

Erik Schönwälder

inkl.
E-Book

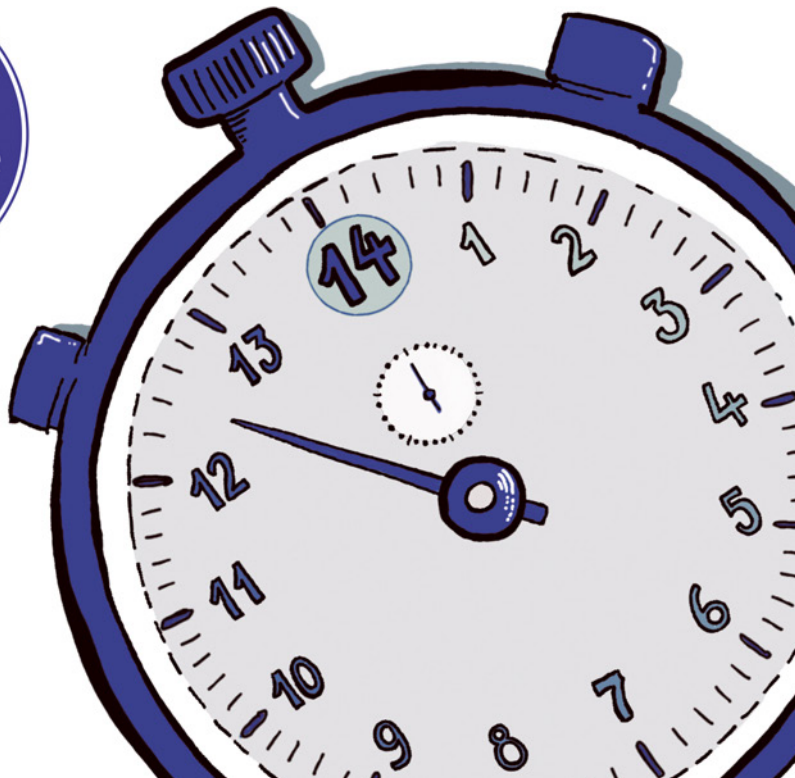
SQL

Schnelleinstieg

Datenbanken abfragen und verwalten
in 14 Tagen

Einfach und ohne Vorkenntnisse

Zahlreiche
Praxisbeispiele



mitp

Inhalt

Einleitung

Programmieren lernen in 14 Tagen	11
Der Aufbau des Buchs	11
Beispieldatenbank	12
Fragen und Feedback	12



Einstieg in die Welt der Daten

1.1 Wozu benötigen wir Datenbanken?	13
1.2 Bestandteile einer Datenbank	14
1.3 Einblick in die verschiedenen Datenbanktechnologien	15
1.4 Datenbanken und SQL, wie hängt das zusammen?	18
1.5 Identifikation von Daten mit Primär- und Fremdschlüsseln	19



Einrichten der Arbeitsfläche

2.1 Der SQL-Dialekt – Ist SQL immer gleich SQL?	23
2.2 Installation von PostgreSQL	26
2.3 Installation von MySQL	28
2.4 Einrichtung von PostgreSQL	33
2.5 Einrichtung von MySQL	37
2.6 Einführung in den Datenbestand	42
2.7 Unterschiede zwischen PostgreSQL und MySQL	44



Die ersten Schritte – das Fundament einer jeden Datenbankabfrage

3.1	Was ist eine Datenbankabfrage?	45
3.2	Daten abfragen mit SELECT und FROM	46
3.3	Doppelte Daten entfernen	48
3.4	Exkursion in die SQL-Syntax	50
3.5	Abfragen anpassen – Spalten und Tabellen umbenennen	52
3.6	Von klein zu groß oder eher andersherum? – Sortieren der Ergebnismenge	53
3.7	Zu viele Ergebnisse? Der LIMIT-Befehl als Lösung	57
3.8	Übungsaufgaben	58
3.9	Lösungen	59



Die Datenflut bezwingen – Daten mit dem WHERE-Befehl filtern

4.1	Was ist eine Bedingung?	61
4.2	Exkursion in die Aussagenlogik – Was ist wahr und was ist falsch? ..	67
4.3	Wenn eine Bedingung nicht ausreicht – Verknüpfungen mehrerer Bedingungen	69
4.4	Von ... bis – Bedingungen mit Bereichsangaben	73
4.5	Bedingungen mit benutzerdefinierten Zeichenketten	78
4.6	Mehr Flexibilität mit regulären Ausdrücken	82
4.7	Übungsaufgaben	88
4.8	Lösungen	89



Gruppierung von Daten und mit Daten rechnen

5.1	Daten in Gruppen ablegen	93
5.2	Aggregatfunktionen	96
5.3	Entfernen von Gruppen	105
5.4	Übungsaufgaben	108
5.5	Lösungen	109

6

Verknüpfen von mehreren Tabellen

6.1	Verbinden von Tabellen	111
6.2	Zu viele Befehle? In dieser Reihenfolge laufen die Befehle ab	121
6.3	Übungsaufgaben	123
6.4	Lösungen	123

7

Verschachteln von Datenbankabfragen und Datenmanipulation

7.1	Komplexe Abfragen mittels Unterabfragen erstellen	125
7.2	Manipulation von numerischen Daten	129
7.3	Manipulation von Zeichenketten	132
7.4	Weitere nützliche Funktionen	136
7.5	Übungsaufgaben	139
7.6	Lösungen	140

8

Verknüpfen von mehreren Ergebnissen

8.1	Vereinigung zweier Ergebnisse	143
8.2	Schnittmenge zweier Ergebnisse	148
8.3	Differenz zweier Ergebnisse	150
8.4	Die symmetrische Differenz	151
8.5	Übungsaufgaben	153
8.6	Lösungen	153



Erstellen der ersten eigenen Datenbank

9.1	Erstellen einer Datenbank	157
9.2	Erstellen einer eigenen Relation	162
9.3	Exkursion in die Datentypen	163
9.4	Einfügen von Werten	168
9.5	Übungsaufgaben	172
9.6	Lösungen	172



