

ralf ADAMS

Mit Beispielen in  
**MySQL/  
MariaDB,  
PostgreSQL  
und T-SQL**

# SQL

Der Grundkurs für  
Ausbildung und Praxis

3. Auflage



Mit Exkurs zu NoSQL

HANSER

Adams  
SQL



### **Bleiben Sie auf dem Laufenden!**

Unser **Computerbuch-Newsletter** informiert Sie monatlich über neue Bücher und Termine. Profitieren Sie auch von Gewinnspielen und exklusiven Leseproben. Gleich anmelden unter:

**[www.hanser-fachbuch.de/newsletter](http://www.hanser-fachbuch.de/newsletter)**





Ralf Adams

# SQL

## **Der Grundkurs für Ausbildung und Praxis**

Mit Beispielen in MySQL/MariaDB,  
PostgreSQL und T-SQL

3., aktualisierte Auflage

HANSER

Der Autor:

*Ralf Adams*, Bochum

Kontakt: [sq1buch@ralfadams.de](mailto:sq1buch@ralfadams.de)

Alle in diesem Buch enthaltenen Informationen, Verfahren und Darstellungen wurden nach bestem Wissen zusammengestellt und mit Sorgfalt getestet. Dennoch sind Fehler nicht ganz auszuschließen. Aus diesem Grund sind die im vorliegenden Buch enthaltenen Informationen mit keiner Verpflichtung oder Garantie irgendeiner Art verbunden. Autor und Verlag übernehmen infolgedessen keine juristische Verantwortung und werden keine daraus folgende oder sonstige Haftung übernehmen, die auf irgendeine Art aus der Benutzung dieser Informationen – oder Teilen davon – entsteht. Ebenso übernehmen Autor und Verlag keine Gewähr dafür, dass beschriebene Verfahren usw. frei von Schutzrechten Dritter sind. Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Buch berechtigt deshalb auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Bibliografische Information der Deutschen Nationalbibliothek:

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Dieses Werk ist urheberrechtlich geschützt.

Alle Rechte, auch die der Übersetzung, des Nachdruckes und der Vervielfältigung des Buches, oder Teilen daraus, vorbehalten. Kein Teil des Werkes darf ohne schriftliche Genehmigung des Verlages in irgendeiner Form (Fotokopie, Mikrofilm oder ein anderes Verfahren) – auch nicht für Zwecke der Unterrichtsgestaltung – reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

© 2020 Carl Hanser Verlag München, [www.hanser-fachbuch.de](http://www.hanser-fachbuch.de)

Lektorat: Sylvia Hasselbach

Copy editing: Sandra Gottmann, Wasserburg

Layout: der Autor mit LaTeX

Umschlagdesign: Marc Müller-Bremer, [www.rebranding.de](http://www.rebranding.de), München

Umschlagrealisation: Max Kostopoulos, unter Verwendung von Grafiken von

© [shutterstock.com](http://shutterstock.com)/Viktorus

Datenbelichtung, Druck und Bindung: Kösel, Krugzell

Ausstattung patentrechtlich geschützt. Kösel FD 351, Patent-Nr. 0748702

Printed in Germany

Print-ISBN: 978-3-446-46110-9

E-Book-ISBN: 978-3-446-46274-8

E-Pub-ISBN: 978-3-446-46324-0

*Dieses Buch möchte ich allen Lehrerinnen und Lehrern der ehemaligen Aufbau-realschule in Eslohe, Sauerland, widmen.*

*Besonders denke ich dabei an meinen Klassenlehrer und späterem Schulleiter, Herrn Schmidt. Durch Ihr stetes Bemühen um jeden Einzelnen sind Sie mir menschlich und heute als Lehrer ein Vorbild.*



# Inhalt

Vorwort zur 3. Auflage .....	XVII
<b>Teil I Was man so wissen sollte .....</b>	<b>1</b>
<b>1 Datenbanksystem .....</b>	<b>3</b>
1.1 Aufgaben und Komponenten .....	3
1.1.1 Datenbank .....	3
1.1.2 Datenbankmanagementsystem .....	5
1.2 Im Buch verwendete Server .....	7
1.2.1 MySQL und MariaDB .....	7
1.2.2 PostgreSQL .....	9
1.2.3 Microsoft SQL Server .....	10
<b>2 Einführung in relationale Datenbanken .....</b>	<b>11</b>
2.1 Was ist eine relationale Datenbank? .....	11
2.1.1 Abgrenzung zu anderen Datenbanken .....	11
2.1.2 Tabelle, Zeile und Spalte .....	13
2.1.3 Schlüssel, Primärschlüssel und Fremdschlüssel .....	16
2.2 Kardinalitäten und ER-Modell .....	22
2.2.1 Darstellung von Tabellen im ER-Modell .....	22
2.2.2 <i>1:1</i> -Verknüpfung .....	24
2.2.2.1 Wann liegt eine <i>1:1</i> -Verknüpfung vor? .....	24
2.2.2.2 Wie kann ich eine <i>1:1</i> -Verknüpfung darstellen? .....	25
2.2.2.3 Kann man die Kardinalität genauer beschreiben? .....	26
2.2.3 <i>1:n</i> -Verknüpfung .....	27
2.2.3.1 Wann liegt eine <i>1:n</i> -Verknüpfung vor? .....	27
2.2.3.2 Wie kann ich eine <i>1:n</i> -Verknüpfung darstellen? .....	28
2.2.3.3 Kann man die Kardinalität genauer beschreiben? .....	28

2.2.4	<i>n:m</i> -Verknüpfung .....	29
2.2.4.1	Wann liegt eine <i>n:m</i> -Verknüpfung vor? .....	29
2.2.4.2	Wie kann ich eine <i>n:m</i> -Verknüpfung darstellen? .....	30
2.2.4.3	Kann man die Kardinalität genauer beschreiben? .....	31
2.2.5	Aufgaben zum ER-Modell .....	31
2.3	Referenzielle Integrität .....	32
2.3.1	Verletzung der referenziellen Integrität durch Löschen .....	33
2.3.2	Verletzung der referenziellen Integrität durch Änderungen .....	34
2.4	Normalformen .....	34
2.4.1	Normalform 1 .....	35
2.4.2	Normalform 2 .....	37
2.4.3	Normalform 3 .....	38
2.4.4	Normalform Rest .....	39
<b>3</b>	<b>Unser Beispiel: Ein Online-Shop .....</b>	<b>41</b>
3.1	Kundenverwaltung .....	41
3.2	Artikelverwaltung .....	42
3.3	Bestellwesen .....	43
<b>Teil II</b>	<b>Datenbank aufbauen .....</b>	<b>45</b>
<b>4</b>	<b>Installation des Servers .....</b>	<b>47</b>
4.1	MySQL unter Windows 10 .....	47
4.2	MariaDB unter Windows 10 .....	53
4.3	Andere Installationen mit Docker .....	57
4.3.1	MySQL .....	58
4.3.2	MariaDB .....	60
4.3.3	PostgreSQL .....	61
4.3.4	Microsoft SQL Server .....	62
<b>5</b>	<b>Datenbank und Tabellen anlegen .....</b>	<b>63</b>
5.1	Die Programmiersprache SQL .....	63
5.2	Anlegen der Datenbank .....	64
5.2.1	Wie ruft man den MySQL Client auf? .....	65
5.2.2	Wie legt man eine Datenbank an? .....	66
5.2.3	Wie löscht man eine Datenbank? .....	68
5.2.4	Wie wird ein Zeichensatz zugewiesen? .....	68
5.2.5	Wie wird eine Sortierung zugewiesen? .....	70
5.3	Anlegen der Tabellen .....	72

