

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

SQL

Ralf ADAMS

Der Grundkurs für
Ausbildung und Praxis

5. Auflage



MIT: SQL Injection abwenden

HANSER



Ihr Plus – digitale Zusatzinhalte!

Auf unserem Download-Portal finden Sie zu diesem Titel kostenloses Zusatzmaterial.

Geben Sie auf **plus.hanser-fachbuch.de** einfach diesen Code ein:

plus-6qrL3-2dsm7



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



Ralf Adams

SQL

Der Grundkurs für Ausbildung und Praxis

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

5. Auflage

HANSER

Der Autor:

Ralf Adams, Bochum

Kontakt: sqibuch@ralfadams.de

Alle in diesem Werk 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 Werk 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 wenig übernehmen Autor und Verlag die Gewähr dafür, dass beschriebene Verfahren usw. frei von Schutzrechten Dritter sind. Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt also 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.

Die endgültige Entscheidung über die Eignung der Informationen für die vorgesehene Verwendung in einer bestimmten Anwendung liegt in der alleinigen Verantwortung des Nutzers.

Aus Gründen der besseren Lesbarkeit wird auf die gleichzeitige Verwendung der Sprachformen männlich, weiblich und divers (m/w/d) verzichtet. Sämtliche Personenbezeichnungen gelten gleichermaßen für alle Geschlechter.

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 – mit Ausnahme der in den §§ 53, 54 URG genannten Sonderfälle –, reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

© 2024 Carl Hanser Verlag München, <http://www.hanser-fachbuch.de>

Lektorat: Sylvia Hasselbach

Copy editing: Sandra Gottmann, Wasserburg

Layout: der Autor mit LaTeX

Coverkonzept: Marc Müller-Bremer, www.rebranding.de, München

Covergestaltung und Titelmotiv: Tom West, unter Verwendung von Grafiken von

© [shutterstock.com](https://www.shutterstock.com)/Viktorus

Druck und Bindung: Hubert & Co. GmbH & Co. KG BuchPartner, Göttingen

Printed in Germany

Print-ISBN: 978-3-446-47913-5

E-Book-ISBN: 978-3-446-47919-7

E-Pub-ISBN: 978-3-446-48028-5

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

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

Inhalt

Vorwort zur 5. Auflage	XVII
Teil I Was man so wissen sollte	1
1 Datenbanksystem	3
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
2 Relationale Datenbanken	11
2.1 Einführung	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	21
2.2.1 Darstellung von Tabellen im ER-Modell	22
2.2.2 1:1-Verknüpfung	23
2.2.2.1 Wann liegt eine 1:1-Verknüpfung vor?	23
2.2.2.2 Wie kann ich eine 1:1-Verknüpfung darstellen?	25
2.2.2.3 Kann ich die Kardinalität genauer beschreiben?	25
2.2.3 1:n-Verknüpfung	26
2.2.3.1 Wann liegt eine 1:n-Verknüpfung vor?	26
2.2.3.2 Wie kann ich eine 1:n-Verknüpfung darstellen?	27
2.2.3.3 Kann ich die Kardinalität genauer beschreiben?	27

2.2.4	<i>n:m</i> -Verknüpfung	28
2.2.4.1	Wann liegt eine <i>n:m</i> -Verknüpfung vor?	28
2.2.4.2	Wie kann ich eine <i>n:m</i> -Verknüpfung darstellen?	30
2.2.4.3	Kann ich die Kardinalität genauer beschreiben?	30
2.2.5	Aufgaben zum ER-Modell	30
2.3	Referenzielle Integrität	31
2.3.1	Verletzung der referenziellen Integrität durch Löschen	32
2.3.2	Verletzung der referenziellen Integrität durch Änderungen	33
2.4	Normalformen	33
2.4.1	Normalform 1	34
2.4.2	Normalform 2	36
2.4.3	Normalform 3	37
2.4.4	Normalform Rest	39
3	Unser Beispiel: Ein Online-Shop	41
3.1	Kundenverwaltung	41
3.2	Artikelverwaltung	42
3.3	Bestellwesen	43
Teil II	Datenbank aufbauen	45
4	Installation des Servers	47
4.1	MySQL unter Windows 11	47
4.2	MariaDB unter Windows 11	51
4.3	Andere Installationen mit Docker	55
4.3.1	MySQL	56
4.3.2	MariaDB	58
4.3.3	PostgreSQL	59
4.3.4	Microsoft SQL Server	60
5	Datenbank und Tabellen anlegen	61
5.1	Die Programmiersprache SQL	61
5.2	Anlegen der Datenbank	62
5.2.1	Wie rufe ich den MySQL Client auf?	63
5.2.2	Wie lege ich eine Datenbank an?	64
5.2.3	Wie lösche ich eine Datenbank?	65
5.2.4	Wie weise ich einen Zeichensatz zu?	66
5.2.5	Wie weise ich eine Sortierung zu?	68