Was nicht in eine Datenbank gehört – Einschränkungen definieren

10.1	Daten mit Nullwerten filtern	176
10.2	Daten mit Nullwerten ersetzen	178
10.3	Daten mit Duplikaten filtern	180
10.4	Filter mit Bedingungen erstellen	181
10.5	Primärschlüssel einer Tabelle festlegen	183
10.6	Fremdschlüssel einer Tabelle festlegen	185
10.7	IDs automatisch hochzählen	187
10.8	Übungsaufgaben	191
10.9	Lösungen	191



Eine bestehende Datenbank mit ihrem Inhalt anpassen

11.1	Aktualisierung bestehender Tabellen	195
11.2	Nachträgliches Anpassen von Datentypen und Einschränkungen	199
11.3	Aktualisierung bestehender Werte	205
11.4	Löschen bestehender Werte	208
11.5	Übungsaufgaben	212
11.6	Lösungen	213

12

Wer darf eigentlich was? – Rechteverwaltung und Zugriffskontrolle

12.1	Kann jeder Nutzer mit der Datenbank arbeiten? – Benutzermanagement in SQL	218
12.2	Vergabe von Rechten	219
12.3	Entzug von Rechten und Löschen von Benutzern	227
12.4	Einfache Nutzerverwaltung mit Benutzerrollen	229
12.5	Übungsaufgaben	231
12.6	Lösungen	231

13

SQL für Fortgeschrittene – Aktion und Reaktion

13.1	Was ist ein Trigger?	233
13.2	Erstellen eines Triggers	234
13.3	Die Reaktion auf ein Ereignis – Trigger-Funktionen	237
13.4	Testen von Triggern	244
13.5	Verändern und Löschen von bestehenden Triggern	245
13.6	Übungsaufgabe	246
13.7	Lösung	247

14

Ausblick – Gibt es noch mehr?

14.1	Views – verschiedene Sichten auf eine Datenbank	250
14.2	Datenzugriffe beschleunigen mit Indizes	253
14.3	Arbeiten auf einer Datenbank mit mehreren Benutzern und Abstürze	258

Stichwortverzeichnis

Einleitung

Programmieren lernen in 14 Tagen

Mit diesem Buch haben Sie sich für einen einfachen, praktischen und fundierten Einstieg in die Welt der Programmierung entschieden. Sie lernen ohne unnötigen Ballast alle Grundlagen, um SQL effektiv in Ihrem Berufs- und Interessensgebiet einzusetzen.

Wenn Sie Zeit genug haben, können Sie jeden Tag ein neues Kapitel durcharbeiten und so innerhalb von zwei Wochen den Umgang mit SQL erlernen. Alle Erklärungen sind leicht verständlich formuliert und setzen keine Vorkenntnisse voraus. Am besten lesen Sie das Buch neben der Computertastatur und probieren die Codebeispiele und Übungen gleich aus.

Der Aufbau des Buchs

Das Buch ist in zwei Hauptteile gegliedert: Datenbankabfragen und Datenbankerstellung. Bevor es allerdings mit dem ersten Teil startet, werden in den einführenden Kapiteln alle wichtigen Grundlagen erklärt: die Funktionsweise von Datenbanken, die Integration von SQL sowie die Installation und Einführung der notwendigen Software. Ausgerüstet mit dem gelernten Fundament, legt das erste Kapitel den Fokus auf das Abfragen von Daten aus einer Datenbank. Schritt für Schritt lernen Sie hier, wie Abfragen formuliert werden, indem Sie Daten filtern, gruppieren oder auch verknüpfen.

Der zweite Teil des Buchs widmet sich der praktischen Erstellung einer eigenen Datenbank samt ihrer Tabellen und Daten. Neben dem gesamten Erstellungsprozess wird hierbei auch ein besonderes Augenmerk auf Sicherheitskonzepte gelegt, wie etwa ein gelungenes Benutzer- und Rechtemanagement sowie die Wartung einer Datenbank durch das Aktualisieren der zugrunde liegenden Strukturen und Daten. Abschließend bieten die letzten beiden Kapitel wertvolle Hinweise und weiterführende Ansätze, wie Sie Ihre SQL-Kenntnisse nach dem Schnelleinstieg weiter vertiefen können.

Auch wenn im Buch ein besonderer Fokus auf die beiden Datenbanksysteme MySQL und PostgreSQL gelegt wird, kann das erlernte Wissen ebenfalls auf andere Systeme übertragen werden.

Ihr Tagespensum schließt mit praktischen Übungen, in denen Sie Ihr neu gewonnenes Wissen vertiefen können. Die Lösungen zu diesen Übungen finden Sie am Ende eines jeden Kapitels.

Am Ende des Buchs finden Sie ein Stichwortverzeichnis, das Ihnen hilft, bestimmte Inhalte schneller zu finden.

Beispieldatenbank

Um stets einen Praxisbezug zu gewährleisten, arbeitet das Buch mit einer Beispieldatenbank, die während der einzelnen Kapitel und auch in den Übungen verwendet wird.

Die Beispieldatenbank sowie die Lösungen zu den Übungen stehen Ihnen auf der Webseite des Verlags unter www.mitp.de/0868 zum Download zur Verfügung.

Eine nähere Beschreibung der Beispieldatenbank und wie sie in MySQL und PostgreSQL eingepflegt werden kann, finden Sie im Kapitel 2.

Fragen und Feedback

Unsere Verlagsprodukte werden mit großer Sorgfalt erstellt. Sollten Sie trotzdem einen Fehler bemerken oder eine andere Anmerkung zum Buch haben, freuen wir uns über eine direkte Rückmeldung an lektorat@mitp.de.

Falls es zu diesem Buch bereits eine Errata-Liste gibt, finden Sie diese unter www.mitp.de/0868 im Reiter DOWNLOADS.

