



Marco Emrich

Datenbanken & SQL für Einsteiger

Datenbankdesign und MySQL in der Praxis

Ein Webmasters Press Lernbuch

Version 3.9.3 vom 03.06.2020

Autorisiertes Curriculum für das Webmasters Europe Ausbildungs- und Zertifizierungsprogramm

Inhaltsverzeichnis

Vorwort	11
1 Einführung	12
1.1 Aufbau	12
1.2 Auf den Schultern von Riesen...	12
1.3 Voraussetzungen	13
2 Datenbanken	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?	17
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	24
2.11 Testen Sie Ihr Wissen	25
3 Datenbanken in der Praxis	26
3.1 Einrichten einer Entwicklungsumgebung	26
3.2 Starten und Stoppen von MySQL	27
3.3 SQL	27
3.4 Was ist nun eigentlich SQL?	31
3.5 BNF	31
3.6 Konfiguration	33
3.7 Zusammenfassung	33
3.8 Testen Sie Ihr Wissen	34
3.9 Übungen	34
4 Datenbankdesign: Das Domänenmodell	35
4.1 Ein Kundengespräch	35
4.2 Das Domänenmodell	37
4.2.1 UML	38
4.2.2 Klassendiagramm	38
4.3 Richtlinien für Klassendiagramme	40
4.4 Mehrere Klassen	40
4.5 Zusammenfassung	40
4.6 Testen Sie Ihr Wissen	41
4.7 Übungen	41

5	Einführung in Draw.io	43
5.1	Vorteile von UML-Werkzeugen	43
5.2	Anlegen eines neuen Projektes in Draw.io	44
5.3	Klassen in Draw.io	45
5.4	Zusammenfassung	46
5.5	Übungen	47
5.6	Zusatzübungen	47
6	Datenbankdesign: Das physische Datenmodell	49
6.1	Physisches Datenmodell	49
6.2	Unterschiede zum Domänenmodell	51
6.3	Vom Domänenmodell zum physischen Datenmodell	51
6.4	Zusammenfassung	52
6.5	Testen Sie Ihr Wissen	53
6.6	Übungen	53
6.7	Zusatzübungen	53
7	Das Relationale Modell	54
7.1	Was ist das relationale Modell?	54
7.2	Aspekte des relationalen Modells	54
7.3	Relationen	55
7.4	Was war nochmal eine Menge?	56
7.5	Mengenschreibweise	57
7.6	Konsequenzen des Relationsbegriffs	57
7.7	Zusammenfassung	59
7.8	Testen Sie Ihr Wissen	60
7.9	Übungen	61
7.10	Zusatzübungen	61
8	Datentypen	63
8.1	Datentypen	63
8.2	Datentypen in MySQL	63
8.2.1	Zeichenketten	64
8.2.2	Zahlen	65
8.2.3	Zeit und Datum	65
8.2.4	Wahrheitswerte	66
8.3	Zurück zum physischen Datenmodell	66
8.4	Zusammenfassung	66
8.5	Testen Sie Ihr Wissen	67
8.6	Übungen	67
8.7	Zusatzübungen	68
9	SQL: Tabellen anlegen und löschen	69
9.1	Tabellen anlegen	69
9.2	DML, DDL und DCL	71
9.3	BNF: Wiederholung und Verschachtelung	71
9.4	Löschen von Tabellen	72
9.5	Zusammenfassung	73
9.6	Testen Sie Ihr Wissen	74
9.7	Übungen	74
9.8	Zusatzübungen	75