5.3.1	Welche Datentypen gibt es?.....	73
5.3.2	Wie legt man eine Tabelle an? .....	74
5.3.3	Wann eine Aufzählung (ENUM) und wann eine neue Tabelle? .....	77
5.3.4	Wann ein DECIMAL, wann ein DOUBLE? .....	78
5.3.5	Wann verwendet man NOT NULL? .....	80
5.3.6	Wie legt man einen Fremdschlüssel fest? .....	82
5.3.7	Wie kann man Tabellen aus anderen herleiten? .....	89
5.3.8	Ich brauche mal eben kurz 'ne Tabelle! .....	90
<b>6</b>	<b>Indizes anlegen .....</b>	<b>93</b>
6.1	Index für Anfänger .....	93
6.1.1	Wann wird ein Index automatisch erstellt? .....	95
6.1.2	Wie kann man einen Index manuell erstellen?.....	97
6.2	Und jetzt etwas genauer .....	99
6.2.1	Wie kann ich die Schlüsseleigenschaft erzwingen?.....	99
6.2.2	Wie kann ich Dubletten verhindern? .....	100
6.2.3	Was bedeutet Indexselektivität? .....	102
6.2.4	Wie kann man einen Index löschen? .....	104
<b>7</b>	<b>Werte in Tabellen einfügen.....</b>	<b>105</b>
7.1	Daten importieren .....	105
7.1.1	Das CSV-Format .....	106
7.1.2	LOAD DATA INFILE .....	107
7.1.3	Was ist, wenn ich geänderte Werte importieren will? .....	111
7.2	Daten anlegen .....	112
7.2.1	Wie legt man mehrere Zeilen mit einem Befehl an? .....	113
7.2.2	Wie kann man eine einzelne Zeile anlegen? .....	114
7.2.3	Vorsicht Constraints!.....	115
7.2.4	Einfügen von binären Daten über einen C#-Client.....	116
7.2.5	Einfügen von binären Daten LOAD FILE .....	119
7.3	Daten kopieren.....	120
<b>Teil III</b>	<b>Datenbank ändern .....</b>	<b>123</b>
<b>8</b>	<b>Datenbank und Tabellen umbauen.....</b>	<b>125</b>
8.1	Eine Datenbank ändern.....	125
8.2	Ein Schema löschen .....	127
8.3	Eine Tabelle ändern .....	129
8.3.1	Wie kann ich den Namen der Tabelle ändern? .....	129

8.3.2	Wie kann ich eine Spalte hinzufügen? .....	131
8.3.3	Wie kann ich die Spezifikation einer Spalte ändern? .....	132
8.3.4	Zeichenbasierte Spalten in der Länge verändern .....	133
8.3.5	Zeichensatz verändern .....	134
8.3.6	Zeichenbasierte Spalten in numerische Spalten verändern .....	134
8.3.7	Numerische Spalten im Wertebereich verändern .....	135
8.3.8	Datum- oder Zeitspalten verändern .....	135
8.3.9	Wie kann ich aus einer Tabelle Spalten entfernen? .....	137
8.4	Eine Tabelle löschen .....	138
8.4.1	Einfach löschen .....	139
8.4.2	Was bedeuten die Optionen CASCADE und RESTRICT? .....	140
<b>9</b>	<b>Werte in Tabellen verändern .....</b>	<b>141</b>
9.1	WHERE-Klausel .....	141
9.1.1	Wie formuliert man eine einfache Bedingung? .....	142
9.1.2	Wird zwischen Groß- und Kleinschreibung unterschieden? .....	143
9.1.3	Wie formuliert man eine zusammengesetzte Bedingung? .....	145
9.2	Tabelleninhalte verändern .....	146
9.2.1	Szenario 1: Einfache Wertzuweisung .....	148
9.2.2	Szenario 2: Berechnete Werte .....	148
9.2.3	Szenario 3: Gebastelte Zeichenketten .....	149
9.2.4	Was bedeutet die Option LOW_PRIORITY? .....	150
9.2.5	Was bedeutet die Option IGNORE? .....	150
9.3	Tabelleninhalte löschen .....	150
9.3.1	Und was passiert bei Constraints? .....	151
9.3.2	Was passiert mit dem AUTO_INCREMENT? .....	152
9.3.3	Was bedeutet LOW_PRIORITY? .....	153
9.3.4	Was bedeutet QUICK? .....	153
9.3.5	Was bedeutet IGNORE? .....	153
9.3.6	Wie kann man eine Tabelle komplett leeren? .....	154
<b>Teil IV</b>	<b>Datenbank auswerten .....</b>	<b>155</b>
<b>10</b>	<b>Einfache Auswertungen .....</b>	<b>157</b>
10.1	Ausdrücke .....	158
10.1.1	Konstanten .....	158
10.1.2	Wie kann man Berechnungen vornehmen? .....	159
10.1.3	Wie ermittelt man Zufallszahlen? .....	160

10.1.4	Wie steckt man das Berechnungsergebnis in eine Variable? .....	161
10.2	Zeilen- und Spaltenwahl .....	162
10.3	Sortierung .....	163
10.3.1	Was muss bei der Sortierung von Texten beachtet werden? .....	165
10.3.2	Wird zwischen Groß- und Kleinschreibung unterschieden? .....	167
10.3.3	Wie werden Datums- und Uhrzeitwerte sortiert? .....	169
10.3.4	Wie kann man das Sortieren beschleunigen? .....	170
10.4	Mehrfachausgaben unterbinden .....	173
10.4.1	Fallstudie: Datenimport von Bankdaten .....	174
10.4.2	Was ist beim DISTINCT bzgl. der Performance zu beachten? .....	176
10.5	Ergebnismenge ausschneiden .....	177
10.5.1	Wie kann man sich die ersten $n$ Datensätze ausschneiden? .....	177
10.5.2	Wie kann man Teilmengen mittendrin ausschneiden? .....	178
10.6	Ergebnisse exportieren .....	179
10.6.1	Wie legt man eine Exportdatei auf dem Server an? .....	179
10.6.2	Wie legt man eine Exportdatei auf dem Client an? .....	180
10.6.3	Wie liest man mithilfe eines C#-Client binäre Daten aus? .....	181
<b>11</b>	<b>Tabellen verbinden .....</b>	<b>183</b>
11.1	Heiße Liebe: Primär-Fremdschlüsselpaare .....	184
11.2	INNER JOIN zwischen zwei Tabellen .....	187
11.2.1	Bauanleitung für einen INNER JOIN .....	188
11.2.2	Abkürzende Schreibweisen .....	192
11.2.3	Als Datenquelle für temporäre Tabellen .....	192
11.2.4	JOIN über Nichtschlüsselpalten .....	195
11.3	INNER JOIN über mehr als zwei Tabellen .....	197
11.4	Es muss nicht immer heiße Liebe sein: OUTER JOIN .....	200
11.5	Narzissmus pur: SELF JOIN .....	205
11.6	Eine Verknüpfung beschleunigen .....	208
<b>12</b>	<b>Differenzierte Auswertungen .....</b>	<b>211</b>
12.1	Statistisches mit Aggregatfunktionen .....	211
12.2	Tabelle in Gruppen zerlegen .....	214
12.3	Gruppenergebnisse filtern .....	218
12.4	Noch Fragen? .....	220
12.4.1	Kann ich nach Ausdrücken gruppieren? .....	220
12.4.2	Kann ich nach mehr als einer Spalte gruppieren? .....	220
12.4.3	Wie kann ich GROUP BY beschleunigen? .....	221
12.4.4	Parallele Bearbeitung – unterschiedliche Ergebnisse? .....	223
12.5	Aufgaben .....	223