Wir wünschen Ihnen viel Erfolg und Spaß bei der Programmierung mit SQL!



Einstieg in die Welt der Daten

»Daten« – kaum ein anderer Begriff erhält so viel Aufmerksamkeit in den Medien. Ob im Zusammenhang mit Datenschutz, Datensicherheit oder Big Data, Daten sind allgegenwärtig. Nicht zu Unrecht kommen infolgedessen zahlreiche Fragen auf: Warum sind Daten so wichtig? Wie können sie abgespeichert werden? Und wie werden sie eigentlich verwaltet?

In diesem Kapitel erhalten Sie eine Einführung in die Welt der Daten. Dabei werden nicht nur die oben angesprochenen Fragen beantwortet, sondern Sie bekommen auch elementare Grundlagen an die Hand, welche für den weiteren Verlauf des Buchs essenziell sind.

1.1 Wozu benötigen wir Datenbanken?

Personenbezogene Daten sind das Gold der heutigen Zeit. Einmal analysiert, geben sie Unternehmen wie Google, Meta oder Amazon die Möglichkeit, personalisierte Werbung oder Produktempfehlungen zu schalten. Auch kleinere Unternehmen setzen Daten gewinnbringend ein, um beispielsweise ihre Arbeitsabläufe zu optimieren oder gezielter Kunden anzuwerben. Unabhängig vom betrachteten Beispiel zeigt sich stets eine zentrale Gemeinsamkeit: die Verwendung von Datenbanken. Mithilfe von Datenbanken können die relevanten Daten gesammelt und gemeinsam abgespeichert werden. Weiterhin bieten sie die Option, abgelegte Daten je nach Art und Umfang der Analyse zu verwalten und abzurufen.

Umgangssprachlich formuliert ist eine Datenbank also ein Ort, an welchem Daten aller Art zentral gespeichert, verwaltet und wieder abgerufen werden können. Die Herkunft der Daten spielt dabei keine Rolle. Während beispielsweise eine Trockenbaufirma Datenbanken für ihre Kunden-, Mitarbeiter- und

Lieferantendaten nutzt, legt eine Shopping-Webseite hier ihre Produkt- und Bestelldaten ab.

1.2 Bestandteile einer Datenbank

Tatsächlich ist Datenbank nicht gleich Datenbank. Eine Datenbank besteht aus mehreren Teilen, deren Bezeichnungen oft durcheinandergeworfen werden. Aber beginnen wir von vorn. Ein *Datenbanksystem*, was fälschlicherweise oft als Datenbank bezeichnet wird, setzt sich aus genau zwei Teilen zusammen:

- der *Datenbank*, auch als *Datenbasis* bezeichnet
- dem *Datenbank-Management-System* (oft abgekürzt mit DBMS)

In der Datenbank werden die Daten physisch abgelegt. Ein direktes Arbeiten auf diesen Daten ist mithilfe der Datenbank jedoch nicht möglich. Hierfür ist eine spezielle Software notwendig, das Datenbank-Management-System. In der Abbildung 1.1 ist dieser Zusammenhang genauer veranschaulicht.

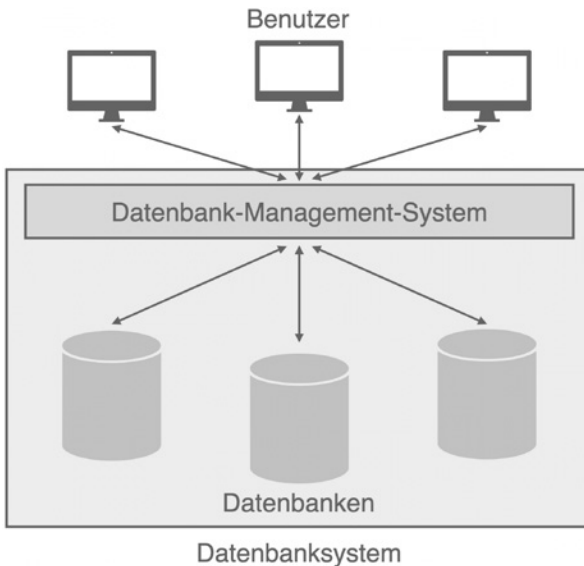


Abb. 1.1: Bestandteile eines Datenbanksystems

Möchte ein Benutzer seine Daten also mithilfe eines Datenbanksystems ablegen, so kann er nicht direkt auf die Datenbank zugreifen. Diese ist lediglich für das Speichern der Daten zuständig und eine Interaktion ist nur über eine

Schnittstelle möglich. Das Datenbank-Management-System repräsentiert diese Schnittstelle und ist für die Verwaltung der Daten verantwortlich. Sollen Daten eingepflegt, modifiziert oder gegebenenfalls gelöscht werden, so interagiert der Benutzer über das Datenbank-Management-System (kurz DBMS) mit der Datenbank.

Neben dem Abspeichern, Verwalten und Abrufen von Daten ist ein DBMS noch für weitere Aufgaben zuständig. Beispielsweise muss es sich auch um die Konsistenz einer Datenbank kümmern. Angenommen eine Firma hat all ihre Daten in einem Datenbanksystem abgelegt. Weiterhin wurde die Regel aufgestellt, dass jeder Mitarbeiter am Ende des Monats mindestens 1.000 Euro Gehalt ausgezahlt bekommt. Die Aufgabe des DBMS ist es nun sicherzustellen, dass diese Regel niemals verletzt wird. Weder das Hinzufügen eines neuen Mitarbeiters mit einem Monatslohn von weniger als 1.000 Euro noch die Reduzierung des Monatslohns unter die Mindestgrenze eines bestehenden Mitarbeiters ist also gestattet.

