beim Einsatz des Kommandozeilen-Clients *mysql* unter Windows ergibt sich ein grundsätzliches, nicht auf direktem Wege lösbares Problem: Da die Eingabeaufforderung auch in ihrer heutigen Form ein Überbleibsel des alten Betriebssystems MS-DOS darstellt, verwendet sie einen anderen Zeichensatz als Windows selbst. Aus diesem Grund werden Umlaute und andere Sonderzeichen, die Sie hier eingeben, in Windows-Anwendungen falsch wiedergegeben und umgekehrt. Im Grunde kann man also keinen anderen Tipp geben, als *mysql* unter Windows nur zur Verwaltung und Überprüfung von Datenbanken zu verwenden. Für die Neueingabe deutschsprachiger Daten müssen Sie statt dessen *phpMyAdmin* oder einen anderen grafischen Client einsetzen.

Für jeden internen Befehl gibt es eine Abkürzung mit vorangestelltem Backslash (`\`), die Sie ohne das übliche Semikolon eingeben können. Statt `HELP;` ist beispielsweise die Kurzfassung `\h` zulässig. Ein weiteres Synonym für `HELP;` ist übrigens das Fragezeichen – sowohl mit (`\?`) als auch ohne Backslash (`?`).

Außer den Kurzbefehlen muss jede Eingabe – ob interner Befehl oder SQL-Abfrage – durch ein Semikolon abgeschlossen werden. Dies erlaubt mehrere Anweisungen in einer einzelnen Zeile. Das folgende Beispiel wählt zunächst die Datenbank `gewinnspiel` als Standard aus und zeigt anschließend die Datensätze der Tabelle `gw_fragen` an:

```
mysql> USE gewinnspiel; SELECT * FROM gw_fragen;
```

Auch die Verteilung langer Eingaben auf beliebig viele Zeilen wird unterstützt. Das nachfolgende Beispiel wählt die Fragetexte und die jeweils richtige Antwort für das Gewinnspiel aus:

```
mysql> SELECT fr_frage, an_text
-> FROM gw_fragen, gw_antworten
-> WHERE fr_id=an_frage AND fr_korrekt=an_antwort;
```

fr_frage	an_text
Wie heißt die Hauptstadt von Italien?	Rom
Welche dieser Hauptstädte hieß einst Lutetia?	Paris
Bratislava ist die Hauptstadt von ...?	Slowakei
Welche dieser Städte ist keine Hauptstadt?	Istanbul

4 rows in set (0.00 sec)

Statt des abschließenden Semikolons können Sie auch die Zeichenfolge `\g` (den Befehl `go`) verwenden. Die spezielle Variante `\G` (`ego`) sorgt dagegen für eine andere Ausgabe: Jedes Feld wird in einer separaten Zeile ausgegeben. Das ist nützlich, wenn die Spalten einer Abfrage zu breit sind, so dass die Tabelle nicht korrekt dargestellt werden kann. Hier die obige Abfrage in diesem Format:

```
mysql> SELECT fr_frage, an_text
-> FROM gw_fragen, gw_antworten
-> WHERE fr_id=an_frage AND fr_korrekt=an_antwort \G
***** 1. row *****
fr_frage: Wie heißt die Hauptstadt von Italien?
an_text: Rom
***** 2. row *****
fr_frage: Welche dieser Hauptstädte hieß einst Lutetia?
an_text: Paris
***** 3. row *****
fr_frage: Bratislava ist die Hauptstadt von ...?
an_text: Slowakei
***** 4. row *****
fr_frage: Welche dieser Städte ist keine Hauptstadt?
an_text: Istanbul
4 rows in set (0.00 sec)
```

Auf Wunsch können Sie das Semikolon auch durch ein anderes Zeichen ersetzen. Verwenden Sie dazu den Befehl `DELIMITER` oder `\d`. In der Langform müssen Sie dazu das bisherige Schlusszeichen anhängen – das folgende Beispiel stellt den Doppelpunkt (`:`) ein:

```
mysql> DELIMITER ;;
```

Zurück zum Semikolon geht es danach wie folgt:

```
mysql> DELIMITER ;;
```

Die Kurzfassung `\d` macht diese Einstellungen übersichtlicher und verständlicher:

```
mysql> \d :
```

```
mysql> \d ;
```

Ein weiteres besonderes Kürzel ist `\c` (`CLEAR`): Es ermöglicht den Abbruch einer bereits begonnenen, insbesondere mehrzeiligen Eingabe. Zum Beispiel:

```
mysql> SELECT * FROM gw_fragen  
-> WHERE \c
```



Die Langfassungen interner Befehle, die nicht für sich stehen, sondern andere Befehle ergänzen oder manipulieren, können Sie nur dann verwenden, wenn Sie den Client mit der speziellen Option `--named-commands` gestartet haben. Ein Beispiel wäre ein ausgeschriebenenes `clear` statt `\c`.

Einen der wichtigsten internen Befehle des Clients haben Sie bereits kennen gelernt: `USE Datenbank` wählt die angegebene Datenbank aus, was erheblich komfortabler ist, als in jeder Abfrage `Datenbank.Tabellenname` schreiben zu müssen. Zum Beispiel:

```
mysql> USE gewinnspiel;  
Database changed
```

Wenn Sie herausfinden möchten, welche Datenbank gerade Standard ist, können Sie per `SELECT`-Abfrage den Wert der Funktion `DATABASE()` ermitteln:

```
mysql> SELECT DATABASE();  
+-----+  
| DATABASE() |  
+-----+  
| gewinnspiel |  
+-----+  
1 row in set (0.01 sec)
```

Sollten Sie keine spezielle Datenbank ausgewählt haben, erhalten Sie die Ausgabe `NULL`.