5.3	Anlegen der Tabellen	70
5.3.1	Welche Datentypen gibt es?	70
5.3.2	Wie lege ich eine Tabelle an?	71
5.3.3	Wann eine Aufzählung (ENUM) und wann eine neue Tabelle?	74
5.3.4	Wann ein DECIMAL und wann ein DOUBLE?	76
5.3.5	Wann verwende ich NOT NULL?	77
5.3.6	Wie lege ich einen Fremdschlüssel fest?	80
5.3.7	Wie kann ich Tabellen aus anderen herleiten?	85
5.3.8	Ich brauche mal eben kurz 'ne Tabelle!	86
6	Indizes anlegen	89
6.1	Index für Anfänger	89
6.1.1	Wann wird ein Index automatisch erstellt?	90
6.1.2	Wie kann ich einen Index manuell erstellen?	93
6.2	Und jetzt etwas genauer	95
6.2.1	Wie kann ich die Schlüsseleigenschaft erzwingen?	95
6.2.2	Wie kann ich Dubletten verhindern?	96
6.2.3	Was bedeutet Indexselektivität?	98
6.2.4	Wie kann ich einen Index löschen?	100
7	Werte in Tabellen einfügen	101
7.1	Daten importieren	101
7.1.1	Das CSV-Format	101
7.1.2	CSV-Daten laden mit LOAD DATA INFILE	103
7.1.3	Was ist, wenn ich geänderte Werte importieren will?	107
7.2	Daten anlegen	108
7.2.1	Wie lege ich mehrere Zeilen mit einem Befehl an?	109
7.2.2	Wie kann ich eine einzelne Zeile anlegen?	110
7.2.3	Vorsicht Constraints!	111
7.2.4	Einfügen von binären Daten über einen C#-Client	112
7.2.5	Einfügen von binären Daten LOAD FILE	115
7.3	Daten kopieren	116
Teil III	Datenbank ändern	119
8	Datenbank und Tabellen umbauen	121
8.1	Eine Datenbank ändern	121
8.2	Eine Datenbank löschen	123

8.3	Eine Tabelle ändern	125
8.3.1	Wie kann ich den Namen der Tabelle ändern?	125
8.3.2	Wie kann ich eine Spalte hinzufügen?	126
8.3.3	Wie kann ich die Spezifikation einer Spalte ändern?	128
8.3.4	Zeichenbasierte Spalten in der Länge verändern	129
8.3.5	Zeichensatz verändern	130
8.3.6	Zeichenbasierte Spalten in numerische Spalten verändern	130
8.3.7	Numerische Spalten im Wertebereich verändern	131
8.3.8	Datum- oder Zeitspalten verändern	131
8.3.9	Wie kann ich aus einer Tabelle Spalten entfernen?	133
8.4	Eine Tabelle löschen	134
8.4.1	Einfach löschen	134
8.4.2	Was bedeuten CASCADE und RESTRICT?	135
9	Werte in Tabellen verändern	137
9.1	WHERE-Klausel	137
9.1.1	Wie formuliere ich eine einfache Bedingung?	138
9.1.2	Wird zwischen Groß- und Kleinschreibung unterschieden?	139
9.1.3	Wie formuliere ich eine zusammengesetzte Bedingung?	140
9.2	Tabelleninhalte verändern	142
9.2.1	Szenario 1: Einfache Wertzuweisung	143
9.2.2	Szenario 2: Berechnete Werte	144
9.2.3	Szenario 3: Gebastelte Zeichenketten	145
9.2.4	Was bedeuten LOW_PRIORITY und IGNORE?	145
9.3	Tabelleninhalte löschen	146
9.3.1	Und was passiert bei Constraints?	147
9.3.2	Was passiert mit dem AUTO_INCREMENT?	148
9.3.3	Was bedeuten LOW_PRIORITY, QUICK und IGNORE?	149
9.3.4	Wie kann ich eine Tabelle komplett leeren?	149
Teil IV	Datenbank auswerten	151
10	Einfache Auswertungen	153
10.1	Ausdrücke	154
10.1.1	Konstanten	154
10.1.2	Wie kann ich Berechnungen vornehmen?	154
10.1.3	Wie ermittle ich Zufallszahlen?	155
10.1.4	Wie stecke ich das Berechnungsergebnis in eine Variable?	157

