



Marco Emrich

# *Datenbanken & SQL für Einsteiger*

*Datenbankdesign und MySQL in der Praxis*

**Ein Webmasters Press Lernbuch**

Version 3.8.0 vom 23.04.2019

Autorisiertes Curriculum für das Webmasters Europe Ausbildungs- und Zertifizierungsprogramm

# Inhaltsverzeichnis

<b>Vorwort</b>	11
<b>1 Einführung</b>	12
1.1 Allgemeines	12
1.2 Auf den Schultern von Riesen...	12
1.3 Voraussetzungen	14
<b>2 Datenbanken</b>	15
2.1 Warum Datenbanken?	15
2.2 Datenbanken im Web	15
2.2.1 Webanwendungen	16
2.2.2 Web-Komponenten	17
2.3 Persistenz	17
2.4 Was ist eine Datenbank?	18
2.4.1 Struktur	18
2.4.2 Datenbankmodelle	18
2.4.3 Das Datenbank-Management-System	19
2.5 Anforderungen an ein DBMS	20
2.6 Verbreitete RDBMS	21
2.7 MySQL	22
2.8 Das Client-Server-Prinzip	23
2.9 Datenbank-Clients	24
2.10 Zusammenfassung	25
2.11 Testen Sie Ihr Wissen	25
<b>3 Datenbanken in der Praxis</b>	27
3.1 Einrichten einer Entwicklungsumgebung	27
3.2 Starten und Stoppen von MySQL	28
3.3 SQL	29
3.4 Was ist nun eigentlich SQL?	32
3.5 BNF	32
3.6 Konfiguration	34
3.7 Zusammenfassung	34
3.8 Testen Sie Ihr Wissen	34
3.9 Übungen	35
<b>4 Datenbankdesign: Das Domänenmodell</b>	36
4.1 Ein Kundengespräch	36
4.2 Das Domänenmodell	38
4.2.1 UML	39
4.2.2 Klassendiagramm	39
4.3 Richtlinien für Klassendiagramme	41
4.4 Mehrere Klassen	41
4.5 Zusammenfassung	41
4.6 Testen Sie Ihr Wissen	42
4.7 Übungen	42

<b>5</b>	<b>Einführung in Draw.io</b>	44
5.1	Vorteile von UML-Werkzeugen	44
5.2	Anlegen eines neuen Projektes in Draw.io	45
5.3	Klassen in Draw.io	46
5.4	Zusammenfassung	47
5.5	Übungen	48
5.6	Zusatzübungen	48
<b>6</b>	<b>Datenbankdesign: Das physische Datenmodell</b>	50
6.1	Physisches Datenmodell	50
6.2	Unterschiede zum Domänenmodell	52
6.3	Vom Domänenmodell zum physischen Datenmodell	52
6.4	Zusammenfassung	53
6.5	Testen Sie Ihr Wissen	54
6.6	Übungen	55
6.7	Zusatzübungen	55
<b>7</b>	<b>Das Relationale Modell</b>	56
7.1	Was ist das relationale Modell?	56
7.2	Aspekte des relationalen Modells	56
7.3	Relationen	57
7.4	Was war nochmal eine Menge?	58
7.5	Mengenschreibweise	59
7.6	Konsequenzen des Relationsbegriffs	59
7.7	Zusammenfassung	61
7.8	Testen Sie Ihr Wissen	62
7.9	Übungen	62
7.10	Zusatzübungen	63
<b>8</b>	<b>Datentypen</b>	64
8.1	Datentypen	64
8.2	Datentypen in MySQL	65
8.2.1	Zeichenketten	65
8.2.2	Zahlen	66
8.2.3	Zeit und Datum	66
8.2.4	Wahrheitswerte	67
8.3	Zurück zum physischen Datenmodell	67
8.4	Zusammenfassung	68
8.5	Testen Sie Ihr Wissen	68
8.6	Übungen	69
8.7	Zusatzübungen	70
<b>9</b>	<b>SQL: Tabellen anlegen und löschen</b>	71
9.1	Tabellen anlegen	71
9.2	DML, DDL und DCL	73
9.3	BNF: Wiederholung und Verschachtelung	73
9.4	Löschen von Tabellen	74
9.5	Zusammenfassung	75
9.6	Testen Sie Ihr Wissen	76
9.7	Übungen	76
9.8	Zusatzübungen	77