<b>13</b>	<b>Auswertungen mit Unterabfragen</b>	<b>225</b>
13.1	Das Problem und die Lösung	225
13.2	Nicht korrelierende Unterabfrage	228
13.2.1	Skalarunterabfrage	228
13.2.1.1	Beispiel 1: Banken mit höchster BLZ	228
13.2.1.2	Beispiel 2: Überdurchschnittlich teure Artikel	229
13.2.1.3	Beispiel 3: Überdurchschnittlich wertvolle Bestellungen	230
13.2.2	Listenunterabfrage	232
13.2.2.1	Beispiel 1: IN()	232
13.2.2.2	Beispiel 2: ALL()	233
13.2.2.3	Beispiel 3: ALL()	234
13.2.2.4	Beispiel 4: ANY()	237
13.2.3	Unterschied zwischen IN(), ALL() und ANY()	238
13.2.4	Unterschied zwischen NOT IN() und <> ALL()	239
13.2.5	Tabellenunterabfrage	239
13.3	Korrelierende Unterabfrage	240
13.3.1	Beispiel 1: Rechnungen mit vielen Positionen	240
13.3.2	Beispiel 2: EXISTS	241
13.4	Fallstudie Datenimport	242
13.5	Wie ticken Unterabfragen intern?	245
13.6	Aufgaben	249
<b>14</b>	<b>Mengenoperationen</b>	<b>251</b>
14.1	Die Vereinigung mit UNION	251
14.2	Die Schnittmenge	254
14.2.1	Mit INTERSECT	254
14.2.2	Mit Unterabfragen	255
14.3	Die Differenzmenge	256
14.3.1	Mit EXCEPT	256
14.3.2	Mit Unterabfragen	257
14.4	UNION, INTERSECT und EXCEPT ... versteh' ich nicht!	258
<b>15</b>	<b>Bedingungslogik</b>	<b>261</b>
15.1	Warum ein CASE?	261
15.2	Einfacher CASE	263
15.3	SEARCHED CASE	265
15.4	Fallbeispiele	267
15.4.1	Lagerbestand überprüfen	267
15.4.2	Kundengruppen ermitteln	268
15.4.3	Aktive Lieferanten ermitteln	271
15.4.4	Aufgaben	272

<b>16</b>	<b>Ansichtssache</b> .....	<b>273</b>
16.1	Was ist eine Ansicht? .....	273
16.1.1	Wie wird eine Ansicht angelegt? .....	274
16.1.2	Wie wird eine Ansicht verarbeitet? .....	276
16.1.3	Wie wird eine Ansicht gelöscht? .....	279
16.1.4	Wie wird eine Ansicht geändert? .....	282
16.2	Anwendungsgebiet: Vereinfachung .....	282
16.3	Anwendungsgebiet: Datenschutz .....	285
16.4	Grenzen einer Ansicht .....	285
<b>17</b>	<b>Exkurs NoSQL</b> .....	<b>289</b>
17.1	Vorbereitung der MySQL-Shell .....	290
17.2	Datenmodellierung des Warenkorbs .....	291
17.2.1	JavaScript Object Notation (JSON) .....	291
17.2.2	Struktur unseres JSON-Dokuments .....	293
17.3	NoSQL: MySQL mit JavaScript-Client .....	294
17.3.1	Anlegen eines Warenkorbs .....	295
17.3.2	Inhalte des Warenkorbs anlegen .....	296
17.3.3	Inhalte des Warenkorbs auswerten .....	299
17.3.4	Inhalte des Warenkorbs verändern .....	302
17.4	NoSQL: klassisches SQL mit JSON-Funktionen .....	304
17.4.1	Anlegen eines Warenkorbs .....	304
17.4.2	Inhalte des Warenkorbs anlegen .....	305
17.4.3	Inhalte des Warenkorbs auswerten .....	307
17.4.4	Inhalte des Warenkorbs verändern .....	308
17.4.5	Inhalte des Warenkorbs löschen .....	311
<b>Teil V</b>	<b>Anweisungen kapseln</b> .....	<b>313</b>
<b>18</b>	<b>Locking</b> .....	<b>315</b>
<b>19</b>	<b>Transaktion</b> .....	<b>319</b>
19.1	Das Problem .....	319
19.2	Was ist eine Transaktion? .....	321
19.3	Isolationsebenen .....	324
19.3.1	READ UNCOMMITTED .....	324
19.3.2	READ COMMITTED .....	326
19.3.3	REPEATABLE READ .....	327
19.3.4	SERIALIZABLE .....	328
19.4	Fallbeispiel in C# .....	329
19.5	Deadlock .....	331

<b>20</b>	<b>STORED PROCEDURE</b>	<b>333</b>
20.1	Einstieg und Variablen	334
20.2	Verzweigung	339
20.2.1	Einfache Verzweigung mit IF	339
20.2.2	Mehrfache Verzweigung mit CASE	342
20.3	Schleifen	345
20.3.1	LOOP-Schleife	346
20.3.2	WHILE-Schleife	348
20.3.3	REPEAT-Schleife	351
20.4	Transaktion innerhalb einer Prozedur	352
20.5	CURSOR	353
20.6	Aufgaben	359
<b>21</b>	<b>Funktion</b>	<b>361</b>
<b>22</b>	<b>TRIGGER</b>	<b>363</b>
22.1	Was ist das?	363
22.2	Ein Beispiel für einen INSERT-Trigger	365
22.3	Ein Beispiel für einen UPDATE-Trigger	366
22.4	Ein Beispiel für einen DELETE-Trigger	368
<b>23</b>	<b>EVENT</b>	<b>371</b>
23.1	Wie legt man ein Ereignis an?	371
23.2	Wie wird man ein Ereignis wieder los?	374
<b>Teil VI</b>	<b>Anhänge</b>	<b>375</b>
<b>24</b>	<b>Datenbank administrieren</b>	<b>377</b>
24.1	Backup und Restore	377
24.1.1	Backup mit mysqldump	377
24.1.2	Restore mit mysqldump	379
24.2	Benutzerrechte	379
24.2.1	Benutzerrechte und Privilegien	379
24.2.2	Benutzer anlegen/Recht zuweisen	382
24.2.2.1	CREATE USER	382
24.2.2.2	GRANT	383
24.2.2.3	REVOKE	385
24.3	MySQL und MariaDB Engines	386