10.2	Zeilen- und Spaltenwahl	158
10.3	Sortierung	159
10.3.1	Was muss ich bei der Sortierung von Texten beachten?	161
10.3.2	Wird zwischen Groß- und Kleinschreibung unterschieden?	163
10.3.3	Wie werden Datums- und Uhrzeitwerte sortiert?	165
10.3.4	Wie kann ich das Sortieren beschleunigen?	166
10.4	Mehrfachausgaben unterbinden	169
10.4.1	Fallstudie: Datenimport von Bankdaten	170
10.4.2	Was muss ich beim DISTINCT bzgl. der Performance beachten?	173
10.5	Ergebnismenge ausschneiden	173
10.5.1	Wie kann ich die ersten n Datensätze ausschneiden?	173
10.5.2	Wie kann ich Teilmengen mittendrin ausschneiden?	174
10.6	Ergebnisse exportieren	176
10.6.1	Wie lege ich eine Exportdatei auf dem Server an?	176
10.6.2	Wie lege ich eine Exportdatei auf dem Client an?	177
10.6.3	Wie lese ich mithilfe eines C#-Client binäre Daten aus?	177
11	Tabellen verbinden	179
11.1	Heiße Liebe: Primär-Fremdschlüsselpaare	180
11.2	INNER JOIN zwischen zwei Tabellen	183
11.2.1	Bauanleitung für einen INNER JOIN	184
11.2.2	Abkürzende Schreibweisen	188
11.2.3	Als Datenquelle für temporäre Tabellen	189
11.2.4	JOIN über Nichtschlüsselspalten	191
11.3	INNER JOIN über mehr als zwei Tabellen	193
11.4	Es muss nicht immer heiße Liebe sein: OUTER JOIN	196
11.5	Narzissmus pur: Self Join	201
11.6	Eine Verknüpfung beschleunigen	205
12	Differenzierte Auswertungen	207
12.1	Statistisches mit Aggregatfunktionen	207
12.2	Tabelle in Gruppen zerlegen	210
12.3	Gruppenergebnisse filtern	214
12.4	Noch Fragen?	216
12.4.1	Kann ich nach Ausdrücken gruppieren?	216
12.4.2	Kann ich nach mehr als einer Spalte gruppieren?	217
12.4.3	Wie kann ich GROUP BY beschleunigen?	217
12.4.4	Parallele Bearbeitung – unterschiedliche Ergebnisse?	219
12.5	Aufgaben	219