<b>10</b>	<b>SQL: Datensätze einfügen und auslesen</b>	78
10.1	Daten einfügen	78
10.2	Daten anzeigen	79
10.3	Zusammenfassung	80
10.4	Testen Sie Ihr Wissen	80
10.5	Übungen	81
10.6	Zusatzübungen	82
<b>11</b>	<b>Schlüssel</b>	83
11.1	Was ist ein Schlüssel?	83
11.2	Schlüssel finden	86
11.3	Primär- und Alternativschlüssel	87
11.4	Künstliche Schlüssel	88
11.4.1	Eigenschaften guter Primärschlüssel	88
11.4.2	Tabellen ohne geeigneten Primärschlüssel	89
11.4.3	Ids im objekt-relationalen Mapping	90
11.5	Zusammenfassung	91
11.6	Testen Sie Ihr Wissen	91
11.7	Übungen	92
11.8	Zusatzübungen	92
<b>12</b>	<b>Schlüssel in SQL</b>	93
12.1	Primärschlüssel anlegen	93
12.2	Alternativschlüssel anlegen	94
12.3	Autoincrement	95
12.4	Schlüssel in BNF	96
12.5	Zusammenfassung	97
12.6	Testen Sie Ihr Wissen	97
12.7	Übungen	98
12.8	Zusatzübungen	98
<b>13</b>	<b>SQL: Ändern bestehender Tabellen</b>	100
13.1	DDL-Änderungen ohne Datenverlust	100
13.2	Spalten hinzufügen mit ADD	101
13.3	Spalten löschen mit DROP	102
13.4	Spalten ändern oder umbenennen mit CHANGE	103
13.5	Tabellen umbenennen mit RENAME	103
13.6	BNF von ALTER	104
13.7	Zusammenfassung	104
13.8	Testen Sie Ihr Wissen	105
13.9	Übungen	105
13.10	Zusatzübungen	106
<b>14</b>	<b>Migrationen</b>	107
14.1	Versionieren des Datenbankschemas	107
14.2	Beispieldaten	109
14.3	Zusammenfassung	110
14.4	Übungen	110
14.5	Zusatzübungen	110

<b>15</b>	<b>SQL: Datensätze gezielt auslesen</b>	112
15.1	SELECT mit Spaltenangabe	112
15.2	Doppelte Datensätze entfernen	113
15.3	Zeilen filtern	114
15.4	Vergleichsoperatoren	114
15.5	Suche mit LIKE	115
15.6	Logische Operatoren	116
15.7	BNF von SELECT	118
15.8	SELECT-FROM-WHERE-Block	118
15.9	Zusammenfassung	118
15.10	Testen Sie Ihr Wissen	119
15.11	Übungen	119
<b>16</b>	<b>Ausdrücke in SQL</b>	121
16.1	Ausdrücke im SELECT	121
16.2	Ausdruck ohne Quelltable	122
16.3	Ausdrücke im WHERE-Teil	123
16.4	Operatoren	123
16.4.1	Arithmetische Operatoren	123
16.4.2	Logische und Vergleichsoperatoren	124
16.4.3	Präzedenz und Klammerung	125
16.5	Funktionen	125
16.5.1	Verschachteln von Funktionen	126
16.6	Umbenennen von Spaltenüberschriften	127
16.7	BNF von SELECT	128
16.8	Zusammenfassung	128
16.9	Testen Sie Ihr Wissen	129
16.10	Übungen	129
16.11	Zusatzübungen	130
<b>17</b>	<b>Aggregatsfunktionen</b>	131
17.1	Zählen von Zeilen	131
17.2	Minimum und Maximum	132
17.3	Durchschnitt und Summe	132
17.4	Zusammenfassung	132
17.5	Testen Sie Ihr Wissen	133
17.6	Übungen	133
<b>18</b>	<b>Datensätze ändern und löschen</b>	134
18.1	Ändern von Datensätzen	134
18.2	Ausdrücke im UPDATE	135
18.3	Löschen von Datensätzen	135
18.4	Zusammenfassung	136
18.5	Testen Sie Ihr Wissen	137
18.6	Übungen	137
18.7	Zusatzübungen	137
<b>19</b>	<b>Sortierung und LIMIT</b>	138
19.1	Sortierung	138
19.2	Sortierung nach mehreren Attributen	139
19.3	Auf- und absteigende Sortierung	140
19.4	Die BNF von ORDER BY	141