Die SQL-Anweisung `SHOW` ermöglicht die Anzeige weiterer Informationen. `SHOW DATABASES` gibt alle auf dem Server verfügbaren Datenbanken aus. Zum Beispiel:

```
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| buchliste |
| essen |
| fachbuecher |
| gewinnspiel |
| mysql |
| reisebuero |
| test |
+-----+
17 rows in set (0.02 sec)
```

SHOW TABLES zeigt dagegen die Tabellen der aktuellen Datenbank an – natürlich nur, sofern eine Standarddatenbank ausgewählt wurde:

```
mysql> SHOW TABLES;
+-----+
| Tables_in_gewinnspiel |
+-----+
| gw_antworten |
| gw_fragen |
| gw_teilnahme |
| gw_teilnehmer |
+-----+
4 rows in set (0.00 sec)
```

Schließlich können Sie sich auch die Struktur einer Tabelle anzeigen lassen. Die zuständige Abfrage besitzt die Syntax `DESCRIBE [Datenbank.]Tabelle`. Den Datenbanknamen benötigen Sie nur, falls sich die gewünschte Tabelle nicht in der aktuellen Datenbank befindet. Hier sehen Sie ein Beispiel:

```
mysql> DESCRIBE gw_fragen;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| fr_id | int(11) | | PRI | NULL | auto_increment |
| fr_frage | varchar(80) | YES | | NULL | |
| fr_korrekt | int(11) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)
```

Mit Hilfe des Befehls `TEE` oder `\T` können Sie auch im laufenden Betrieb eine Datei festlegen, an die sämtliche Befehlseingaben und Ergebnisse angehängt werden. Zum Beispiel:

```
mysql> TEE mysql.log;
```

Wenn Sie die Dialoge nicht mehr in eine Datei kopieren möchten, können Sie den Befehl `NOTE` oder `\t` verwenden:

```
mysql> NOTE;
```

Sehr interessant ist auch der Befehl `SOURCE Dateiname`: Er dient dazu, SQL-Abfragen aus einer externen (Text-)Datei zu lesen und auszuführen. Das folgende Beispiel bezieht sich auf die Erstellung der ab dem nächsten Kapitel benötigten Datenbank *reisebuero*:

```
mysql> SOURCE reisebuero.sql;
```

Die Datei *reisebuero.sql* finden Sie übrigens im Verzeichnis *databases* auf der beiliegenden CD-ROM.

Alternativ können Sie SQL-Quelldateien auch per Eingabeumleitung an den `mysql`-Client übergeben. Zum Beispiel:

```
$ mysql -u root -p <reisebuero.sql
Enter password:
```

Speziell für solche Quelldateien ist es übrigens nützlich, jede einzelne Abfrage mit `\p` (`print`) vor dem Semikolon abzuschließen – dadurch werden nicht nur Ergebnisse wie »Database created«, sondern auch die Abfragen selbst angezeigt. Das folgende Beispiel ist die erste Zeile von *reisebuero.sql* – sie löscht diese Datenbank, falls sie bereits vorhanden sein sollte:

```
mysql> DROP DATABASE IF EXISTS reisebuero \p;
```

SQL-Dateien als »Futter« für `SOURCE` können Sie mit Hilfe des Kommandozeilen-tools *mysqldump* erstellen. Auf diese Weise lassen sich existierende Datenbanken leicht exportieren.

`STATUS` oder `\s` liefert Statusinformationen über den MySQL-Server und -Client. Zum Beispiel:

```
mysql> STATUS;
-----
mysql Ver 14.7 Distrib 4.1.13, for Win95/Win98 (i32)

Connection id:          7
Current database:
Current user:           root@localhost
SSL:                    Not in use
Using delimiter:       ;
Server version:        4.1.13-nt
Protocol version:      10
Connection:            localhost via TCP/IP
Server characterset:   latin1
Db characterset:       latin1
Client characterset:   latin1
Conn. characterset:    latin1
TCP port:              3306
Uptime:                2 hours 54 min 59 sec

Threads: 1 Questions: 41 Slow queries: 0 Opens: 13 Flush tables: 1 Open tables: 1
Queries per second avg: 0.004
-----
```

Um Ihre Arbeit mit dem Client zu beenden, können Sie zu guter Letzt `EXIT` oder `QUIT` (Kurzfassung jeweils `\q`) eingeben:

```
mysql> EXIT;  
Bye
```

phpMyAdmin

In Kapitel 2 wurde bereits beschrieben, wie Sie den webbasierten Client *phpMyAdmin* einrichten und starten können. Wenn Sie diese Installationsanleitung befolgt haben, müsste phpMyAdmin unter `http://localhost/phpmyadmin/` zur Verfügung stehen, andernfalls müssen Sie die URL gemäß Ihrer eigenen Konfiguration eingeben. Je nachdem, wie Sie die Benutzeranmeldung eingerichtet haben, müssen Sie nach Aufruf der URL Benutzername und Kennwort eingeben oder auch nicht. Anschließend gelangen Sie auf die in Abbildung 4-1 gezeigte phpMyAdmin-Startseite.

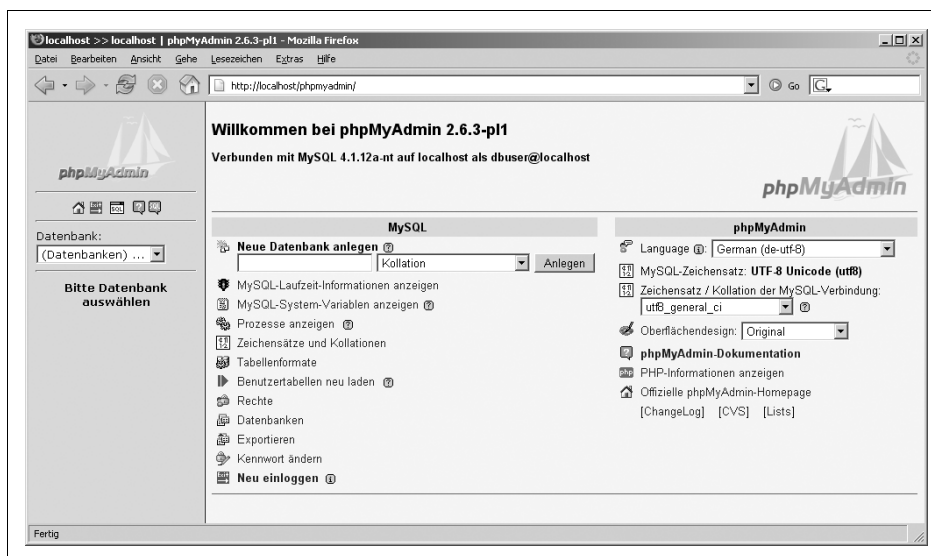


Abbildung 4-1: Die Startseite von *phpMyAdmin*

Der Bildschirm ist in zwei Frames unterteilt: Im schmalen linken Frame können Sie eine Datenbank aus dem Pull-down-Menü auswählen. Rechts finden Sie auf der Startseite die beiden Rubriken *MySQL* mit grundlegenden Operationen und Informationen zum Datenbankserver sowie *phpMyAdmin* mit einigen Grundeinstellungen für den phpMyAdmin-Client selbst. Tabelle 4-1 zeigt zunächst eine Übersicht über die einzelnen Punkte der Spalte *MySQL*. Falls verfügbar, wird zusätzlich der entsprechende Befehl aufgelistet, den Sie für dasselbe Ergebnis im `mysql`-Kommandozeilenclient eingeben müssten. Einige dieser Optionen werden in Kapitel 8, *MySQL-Administration*, genauer behandelt.