13	Auswertungen mit Unterabfragen	221
13.1	Das Problem und die Lösung	221
13.2	Nicht korrelierende Unterabfrage	224
13.2.1	Skalarunterabfrage	224
13.2.1.1	Beispiel 1: Banken mit höchster BLZ	224
13.2.1.2	Beispiel 2: Überdurchschnittlich teure Artikel	225
13.2.1.3	Beispiel 3: Überdurchschnittlich wertvolle Bestellungen	226
13.2.2	Listenunterabfrage	227
13.2.2.1	Beispiel 1: IN ()	227
13.2.2.2	Beispiel 2: ALL ()	229
13.2.2.3	Beispiel 3: ALL ()	230
13.2.2.4	Beispiel 4: ANY ()	232
13.2.2.5	Unterschied zwischen IN (), ALL () und ANY ()	234
13.2.2.6	Unterschied zwischen NOT IN () und <> ALL ()	235
13.2.3	Tabellenunterabfrage	235
13.3	Korrelierende Unterabfrage	236
13.3.1	Beispiel 1: Rechnungen mit vielen Positionen	236
13.3.2	Beispiel 2: EXISTS	237
13.4	Common Table Expression versus Unterabfrage	237
13.5	Fallstudie Datenimport	239
13.6	Wie ticken Unterabfragen intern?	243
13.7	Aufgaben	244
14	Mengenoperationen	245
14.1	Die Vereinigung mit UNION	245
14.2	Die Schnittmenge	248
14.2.1	Mit INTERSECT	248
14.2.2	Mit Unterabfragen	249
14.3	Die Differenzmenge	250
14.3.1	Mit EXCEPT	250
14.3.2	Mit Unterabfragen	251
14.4	UNION, INTERSECT und EXCEPT ... versteh' ich nicht!	252
15	Bedingungslogik	255
15.1	Warum ein CASE?	255
15.2	Einfacher CASE	257
15.3	Searched CASE	259
15.4	Fallbeispiele	261
15.4.1	Lagerbestand überprüfen	261

15.4.2	Kundengruppen ermitteln	262
15.4.3	Aktive Lieferanten ermitteln	265
15.4.4	Aufgaben	266
16	Ansichtssache	267
16.1	Was ist eine Ansicht?	267
16.1.1	Wie lege ich eine Ansicht an?	268
16.1.2	Wie wird eine Ansicht verarbeitet?	270
16.1.3	Wie lösche ich eine Ansicht?	273
16.1.4	Wie ändere ich eine Ansicht?	276
16.2	Anwendungsgebiet: Vereinfachung	276
16.3	Anwendungsgebiet: Datenschutz	279
16.4	Grenzen einer Ansicht	279
17	Exkurs NoSQL	283
17.1	Vorbereitung der MySQL-Shell	283
17.2	Datenmodellierung des Warenkorbs	285
17.2.1	JavaScript Object Notation (JSON)	285
17.2.2	Struktur unseres JSON-Dokuments	286
17.3	NoSQL: MySQL mit JavaScript-Client	287
17.3.1	Anlegen eines Warenkorbs	288
17.3.2	Inhalte des Warenkorbs anlegen	289
17.3.3	Inhalte des Warenkorbs auswerten	293
17.3.4	Inhalte des Warenkorbs verändern	296
17.4	NoSQL: klassisches SQL mit JSON-Funktionen	297
17.4.1	Anlegen eines Warenkorbs	298
17.4.2	Inhalte des Warenkorbs anlegen	298
17.4.3	Inhalte des Warenkorbs auswerten	300
17.4.4	Inhalte des Warenkorbs verändern	302
17.4.5	Inhalte des Warenkorbs löschen	305
Teil V	Anweisungen kapseln	307
18	Locking	309
19	Transaktionen	313
19.1	Das Problem	313
19.2	Was ist eine Transaktion?	315
19.3	Isolationsebenen	318

19.3.1	READ UNCOMMITTED	319
19.3.2	READ COMMITTED	320
19.3.3	REPEATABLE READ	321
19.3.4	SERIALIZABLE	322
19.4	Fallbeispiel in C#	323
19.5	Deadlock	325
20	Prozeduren	327
20.1	Einstieg und Variablen	328
20.2	Verzweigung	333
20.2.1	Einfache Verzweigung mit IF	333
20.2.2	Mehrfache Verzweigung mit CASE	336
20.3	Schleifen	339
20.3.1	LOOP-Schleife	340
20.3.2	WHILE-Schleife	342
20.3.3	REPEAT-Schleife	345
20.4	Transaktion innerhalb einer Prozedur	346
20.5	Cursor	347
20.6	Aufgaben	354
21	Funktionen	355
22	Auslöser	357
22.1	Was ist das?	357
22.2	Ein Beispiel für einen INSERT-Trigger	359
22.3	Ein Beispiel für einen UPDATE-Trigger	360
22.4	Ein Beispiel für einen DELETE-Trigger	362
23	Ereignisse	365
23.1	Wie lege ich ein Ereignis an?	365
23.2	Wie werde ich Ereignisse wieder los?	368
Teil VI	Anhänge	369
24	Datenbank administrieren	371
24.1	Daten sichern und wiederherstellen	371
24.1.1	Backup	371
24.1.2	Restore	373
24.2	Benutzerrechte	373
24.2.1	Benutzerrechte und Privilegien	373