19.5	Limit	141
19.6	BNF von LIMIT	142
19.7	Zusammenfassung	142
19.8	Testen Sie Ihr Wissen	142
19.9	Übungen	143
<b>20</b>	<b>Die Null-Markierung</b>	<b>144</b>
20.1	NULL-Marker	144
20.2	NULL-Operatoren	145
20.3	NULL-Marker für »nicht existent«	146
20.4	NULL im relationalen Modell	147
20.5	Not-Null-Spalten	147
20.6	NOT NULL in UML	148
20.7	Vorgeschlagene Vorgehensweise	148
20.8	Zusammenfassung	149
20.9	Testen Sie Ihr Wissen	149
20.10	Übungen	150
20.11	Zusatzübungen	150
<b>21</b>	<b>Beziehungen im Domänenmodell</b>	<b>151</b>
21.1	Beziehungen modellieren und benennen	151
21.2	Kardinalität	152
21.3	Zusammenfassung	153
21.4	Testen Sie Ihr Wissen	154
21.5	Übungen	154
21.6	Zusatzübungen	154
<b>22</b>	<b>1:n-Beziehungen in SQL</b>	<b>155</b>
22.1	Fremdschlüssel	155
22.2	Die 1:n-Beziehung in SQL	157
22.3	JOIN und WHERE	158
22.4	BNF	159
22.5	Zusammenfassung	159
22.6	Testen Sie Ihr Wissen	159
22.7	Übungen	160
<b>23</b>	<b>n:m-Beziehungen in SQL</b>	<b>161</b>
23.1	Die Zwischentabelle	161
23.2	Die n:m-Beziehung in SQL	163
23.3	n:m-Beziehungen in Entities überführen	164
23.4	BNF	165
23.5	Zusammenfassung	166
23.6	Testen Sie Ihr Wissen	166
23.7	Übungen	166
23.8	Zusatzübungen	167
<b>24</b>	<b>Anhang A: Befehlsübersicht</b>	<b>168</b>
<b>25</b>	<b>Anhang B: Agilität in der Webentwicklung</b>	<b>170</b>

<b>26</b>	<b>Anhang C: Quellen &amp; Literaturhinweise</b>	176
26.1	APA-Style	176
26.2	Quellen	176
26.3	Empfehlungen	179
	<b>Lösungen der Übungsaufgaben</b>	181
	<b>Lösungen der Wissensfragen</b>	191
	<b>Index</b>	203

# Vorwort

Kaum noch eine Website kann es sich heutzutage erlauben, mit rein statischen Inhalten aufzuwarten. So ist im *Web 2.0* die Mitarbeit der Benutzer gefragt. Onlineshops aktualisieren ständig ihre Inhalte. Selbst anscheinend statische Webseiten verwenden meist ein ausgeklügeltes **CMS** (Content Management System), um die Pflege zu erleichtern.

Das moderne *World Wide Web* verarbeitet täglich eine gigantische Datenflut, die außerdem stetig wächst. Wohin also mit all den Daten?

Ein Konzept, das sich seit fast 40 Jahren als **das** Konzept der Datenhaltung schlechthin erwiesen hat, hat auch die Webentwicklung im Sturm erobert. Die Rede ist vom **relationalen Datenbankmodell**. Ein Modell, das allen Versuchen es abzulösen trotzte und bis heute als stabil, fundiert und ausgereift gilt. Wenn der führende Datenbankexperte *C. J. Date* recht behält, wird das relationale Datenbankmodell auch in 100 Jahren noch vorherrschend sein (Date 2005).

*Hundert Jahre in die Zukunft blickend, erwarte ich, dass Datenbanksysteme immer noch auf Codd's relationalem Modell basieren. Warum? Die Grundlagen des Modells – Mengenlehre und Prädikatenlogik – sind ihrerseits grundlegende. Elemente der Prädikatenlogik im Besonderen reichen gut 2000 Jahre zurück – wenigstens bis zu Aristoteles (384-322 v. Chr.).*

Keine Sorge, Sie lernen hier nicht nur 40 Jahre alte Konzepte, sondern gewinnen auch moderne Erkenntnisse. Erkenntnisse, die erst kürzlich den Schritt in den Mainstream der Softwareentwicklung geschafft haben und die den Unternehmen, die sie bereits anwenden, enorme Wettbewerbsvorteile bescheren. Dazu zählen beispielsweise Datenbank-Migrationen oder objekt-relationales Mapping. Auch die Datenbanktheorie lernen Sie in der aktuellsten Version kennen. Sie bekommen die Mittel an die Hand, die Sie benötigen, um erfolgreich moderne **datenbankgestützte Webanwendungen** zu entwickeln.

Ich werde Ihnen sowohl ein tiefes Verständnis des relationalen Datenbankmodells als auch alle wichtigen Handgriffe für die Praxis vermitteln. Sie werden von meiner langjährigen Berufserfahrung in der Webentwicklung profitieren und lernen, wie Sie typische Klippen gekonnt umschiffen.

Für Feedback, Fragen und auch Kritik habe ich immer ein offenes Ohr unter [m.emrich@webmasters.de](mailto:m.emrich@webmasters.de).