Tabelle 4-1: »MySQL«-Optionen auf der phpMyAdmin-Startseite

Funktion	Kommandozeilenvariante	Beschreibung
Neue Datenbank anlegen	CREATE DATABASE ...	Eine neue Datenbank mit der angegebenen Kollation (Sprach-Sortierreihenfolge) erstellen
MySQL-Laufzeit-Informationen anzeigen	SHOW STATUS	Ausführliche Variante der Statusanzeige (siehe vorigen Abschnitt)
MySQL-System-Variablen anzeigen	SHOW VARIABLES	In Systemvariablen gespeicherte MySQL-Einstellungen und -Eigenschaften anzeigen
Prozesse anzeigen	SHOW PROCESSLIST	Alle aktiven Verbindungen des MySQL-Servers anzeigen
Zeichensätze und Kollationen	SHOW CHARSET/SHOW COLLATION	Übersicht aller Zeichensätze und Sprach-Sortierreihenfolgen (Kollationen)
Rechte	SHOW PRIVILEGES	Übersicht über alle MySQL-Benutzer und ihre Berechtigungen
Datenbanken	SHOW DATABASES	Liste aller Datenbanken des MySQL-Servers
Exportieren	Dienstprogramm <i>mysqldump</i> u.a. (siehe Kapitel 8, <i>MySQL-Administration</i>)	Datenbanken in verschiedenen Formaten exportieren
Kennwort ändern	SET PASSWORD ...	Kennwort des aktuellen Benutzers ändern (für aktuelle PHP/phpMyAdmin-Versionen muss die Option <i>MySQL 4.0 kompatibel</i> gewählt werden)
Neu einloggen	CONNECT	Neue Verbindung zum MySQL-Server herstellen

In der Spalte *phpMyAdmin* können Sie dagegen Einstellungen für den Client selbst vornehmen und einige Informationen erhalten:

- Unter *Language* wird die Sprache von phpMyAdmin selbst eingestellt – für alle Beispiele in diesem Buch wurde natürlich *German* gewählt, aber wie Sie sehen, unterstützt das Tool noch zahlreiche weitere Sprachen.
- *MySQL-Zeichensatz* zeigt den (hier nicht änderbaren) Zeichensatz an, den der Server selbst standardmäßig verwendet.
- *Zeichensatz/Kollation der MySQL-Verbindung* ermöglicht Ihnen die Auswahl, welchen Zeichensatz und welche Kollation phpMyAdmin bei der Kommunikation mit dem Server verwenden soll.
- Unter *Oberflächendesign* können Sie ein Farbschema für phpMyAdmin auswählen; mitgeliefert werden die beiden Einstellungen *Original* und *Darkblue/orange*. Die Auswahl hat nicht nur ästhetische Bedeutung, sondern ermöglicht je nach Umgebungslicht und Monitortyp eventuell eine bessere Kontrastwirkung. Im Internet sind weitere Themen erhältlich.
- Die über den gleichnamigen Link erreichbare *phpMyAdmin-Dokumentation* wird automatisch mit phpMyAdmin geliefert.
- *PHP-Informationen anzeigen* öffnet ein neues Browserfenster, in dem die Ausgabe des Befehls `phpinfo()` angezeigt wird (siehe Kapitel 2, *Installation und Inbetriebnahme*).

- Die restlichen Links verweisen auf die Websites des phpMyAdmin-Projekts selbst: Neben der offiziellen *Homepage* sind auch das *ChangeLog* (Liste der Änderungen pro Version), das *CVS-Repository* (Quellcode-Archiv mit Versionsverwaltung) sowie *Lists* (die phpMyAdmin-Mailing-Listen; siehe Anhang D, *Ressourcen und Tools*) zu erreichen.

Unter den beiden Spalten werden eventuell Konfigurations- oder Sicherheitsprobleme von phpMyAdmin oder der zugrunde liegenden PHP-Installation angezeigt. Besonders kritische Fehler (zum Beispiel Zugriff mit einem leeren Passwort) werden in Rot dargestellt und sollten unverzüglich behoben werden.

Sobald Sie links eine Datenbank auswählen, werden die Namen aller ihrer Tabellen als Hyperlinks angezeigt. Auch im rechten Frame erscheint eine Übersicht über sämtliche Tabellen. In Abbildung 4-2 wird dies für die Datenbank *gewinnspiel* aus dem vorigen Kapitel dargestellt. In der Spalte *Aktion* finden Sie Schaltflächen für den schnellen Zugriff auf sechs wichtige Tabellenoperationen: *Anzeigen* stellt die Inhalte der Tabelle dar, mittels *Suche* können Sie die Tabelle nach diversen Kriterien durchsuchen, *Einfügen* dient dem Hinzufügen neuer Datensätze, *Struktur* ermöglicht Änderungen an der Tabellenstruktur selbst (Namen, Datentypen und Reihenfolge der Spalten und so weiter), *Leeren* löscht sämtliche Inhalte der Tabelle und *Löschen* sogar die Tabelle selbst. Für die beiden letzteren Operationen wird allerdings zuvor eine JavaScript-basierte Sicherheitsabfrage durchgeführt.