24.2.2	Benutzer anlegen und Rechte zuweisen	376
24.2.2.1	CREATE USER bzw. CREATE ROLE	376
24.2.2.2	Rechte vergeben mit GRANT	378
24.2.2.3	Rechte entziehen mit REVOKE	379
24.3	MySQL und MariaDB Engines	380
25	SQL-Injection	383
25.1	Das Problem	383
25.2	Beispiel: Suchmaske	384
25.3	Gegenmaßnahmen	391
25.3.1	Never Trust an Unknown Input	391
25.3.2	Trennung von Anweisungen und Daten	392
25.3.2.1	Verwenden Sie Prozeduren oder Funktionen	392
25.3.2.2	Verwenden Sie Prepared Statements	392
25.3.3	Ich darf nur, was ich soll	393
25.3.4	Nichtssagende Fehlermeldungen	394
26	SQL-Referenz	395
26.1	Datentypen	395
26.1.1	Numerische Datentypen	395
26.1.1.1	Ganze Zahlen	395
26.1.1.2	Gebrochene Zahlen	396
26.1.2	Zeichen-Datentypen	397
26.1.3	Datums- und Zeit-Datentypen	398
26.1.4	Binäre Datentypen	401
26.1.5	JSON	401
26.1.6	Räumliche Datentypen	402
26.1.7	Standardwerte	403
26.1.8	Zusätze für Datentypen	404
26.2	Operatoren und Funktionen	406
26.2.1	Mathematische Operatoren	406
26.2.2	Mathematische Funktionen	406
26.2.3	Aggregatfunktionen	409
26.3	Bedingungen	412
26.3.1	Vergleichsoperatoren	412
26.3.2	Logikoperatoren	415
26.3.2.1	NOT, Negation, \neg	415
26.3.2.2	AND, Konjunktion, \wedge	415
26.3.2.3	OR, Disjunktion, \vee	416

26.3.2.4	XOR, Antivalenz, ⊗	416
26.3.2.5	Verallgemeinerung auf mehr als zwei Operanden	416
26.4	Befehle	417
26.4.1	Data Definition Language	417
26.4.2	Data Manipulation Language	429
26.4.3	Benutzerverwaltung	433
27	Ausgewählte Quelltexte	437
27.1	DOUBLE versus DECIMAL	437
27.2	Rundungsfehler	441
27.3	NULL versus NOT NULL	441
27.4	Suchen mit und ohne Index	444
27.5	Messen der Performance der Einfügeoperation	447
27.6	Messen der Indexselektivität	450
27.7	Sortieren ohne und mit Index	452
28	Quelltexte	457
28.1	MySQL/MariaDB	457
28.1.1	Quelltexte zu Teil II	457
28.1.2	Quelltexte zu Teil III	469
28.1.3	Quelltexte zu Teil IV	473
28.1.4	Quelltexte zu Teil V	517
28.1.5	Quelltexte zu Teil VI	531
28.2	PostgreSQL	536
28.2.1	Quelltexte zu Teil II	536
28.2.2	Quelltexte zu Teil III	545
28.2.3	Quelltexte zu Teil IV	549
28.2.4	Quelltexte zu Teil V	579
28.3	Microsoft SQL Server	584
28.3.1	Quelltexte zu Teil II	584
28.3.2	Quelltexte zu Teil III	595
28.3.3	Quelltexte zu Teil IV	600
Literatur		635
Stichwortverzeichnis		641

Vorwort zur 5. 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, der dann durch berufliche Praxis ausgebaut werden kann. Trotzdem vertiefe ich an vielen Stellen im Buch den Einblick in SQL oder den MySQL Server – zum einen, um zu zeigen, dass ich auch ein bisschen was draufhabe, zum anderen, um Neugierde und Jagdtrieb beim Leser¹ 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 so viel 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.