<b>25</b>	<b>Rund um den MySQL Client</b> .....	<b>389</b>
25.1	Aufruf(parameter) .....	389
25.2	Befehle .....	392
<b>26</b>	<b>SQL-Referenz</b> .....	<b>397</b>
26.1	Datentypen .....	397
26.1.1	Numerische Datentypen .....	397
26.1.1.1	Ganze Zahlen .....	397
26.1.1.2	Gebrochene Zahlen .....	398
26.1.2	Zeichen-Datentypen .....	399
26.1.3	Datums- und Zeit-Datentypen .....	400
26.1.4	Binäre Datentypen .....	403
26.1.5	JSON .....	404
26.1.6	Räumliche Datentypen .....	404
26.1.7	Standardwerte .....	405
26.1.8	Zusätze für Datentypen .....	406
26.2	Operatoren und Funktionen .....	408
26.2.1	Mathematische Operatoren .....	408
26.2.2	Mathematische Funktionen .....	408
26.2.3	Aggregatfunktionen .....	411
26.3	Bedingungen .....	414
26.3.1	Vergleichsoperatoren .....	414
26.3.2	Logikoperatoren .....	416
26.3.2.1	NOT, Negation, $\neg$ .....	416
26.3.2.2	AND, Konjunktion, $\wedge$ .....	417
26.3.2.3	OR, Disjunktion, $\vee$ .....	418
26.3.2.4	XOR, Antivalenz, $\otimes$ .....	418
26.4	Befehle .....	419
26.4.1	Data Definition Language .....	419
26.4.2	Data Manipulation Language .....	431
26.4.3	Benutzerverwaltung .....	435
<b>27</b>	<b>Ausgewählte Quelltexte</b> .....	<b>439</b>
27.1	DOUBLE versus DECIMAL .....	439
27.2	Rundungsfehler .....	443
27.3	NULL versus NOT NULL .....	444
27.4	Suchen mit und ohne Index .....	446
27.5	Messen der Performance der Einfügeoperation .....	449
27.6	Messen der Indexselektivität .....	452
27.7	Sortieren ohne und mit Index .....	454

<b>28 Rund ums Zeichen</b> .....	<b>457</b>
28.1 Für Deutsch relevante Zeichensätze .....	457
28.2 Für Deutsch relevante Sortierungen .....	458
<b>29 Quelltexte</b> .....	<b>461</b>
29.1 MySQL/MariaDB .....	461
29.1.1 Quelltexte zu Teil II .....	461
29.1.2 Quelltexte zu Teil III .....	473
29.1.3 Quelltexte zu Teil IV .....	477
29.1.4 Quelltexte zu Teil V .....	521
29.2 PostgreSQL .....	536
29.2.1 Quelltexte zu Teil II .....	536
29.2.2 Quelltexte zu Teil III .....	545
29.2.3 Quelltexte zu Teil IV .....	549
29.2.4 Quelltexte zu Teil V .....	580
29.3 Microsoft SQL Server .....	584
29.3.1 Quelltexte zu Teil II .....	584
29.3.2 Quelltexte zu Teil III .....	595
29.3.3 Quelltexte zu Teil IV .....	600
<b>Literatur</b> .....	<b>635</b>
<b>Stichwortverzeichnis</b> .....	<b>639</b>

# Vorwort zur 3. Auflage

Und noch'n SQL-Buch. Es gibt so viele SQL-Bücher, dass man berechtigt die Frage stellen kann, warum man noch eines braucht. Ich kann die Frage nur indirekt beantworten. Als Lehrer für Anwendungsentwicklung an einem Berufskolleg habe ich über Jahre erlebt, dass die Auszubildenden sich sehr mit den üblichen Büchern abmühen.

Die fachliche Qualität dieser Bücher ist unbestritten. Aber die Sprache ist meist von *IT-Profi* zu *IT-Profi*, und genau damit sind Auszubildende und Berufsanfänger oft überfordert – zumindest wird der Einstieg erschwert.

Ich habe daher begonnen, leicht verständliche Skripte zu schreiben, aus denen sich dieses Buch speist. Dabei werden Befehle didaktisch reduziert und Beispiele möglichst lebensnah ausgesucht. Fachbegriffe werden nur verwendet, wenn sie IT-sprachlicher Umgang sind; akademische Begriffe werden vermieden, wobei ich ihre Berechtigung nicht in Abrede stellen möchte.

Primärziel ist ein möglichst umfangreicher Ersteinstieg (Grundkurs), der dann durch berufliche Praxis ausgebaut werden kann. Trotzdem vertiefe ich an vielen Stellen im Buch den Einblick in SQL oder den MySQL Server (Vertiefendes) – zum einen, um zu zeigen, dass ich auch ein bisschen was drauf habe, zum anderen, um Neugierde und Jagdtrieb beim Leser<sup>1</sup> zu wecken.

Ein weiterer Grund für dieses Buch ist, dass es mir großen Spaß gemacht hat, es zu schreiben. Ich hoffe, dass es Ihnen genau soviel Spaß macht, es zu lesen und damit zu arbeiten. Falls Sie mich fachlich korrigieren oder ergänzen möchten, senden Sie mir doch bitte eine E-Mail an [sqlbuch@ralfadams.de](mailto:sqlbuch@ralfadams.de).

Der Titel des Buches ist SQL und nicht MySQL. Ich habe deshalb an vielen Stellen den Unterschied zwischen SQL-Standard und seinen Dialekten aufgezeigt. Trotzdem wird es schwer sein, die Beispiele *einfach so* auf andere DBMS zu übertragen. Auf jeden Fall werden Sie ein Verständnis für den allgemeinen Aufbau und die Funktionsweise der Befehle erwerben, sodass Sie leicht die verschiedenen SQL-Dialekte adaptieren können.

- Bitte beachten Sie, dass die Pfadangaben in den Skripten mit LOAD DATA INFILE angepasst werden müssen, je nachdem, wo Sie die Daten entpacken.

<sup>1</sup> Der besseren Lesbarkeit wegen verzichte ich auf weiblich/männlich-Konstruktionen. Bitte verstehen Sie dies nicht als stillschweigende Hinnahme des geringen Frauenanteils in den IT-Berufen.

- Ich habe angefangen, für die Aufgaben Musterlösungen bei YouTube (<http://www.youtube.com/channel/UCu4ZybNXw1y4Rs4Mgx-4HKw>) einzustellen. In diesen Videos kann ich einfach besser erklären, worauf es bei den Lösungen ankommt.
- Beim Test der Skripte unter MySQL 5.6.19 ist ein Fehler des Servers aufgetreten (siehe [Ada14]).
- Die im Internet unter <http://downloads.hanser.de> verfügbaren Skripte, Beispiele und Musterlösungen sind auf folgenden Servern getestet worden: *MySQL Community Server 8.0.17*, *MariaDB 10.4.6*, *PostgreSQL 11.4* und *MS SQL Server 14.0.3038.14*. Alle Server liefen in einer Dockerinstanz unter Ubuntu 18.10.
- Zwar haben sich MariaDB und MySQL auseinander entwickelt, aber bei den hier vorgestellten Befehlen konnte ich keine Inkompatibilitäten feststellen.
- Für alle Quelltexte, die bis einschließlich Kapitel 16 vorgestellt werden, gibt es Varianten in MySQL/MariaDB, PostgreSQL und T-SQL. Nur bei ganz wenigen Ausnahmen, die durch die jeweiligen Dialekte oder Eigenheiten begründet sind, musste ich auf eine Transkription verzichten.
- Neu beschrieben werden *Generierte Spalten*, die Option `WITH ROLLUP` bei Gruppierungen und *Common Table Expression* (einfach und rekursiv).
- Auch neu ist ein Kapitel über NoSQL, welches das Spektrum abrunden soll. Ursprünglich hatte ich vor, die NoSQL-Inhalte über das Buch auf die passenden Stellen zu verteilen. So sollte alles um Anlegen, Ändern und Löschen und alles über die Auswertungen in den entsprechenden Abschnitten behandelt werden. Das hat sich als unpraktisch erwiesen und deshalb habe ich alles zum Thema NoSQL in einem Kapitel zusammengefasst.