1.3 Einblick in die verschiedenen Datenbanktechnologien

So wie es etliche Anwendungsfälle gibt, in welchen Datenbanksysteme eingesetzt werden können, so gibt es auch etliche Arten bzw. Technologien, um ein Datenbanksystem aufzubauen. Das wohl bekannteste und in diesem Buch auch relevante Datenbanksystem ist das *relationale Datenbanksystem*. Wie der Name schon vermuten lässt, werden in einem relationalen Datenbanksystem relationale Daten, also Daten in Tabellenform, abgespeichert und alle Daten durch Relationen bzw. Tabellen beschrieben. Die Begriffe *Tabelle* und *Relation* werden in diesem Kontext oft synonym verwendet.

Eine Tabelle besteht aus mehreren Spalten, oftmals auch als *Attribute* bezeichnet, die die Daten beschreiben. Die Zeilen einer Tabelle, sogenannte *Entitäten* oder *Tupel*, stellen dann die Ausprägungen bzw. die einzelnen Bausteine der Daten dar. In der Abbildung 1.2 ist dieses abstrakte Konzept anhand eines Beispiels präsentiert.

Kunden

Kunden-ID	Vorname	Nachname	Alter
1	Sophie	Müller	24
2	Sven	Buschmann	76
3	Ralf	Schmaler	56

Zeile / Entität

Spalte / Attribut

Abb. 1.2: Beispiel einer relationalen Datenbank – Kunden-Relation

Die Tabelle bzw. Relation namens »Kunden« beinhaltet die Kundeninformationen eines Onlineshops. Zu einem Kunden werden die Attribute »Vorname«, »Nachname« und »Alter« abgespeichert. In der Tabelle sind diese Eigenschaften durch Spalten repräsentiert. Weiterhin ist jedem Kunden eine eindeutige »Kunden-ID« zugeordnet. Das Konzept hinter den IDs wird im Abschnitt 1.5 detailliert erläutert und ist an dieser Stelle noch unbedeutend. Die spezifischen Kunden, folglich die einzelnen Individuen, werden über die Zeilen ausgedrückt. Beispielsweise beschreibt die erste Zeile die 24-jährige Sophie Müller mit der ID 1 oder die dritte Zeile den 56-jährigen Ralf Schmaler mit der ID 3.

Zusammengefasst werden die Merkmale eines Kunden, oder allgemeiner gesprochen einer Entität, durch die einzelnen Spalten einer Tabelle beschrieben. Die individuellen Ausprägungen der Daten, im gezeigten Beispiel die Kunden, werden dann über die Zeilen einer Tabelle dargestellt.

Ein relationales Datenbanksystem besitzt allerdings in den seltensten Fällen nur eine Tabelle. Meistens besteht ein solches System aus Dutzenden Tabellen, die untereinander durch Beziehungen verknüpft sind. So kann es zum Beispiel sein, dass der Onlineshop aus der letzten Abbildung neben der Kunden-Relation noch eine weitere Bestellungen-Relation mit den Attributen »Bestell-ID«, »Kunden-ID«, »Produktname« und »Menge« beinhaltet.

Bestellungen

Bestell-ID	Kunden-ID	Produktname	Menge
1	3	T-Shirt	4
2	6	Hose	1
3	4	Socken	10

Abb. 1.3: Bestellungen-Relation

Die in Abbildung 1.3 dargestellte Bestellungen-Relation dokumentiert alle getätigten Bestellungen. Dabei gibt die Kunden-ID Auskunft über den Kunden, der die Bestellung aufgegeben hat. Ebenso erfasst die Relation den Produktnamen, welches Produkt bestellt wurde, sowie die Menge, in der das Produkt geordert wurde. Durch die Kunden-ID stehen die Kunden- und Bestellungen-Relationen nun miteinander in Beziehung. Durch den Zusammenschluss gleicher IDs können Tupel bzw. Entitäten beider Relationen miteinander verknüpft werden, wodurch zusätzliche Informationen abgespeichert werden.

Laut der ersten Zeile der Bestellungen-Relation hat der Kunde mit der ID 3 vier T-Shirts bestellt. Mithilfe der ID 3 können Sie nun leicht in der Kunden-Relation nach der entsprechenden Entität suchen. Resultierend stellen Sie fest, dass der 56-jährige Ralf Schmalder der Absender der entsprechenden Bestellung ist.

Neben relationalen Datenbanksystemen existiert noch eine Vielzahl weiterer Systeme. Typischerweise werden diese unter dem Begriff *NoSQL* zusammengefasst. Diese Systeme grenzen sich von der relationalen Welt ab und beschäftigen sich folglich mit Daten, die ohne eine tabellarische Struktur auskommen.

Namentlich zu erwähnen sind hier sogenannte *Document-Stores*, die schemafreie Dokumente abspeichern. Schemafrei bedeutet, dass Kunden beispielsweise nicht immer mit den gleichen Merkmalen wie in einer Tabelle ausgestattet sein müssen. So kann ein Kunde auch gern mal keinen Vor- und Nachnamen, aber dafür einen Künstlernamen besitzen oder anders als die anderen Kunden eine zusätzliche Präferenzliste mitführen, in welcher die Favoritenprodukte abgelegt sind.

In *Key-Value-Stores* dagegen werden Datenpunkte immer in Verbindung mit einem Schlüssel abgespeichert. Der Schlüssel ist zu einem späteren Zeitpunkt wichtig, um die Datenpunkte wieder abzurufen. Nicht nur ist dieses System aufgrund seiner Einfachheit sehr beliebt, es glänzt zusätzlich noch mit einem hohen Effizienz- und Flexibilitätsgrad.

Auch *Graphdatenbanken* sollten in diesem Kontext erwähnt werden. Hier werden die einzelnen Entitäten durch Knoten in einem Netzwerk aus Beziehungen repräsentiert. Insbesondere soziale Strukturen lassen sich hiermit hervorragend abbilden.

1.4 Datenbanken und SQL, wie hängt das zusammen?

Ein SQL-Buch beinhaltet neben den grundlegenden Konzepten wie der Unterscheidung verschiedener Datenbankarten oder den Bestandteilen einer Datenbank natürlich auch den essenziellsten Schwerpunkt: SQL selbst. Wie kann SQL also in den gerade erläuterten Datenbankkontext gebracht werden, und was ist SQL eigentlich?