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.

¹ 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.

Wenn Sie auf

plus.hanser-fachbuch.de

den Code

plus-6qrL3-2dsm7

eingeben, können Sie Skripte, Beispiele und Musterlösungen downloaden. Diese wurden auf folgenden Servern getestet: *MySQL Community Server 8.0.33*, *MariaDB 10.6.14*, *MariaDB 11.0.2*, *PostgreSQL 15.3* und *MS SQL Server 2022 16.x*. Alle Server liefen in einer Dockerinstanz unter Debian 12 (Bookworm). 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 Dialekte oder Eigenheiten begründet sind, musste ich auf eine Transkription verzichten.

Seit kurzem gibt es außerdem im Hanser Verlag den Hanser eCampus, das adaptive Kursangebot für die Hochschullehre und die Unternehmensweiterbildung. Hier ist der E-Learning-Kurs *Datenbankgrundlagen* erschienen, der auf Teil I dieses Buches basiert. Weitere Infos zum Kurs und eine Demoversion finden Sie hier:

<https://www.hanser-ecampus.de/kurse/informatik/datenbankgrundlagen>.

Danksagung

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

Besonders will ich mich bei meinen Schülerinnen und Schülern der Technischen Beruflichen Schule 1 in Bochum (<http://www.tbs1.de>) bedanken. 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 aber die 5. Auflage dieses Buchs erscheinen kann, ist in erster Linie Ihnen, liebe Leserinnen und Leser, zu verdanken; dafür ein herzliches *Dankeschön!*

Ralf Adams, September 2023

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
 - Datenbank
 - Datenbankmanagementsystem
 - Datenbanksystem

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](#) 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](#)). 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.

Üblicherweise werden in einer Datenbank folgende Datenobjekte vorkommen¹:

- *Tabellen*: Bei einer relationalen Datenbank werden die Daten in Tabellen organisiert (Kundentabelle, Artikeltable, Filmtabelle usw.). Deshalb sind die Tabellen das Herz-

¹ Die Liste ist nicht vollständig. Ich werde mich aber in diesem Buch auf diese beschränken.