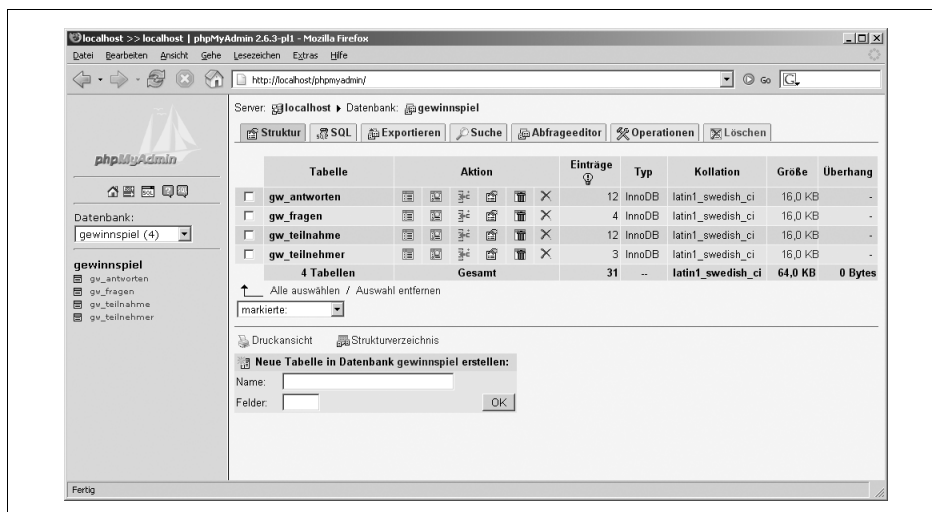


Abbildung 4-2: Übersichtsseite einer Datenbank in phpMyAdmin

Unter der Tabellenübersicht finden Sie ein Formular zum Erstellen einer neuen Tabelle innerhalb der aktuellen Datenbank. Dazu müssen Sie einen Namen und die gewünschte Anzahl der Felder pro Datensatz eingeben. Sobald Sie auf *OK* klicken, gelangen Sie auf eine neue Seite, die weiter unten beschrieben wird.



In vielen Browsern (zum Beispiel in Firefox, der für die Beispiele in diesem Buch verwendet wird, oder auch im Microsoft Internet Explorer) werden die Funktionen aller phpMyAdmin-Schaltflächen als Tooltips angezeigt, sobald Sie mit der Maus darüber fahren. Sämtliche Schaltflächenbezeichnungen im vorliegenden Buch beziehen sich auf diese Tooltips.

Tabellenoptionen

Wenn Sie im linken Frame eine Tabelle auswählen, sieht der Hauptframe so aus, wie es in Abbildung 4-3 für die Tabelle `gw_fragen` gezeigt wird. Die Seite entspricht der Auswahl *Struktur* auf der Hauptseite einer Datenbank.

The screenshot shows the phpMyAdmin interface for the 'gewinnspiel' database, specifically the 'gw_fragen' table. The main table lists columns: 'fr_id' (int(11), auto_increment), 'fr_frage' (varchar(80), latin1_swedish_ci), and 'fr_korrekt' (int(11)). Below this, there are sections for 'Indizes' (showing a primary key on 'fr_id'), 'Speicherplatzverbrauch' (storage usage), and 'Zeilenstatistik' (row statistics). At the bottom, there is a text area for SQL queries and a 'Felder:' dropdown menu.

Abbildung 4-3: Startseite einer Datenbanktabelle in phpMyAdmin

Ganz oben auf der Seite befinden sich einige Links auf wichtige Datenbankbereiche und Tabellenoptionen. In der ersten Zeile werden Links auf die aktuelle Hierarchie aus Host, Datenbank und Tabelle angezeigt. Die zweite Zeile enthält Links auf diverse Seiten zur Tabellenbearbeitung:

- *Struktur* (aktuelle Seite): Stellt den Grundaufbau, aber nicht den Inhalt der Tabelle dar.

- *Anzeigen*: Stellt die Inhalte der Tabelle mit Blätter- und Sortiermöglichkeiten dar, standardmäßig je 30 Datensätze pro Seite.
- *SQL*: Ermöglicht die manuelle Eingabe oder den Datei-Upload von SQL-Abfragen. Auf der Struktur-Seite befindet sich ebenfalls ein solches Formular, das in Kürze beschrieben wird.
- *Suche*: Ein Formular mit verschiedenen Optionen zur Suche in der Datenbank.
- *Einfügen*: Neue Datensätze in die aktuelle Tabelle eingeben.
- *Exportieren*: Exportiert die Struktur und/oder die Daten der Tabelle in verschiedenen Formaten mit diversen Optionen.
- *Operationen*: Änderung von Tabelleneigenschaften wie Name, Typ oder Kollation.
- *Leeren*: Entfernen aller Inhalte der Tabelle (mit Sicherheitsabfrage).
- *Löschen*: Entfernen der gesamten Tabelle und ihrer Inhalte, ebenfalls mit Nachfrage zur Absicherung.

Einige der genannten Optionen werden im Folgenden näher erläutert, andere erst in späteren Kapiteln, zu deren Thematik sie besser passen.

Tabellenstruktur

Wie bereits erwähnt, stellt die erste Tabellenbearbeitungsseite die Struktur einer Tabelle dar. Sie erscheint, wenn Sie im linken Frame eine Tabelle auswählen oder wenn Sie den Link *Struktur* auf einer anderen Tabellen-Seite anklicken. In Abbildung 4-4 sehen Sie die Strukturansicht der Tabelle *gw_antworten*.

Oben werden zunächst die wichtigsten Eigenschaften aller Tabellenspalten angezeigt:

- *Feld*: Name der Spalte
- *Typ*: SQL-Datentyp der Spalte
- *Kollation*: Sprach-Sortierreihenfolge von Spalten mit Textdatentypen
- *Attribute*: Zusatzeigenschaften einer Spalte, zum Beispiel `UNSIGNED` (vorzeichenlos)
- *Null*: Gibt an, ob Felder in der Spalte leer bleiben dürfen oder nicht.
- *Standard*: Standardwert für die Felder der Spalte, falls definiert.
- *Extra*: Falls die Spalte automatisch hochgezählt wird, steht hier `auto_increment`.
- *Aktion*: Schaltflächen für Operationen mit dieser Spalte. Von links nach rechts stehen folgende Optionen zur Verfügung: *Ändern* – Spalteneigenschaften modifizieren, *Löschen* – Spalte und ihre Inhalte entfernen, *Primärschlüssel* – diese Spalte als Primärschlüssel der Tabelle einstellen, *Index* – einen Index für die Spalte einrichten, *Unique* – festlegen, dass jedes Feld der Spalte einen ein-