10	SQL: Datensätze einfügen und auslesen	76
10.1	Daten einfügen	76
10.2	Daten anzeigen	77
10.3	Zusammenfassung	78
10.4	Testen Sie Ihr Wissen	78
10.5	Übungen	79
10.6	Zusatzübungen	80
11	Schlüssel	81
11.1	Was ist ein Schlüssel?	81
11.2	Schlüssel finden	84
11.3	Primär- und Alternativschlüssel	85
11.4	Künstliche Schlüssel	86
11.4.1	Eigenschaften guter Primärschlüssel	86
11.4.2	Tabellen ohne geeigneten Primärschlüssel	87
11.4.3	Ids im objekt-relationalen Mapping	88
11.5	Zusammenfassung	88
11.6	Testen Sie Ihr Wissen	88
11.7	Übungen	89
11.8	Zusatzübungen	90
12	Schlüssel in SQL	91
12.1	Primärschlüssel anlegen	91
12.2	Alternativschlüssel anlegen	92
12.3	Autoincrement	93
12.4	Schlüssel in BNF	94
12.5	Zusammenfassung	95
12.6	Testen Sie Ihr Wissen	96
12.7	Übungen	96
12.8	Zusatzübungen	97
13	SQL: Ändern bestehender Tabellen	98
13.1	DDL-Änderungen ohne Datenverlust	98
13.2	Spalten hinzufügen mit ADD	99
13.3	Spalten löschen mit DROP	100
13.4	Spalten ändern oder umbenennen mit CHANGE	101
13.5	Tabellen umbenennen mit RENAME	101
13.6	BNF von ALTER	102
13.7	Zusammenfassung	102
13.8	Testen Sie Ihr Wissen	103
13.9	Übungen	103
13.10	Zusatzübungen	104
14	Migrationen	105
14.1	Versionieren des Datenbankschemas	105
14.2	Beispieldaten	107
14.3	Zusammenfassung	108
14.4	Übungen	108
14.5	Zusatzübungen	108

15	SQL: Datensätze gezielt auslesen	110
15.1	SELECT mit Spaltenangabe	110
15.2	Doppelte Datensätze entfernen	111
15.3	Zeilen filtern	112
15.4	Vergleichsoperatoren	112
15.5	Suche mit LIKE	113
15.6	Logische Operatoren	114
15.7	BNF von SELECT	116
15.8	SELECT-FROM-WHERE-Block	116
15.9	Zusammenfassung	116
15.10	Testen Sie Ihr Wissen	117
15.11	Übungen	117
16	Ausdrücke in SQL	119
16.1	Ausdrücke im SELECT	119
16.2	Ausdruck ohne Quelltable	120
16.3	Ausdrücke im WHERE-Teil	121
16.4	Operatoren	121
16.4.1	Arithmetische Operatoren	121
16.4.2	Logische und Vergleichsoperatoren	122
16.4.3	Präzedenz und Klammerung	123
16.5	Funktionen	123
16.5.1	Verschachteln von Funktionen	124
16.6	Umbenennen von Spaltenüberschriften	125
16.7	BNF von SELECT	126
16.8	Zusammenfassung	126
16.9	Testen Sie Ihr Wissen	127
16.10	Übungen	128
16.11	Zusatzübungen	129
17	Aggregatsfunktionen	130
17.1	Zählen von Zeilen	130
17.2	Minimum und Maximum	131
17.3	Durchschnitt und Summe	131
17.4	Zusammenfassung	131
17.5	Testen Sie Ihr Wissen	132
17.6	Übungen	132
18	Datensätze ändern und löschen	133
18.1	Ändern von Datensätzen	133
18.2	Ausdrücke im UPDATE	134
18.3	Löschen von Datensätzen	134
18.4	Zusammenfassung	135
18.5	Testen Sie Ihr Wissen	136
18.6	Übungen	136
18.7	Zusatzübungen	136
19	Sortierung und LIMIT	137
19.1	Sortierung	137
19.2	Sortierung nach mehreren Attributen	138
19.3	Auf- und absteigende Sortierung	139
19.4	Die BNF von ORDER BY	140