## Danksagung

Als Erstes möchte ich mich bei Frau Sylvia Hasselbach vom Hanser Verlag dafür bedanken, dass sie diese Neuauflage – wie schon die Voraufgabe – angestoßen und vorangetrieben hat. Frau Rothe und Frau Gottmann haben sprachliche Ausrutscher und flapsige Formulierungen glatt gebügelt. Das Layout wurde von Frau Irene Weilhart betreut.

Ich möchte meinen Kollegen Dr. Andreas Alef und Marco Bakera, mit denen ich das Vergnügen habe, an der Technischen Beruflichen Schule 1 in Bochum (<http://www.tbs1.de>) zu unterrichten, für ihre kritischen und aufmunternden Kommentare danken.

Besonders will ich meine Schülerinnen und Schüler erwähnen. Die hier vorgestellten Beispiele und Konzepte sind in großen Teilen durch ihre schonungslose Kritik an bestehenden Lehrmaterialien entstanden. Das penetrante *Kapier ich nicht!* hat mich immer weiter angespornt, es noch verständlicher zu versuchen. Falls dieses Buch SQL gut vermittelt, ist das auch deren Verdienst.

Dass nun die 3. Auflage dieses Buchs erscheinen kann, ist aber in erster Linie Ihnen, liebe Leserinnen und Leser, zu verdanken; dafür ein herzliches *Dankeschön!*

Ralf Adams, Oktober 2019

# TEIL I

Was man so wissen sollte



# 1

# Datenbanksystem

## ■ 1.1 Aufgaben und Komponenten



Es werden die wichtigsten Aufgaben und Komponenten eines Datenbanksystems vorgestellt. Die Begriffe werden lediglich eingeführt, weil sich ein detailliertes Verständnis erst in den nachfolgenden Kapiteln entwickeln kann.

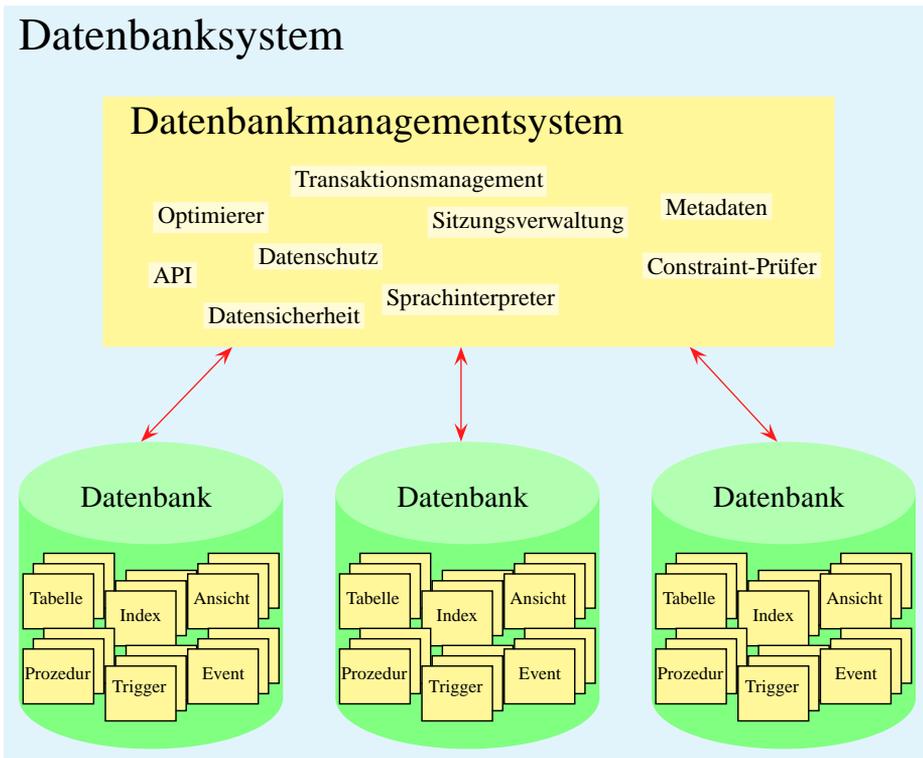
- Grundkurs
  - Datenbanksystem
  - Datenbank
  - Datenbankmanagementsystem

Ein Datenbanksystem besteht aus einem Datenbankmanagementsystem (DBMS) und den Datenbanken (DB). Beide Komponenten sind in der Praxis eng miteinander verzahnt, sollten aber gedanklich unterschieden werden.

In [Bild 1.1 auf der nächsten Seite](#) ist der Aufbau eines Datenbanksystems schematisch dargestellt. Die Datenbanken enthalten die eigentlichen Daten und unmittelbar damit verknüpfte Datenobjekte wie z.B. eine Ansicht (siehe [Kapitel 16 auf Seite 273](#)). Über eine Kommunikationsschnittstelle werden diese Datenobjekte vom DBMS verwaltet. Das DBMS selbst besteht wiederum aus vielen kleinen Komponenten, die jeweils auf eine Aufgabe spezialisiert sind.

### 1.1.1 Datenbank

Die Aufgabe der Datenbank ist die logische und physische Verwaltung der Daten und damit eng verbundener Datenobjekte. Alle diese Datenobjekte können vom Programmierer angelegt, geändert und gelöscht werden. Die Änderungen beziehen sich sowohl auf die Struktur als auch auf den Inhalt. So können einer Tabelle neue Spalten (z.B. zweiter Vorname bei einer Adresse) als auch neue Zeilen (z.B. eine neue Adresse) hinzugefügt werden.



**Bild 1.1** Aufbau eines Datenbanksystems

Üblicherweise werden in einer Datenbank folgende Datenobjekte vorkommen<sup>1</sup>:

- **Tabellen:** Bei einer relationalen Datenbank werden die Daten in Tabellen organisiert (Kundentabelle, Artikeltable, Filmtabelle usw.). Deshalb sind die Tabellen das Herzstück einer Datenbank. Alle anderen Datenbankobjekte sind aus diesen Tabellen abgeleitet oder verwenden diese.
- **Temporäre Tabellen:** Sie werden explizit vom Programmierer oder implizit vom Optimierer angelegt, um Zwischenergebnisse wiederverwendbar zu machen. In der Regel werden diese automatisch nach Beendigung einer Sitzung gelöscht.
- **Indizes:** Diese erlauben eine erhebliche Beschleunigung bestimmter Auswertungen. Die Daten aus den Tabellen werden dabei in frei wählbaren, aber festgelegten Reihenfolgen sortiert.
- **Ansichten:** Auf Vorrat gebastelte Auswertungen, die wie Tabellen verwendet werden können, ohne dabei einen eigenen Datenbestand aufzubauen.
- **Prozeduren:** Kleine, selbst geschriebene SQL-Programme, die auf dem Server ausgeführt werden.

<sup>1</sup> Die Liste ist nicht vollständig. Ich werde mich aber auf diese beschränken.