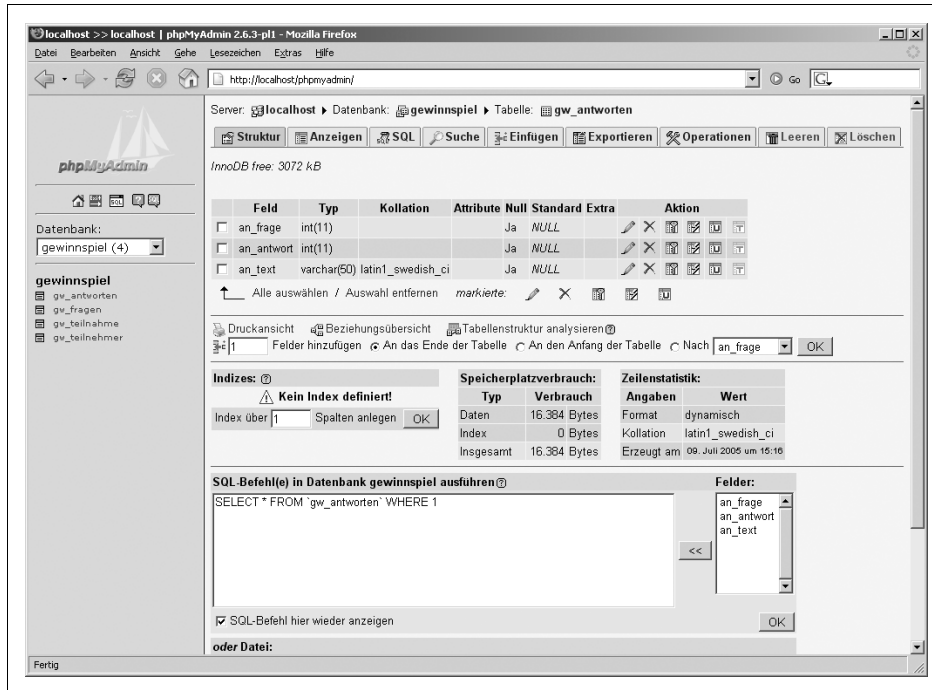


Abbildung 4-4: Die Struktur der Tabelle gw_antworten in phpMyAdmin

deutigen Wert enthalten muss, *Volltext* – einen Volltextindex erstellen (nur für einfache Texttypen in *MyISAM*-Tabellen).



Die einzelnen Optionen dieser Übersicht werden weiter unten genauer erläutert, wenn es darum geht, sie manuell einzugeben beziehungsweise festzulegen.

Unter der Tabellenansicht finden Sie einen Satz Schaltflächen, die den soeben beschriebenen Aktionen entsprechen. Auf diese Weise lassen sich die Aktionen auf mehrere per Kontrollkästchen ausgewählte Spalten gleichzeitig anwenden. In diesem Zusammenhang sind auch die beiden Links *Alle auswählen* und *Auswahl entfernen* sehr praktisch.

Im nächsten Abschnitt finden Sie zunächst drei Links: *Druckansicht* stellt die Tabellenstruktur auf einer nüchternen, druckoptimierten Seite ohne Einstellmöglichkeiten dar und enthält zusätzlich die passende Schaltfläche zum Drucken. Die *Beziehungsübersicht* stellt definierte Verknüpfungen zur referenziellen Integrität dar – Näheres darüber erfahren Sie in Kapitel 5. Der nächste Punkt, *Tabellenstruktur analysieren*, vergleicht die Datentypen der Tabelle mit den tatsächlich enthaltenen Feldwerten und macht gegebenenfalls Verbesserungsvorschläge. Auch darauf wird im nächsten Kapitel genauer eingegangen.

Als Nächstes erhalten Sie die Möglichkeit, eine frei wählbare Anzahl neuer Spalten (*Felder*) zur Tabelle hinzuzufügen. Sie können sich aussuchen, ob sie am Anfang der Tabelle, an ihrem Ende oder hinter einer bestimmten vorhandenen Spalte eingefügt werden sollen. Wenn Sie die gewünschten Einstellungen vorgenommen und die Schaltfläche *OK* betätigt haben, gelangen Sie auf eine neue Seite, die der weiter unten beschriebenen Eingabeseite für neue Tabellen entspricht.

Der nachfolgende Abschnitt gibt Auskunft über die in der Tabelle definierten *Indizes*. Ausführliche Informationen über Schlüssel und Indizes finden Sie im nächsten Kapitel.

Im letzten Abschnitt finden Sie ein umfangreiches Textfeld zur freien Eingabe von SQL-Abfragen. Rechts daneben befindet sich ein Auswahlfeld mit sämtlichen Spalten der aktuellen Tabelle. Wenn Sie darin einen Feldnamen anklicken und anschließend die daneben liegende Pfeil-Schaltfläche betätigen, wird dieser Name – ordentlich in `Backticks` eingefasst – an die Cursorposition des Eingabefeldes übernommen. Der automatisch vorgegebene Eintrag dient der Auswahl sämtlicher Spalten (*) und aller Datensätze (Kriterium `WHERE 1`); er entspricht der Tabellenoption *Anzeigen*.

Tabelleninhalt

Wenn Sie eine Tabelle ausgewählt haben und den Link *Anzeigen* anklicken, werden die Inhalte der Tabelle angezeigt; in Abbildung 4-5 sehen Sie die *Tabellengw_fragen* als Beispiel. Oben erscheint der Wortlaut der SQL-Abfrage, die zum aktuellen Abschnitt der Tabellendaten geführt hat. Per Voreinstellung werden jeweils 30 Datensätze pro Seite angezeigt, was durch eine `LIMIT`-Klausel erreicht wird.

Über den eigentlichen Daten (und ein weiteres Mal darunter) können Sie festlegen, welche und wie viele Datensätze angezeigt werden sollen; ein Klick auf die Schaltfläche *Zeige* bestätigt Ihre Auswahl. Besitzt die Tabelle mehr Zeilen, als bei der gewählten Anzahl auf die aktuelle Seite passen, werden zusätzliche Schaltflächen zum Blättern bereitgestellt: `<` und `>` blättern je eine Seite zurück beziehungsweise vor, während `<<` und `>>` an den Anfang beziehungsweise das Ende der derzeitigen Sortierreihenfolge springen.