Ich freue mich, Sie nun in die spannende Welt hinter den Kulissen der datenbankgestützten Webanwendungen mitzunehmen.

Viel Spaß beim Durcharbeiten dieses Buches und viel Erfolg beim Bau Ihrer Webanwendungen!

Ihr Marco Emrich



# Datenbanken in der Praxis

# 3

## In dieser Lektion lernen Sie

- welche Programme Sie zur Datenbankentwicklung benötigen.
- wie Sie Datenbanken erstellen, auflisten und löschen.
- wie Sie in BNF jemanden an einen Löwen verfüttern.

## 3.1 Einrichten einer Entwicklungsumgebung

Damit Sie Datenbanken erstellen können, müssen Sie zumindest ein DBMS und einen Datenbank-Client installieren. MySQL ist sowohl unter der OpenSource-Lizenz *GPL* als auch kommerziell erhältlich. In diesem Buch setzen Sie das freie DBMS *MySQL* ein. *MySQL* ist mit über 5 Millionen aktiven Installationen (*Stand 2004*<sup>26</sup>) das am weitesten verbreitete Open-Source DBMS. Es ist eine kostengünstige und effektive Lösung für kleine und mittlere Projekte und wird oft sogar von großen Organisationen wie z.B. *Yahoo!*, *Slashdot* oder der *NASA* eingesetzt.



**Abb. 12** XAMPP  
Logo

Eine vollständige Entwicklungsumgebung besteht, neben dem DBMS, aus weiteren Paketen wie z.B. Webserver, Programmiersprachen, Editor und Datenbank-Client.

Die Installation und Konfiguration all dieser Werkzeuge, Sprachen etc. ist nicht immer ganz einfach. Deswegen gibt es **Distributionen**, die eine Auswahl an Paketen bündeln und somit sowohl eine einfache Installation als auch eine bereits angepasste Konfiguration bieten.

Ich empfehle Ihnen, eine der kostenlosen Distributionen **XAMPP** (für Windows) oder **MAMP** (für Mac OS X) zu verwenden. Beide Distributionen enthalten unter anderem folgende Pakete:

- *MySQL* (DBMS)
- *Apache* (Webserver)
- *PHP* (Programmiersprache)



**Abb. 13** MAMP  
Logo

Unter <http://www.apachefriends.org/en/xampp.html> erhalten Sie *XAMPP* und die dazugehörige Installationsanleitung, *MAMP* entsprechend unter [www.mamp.info](http://www.mamp.info)<sup>27</sup>.

Sollten Sie unter einem Debian-basierten Linux – wie z.B. *Ubuntu* – arbeiten, installieren Sie *MySQL* mittels: `apt-get install mysql-server mysql-client`.

26. [http://en.wikipedia.org/wiki/MySQL\\_AB](http://en.wikipedia.org/wiki/MySQL_AB)

27. <http://www.mamp.info>



### Achtung

XAMPP und MAMP sind Distributionen, die ausschließlich für die Entwicklung verwendet werden sollten. In Ihrer Basiskonfiguration erfüllen Sie nicht die Sicherheitsanforderungen, die notwendig sind, um einen Produktivbetrieb (insbesondere im Internet) zu gewährleisten.

Für den produktiven Betrieb von Webanwendungen sind meist Linuxserver im Einsatz. Es empfiehlt sich, im Produktivbetrieb eine Linuxinstallation mit den entsprechenden Paketen zu verwenden, die vorher sorgfältig manuell konfiguriert und auf Sicherheitsaspekte getestet wurde. Wenn Sie später Anwendungen bei einem Webhoster betreiben, wird dieser das für Sie übernehmen.

### Übung 1: MySQL-Installation

- Installieren Sie nun MySQL für Ihr Betriebssystem.