19.5	Limit	140
19.6	BNF von LIMIT	141
19.7	Zusammenfassung	141
19.8	Testen Sie Ihr Wissen	141
19.9	Übungen	142
20	Die Null-Markierung	143
20.1	NULL-Marker	143
20.2	NULL-Operatoren	144
20.3	NULL-Marker für »nicht existent«	145
20.4	NULL im relationalen Modell	146
20.5	Not-Null-Spalten	146
20.6	NOT NULL in UML	147
20.7	Vorgeschlagene Vorgehensweise	147
20.8	Zusammenfassung	148
20.9	Testen Sie Ihr Wissen	148
20.10	Übungen	148
20.11	Zusatzübungen	149
21	Beziehungen im Domänenmodell	150
21.1	Beziehungen modellieren und benennen	150
21.2	Kardinalität	151
21.3	Zusammenfassung	152
21.4	Testen Sie Ihr Wissen	153
21.5	Übungen	153
21.6	Zusatzübungen	153
22	1:n-Beziehungen in SQL	154
22.1	Fremdschlüssel	154
22.2	Die 1:n-Beziehung in SQL	156
22.3	JOIN und WHERE	157
22.4	BNF	158
22.5	Zusammenfassung	158
22.6	Testen Sie Ihr Wissen	158
22.7	Übungen	158
23	n:m-Beziehungen in SQL	160
23.1	Die Zwischentabelle	160
23.2	Die n:m-Beziehung in SQL	162
23.3	n:m-Beziehungen in Entities überführen	163
23.4	BNF	164
23.5	Zusammenfassung	165
23.6	Testen Sie Ihr Wissen	165
23.7	Übungen	165
23.8	Zusatzübungen	165
24	Anhang A: Befehlsübersicht	167
25	Anhang B: Agilität in der Webentwicklung	169

26	Anhang C: Quellen & Literaturhinweise	175
26.1	APA-Style	175
26.2	Quellen	175
26.3	Empfehlungen	178
	Lösungen der Übungsaufgaben	180
	Lösungen der Wissensfragen	190
	Index	200

Vorwort

Kaum noch eine Website kann es sich heutzutage erlauben, mit rein statischen Inhalten aufzuwarten. So ist auf vielen Websites die Interaktion mit den Besuchern 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.

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

Viel Spaß beim Lernen und viel Erfolg beim Bau Ihrer Webanwendungen!

Ihr Marco Emrich

3

Datenbanken in der Praxis

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 dieser Class setzen Sie das freie DBMS *MySQL* ein. *MySQL* ist das am weitesten verbreitete Open-Source DBMS²⁶. 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.

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, die kostenlose Distribution **MAMP** (für macOS und Windows) zu verwenden. Die Distribution enthält unter anderem folgende Pakete:

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



Abb. 8 MAMP Logo

Unter www.mamp.info²⁷ erhalten Sie *MAMP* und die dazugehörige Installationsanleitung.

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



Achtung

MAMP ist eine Distribution, die ausschließlich für die Entwicklung verwendet werden sollten. In Ihrer Basiskonfiguration erfüllt Sie nicht die Sicherheitsanforderungen, die notwendig sind, um einen Produktivbetrieb (insbesondere im Internet) zu gewährleisten.

Für den produktiven Betrieb von Webanwendungen sind meist Unix- oder Linux-basierte Server im Einsatz. Es empfiehlt sich, im Produktivbetrieb eine Linux-Installation 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.

26. Quelle: DB-Engines Ranking (<https://db-engines.com/de/ranking>) abgerufen am 24.02.2020

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

Übung 1: MySQL-Installation

- Installieren Sie nun MySQL für Ihr Betriebssystem.

3.2 Starten und Stoppen von MySQL

unter Windows und macOS

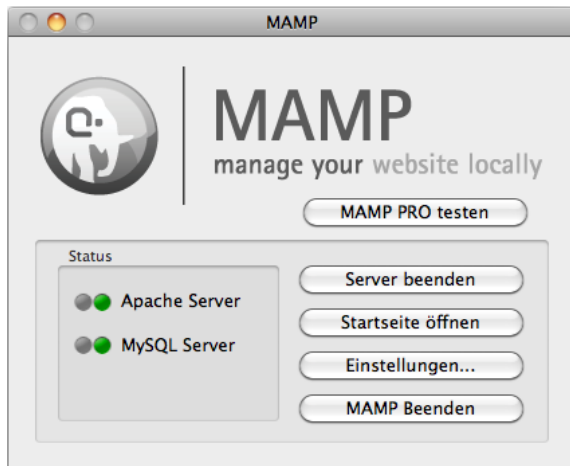


Abb. 9 Das MAMP Control-Panel

MAMP enthält 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.
- ▶ **macOS/MAMP:** Fügen Sie den Pfad²⁸ `/Applications/MAMP/Library/bin/mysql` zu den Systempfaden hinzu. Fall Sie auf Schwierigkeiten stoßen, hilft Ihnen vielleicht dieser [Blogbeitrag](#)²⁹ weiter.
- ▶ **Windows/MAMP:** 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:\MAMP\bin\mysql\bin`. Starten Sie anschließend Windows neu.