- *Trigger*: Auch kleine, selbst geschriebene Programme, die aber automatisch aufgerufen werden, wenn Daten in den entsprechenden Tabellen verändert werden.
- *Ereignisse*: Schon wieder kleine, selbst geschriebene Programme, die zeitgesteuert aufgerufen werden.



#### Definition 1: Datenbank (eng)

Unter einer *Datenbank* im engeren Sinn versteht man eine Informationssammlung, die so strukturiert ist, dass sie zweckgebunden und effizient IT-gestützt verwaltet und ausgewertet werden kann.

Entscheidend ist, dass die Daten *strukturiert* sind! Jede Informationssammlung, die eine gewisse Struktur hat, kann letztlich von einem Computerprogramm verwaltet und ausgewertet werden. Fehlt die Struktur, sieht das Ganze schon anders aus.



**Aufgabe 1.1:** Überlegen Sie sich mindestens zwei Beispiele für strukturierte und unstrukturierte Datensammlungen.

In MySQL und MariaDB werden die Datenbanken durch die Storage Engines (z.B. InnoDB bzw. XtraDB) realisiert. Diese legen auf den Festspeicherplatten die Dateien an, in denen die Daten abgespeichert werden. Auch die Zugriffskontrolle erfolgt durch die Storage Engines.

## 1.1.2 Datenbankmanagementsystem

Oben habe ich erwähnt, dass ein Datenbanksystem aus der Datenbank und einem Datenbankmanagementsystem besteht.



#### Definition 2: Datenbankmanagementsystem (DBMS)

Eine Toolsammlung zur Verwaltung, Bearbeitung und Auswertung einer Datenbank nennt man *Datenbankmanagementsystem* (DBMS).

Dies sind die Aufgaben eines DBMS<sup>2</sup>:

- *Sprachinterpreter*: Herzstück des DBMS ist der Interpreter<sup>3</sup>. Dieser übersetzt die Befehle in einen ausführbaren Code. Die Sprache, die wir hier im Buch verwenden werden, ist SQL. Es gibt und gab aber auch andere Datenbankabfragesprachen wie zum Beispiel dBase, VB für MS-Access, OO-SQL, Sequel usw.
- *Optimierer*: Die Ausführung eines SQL-Befehls kann oft auf verschiedene Art und Weise passieren. Der Optimierer versucht, anhand von Schätzungen und Algorithmen einen Plan für die Ausführung anzulegen, der möglichst schnell abgearbeitet werden kann.

<sup>2</sup> Je nach Hersteller oder Lesart finden Sie andere Aufgabensammlungen, aber mit dieser kommen wir schon sehr weit.

<sup>3</sup> Es ist müßig, darüber zu streiten, ob es sich um einen Interpreter oder einen Compiler oder einen Jitter handelt. Warum? Weil es keinen interessiert ;-)

- *Sitzungsverwaltung*: Wann immer ein Befehl an den Server gesendet werden soll, muss man sich in einer Sitzung (engl. *session*) befinden. Dazu muss zuerst eine Sitzung geöffnet werden. Jetzt können beliebig viele SQL-Befehle gesendet und Daten empfangen werden. Zum Schluss wird die Sitzung serverseitig – z.B. durch einen Timeout – oder clientseitig beendet.
- *Randbedingungsprüfer*: Für Tabellen können Randbedingungen (engl. *constraints*) formuliert werden, die immer gelten müssen. Würde die Ausführung eines Befehls dazu führen, dass diese Randbedingungen nicht erfüllt sind, wird die Ausführung des Befehls in der Regel verweigert.
- *Datenschutz*: Durch die Vergabe von Zugriffsrechten kann das Recht auf lesende und schreibende Zugriffe so wie auf das Ausführen von Prozeduren passgenau zugeschnitten werden.
- *Datensicherheit*: Der Verlust von Daten ist der GAU<sup>4</sup> schlechthin. Das DBMS muss sicherstellen, dass nicht durch Serverabsturz oder Ähnliches Daten verloren gehen.
- *Transaktionsmanagement*: Transaktionen ermöglichen parallelen Zugriff und eine Art *undo* im Fehlerfall. Das zu gewährleisten, erfordert eine Menge Mühe. Die Qualität des Transaktionsmanagements ist oft ein entscheidendes Merkmal eines DMBS.
- *API*<sup>5</sup>: Die Daten werden in der Regel durch eine oder mehrere Anwendungen (Clients) bearbeitet. Damit die Anwendung auf das DBMS zugreifen kann, braucht es eine Schnittstelle, über die es zu den Daten gelangt. Der MySQL oder MariaDB Server bietet beispielsweise APIs für C, C++, C#/ .NET, PHP, Perl, Python und Tcl. Auch stehen APIs für JDBS<sup>6</sup> und ODBC<sup>7</sup> zur Verfügung. Man nennt sie *Konnektoren*.
- *Metadaten*: Verwaltungsinformationen, Statistiken etc., eben der ganze Rest.

Der Begriff *Datenbankmanagementsystem* wird oft anstelle von *Datenbanksystem* verwendet. Gerade die schematischen Darstellungen in den Dokumentationen der Hersteller unterscheiden nicht zwischen diesen beiden Begriffen.

Sind die Datenbanken eines Datenbankmanagementsystems in Form von Tabellen organisiert, so handelt es sich um ein *relationales Datenbankmanagementsystem (RDBMS)*.

Und noch der Vollständigkeit halber:



### Definition 3: Datenbanksystem

Ein System, welches die Datenbanken und das dazugehörige Datenbankmanagementsystem als Komplettpaket anbietet, nennt man *Datenbanksystem*.

<sup>4</sup> Abkürzung für: Größter anzunehmender Unfall

<sup>5</sup> Abkürzung für: Application Programming Interface; engl. für Programmierschnittstelle

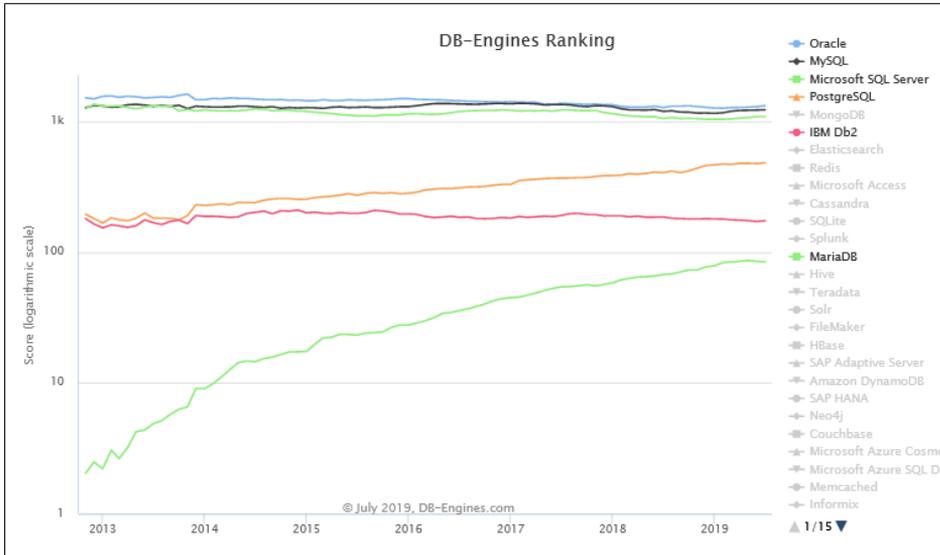
<sup>6</sup> Abkürzung für: Java Database Connectivity: Programmierschnittstelle für JAVA

<sup>7</sup> Abkürzung für: Open Database Connectivity: Eine offene standardisierte Schnittstelle. Sie wird von fast allen Datenbanksystemherstellern angeboten. Wer ODBC-Programme schreibt, kann leicht zwischen verschiedenen Datenbanksystemherstellern wechseln.