## 3.2 Starten und Stoppen von MySQL

unter Windows und MAC OS X

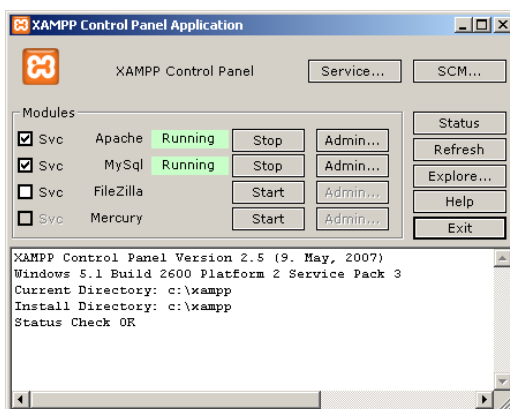


Abb. 14 Das XAMPP Control-Panel

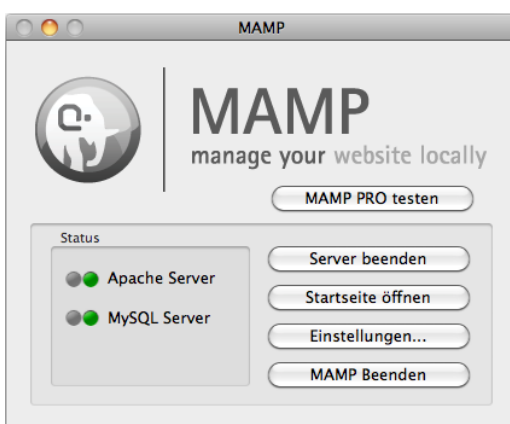


Abb. 15 Das MAMP Control-Panel

Sowohl MAMP als auch XAMPP enthalten ein Control Panel. Mit diesem können Sie den MySQL-Server jederzeit starten und stoppen.

unter Linux

Unter Linux können Sie MySQL mit dem Befehl

```
/etc/init.d/mysql start
```

starten und mit

```
/etc/init.d/mysql stop
```

wieder anhalten.

### Übung 2: MySQL-Server starten und stoppen

Probieren Sie nun das Starten und Stoppen aus! Achten Sie darauf, dass der Server nach der Übung wieder gestartet ist!

## 3.3 SQL

Innerhalb des DBMS gibt es mehrere Datenbanken. Wenn Sie wissen möchten, welche Datenbanken es bereits gibt, können Sie das DBMS (oder genauer den DB-Server) einfach fragen. Sie können eine Frage nicht auf Deutsch stellen. Verwenden Sie die Sprache, die der Server versteht. Praktischerweise verstehen fast alle modernen relationalen DBMS die Sprache *SQL*. Dazu benötigen Sie einen Client, der die Anfrage stellen kann. Später werden Sie Ihre eigenen Clients in Form von Webanwendungen programmieren. Im einfachsten Fall reicht uns zum Testen und Üben aber der von MySQL mitgelieferte Kommandozeilen-Client. Damit Sie diesen Client von überall aus aufrufen können, müssen Sie ihn noch in den Pfad der ausführbaren Dateien eintragen.

- **Linux/MySQL:** Geschieht automatisch. Sie müssen nichts tun.
- **MAC OS X/MAMP:** Fügen Sie den Pfad<sup>28</sup> `/Applications/MAMP/Library/bin/mysql` zu den Systempfaden hinzu. Fall Sie auf Schwierigkeiten stoßen, hilft Ihnen vielleicht dieser [Blogbeitrag](#)<sup>29</sup> weiter.
- **Windows/XAMPP:** Drücken Sie **win+Pause**, um die Systemeigenschaften zu öffnen. Wählen Sie den Reiter *Erweitert* bzw. *erweiterte Systemeinstellungen* und klicken Sie dort auf *Umgebungsvariablen*. Bearbeiten Sie die Variable `PATH` und ergänzen Sie `;%C:\xampp\mysql\bin`.

### Übung 3: MySQL-Client starten

Starten Sie jetzt den Kommandozeilenclient:

1. Öffnen Sie eine **Kommandozeile** in einem **Konsolen-** bzw. **Terminal-Fenster**.

**Windows:** Entweder per *Start > Programme > Zubehör > Eingabeaufforderung* oder durch Drücken der Taste **win+R** (run) und Eingabe von `cmd`.

**MAC OS X:** Öffnen Sie *Spotlight* mit der Tastenkombination **cmd-Space**. Geben Sie *Terminal* ein.

**Linux:** Unter Ubuntu dient die Tastenkombination **ctrl-alt-T** als Shortcut.

2. Starten Sie den MySQL-Client in der Konsole mittels `mysql -u root`.

`mysql` ist das Programm. Die Angabe `-u root` bedeutet, dass es unter dem Benutzer (user) *root* gestartet wird. Dieser Benutzer besitzt alle Rechte und ist bereits angelegt. Normalerweise benötigen

28. <https://sites.google.com/site/mamppro/de/mamp/howtos/mysql-ueber-die-kommandozeile-nutzen>

29. <http://macokratie.blogspot.com/2014/10/path-umgebungsvariable-shell-variable.html>

Sie noch ein Passwort. Nach der Installation ist das Passwort aber standardmäßig deaktiviert. Da Sie die Datenbank im Moment nur lokal zur Entwicklung einsetzen, stellt das kein Problem da. Für den Produktivbetrieb später müssen Sie selbstverständlich ein Passwort festlegen.

Sie sollten nun den MySQL-Prompt `mysql>` sehen, der es Ihnen erlaubt, SQL-Befehle einzugeben.