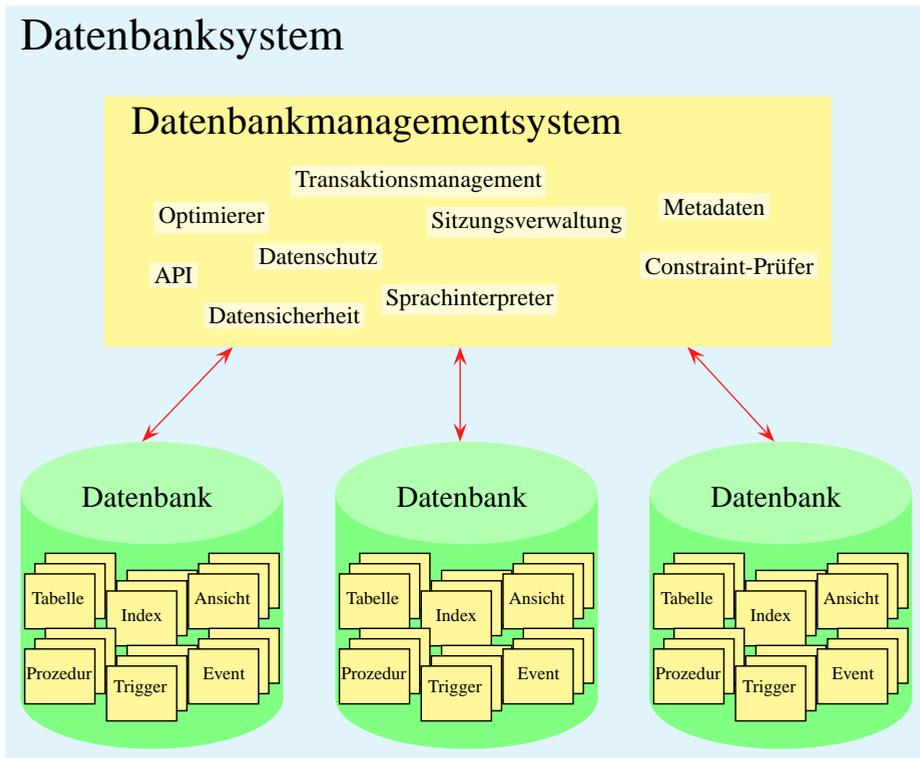


Bild 1.1 Aufbau eines Datenbanksystems

stück 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.
- *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 DBMS²:

- *Sprachinterpret*: Herzstück des DBMS ist der Interpret³. 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.
- *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.

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

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

- *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 GAU⁴ 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*⁵: 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. Diese APIs werden in der MySQL-/MariaDB-Welt *Konnektoren* genannt. Der MySQL oder MariaDB Server bietet beispielsweise APIs für C, C++, C#.NET, Node.js PHP, Perl, Python, Ruby und Tcl. Auch stehen APIs für JDBS⁶ und ODBC⁷ zur Verfügung. MariaDB kennt zusätzlich noch Schnittstellen zu Erlang, Jupyter, Excel, Swift und R.
- *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)*; bei objektorientierten Datenbanken spricht man analog von *objektorientierten Datenbankmanagementsystemen (OODBMS)*.

Und noch der Vollständigkeit halber:



Definition 3: Datenbanksystem

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

⁴ Abkürzung für: Größter anzunehmender Unfall

⁵ Abkürzung für: Application Programming Interface; engl. für Programmierschnittstelle

⁶ Abkürzung für: Java Database Connectivity: Programmierschnittstelle für JAVA

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

1.2 Im Buch verwendete Server



Kurzporträts der verwendeten SQL Server: Hersteller und Geschichte

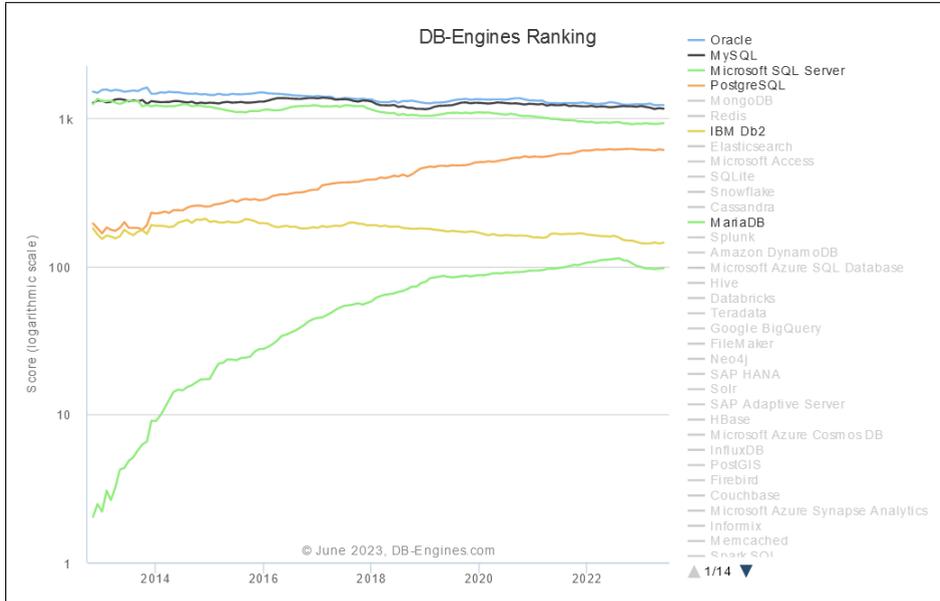


Bild 1.2 Ranking einiger RDBMSs, Quelle [DE23]

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 seit 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.