## 1.2 Im Buch verwendete Server



Kurzporträts der verwendeten SQL Server: Hersteller und Geschichte



**Bild 1.2** Ranking einiger RDBMS, Quelle [DE19]

In der ersten Auflage dieses Buchs habe ich fast ausschließlich MySQL/MariaDB als Plattform genutzt. Um SQL breiter vorstellen zu können, wurden in der zweiten Auflage die meisten Beispiele auch in PostgreSQL angeboten. Konsequenterweise wird in der dritten Auflage ein weiteres System bedient: der MS SQL Server. Wie Sie Bild 1.2 entnehmen können, decke ich damit eine Vielzahl von Installationen und SQL-Dialekten ab.

### 1.2.1 MySQL und MariaDB

Die schwedische Firma MySQL AB hat MySQL von 1994 bis 2008 entwickelt. Die ursprüngliche Intention war eine verbesserte und beschleunigte Verarbeitung eines selbst entwickelten Tabellensystems mit dem Namen ISAM. Dazu wurde mSQL<sup>8</sup> genutzt. Von 2008 bis 2010 wurde MySQL AB von Sun Microsystems gepflegt, und seit Januar 2010 wird MySQL unter dem Schirm von Oracle weiterentwickelt.

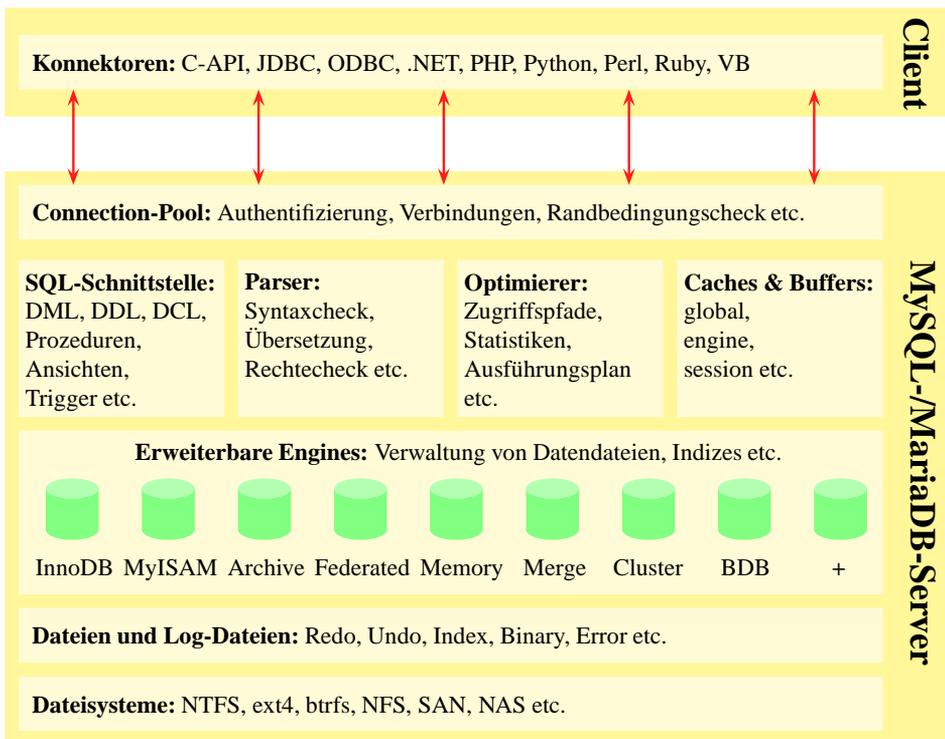
Der Name *MySQL* kommt nicht vom englischen *my* (mein). Einer der Firmengründer, Michael Widenius, hat sympathischerweise den Vornamen *My* seiner Tochter verwendet. Der Name des Delphins im Logo ist Sakila. Er wurde in einem Wettbewerb ermittelt, den der

<sup>8</sup> Kein Tippfehler! Siehe [Ltd10]

Open Source-Entwickler Ambrose Twebaze aus Uganda gewann. Sakila ist ein Mädchenname in der Sprache *siSwati* und auch der Name einer Stadt in Tansania.

Nach der Übernahme von MySQL durch Oracle haben die Spannungen zwischen den Entwicklern und Oracle ständig zugenommen, sodass der *Erfinder* von MySQL – Michael Widenius – sich mit der neu erstellen Engine Aria von MySQL abgespalten und 2009 das Projekt MariaDB ins Leben gerufen hat. Wie MySQL ist auch dieses Projekt nach einer Tochter von Widenius benannt. Wegen seiner offeneren Lizenzpolitik und der schnelleren Umsetzung von Neuerungen und Fehlerkorrekturen hat MariaDB an vielen Stellen – aber nicht, wie oft behauptet, an den meisten – MySQL abgelöst (siehe [Bild 1.2 auf der vorherigen Seite](#)).

MySQL und MariaDB sind Client-Server-Datenbanksysteme. Ein Server stellt alleine oder im Verbund mit anderen Servern den Anwendungen (Clients) die Datenbankdienste zur Verfügung. Der MySQL-/MariaDB-Server besteht – wie jedes DBMS – aus vielen Komponenten, die hier kurz angerissen werden.



**Bild 1.3** Komponenten von MySQL und MariaDB

- *Konnektoren* (Verbinder): Der Client baut über die Konnektoren eine Sitzung zum Server auf. Dies erfolgt über das TCP/IP-Protokoll und in der Regel über den Port 3306.
- *Connection-Pool*: Hier werden die Verbindungen zu den Clients verwaltet. Beim Verbindungsaufbau wird anhand des Benutzernamens und des Passworts die Verbindungsanfrage authentifiziert. Ist die Anzahl der maximal verfügbaren Verbindungen (Connections Limits) nicht überschritten, wird ein Verbindungsthread eingerichtet. Ebenso wer-

den die anderen Grenzwerte für die Verbindung überwacht: Datenübertragungsvolumen, Timeout etc.

- *SQL-Schnittstelle*: Hier werden die SQL-Befehle entgegengenommen. Sie werden dann zum Parser weitergereicht.
- *Parser*: Der Parser überprüft die Syntax eines Befehls und ob man die Ausführungsrechte für diesen Befehl hat.
- *Optimizer* (Optimierer): Anhand von Schätzungen, Statistiken und Algorithmen wird für nicht triviale Befehle ein Ausführungsplan erstellt. Dieser Ausführungsplan berücksichtigt ggf. im Cache vorhandene Ergebnisse.
- *Caches und Buffers* (Zwischenspeicher): Daten und Ergebnisse können in Zwischenspeichern aufgehoben werden. Diese sind nur in der Sitzung verfügbar, die diese erstellt hat (lokal) oder in allen Sitzungen (global).
- *Storage Engines*: Die Motoren des Servers. Hier werden die Daten tatsächlich verarbeitet. Jede Engine ist dabei für bestimmte Aufgabenstellungen besonders gut geeignet. Der große Vorteil von MySQL und MariaDB ist, dass jeder mit einem besonderen Bedarf eine Engine bauen kann. Er muss *nur* die Schnittstellen beachten (siehe [MyS19b]) und kann dann seine Speziallösungen anbieten.
- *File System* (Dateisystem): Je nach Betriebssystem werden hier Daten in unterschiedlichen Dateisystemen (NTFS, BTRFS, EXT4 etc.) abgelegt. Bis auf die Frage, ob das Dateisystem zwischen Groß- und Kleinschreibung unterscheidet, spielt dieses für die SQL-Programmierung keine Rolle.
- *Management Services & Utilities*: Parallel dazu gibt es die vielen kleinen Helferlein, ohne die nichts geht: für das Sichern und Wiederherstellen von Daten, Datenreplikation, Administration und Konfiguration, Datenmigration und Metadaten.