In der nächsten Zeile können Sie sich weitere Darstellungsoptionen aussuchen: Der Standardmodus *untereinander* stellt die Datensätze in vertikaler und die Spalten in horizontaler Ansicht dar. Die Option *horizontal (gedrehte Bezeichner)* dreht die Spaltenbezeichnungen um 90°, was bei Spalten mit schmalen Inhalten einiges an Platz spart – allerdings funktioniert das bisher nur im Internet Explorer. Der Modus *nebeneinander* schließlich vertauscht die Tabellenachsen – die Datensätze stehen in Spalten, ihre einzelnen Felder untereinander.

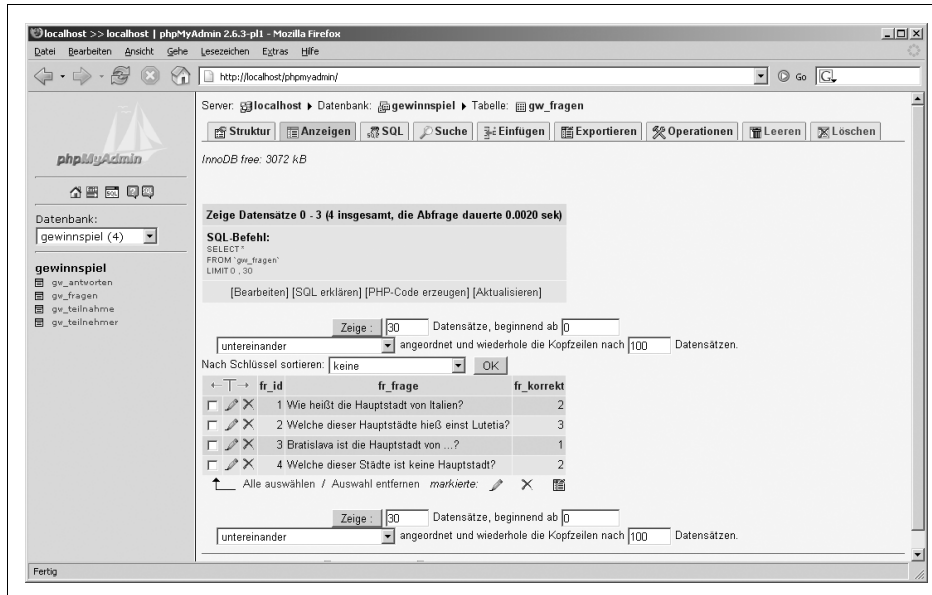


Abbildung 4-5: Darstellung des Tabelleninhalts von gw_fragen in phpMyAdmin

Als Nächstes geht es um die Sortierreihenfolge der angezeigten Datensätze: Das Pull-down-Menü *Nach Schlüssel sortieren* ermöglicht die Auswahl des Primärschlüssels und anderer indizierter Spalten als Sortierkriterium. Viel intuitiver und einfacher ist es dagegen, den Titel der Spalte anzuklicken, nach der sortiert werden soll. Sobald Sie dieselbe Spalte zum zweiten Mal anklicken, wird übrigens die Sortierrichtung umgekehrt; ein kleines Dreieck neben dem Spaltentitel markiert die aktuelle Einstellung.



Oft ist es wünschenswert, mehrere Datensätze, bei denen das aktuelle Sortierkriterium denselben Wert hat, in sich nach einem zweiten Kriterium zu sortieren. Ein häufiger Fall ist etwa das Sortieren nach Vornamen bei identischem Nachnamen. In phpMyAdmin funktioniert dies leider nur per manueller Eingabe einer SQL-Abfrage. Näheres erfahren Sie in Kapitel 6.

Für jede Zeile wird eine Checkbox zum Auswählen angezeigt, damit Sie mit Hilfe der Schaltflächen unter der Tabelle mehrere Datensätze *bearbeiten* (Stiftsymbol), *löschen* (rotes X) beziehungsweise *exportieren* (Tabellensymbol) können. Die Link-Symbole zum *Ändern* sowie zum *Löschen* finden Sie zusätzlich bei jedem einzelnen Datensatz.

Tabellenoperationen

Wenn Sie den Link *Operationen* am oberen Rand von Tabellenseiten anklicken, können Sie einige interessante Änderungen an Tabellen durchführen. Abbildung 4-6 zeigt alle Optionen im Überblick.