SQL steht für »Structured Query Language« und ist eine Abfragesprache. Abfragesprachen sind ähnlich wie Programmiersprachen, nur dass sie einen geringeren Funktionsumfang besitzen und für einen spezifischen Anwendungsfall bestimmt sind. Während Programmiersprachen wie C++ oder Java quasi uneingeschränkt in verschiedensten Szenarien Anwendung finden, widmet sich eine Abfragesprache nur einer konkreten Problematik, nämlich dem Verwalten einer Datenbank. Wie bereits im Abschnitt 1.2 dargestellt, wird eine Datenbank über das Datenbank-Management-System verwaltet. Ein Datenbank-Management-System wiederum nutzt hierfür eine Abfragesprache. In relationalen Datenbanksystemen ist diese Abfragesprache standardmäßig SQL.

Zusammengefasst ist SQL also eine Abfragesprache, die zum Einfügen, Verändern, Löschen und Abrufen von Daten in einem relationalen Datenbanksystem verwendet wird.

1.5 Identifikation von Daten mit Primär- und Fremdschlüsseln

Im Abschnitt 1.3 wurden zur Erklärung eines relationalen Datenbanksystems folgende Beispielrelationen präsentiert:

Kunden

Kunden-ID	Vorname	Nachname	Alter
1	Sophie	Müller	24
2	Sven	Buschmann	76
3	Ralf	Schmalzer	56

Bestellungen

Bestell-ID	Kunden-ID	Produktname	Menge
1	3	T-Shirt	4
2	6	Hose	1
3	4	Socken	10

Abb. 1.4: Kunden- und Bestellungen-Relation

1.5.1 Primärschlüssel

Die Spalte »Kunden-ID« spielt eine bedeutende Rolle, da sie als Primärschlüssel in der Kunden-Relation fungiert. Als Primärschlüssel wird in einer Relation ein Attribut bzw. eine Gruppe von Attributen bezeichnet, die die Entitäten der Relation, folglich die Tabellenzeilen, eindeutig kennzeichnen. In der gegebenen Kunden-Relation repräsentiert beispielsweise die Kunden-ID einen solchen Primärschlüssel. Mithilfe dieses Attributs kann unkompliziert auf die einzelnen Zeilen der Kunden-Relation zugegriffen werden. Wenn beispielsweise die Kunden-ID 3 gegeben ist, so kann mit dieser eindeutig auf den 56-jährigen Ralf Schmalzer verwiesen werden.

Aber warum überhaupt eine separate ID? Könnte nicht einfach der Vor- und Nachname eines Kunden als Primärschlüssel dienen? Nun ja, es ist natürlich denkbar, dass mehrere Kunden denselben Namen besitzen. Wenn folglich mehrere »Ralf Schmalzer« in der Relation abgespeichert werden, kann mit Vor- und Nachnamen nicht mehr eindeutig auf eine Entität bzw. eine Zeile gezeigt werden. Welcher Ralf Schmalzer ist denn gemeint? Um diesem Problem aus dem Weg zu gehen, wird üblicherweise ein künstlicher Primärschlüssel wie eine ID oder eine Nummer als zusätzliches Attribut herangezogen. Wird schlussfolgernd eine neue Entität in die entsprechende Relation eingepflegt, so wird dieser Entität eine eindeutige und noch nicht verwendete ID zugeordnet.

Somit kann sichergestellt werden, dass jede Entität ohne Doppeldeutigkeiten über den Primärschlüssel identifiziert wird.

1.5.2 Zusammengesetzte Primärschlüssel

Wie bereits in der Definition eines Primärschlüssels erwähnt, kann ein Primärschlüssel auch aus mehr als einem Attribut bestehen. Dies ist insbesondere dann erforderlich, wenn ein einzelnes Attribut nicht zur Beschreibung der gesamten Relation ausreicht.

Bibliothek

Ausleih-ID	Benutzer-ID	Datum	Titel
100	19	12.10.2024	Sofort zum SQL-Profi werden
100	19	12.10.2024	Python für Anfänger
201	6	14.10.2024	Wie funktioniert das Internet

Abb. 1.5: Bibliothek-Relation

In der in Abbildung 1.5 dargestellten »Bibliothek-Relation« werden Ausleihvorgänge gespeichert. Dabei besitzt jeder Vorgang die Attribute:

- »Ausleih-ID«, um den Vorgang eindeutig zu identifizieren
- »Benutzer-ID«, um den Benutzer, der Bücher ausleiht, zu kennzeichnen
- »Datum«, um die Zeit, wann der Vorgang stattgefunden hat, zu beschreiben
- »Titel«, um die geliehene Literatur zu definieren

Eine Zeile der Bibliothek-Relation beschreibt den Ausleihprozess von genau einem Buch. In der ersten Zeile leiht sich beispielsweise der Benutzer mit der ID 19 das Buch »Sofort zum SQL-Profi werden« aus. Dieser Ausleihvorgang wird dabei mit der Ausleih-ID 100 identifiziert. Unglücklicherweise reicht hier die alleinige Ausleih-ID nicht als Primärschlüssel aus. Während eines Ausleihvorgangs kann ein Benutzer natürlich mehrere Bücher gleichzeitig ausleihen, wodurch die Ausleih-ID nicht eindeutig auf eine Entität der Relation zeigt. Ist mit der ID 100 nun die erste oder die zweite Zeile gemeint?

Diese Problematik kann über verschiedene Lösungsansätze adressiert werden. Eine Möglichkeit besteht darin, eine weitere ID hinzuzufügen, die jedes

einzelne ausgeliehene Buch identifiziert. Dadurch können verschiedene IDs für ein und denselben Ausleihvorgang vergeben werden und es treten keine Duplikate auf.