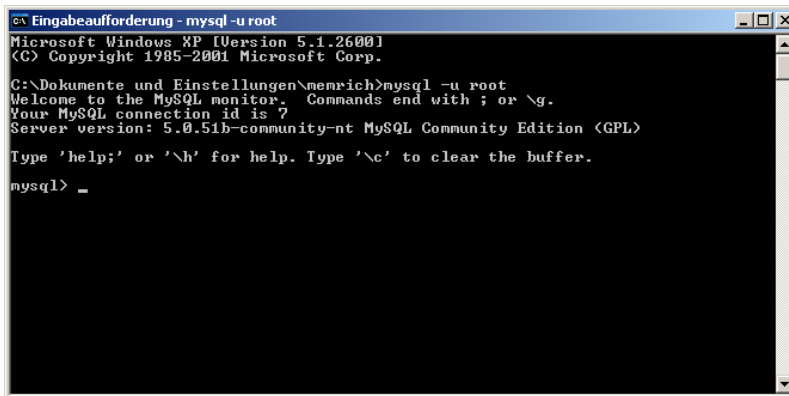


Abb. 16 MySQL-Kommandozeilen-Client

MySQL liefert bereits einige Datenbanken mit. Welche das sind, können Sie mit einem SQL-Befehl herausfinden.

#### Übung 4: Vorhandene Datenbanken auflisten

Geben Sie Ihren ersten SQL-Befehl ein:

```
SHOW DATABASES;
```

Achten Sie insbesondere darauf, dass Sie den Befehl mit einem Semikolon »;« abschließen. Sie sollten folgende Liste erhalten:

```
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| cdcol      |
| mysql     |
| phpmyadmin |
| test      |
| webauth   |
+-----+
6 rows in set (0.00 sec)
mysql> _
```

Abb. 17 Ausgabe von `show databases`

Falls Sie eine Datenbank mehr oder weniger als in Abb. 17 sehen, ist das nicht weiter schlimm. Die mitgelieferten Datenbanken können sich je nach MySQL-Version und Distribution (MAMP, XAMPP, ...) geringfügig unterscheiden.

In Zukunft werde ich Ihnen übrigens bei MySQL-Ausgaben statt eines Screenshots lediglich die reine Ausgabe zeigen – immer direkt nach der dazugehörigen SQL-Anweisung. Der Informationsgehalt ist der gleiche, die Ausgabe in Codeform ist aber besser lesbar. Die obige Ausgabe lässt sich also auch so darstellen:

```
SHOW DATABASES;
```

```

+-----+
| Database |
+-----+
| information_schema |
| cdcol |
| mysql |
| phpmyadmin |
| seminarverwaltung |
| test |
| webauth |
+-----+
7 rows in set (0.00 sec)

```

Ihr erstes Beispielprojekt ist eine *Seminarverwaltung* für eine private Akademie – mehr dazu erfahren Sie in der nächsten Lektion. Für diese Seminarverwaltung benötigen Sie eine eigene Datenbank.

### Übung 5: Datenbank erstellen

Legen Sie die Datenbank *seminarverwaltung* mit folgenden SQL-Befehl an:

```
CREATE DATABASE seminarverwaltung;
```

```
Query OK, 1 row affected (0.01 sec)
```

Ein SQL-Befehl wird auch als **Anfrage** (engl. **Query**) bezeichnet. Sie erhalten die Bestätigung `Query OK` immer dann, wenn der Befehl etwas verändert. Die Zeit in Klammern `(0.01 sec)` sagt Ihnen, dass das Anlegen der neuen Datenbank insgesamt gerade mal eine hundertstel Sekunde gedauert hat.

Um zu prüfen, ob es wirklich funktioniert hat, können Sie sich auch nochmals die vorhandenen Datenbanken anzeigen lassen.

### Übung 6: Datenbankerstellung prüfen

Geben Sie die vorhandenen Datenbanken aus:

```
SHOW DATABASES;
```

```

+-----+
| Database |
+-----+
| information_schema |
| cdcol |
| mysql |
| phpmyadmin |
| seminarverwaltung |
| test |
| webauth |
+-----+
7 rows in set (0.00 sec)

```

Sie sollten sehen, dass Ihre neue Datenbank *seminarverwaltung* nun dabei ist.

Bei dieser Gelegenheit können Sie noch die überflüssige Datenbank *cdcol* löschen. *cdcol* ist eine Beispieldatenbank. Sie benötigen sie nicht weiter.