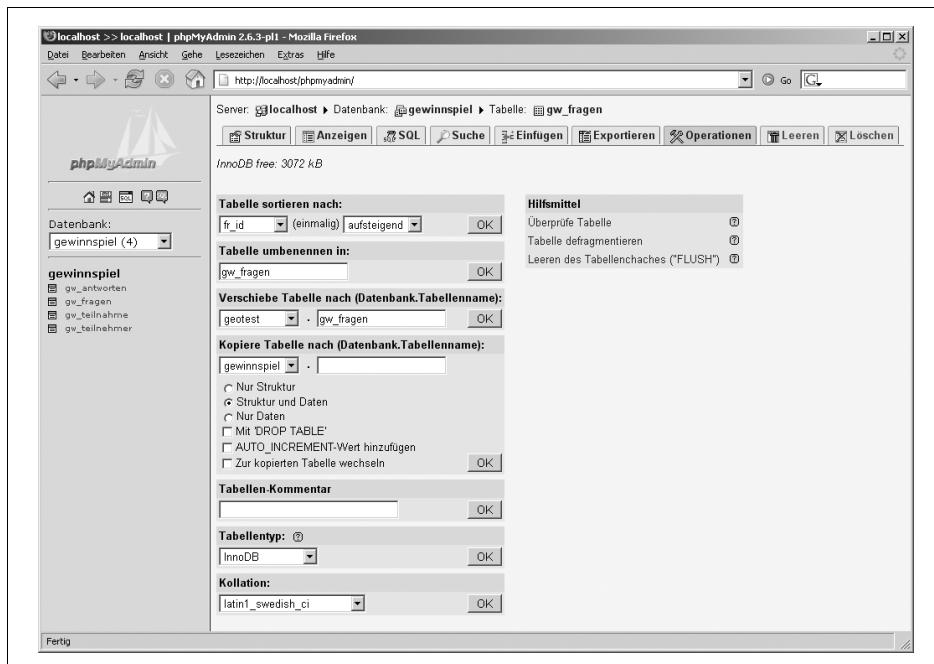


Abbildung 4-6: phpMyAdmin-Tabellenoperationen

Tabelle sortieren nach weist der Tabelle eine automatische Vorsortierung nach dem gewählten Kriterium zu. Das kann nützlich sein, um die Ausgabe der Tabelle in der gewünschten Standardsortierung zu beschleunigen. Mit Hilfe der Option *Tabelle umbenennen in* können Sie der Tabelle einen neuen Namen zuweisen. *Verschiebe Tabelle nach* geht über das Umbenennen innerhalb derselben Datenbank hinaus und ermöglicht auch den Umzug der Tabelle in eine andere Datenbank auf demselben MySQL-Server.

Die Option *Kopiere Tabelle nach* besitzt mehr Einstellmöglichkeiten als die bisher genannten Punkte: Unter der Angabe der Zieldatenbank und -tabelle können Sie bestimmen, ob *Nur Struktur*, *Struktur und Daten* oder *Nur Daten* kopiert werden sollen. Letzteres erfordert als Ziel eine Tabelle mit identischen Spaltendefinitionen. Die Zusatzoption *Mit 'DROP TABLE'* löscht eine eventuell vorhandene Tabelle gleichen Namens. *AUTO_INCREMENT-Wert hinzufügen* erstellt einen nummerierten Schlüssel für die neuen Daten. *Zur kopierten Tabelle wechseln* zeigt nach getaner Arbeit die soeben erstellte Kopie an.

Als Nächstes können Sie den *Tabellen-Kommentar* – eine beliebige Beschreibung– ändern. Darunter lässt sich der Typ der Tabelle nachträglich modifizieren. Näheres über die verschiedenen MySQL-Tabellentypen erfahren Sie im nächsten Kapitel. Zuletzt kann die voreingestellte *Kollation* der Tabelle geändert werden; Spalten, die eine eigene Einstellung besitzen, sind davon nicht betroffen.



Für die gesamte Datenbank gibt es eine vergleichbare Seite. Sie erreichen sie, wenn Sie den Link *Operationen* auf der Übersichtsseite einer Datenbank anklicken. Dort können Sie eine neue Tabelle erstellen und die Datenbank umbenennen oder kopieren. Außerdem lässt sich die Kollation der Datenbank selbst ändern, was nur Tabellen betrifft, für die keine explizite Kollation definiert wurde.

Datenbanken und Tabellen erstellen

Wechseln Sie auf die Startseite (Haus-Symbol im linken Frame), wenn Sie eine neue Datenbank anlegen möchten. Sie müssen einen Namen für die Datenbank angeben; natürlich darf noch keine Datenbank mit dem angegebenen Namen existieren. Zusätzlich können Sie eine *Kollation*, also eine Standard-Sortierreihenfolge, auswählen. Sobald Sie auf *OK* klicken, erscheint die Übersichtsseite der neuen Datenbank – natürlich noch ohne Tabellen.

Als Beispiel soll eine separate Datenbank für eine Promotion-Aktion erstellt werden: eine Liste kostenloser Zusatzleistungen, die ab einem angegebenen Buchungspreis gewährt werden. Im nächsten Kapitel wird die entsprechende Tabelle in eine andere Datenbank übernommen. Geben Sie den Datenbanknamen *promo* ein. Für die Sprache Deutsch gibt es zwei verschiedene Kollationen: *latin1_german1_ci* sortiert gemäß Wörterbuch, so dass die Umlaute *ä*, *ö* und *ü* mit den Buchstaben *a*, *o* und *u* gleichgesetzt werden. *latin1_german2_ci* ist dagegen die Telefonbuch-Reihenfolge, die die Umlaute als *ae*, *oe* und *ue* einsortiert. Im vorliegenden Fall sollten Sie *latin1_german1_ci* wählen.

Im nächsten Schritt können Sie die erste Tabelle in der neuen Datenbank erstellen. Geben Sie dazu unter *Neue Tabelle in der Datenbank promo erstellen* einen Tabellennamen (hier *pr_angebote*) und die Anzahl der Datenfelder beziehungsweise Spalten (3) ein. Nach einem Klick auf *OK* wird eine Seite zur Definition der Tabellenspalten angezeigt (siehe Abbildung 4-7).

Das Eingabeformular enthält für jedes Feld die folgenden Eingabemöglichkeiten:

- *Feld*: Name der Spalte
- *Typ*: Datentyp der Spalte – eine genaue Beschreibung aller SQL-Datentypen folgt im nächsten Kapitel
- *Länge/Set*: Byte- oder Zeichenanzahl des Datentyps beziehungsweise Werteliste von Aufzählungstypen