In den nachfolgenden Kapiteln werden wir gemeinsam eine Datenbank planen, installieren, einrichten, verändern und verwenden. Dabei werden die vielen Fragen beantwortet, die Sie nach dieser kurzen Einführung mit Sicherheit haben werden, also nur Geduld ...

## 1.2.2 PostgreSQL

Geboren wurde PostgreSQL 1986 als Universitätsprojekt an der Universität von Kalifornien, Berkeley. Professor Michael Stonebraker<sup>9</sup> begann das Projekt als Nachfolgeprojekt der Ingres-Datenbank, welche ebenfalls noch heute verwendet wird. Daher leitet sich auch der Name her: *post ingres*. In den nächsten acht Jahren wurde Postgres von Prof. Stonebraker und seinen Studenten immer weiter entwickelt.

Bis 1995 verstand Postgres allerdings kein SQL, sondern nur eine eigene Sprache namens POSTQUEL. Die beiden Doktoranden Andrew Yu und Jolly Chen haben Postgres um SQL erweitert und als Postgres95 veröffentlicht.

1996 wurde Postgres95 als Open-Source-Projekt der Netzgemeinde zur Verfügung gestellt und wird seitdem sehr stark von Unterstützern weiterentwickelt und gefördert. Ebenso

---

<sup>9</sup> Prof. Stonebraker ging später mit einem Fork von Postgres – Illustra – zu Informix, welche diesen in den Universal Server integrierte. Auch diese Datenbank besteht heute noch.

wurde das Erscheinungsjahr aus dem Produktnamen entfernt und die Datenbank heißt seitdem PostgreSQL.

Weitere Informationen über die Geschichte von PostgreSQL und die aktuellen Features des Servers können Sie auf der Homepage des Projekts (<https://www.postgresql.org>) nachlesen.

Von <http://www.postgresql.org/download/> können Sie die aktuelle PostgreSQL-Version für Linux-Distributionen und Windows herunterladen. Eine sehr ausführliche Online-Dokumentation steht unter <http://www.postgresql.org/docs/> zur Verfügung. Dort wird auch auf eine Reihe von weiteren Dokumentationen wie beispielsweise zum Thema *Sicherheit* verwiesen.

### 1.2.3 Microsoft SQL Server

Anders als bei den anderen Datenbankservern kann ich hier keine Geschichten über Töchter oder Universitätslaufbahnen erzählen. Die Entwicklung des MS SQL Servers ist da schon etwas nüchterner ([Gar16]).

In Kooperation mit Sybase brachte Microsoft 1989 die erste Version seines Servers auf den Markt. Zielplattform war das Betriebssystem OS/2. Die Kooperation sah so aus, dass beide gemeinsam am Sybase Server arbeiteten und Sybase das Produkt unter seinem Namen auf UNIX-basierenden Systemen anbot und Microsoft auf OS/2.

Spätestens seit 1992 speist sich der Quelltext beider Server-Produkte (Sybase 4.0 und MS SQL Server 4.2) gleichwertig aus beiden Firmen. Ab 1993 ist nicht mehr OS/2, sondern Windows NT die Zielplattform des MS SQL Servers. Mit der Portierung auf Windows NT wurden erhebliche Anpassungen im Quelltext notwendig, die nicht mehr für das Sybase-Produkt verwendet werden konnten; die beiden Systeme begannen sich voneinander zu entfernen. Besonders der Wunsch von Sybase, dass der Quelltext möglichst plattformneutral bleiben sollte, widersprach den strategischen Zielen Microsofts, die eine volle Unterstützung der damals wirklich bedeutsamen Neuerungen von Windows NT anstrebten.

Microsoft entschied, dass der SQL Server ein zentrales Produkt für die Windows NT Strategie werden sollte. Daher wurden von anderen Datenbankherstellern erfahrene Entwickler angeworben. Dieses geballte Know-how mündete 1998 in die Serverversion 7.0 mit dem Arbeitstitel *Sphinx*. Mit dieser völlig überarbeiteten Version wurde auch die Zusammenarbeit mit Sybase überflüssig und beide Produkte sind seitdem voneinander unabhängig.

Der MS SQL Server wurde in der Folge immer weiter ausgebaut: Es kamen Sprachelemente hinzu, die Sicherheit wurde verbessert, die Performance gesteigert, die Stabilität erhöht, andere Betriebssysteme werden unterstützt usw. Besonders das grafische Frontend und die hohe Integration in die Visual Studio Entwicklungsumgebung werden von vielen Entwicklern geschätzt.

Anders als MySQL/MariaDB oder PostgreSQL kommt der MS SQL Server aber nicht aus der Open-Source- oder Uni-Ecke. Er ist ein rein kommerzielles Produkt und hat daher in den entsprechenden Kreisen ein Imageproblem.

# 2

## Einführung in relationale Datenbanken

### ■ 2.1 Was ist eine relationale Datenbank?



Worüber reden wir hier überhaupt? Relationale Datenbanken werden in Abgrenzung zu anderen Datenbanken eingeführt. Der Begriff einer Tabelle wird intensiv anhand eines Beispiels besprochen. Ebenso wird erklärt, wie Tabellen untereinander in Kontakt bleiben.

- Grundkurs
  - Abgrenzung relationale Datenbank zur hierarchischen Datenbank
  - Grundsätzlicher Aufbau einer Tabelle
  - Schlüssel und Primärschlüssel
  - Fremdschlüssel und Verknüpfungen
- Vertiefendes
  - COBOL-Daten als hierarchische Datenbank
  - XML als hierarchische Datenbank
  - Formale Definitionen zu den Grundbegriffen
  - Viele Worte für eine Sache: Begriffsübersicht
  - Starke und schwache Entitätentypen

#### 2.1.1 Abgrenzung zu anderen Datenbanken

Als *Erfinder* relationaler Datenbanken gilt Dr. E. F. Codd. Dieser hatte in einem Paper<sup>1</sup> den Begriff eingeführt und die Vorteile seiner Lösung beschrieben.

Der Begriff *relationale Datenbank* selbst ist für uns ein bisschen missverständlich. Die Bedeutung wird meist fälschlicherweise wie folgt beschrieben: *Zwischen den Daten bestehen Beziehungen (Relationen); daher der Name.* Nun ist aber der englische Begriff für Beziehung *relationship* und nicht *relation*; somit kann das nicht die Bedeutung sein.

*Relation* ist ein wohldefinierter Fachbegriff und kann umgangssprachlich als *Tabelle* übersetzt werden. Relationale Datenbanken sind also Datenbanken, die in Tabellen organisiert

<sup>1</sup> Siehe [Cod70]