### Übung 7: Datenbank Löschen

1. Löschen Sie die Datenbank *cdcol* mit folgenden Befehl:

```
DROP DATABASE cdcol; Query OK, 0 rows affected (0.00 sec)
```

2. Ein erneutes Auflisten der Datenbanken sollte zeigen, dass die *cdcol* tatsächlich gelöscht wurde.

```
SHOW DATABASES;
```

```
+-----+
| Database          |
+-----+
| information_schema |
| mysql             |
| phpmyadmin        |
| seminarverwaltung |
| test              |
| webauth           |
+-----+
6 rows in set (0.00 sec)
```

## 3.4 Was ist nun eigentlich SQL?

Kurz gesagt ist SQL eine sogenannte **Abfragesprache**. Sie wird von vielen DBMS verwendet, um mit einer Anwendung (d.h. einem Client) zu kommunizieren. Abfragesprache bedeutet, dass Sie damit das DBMS ansteuern können. Sie können Daten in jede Richtung übertragen oder Befehle zur Verwaltung der Daten absenden. Auch Konfigurationseinstellungen lassen sich per SQL übermitteln.

Heute wird SQL als Abkürzung für **Structured Query Language** gedeutet. Ursprünglich kommt der Begriff aber von **SEQUEL** (Structured English Query Language bzw. Nachfolger). Die Buchstaben dazwischen wurden wegen einer Markenrechtsverletzung gestrichen.

SQL ist aber keine Programmiersprache – Sie können keine vollständigen Anwendungen in SQL schreiben. Dazu müssen Sie einen Client in einer anderen Sprache programmieren (wie z. B. PHP oder Ruby). Von dieser Sprache aus greifen Sie dann mit Hilfe von eingebundenem SQL auf das DBMS zu.

Es gibt einen SQL-Standard, der die Sprache grundsätzlich beschreibt. Kein DBMS hält sich aber genau an den Standard. Stattdessen verwendet jedes DBMS seinen eigenen Dialekt. Der lässt dann meist große Teile des Standards außer Acht, bietet dafür aber viele eigene Erweiterungen. Dies gilt insbesondere auch für MySQL.

## 3.5 BNF

Um SQL-Befehle in einer allgemeinen Form angeben zu können, verwende ich die sogenannte **BNF**. BNF steht für **Backus-Naur-Form**, nach den Namen der beiden Entwickler. Es handelt sich dabei um eine sogenannte **Metasprache**, mit der sich andere Sprachen, wie z. B. Programmiersprachen beschreiben und definieren lassen. Sie lernen im Laufe dieses Buches immer mal wieder neue BNF-Symbole kennen. Von der BNF gibt es viele verschiedene Versionen. Einen guten Überblick finden Sie in [Braun \(2007\)](#)<sup>30</sup>. Dieses Buch verwendet die BNF-Version von MySQL. Dadurch können Sie jederzeit das offi-

30. <http://www-cgi.uni-regensburg.de/~brf09510/grammartypes.html>

zielle MySQL-Handbuch (MySQL AB, 2008)<sup>31</sup> Version ist angelehnt an den ISO/IEC 14077-Standard der BNF, der auch im SQL-Standard Verwendung findet.

In der einfachsten Form lässt sich ein Befehl durch zwei Konzepte beschreiben – durch **Schlüsselwörter** und andere fest-definierte Anweisungen auf der einen Seite und **Platzhalter** auf der anderen. Schlüsselwörter und andere fest-definierte Anweisungen müssen Sie exakt notieren. Für Platzhalter dürfen Sie eigene Werte einsetzen. Zeichen, die Sie direkt angeben, nennt die BNF **Terminale**. Das bedeutet, sie sind »zu Ende«, im Sinne von »sie werden nicht weiter ersetzt«. Die Platzhalter, die Sie später ersetzen, nennt die BNF dementsprechend **Nichtterminale** oder **Variablen**. Zur Unterscheidung setzen Sie einfach Groß- und Kleinschreibung ein. Terminale schreiben Sie groß, Nichtterminale dagegen klein.

### Beispiel

Betrachten Sie etwa folgenden BNF-Ausdruck:

```
ICH WERDE DICH DEN gefrässige_tierart ZUM FRAß VORWERFEN.
```

Der folgende BNF-Ausdruck etwa könnte wahlweise von einem wütenden Römer oder einem wütenden Bauern stammen. Es kommt eben darauf an, ob Sie für `gefrässige_tierart` Löwen oder Schweine einsetzen. `gefrässige_tierart` ist hier ein Nichtterminal, alle anderen Wörter dieses Ausdrucks sind Terminale. Es lassen sich beliebig viele Nichtterminale in einem Ausdruck verwenden, wie z.B. in folgendem:

```
HEUTE ABEND GEHE ICH ZU MEINEM FREUND name_eines_freundes NACH ort AUF DIE FEIER,  
ANLÄSSLICH SEINES freudiges_ereignis UND WERDE ZIEMLICH VIEL getränk TRINKEN.
```

Beachten Sie, dass die Variablen ein Underscore-Zeichen »\_« verwenden, um einzelne Wörter zu verbinden. Sie können kein Leerzeichen verwenden, da das Wort sonst als zwei Variablen gelesen wird. Schließlich könnten zwei Variablen direkt aufeinander folgen.