Wollen Sie auf die Installation vom Servern verzichten, können Sie auch Online diverse Server verwenden, um SQL-Code auszuprobieren. Natürlich gibt es dabei Einschränkungen. Ich empfehle: <http://sqlfiddle.com/> und <https://dbfiddle.uk/>.

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 (kein Tippfehler!) genutzt. Von 2008 bis 2010 wurde MySQL 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 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 Bugfixes hat MariaDB an vielen Stellen – aber nicht, wie oft behauptet, an den meisten – MySQL abgelöst (siehe [Bild 1.2](#)).

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.

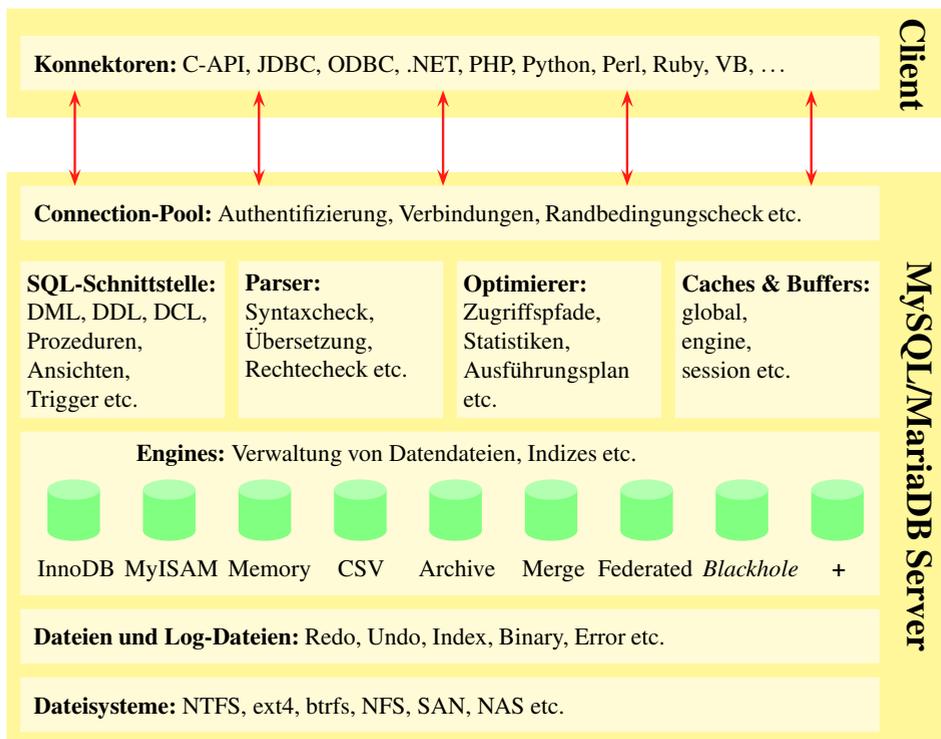


Bild 1.3 Komponenten von MySQL und MariaDB

- *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.⁸
- *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 [MyS21e]) 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.

Zur Geschichte von MySQL gibt es einen netten Absatz in Wikipedia (<https://de.wikipedia.org/wiki/MySQL#Geschichte>) und Dokumentation und Downloads finden Sie auf <http://www.mysql.com>. Die Homepage von MariaDB ist unter <https://mariadb.org> zu finden. Eine kurze, aber gute Übersicht bezüglich der aktuellen Unterschiede zwischen MariaDB und MySQL, finden Sie unter [Ahm22].

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 Projekt an der Universität von Kalifornien, Berkeley. Professor Michael Stonebraker⁹ 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,

⁸ MySQL hat ab Version 8.0 den Cache abgeschafft ([MyS18c]); MariaDB verwendet ihn weiterhin ([Mar23b]).

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

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 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.

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 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. 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.

Der MS SQL Server kann in verschiedenen Varianten von <https://www.microsoft.com/de-de/sql-server/sql-server-downloads> bezogen werden; die Dokumentation liegt in <https://docs.microsoft.com/de-de/sql/?view=sql-server-ver15>.