Alternativ kann die Relation in ihrer aktuellen Form beibehalten werden. Statt den Primärschlüssel jedoch nur mit einem Attribut zu bilden, werden mehrere Attribute als Primärschlüssel gewählt. Resultierend ist der neue Primärschlüssel ein zusammengesetzter und besteht aus der Ausleih-ID und dem Titel. Mit einem Paar (Ausleih-ID, Titel) kann nun eindeutig auf eine Entität geschlossen werden. Während die Ausleih-ID 100 und der Titel »Sofort zum SQL-Profi werden« auf die erste Zeile verweist, beschreibt das Paar 100 und »Python für Anfänger« die zweite Zeile.



Der zusammengesetzte Primärschlüssel (Ausleih-ID, Titel) funktioniert nur unter der Annahme, dass ein Benutzer kein Buch doppelt in einem Ausleihvorgang ausleihen kann. Ansonsten könnte das Paar (100, »Sofort zum SQL-Profi werden«) nicht eindeutig auf eine Entität verweisen. Um diesen Fall zu lösen, müsste eine weitere ID hinzugefügt werden.

1.5.3 Fremdschlüssel

Neben der Idee des Primärschlüssels existiert das Konzept des Fremdschlüssels. Ein Fremdschlüssel ist ein Attribut einer Relation, welches gleichzeitig als Primärschlüssel einer anderen Relation fungiert. In den in der Abbildung 1.4 dargestellten Relationen ist die Kunden-ID in der Bestellungen-Relation folglich ein Fremdschlüssel, während die Bestell-ID den Primärschlüssel der Relation repräsentiert. Auf diese Weise werden mithilfe von Fremdschlüsseln Beziehungen zwischen verschiedenen Relationen etabliert.



Eine Relation kann beliebig viele Fremdschlüssel beinhalten, ohne Limits. Das bedeutet, dass eine Relation sowohl keinen als auch fünf oder sogar zehn Fremdschlüssel führen kann.

Stichwortverzeichnis

A

Abfrage	46
beschleunigen	254
verschachteln	125
Abfragesprache	18
Abstürze	258, 260
ADD COLUMN	196, 199
ADD CONSTRAINT	201
AFTER	235, 236, 243
AGE	137
Aggregatfunktionen	96, 103, 104, 130
ohne GROUP BY	104
ORDER BY	103
Aktion	234, 235, 236
ALL	145
ALL PRIVILEGES	226, 228
ALTER TABLE	196, 197, 199
AND	67, 68, 69, 114
AND-Operator	68, 69
Anmeldung	221
AS	52, 53, 97, 238, 250
ASC	55
ascending	55
ASCII-Tabelle	65, 66
Attribut	15, 19, 162, 168
mehrere gleichzeitig ändern	207
Ausführungszeit	256
Aussagenlogik	67
Operatoren	68
AUTO_INCREMENT	190
Automatisierung	234
AVG	101, 146

B

Baum	256
B-Baum-Index	255
Bedingung	61, 67, 69, 181, 240
Bereichsangaben	73
Filter erstellen	181
Verknüpfung	69
verschachteln	72
zusammengesetzte	72
Befehle	
mehrere gleichzeitig ausführen	171
Reihenfolge	121
BEFORE	235, 243
BEGIN	238, 239, 259
Benutzer	
Eigentum	228
Erstellung	218
löschen	228
löschen in PostgreSQL	228
mehrere	258
Benutzerkonto	218, 221, 229
einrichten in MySQL	223
einrichten in PostgreSQL	222
wechseln	226
Benutzermanagement	218, 251, 258
Benutzerrollen	229
aktivieren in MySQL	230
Benutzer verknüpfen	230
entziehen	230
in MySQL	230
in PostgreSQL	229
löschen	230
Rechte zuweisen	230
Berechnung	95, 96
Bereichsangaben	73

Beschleunigen 253, 256
BETWEEN 73, 75, 76
 NOT-Operator 76
BIGINT 164, 188
BIGSERIAL 188, 189
BOOLEAN 166
Buchstaben 65

C

CASCADE 210, 211, 246
CAST 83, 88
CEIL 130
CHAR 165
Charakterklasse 84, 87
 Konstruktion 84
CHECK 181, 201, 202
Closed Source 24
COMMIT 259
CONCAT 134, 135
COUNT 96
CREATE DATABASE 158
CREATE INDEX 256
CREATE OR REPLACE FUNCTION 237,
 246
CREATE OR REPLACE VIEW ... 252, 253
CREATE ROLE 229
CREATE TABLE 162
 Datentypen 166
CREATE TRIGGER 235
 MySQL 239
 PostgreSQL 239
CREATE TRIGGER-Befehl 243
CREATE USER 218, 219
CREATE VIEW 250
CURRENT_DATE 136
CURRENT_TIME 136

D

Dachsymbol 86
DATE 165

Daten

 ändern 205
 löschen 208
 unvollständige 169
Datenbank 13, 14
 anpassen 195
 anzeigen 158
 anzeigen mit PostgreSQL 158
 auswählen mit MySQL 160
 Einschränkungen 175, 176
 erstellen 158
 Erstellung 157
Datenbankabfrage 45
Datenbank-Management-System 14, 15
Datenbanksystem 14, 254
 Index 255
Datenbasis 14
Datenbestand 42
Datenmanipulation 125
Datentyp 84, 162, 163, 200
 ändern in PostgreSQL 200
 Anpassen 196, 200
 Datumsangaben 165
 nachträglich ändern 199
 numerische 163
 PostgreSQL 84
 Wahrheitswerte 165
 Zeichenketten 165
 Zeiten 165
Datum 136, 148
Datumsangaben 75, 84
DBMS 14, 15, 23
DECIMAL 164
DEFAULT 178, 202
DELETE FROM 208
Delimiter 238
DELIMITER 239, 240
DESC 55
descending 55