### Beispiel

```
ZU MEINEM GEBURTSTAG WÜNSCHE ICH MIR EIN farbe fahrzeugtyp.
```

Diese Variablen können Sie z. B. mit *rotes Fahrrad* oder *blaues Auto* ausprägen.

### Datenbanken zeigen, anlegen und löschen in BNF

Da Sie die BNF nun kennen, kann ich Ihnen auch die BNF-Form der drei SQL-Befehle zeigen, die Sie schon verwendet haben.

Bestehende Datenbanken anzeigen:

```
SHOW DATABASES;
```

Hier gibt es keinen Unterschied zum SQL, da Sie nur Schlüsselwörter verwenden.

Eine Datenbank anlegen:

```
CREATE DATABASE db_name;
```

Eine Datenbank löschen:

---

31. <http://dev.mysql.com/doc>

```
DROP DATABASE db_name;
```

### 3.6 Konfiguration

Um von Anfang an durchgängig Internationalisierung zu ermöglichen, empfehle ich Ihnen, MySQL auf die Zeichenkodierung *UTF-8* umzustellen. Außerdem sollten Sie den SQL-Modus umstellen, damit sich MySQL möglichst konform zum SQL-Standard verhält.

#### Übung 8:

1. **Windows/XAMPP, Linux:** Öffnen Sie die MySQL-Konfigurationsdatei in einem Texteditor. Sie finden diese Datei entweder unter *C:\xampp\mysql\bin\my.ini* (XAMPP), oder unter */etc/mysql/my.cnf* (Linux).

**Mac OS X/MAMP:** Legen Sie die Datei *my.cnf* unter */Applications/MAMP/conf* neu an.

2. Tragen Sie folgende Konfigurationen ein:

```
[mysqld] character-set-server=utf8 sql-mode="TRADITIONAL"
```

3. Starten und Stoppen Sie den MySQL-Server.

### 3.7 Zusammenfassung

- MAMP und XAMPP helfen Ihnen dabei, schnell eine funktionierende Entwicklungsumgebung einzurichten.
- Verwenden Sie SQL, um mit dem DBMS zu kommunizieren.
- Die BNF ist eine Metasprache, die die Syntax eines SQL-Befehls darstellen kann. Lesen Sie BNF-Ausdrücke, wenn Sie neue SQL-Befehle verstehen möchten!

#### BNF

Konzept	Darstellung
Terminale	Wort in Großbuchstaben
Nichtterminale (Variablen)	Wort in Kleinbuchstaben

**Tabelle 3.1** Bisherige BNF-Konzepte

#### SQL

Zweck	BNF
Datenbank anlegen	CREATE DATABASE db_name
Datenbank löschen	DROP DATABASE db_name
Datenbanken auflisten	SHOW DATABASES

**Tabelle 3.2** Bisheriges SQL

### 3.8 Testen Sie Ihr Wissen

1. Was ist SQL?



*Bitte ankreuzen:*

- eine Programmiersprache
- eine Datenbank-Abfragesprache.
- ein DBMS
- eine Datenbank

2. Eine Bündelung von verschiedenen Software-Paketen mit gemeinsamen Installer nennt man?

*Bitte ankreuzen:*

- Distribution
- Packet
- Repository
- Datenbank

3. Symbole in der BNF, die sich weiter ersetzen lassen, heißen...

*Bitte ankreuzen:*

- Terminale
- Nichtterminale
- Variablen
- Operatoren

4. Welchen Bezug hat MySQL zu SQL?

## 3.9 Übungen

### Übung 9: BNF ausprägen

Entwickeln Sie jeweils zwei Ausprägungen der folgenden BNF-Ausdrücke:

- ▶ farbe IST MEINE LIEBLINGSFARBE.
- ▶ IN MEINEM NÄCHSTEN JOB WERDE ICH MINDESTENS gehalts\_betrag währung VERDIENEN.

### Übung 10: Datenbanken Löschen und Anlegen

1. Zeigen Sie erneut alle Datenbanken an.
2. Löschen Sie die Datenbank `test`.
3. Legen Sie folgende neuen Datenbanken an:
  - fluggesellschaft
  - filmverleih
  - partnervermittlung
4. Prüfen Sie, ob das Anlegen der neuen Datenbanken gelungen ist.