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`.

macOS: 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`.
3. Wenn Sie den folgenden Fehler erhalten, ist ein Passwort voreingestellt. `ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: NO)`
Verwenden Sie nachfolgenden Befehl und geben bei der Aufforderung das Passwort *root* ein.
`mysql -u root -p`

`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 Sie noch ein Passwort. Nach der Installation ist das Passwort aber standardmäßig deaktiviert. Teilweise wird das Passwort *root* als Standardpasswort gesetzt. Dies kommt aber auf die Version der Distribution an. Da Sie die Datenbank im Moment nur lokal zur Entwicklung einsetzen, stellt das kein Problem dar. Für den Produktivbetrieb später müssen Sie selbstverständlich ein sicheres Passwort festlegen.

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

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

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

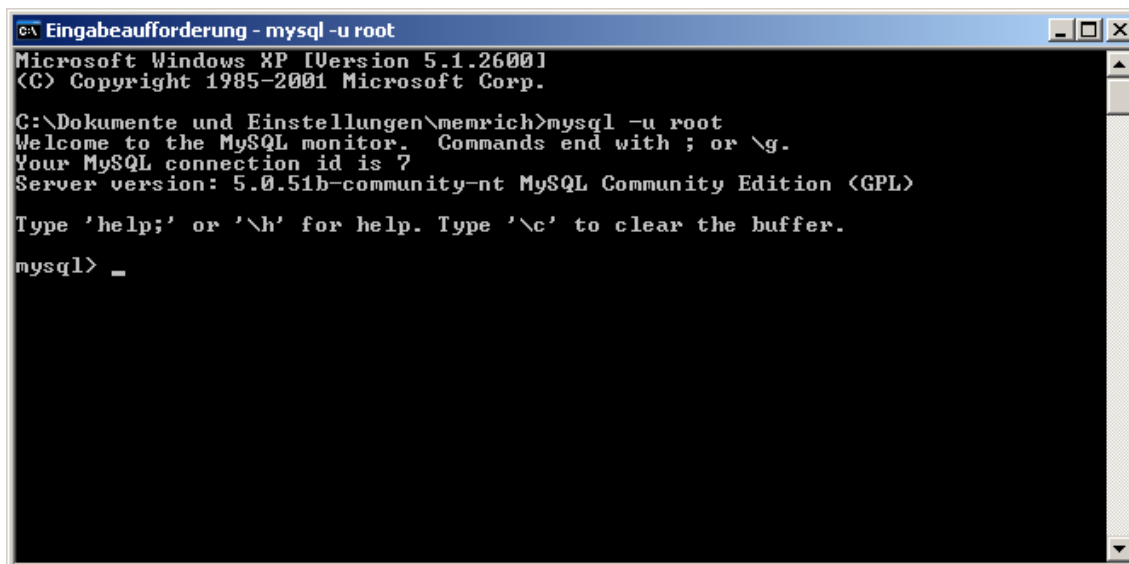


Abb. 10 MySQL-Kommandozeilen-Client

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. 11 Ausgabe von `show databases`

Falls Sie eine Datenbank mehr oder weniger als in Abb. 11 sehen, ist das nicht weiter schlimm. Die mitgelieferten Datenbanken können sich je nach MySQL-Version und Distribution 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 \(2007\)](#)³⁰. Wir verwenden hier die BNF-Version von MySQL.

30. <https://homepages.uni-regensburg.de/~brf09510/grammartypes.html>

Dadurch können Sie jederzeit die [offizielle MySQL-Dokumentation](#)³¹ verwenden. Sie 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. <https://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/MAMP:** Öffnen Sie die MySQL-Konfigurationsdatei in einem Texteditor. Sie finden diese Datei unter `|C:\MAMP\conf\mysql\my.ini`.

macOS MAMP: Legen Sie die Datei `my.cnf` unter `/Applications/MAMP/conf` neu an.

Linux: Öffnen Sie die MySQL-Konfigurationsdatei in einem Texteditor. Sie finden diese Datei unter `/etc/mysql/my.cnf` (Linux).

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 hilft 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

3.9 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.