Differenz	150, 152	Ergebnisrelation	127
MySQL	151	Ergebnisse	
symmetrische	151	Vereinigung	143
DISTINCT	48, 99	verknüpfen	143
Document-Stores	17	Event	235, 236
Dollarzeichen	86	Reaktion	237
Doppelte Daten entfernen	48	EXCEPT	150
DOUBLE	164	EXECUTE PROCEDURE	239
DOUBLE PRECISION	164	EXPLAIN ANALYZE	256
DROP COLUMN	197, 199		
DROP CONSTRAINT	202	F	
DROP INDEX	257	False	43
DROP OWNED BY	229, 230	FALSE	205
DROP PRIMARY KEY	204	Fehlerprävention	176
DROP TABLE	177, 199	Fehlertoleranz	176
DROP TRIGGER	246	Filter	61
DROP USER	228	Filterbedingungen	67
DROP VIEW	253	Flexibilität	249
Duplikate	48, 50, 145	Fließkommazahl	164
entfernen	48, 150	FLOAT	164
filtern	180	FLOOR	130
finden	149	FOR	240
Durchschnitt	101, 145, 146	FOR EACH ROW	235, 236
E		FOR EACH STATEMENT	235
Echtzeitanwendungen	254	FOREIGN KEY	185, 201, 202
Einschränkungen	181	Fremdschlüssel	21, 185, 209, 211
hinzufügen	201	festlegen	185
löschen	202	Fremdschlüsselbeziehung	209
löschen in MySQL	204	anpassen	210
löschen in PostgreSQL	204	Modi	211
mehrere	181	FROM	46, 112
Namen	201	FULL OUTER JOIN	118
Elemente zählen	96		
END	238, 239	G	
Entität	15, 19	Ganzzahl	164, 188
Ergebnismenge		Gleichheit	62
Anzahl der Zeilen festlegen	57	GRANT	219, 220, 252
beschränken nach Bedingung	61	GRANT ALL	226
Größe festlegen	57	MySQL	226
		GRANT ALL PRIVILEGES	226

GRANT DELETE	226
GRANT INSERT	226
GRANT UPDATE	226
Graphdatenbanken	18
Größer als	64
Größer gleich	66
Groß- und Kleinschreibung	132
GROUP BY	94, 101
Gruppen	93
entfernen	105
erzeugen	94
Gruppierung	93

H

HAVING	106, 107
Skalarfunktionen	138
Hierarchie	220
Horizontal	144

I

ID	187
automatisch hochzählen	187
IDENTIFIED BY	218
IF-THEN-ELSE	240
ILIKE	81
Umlaute	82
IN	76
Index	254, 256
ändern	257
erstellen	256
löschen	257
Risiken	257
Inkonsistenz	259
INNER JOIN	116, 117
INSERT INTO	168, 169, 171, 205
INT	166
INTEGER	163, 188
Integrität	163, 182, 201
INTERSECT	149

J

JOIN	112, 116, 144
Arten	117
mehrere gleichzeitig	120
MySQL	119
Punktschreibweise	115
Joins	116

K

Key-Value-Stores	18
Klammern	72
Kleiner als	64
Kleiner gleich	66
Konsistenz	182, 201

L

LANGUAGE	238
LEFT	133
LEFT OUTER JOIN	118
LENGTH	135, 138
Lesebefehl	226
Lesezugriff	220
MySQL	221
PostgreSQL	220
LIKE	78, 81
reguläre Ausdrücke	83
LIMIT	57, 58
MySQL	58
Linux	26
LOG	131
Logarithmen	131
LOWER	132

M

macOS	26, 28
Manipulation	129
MariaDB	25
MAX	102
Maximum	102
MAX_QUERIES_PER_HOUR	219

Mehrbenutzerbetrieb	258	ON ALL TABLES	220
Mehrere Spalten	55	ON DELETE	210
Microsoft SQL Server	24	ON UPDATE	211
MIN	102	Open Source	24
Minimum	102	Operator	
MODIFY COLUMN	200, 202	AND	68
Muster	78, 83	NOT	68
MySQL . 23, 24, 29, 39, 40, 41, 58, 85, 158,		OR	68
189, 190		Verknüpfen	72
Datenbank erstellen	39	XOR	68
Einrichtung	37	OR	67, 68, 70
Installation	28	Oracle Database	24
LIMIT	58	ORDER BY	54, 103
Syntax	51	OR-Operator	68, 69
XOR	73	OUTER JOIN	117
XOR-Operator	72	OWNER TO	227
MySQL Workbench	32, 37, 158		
Anzeige aktualisieren	160	P	
Datenbanken anzeigen	158	Passwort	
N		festlegen in MySQL	218
Namen		festlegen in PostgreSQL	218
MySQL	204	Performance	163
PostgreSQL	204	pgAdmin	35, 44, 158
NEW	240, 242	Anzeige aktualisieren	160
NO ACTION	211	Datenbanken anzeigen	158
NoSQL	17	Datenbank erstellen	34
NOT	67, 68, 71, 76	pgAdmin4	33
NOT NULL	176, 180, 202	Phishing	218
NOT-Operator	68, 71, 77	PK 202	
Klammern	72	Platzhalter	84
NULL	117, 169, 176, 177, 178	plpgsql	238
Nullwerte	176, 178	PostgreSQL 23, 25, 26, 28, 35, 162, 189	
NUMERIC	164	Datumstypen	83
Numerische Daten	130	Einrichtung	33
O		Entwicklungsumgebung	33
OFFSET	57	Installation	26
OLD	240, 242	LIKE	83
ON	112, 113	Syntax	51
		Potenzen	131