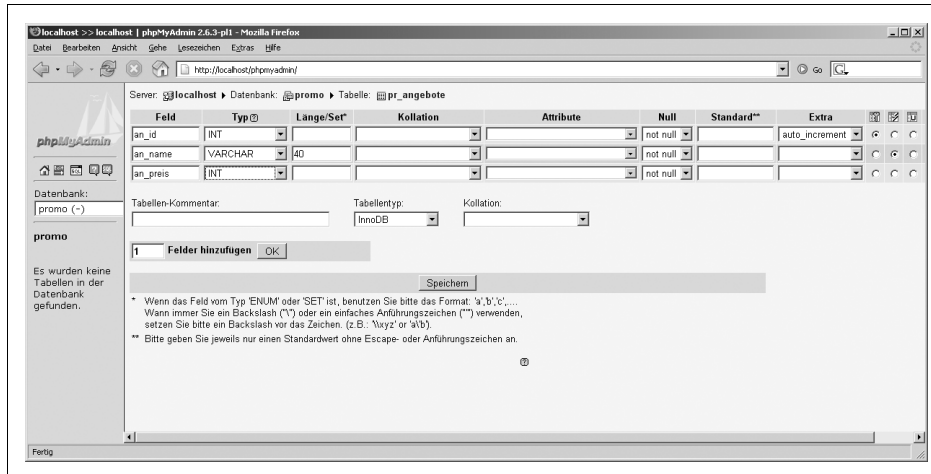


Abbildung 4-7: Definition der Felder einer neuen Tabelle in phpMyAdmin

- **Kollation:** Sortierreihenfolge gemäß Zeichensatz und Sprache
- **Attribute** – Auswahlmöglichkeiten für Zahlentypen: *UNSIGNED* (vorzeichenlos) und *UNSIGNED ZEROFILL* (vorzeichenlos, führende Stellen bis zur Gesamtbreite mit Nullen auffüllen)
- **Null:** Gibt an, ob das Feld leer bleiben darf (*null*) oder nicht (*not null*, Standard)
- **Standard:** Automatischer Vorgabewert für Felder der Spalte
- **Extras:** Auswahl des Attributs *auto_increment* für automatisch durchnummerierte (Schlüssel-)Felder
- **Indizes** zur Auswahl: *Primärschlüssel*, normaler *Index*, *unique* (jedes Feld der Spalte muss einen einmaligen Wert besitzen), kein Index (Standard), *Volltext* (nur in MyISAM-Tabellen möglich)

Füllen Sie das Formular gemäß den Angaben in Tabelle 4-2 aus.

Tabelle 4-2: Spalten der neuen Tabelle *pr_angebote*

Spaltenname	Datentyp	Weitere Optionen
<i>an_id</i>	INT	<i>auto_increment</i> , Primärschlüssel
<i>an_name</i>	VARCHAR	Länge 40, Index
<i>an_preis</i>	INT	–

Den *Tableentyp* können Sie auf *Standard* (ergibt unter Unix *MyISAM* und unter Windows *InnoDB*) stehen lassen. Wenn Sie möchten, können Sie aber auch explizit *MyISAM* wählen, da für diese Tabelle keine Transaktionen vorgesehen sind. Zusätzlich können Sie einen beliebigen *Kommentar* für die Tabelle eingeben sowie

eine *Kollation* auswählen. Letzteres ist hier nicht erforderlich, da die Kollation der Datenbank automatisch für alle Tabellen gilt, die keine eigene Einstellung besitzen. Klicken Sie zum Schluss auf *Speichern*, um die Einstellungen zu akzeptieren. Es erscheint die Strukturseite der Tabelle; zusätzlich wird die soeben ausgeführte SQL-Abfrage zur Tabellenerstellung angezeigt.

Der letzte Schritt besteht natürlich darin, Datensätze in die Tabelle einzufügen. Die klassische Methode ist die manuelle Eingabe. Klicken Sie dazu auf den Link *Einfügen* am oberen Fensterrand. Es erscheint eine Eingabeseite wie in Abbildung 4-8.

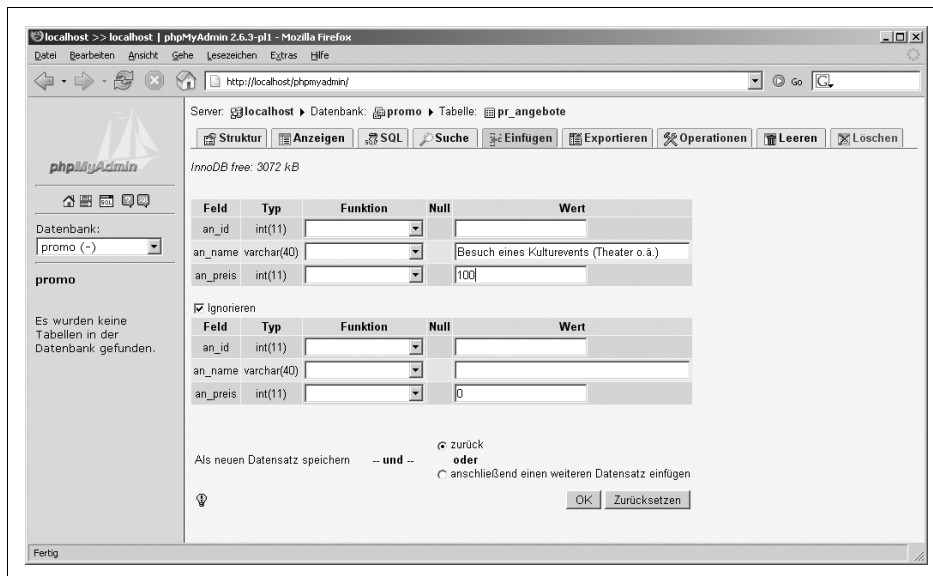


Abbildung 4-8: Eingabe eines neuen Datensatzes in phpMyAdmin

Hier können Sie jeweils einen Datensatz eingeben; nach Entfernen der Auswahl *Ignorieren* auch zwei. Wenn die Eingabe beendet ist, können Sie sich aussuchen, ob Sie mittels *zurück* zur vorigen Ansicht wechseln oder *anschließend einen weiteren Datensatz einfügen* möchten. Geben Sie die in Tabelle 4-3 gezeigten Datensätze ein, indem Sie vor dem Klick auf *OK* wählen, dass Sie einen weiteren Datensatz eingeben möchten. Nach der ersten Auswahl dieser Option sollte sie automatisch bestehen bleiben. Das nicht angegebene Feld *an_id* können (und sollten!) Sie jeweils leer lassen, da es per *auto_increment* automatisch ausgefüllt wird.

Tabelle 4-3: Datensätze für die Tabelle *pr_angebote*

an_name	an_preis
Besuch eines Kulturevents (Theater o.ä.)	100
Abendessen im 5-Sterne-Restaurant	150

Tabelle 4-3: Datensätze für die Tabelle *pr_angebote* (Fortsetzung)

an_name	an_preis
Kostenlose Stadtrundfahrt	200
ÖPNV-Ticket für Aufenthaltsdauer	250
Kostenloser Mietwagen (2 Tage)	300



Die angesprochenen Funktionen zur nachträglichen Änderung von Tabellenstrukturen und -inhalten sind übrigens weitgehend mit den hier vorgestellten Neueingabeseiten identisch – natürlich mit dem einen Unterschied, dass die Eingabefelder bereits Werte enthalten, die Sie ändern können.