POWER	131	REFERENCES	186, 210
Primärschlüssel ..	19, 21, 42, 183, 186, 187, 202	Regeln	176
mehrere Attribute	183	REGEXP	85
MySQL	190	Reguläre Ausdrücke	82, 86, 87
PostgreSQL	190	PostgreSQL	85
zusammengesetzter	20, 184	Sonderzeichen	87
PRIMARY KEY	183, 201, 202	Relation	15, 115
Pseudodatentyp	187	Namen anzeigen	203
public	220	Reihenfolge der Erstellung	186
Punktschreibweise	51, 53, 115	vertikal verbinden	144
Q		Relationales Datenbanksystem .	15, 16, 19, 23
Quadratwurzel	131	Relationen	
Quantifier	84, 87	anzeigen in PostgreSQL	168
Query Tool	34, 35, 36, 160, 221	anzeigen mit MySQL	167
R		verknüpfen	113
Reaktion	234, 235, 236, 237, 240	RENAME COLUMN	198
Beispiel	241	RENAME TO	198
einbinden in MySQL	243	REPLACE	134
einbinden in PostgreSQL	242	RETURN	238, 243
Formulierung	240	RETURNS TRIGGER	238
MySQL	236, 237, 239	REVOKE	228
PostgreSQL	237	RIGHT	133
REAL	164	RIGHT OUTER JOIN	117
Rechnen	131	RLIKE	85, 88
Rechte	226	ROLLBACK	260
entziehen in MySQL	228	Rolle	229
entziehen in PostgreSQL	228	Rollenkonzept	229
Entzug	227	ROUND	102, 130
MySQL	226	Rückgabewert	238
Relationen anderer Benutzer verän- dern	227	Runden	102, 130
Relationen erstellen	227	S	
testen	225	Schema	162, 220
Rechteverteilung	218, 219	Schema-Erstellung	162
MySQL	220	Schleifen	240
PostgreSQL	220	Schnittmenge	148
Rechteverwaltung	217	MySQL	149
		Schreibbefehl	226
		SELECT	46

Semikolon	196	SQL-Syntax	50
SERIAL	187	SQRT	131
Duplikate	189	Stack Builder	26
MySQL	189	START TRANSACTION	259
NULL	189	Stern	47, 221
PostgreSQL	189	Sternsymbol	99
SET	202, 206, 207	Structured Query Language	18
SET NULL	211	Subqueries	125
SET ROLE	230	SUBSTRING	134
SHOW DATABASES	158	Suchen	255
SHOW TABLES	167	Muster	78
Sicherheit	217	Zeichenketten	79
Skalarfunktion	129, 136, 138	SUM	100
PostgreSQL	134	Summe	100
SMALLINT	164	Superuser	227
SMALLSERIAL	189	Symmetrische Differenz	
SMALLSERIAL für SMALLINT	188	MySQL	152
Sortieren	54	Syntax	50
absteigend	55		
auf Spalten verweisen	56	T	
aufsteigend	55	Tabelle	15
Ergebnismenge	53	anpassen	195
mehrere Spalten	55	Ausgabe umbenennen	52
Sortierung		Einschränkungen hinzufügen	201
alphabetisch	65	erstellen	157
Spalte		löschen	199
Ausgabe umbenennen	52	mehrere abfragen	111
Datentyp nachträglich ändern	199	Namen ändern	198
hinzufügen	197	verknüpfen	111
löschen	197	Verknüpfung	113
mehrere angeben	55	TEXT	165
mehrere gleichzeitig ändern	207	TIME	166
nachträglich einfügen	196	TIMESTAMP	166
sortieren	54	TIMESTAMPDIFF	137
umbenennen	196, 198	Timing	235, 236
verweisen auf	56	TO	198
Speichereffizienz	163	Transaktion	259
SQL	18, 23	ROLLBACK	261
Standard	23	rückgängig machen	261
SQL-Dialekt	23	Transaktionsmanagement	259

MySQL	260	UPPER	132
PostgreSQL	260	USE	160
Trennzeichen	240	V	
Trigger	233, 234, 236	VALID UNTIL	219
ändern	245	VALUES	168, 171
erstellen	234	VARCHAR	165, 167
löschen	245	Verbindungsvorschrift	114
löschen in MySQL	246	Vereinigung	145
löschen in PostgreSQL	246	Verknüpfen	72
mehrere gleichzeitig in Postgre- SQL	236	Verknüpfung	
testen	244	Operatoren	114
Trigger-Funktionen	237, 242	Tabellen	113
ändern	246	Verschachteln	125
erstellen in PostgreSQL	243	Vertikal	144
löschen	246	Views	250, 251
PostgreSQL	237	ändern	252
Syntax in PostgreSQL	237	ändern in PostgreSQL	253
Trennzeichen in MySQL	240	Anwendungsfälle	251
True	43	löschen	252, 253
TRUE	205	Rechte in MySQL	252
Tupel	15, 145	Rechte in PostgreSQL	252
U		Sicherheit	251
Uhrzeit	136	W	
Umbenennung	52, 198	Wartung	233
Ungleichheit	63	Werte	
UNION	119, 144	ändern	205
Einschränkungen	147	einfügen	168
PostgreSQL	148	löschen	208, 209
Spaltenanzahl	148	WHERE	61, 107, 206, 208
Spaltenname	146	BETWEEN	74
UNION ALL	145	Gleichheit	62
UNIQUE	180, 189, 201, 202	größer als	64
MySQL	189	größer gleich	66
PostgreSQL	189	IN	77
Unterabfragen	125, 126	kleiner als	64
versus JOINS	129	kleiner gleich	66
UPDATE	205, 206	reguläre Ausdrücke	88
		Ungleichheit	63

WHERE-Befehl ... 63, 71, 72, 75, 76, 86, 87
 WHILE 240
 Windows 26, 28
 WITH PASSWORD 218
 Workbench 32, 38, 40, 44

X

XOR 67, 68
 MySQL 73
 PostgreSQL 72
 XOR-Operator 68, 72

Z

Zeichenketten 62, 78, 81, 132, 165, 167
 Anfang 133
 austauschen 135
 Ende 133
 größer als 64
 Informationen 135

kleiner als 64
 Länge 135
 MySQL 81
 PostgreSQL 62, 81
 suchen 78
 Teil abtrennen 134
 verändern 133
 zusammenfügen 134

Zeilen

mehrere gleichzeitig aktuali-
 sieren 206
 modifizieren 205
 teilweise löschen 208
 zählen 99

Zeitspanne

MySQL 137
 PostgreSQL 137

Zugriffskontrolle 217