

Kennzeichnet neue Begriffe, URLs, E-Mail-Adressen, Dateinamen und Dateierweiterungen.

Proportionalschrift

Wird für Programmlistings benutzt und kennzeichnet innerhalb des Texts Programmelemente wie Variablen- oder Funktionsnamen, Datenbanken, Datentypen, Umgebungsvariablen, Ausdrücke und Schlüsselwörter.

Proportionalschrift **fett**

Zeigt Befehle oder andere Texte, die unverändert von der Benutzerin oder dem Benutzer eingetippt werden sollen, oder Werte, die durch den Kontext bestimmt sind.



Dieses Element kennzeichnet einen Tipp oder Vorschlag.



Dieses Element kennzeichnet einen allgemeinen Hinweis.



Dieses Element kennzeichnet eine Warnung.

Die Codebeispiele verwenden

Dieses Buch soll Ihnen bei Ihrer Arbeit helfen. Falls Beispielcode mit diesem Buch angeboten wird, können Sie diesen im Allgemeinen in Ihren Programmen und

Dokumentationen einsetzen. Sie müssen uns nicht um Erlaubnis ersuchen, es sei denn, Sie reproduzieren einen bedeutsamen Teil des Codes. Zum Beispiel brauchen Sie keine Erlaubnis, wenn Sie ein Programm schreiben, das einige Codeausschnitte aus diesem Buch verwendet. Wenn Sie jedoch Beispiele aus O'Reilly-Büchern verkaufen oder vertreiben wollen, brauchen Sie eine Erlaubnis. Beantworten Sie eine Frage, indem Sie dieses Buch und Beispielcode daraus zitieren, brauchen Sie keine Erlaubnis. Nehmen Sie dagegen eine bedeutende Menge an Beispielcode aus diesem Buch in die Dokumentation Ihres Produkts auf, ist eine Erlaubnis erforderlich.

Wir wissen eine Quellenangabe zu schätzen, fordern sie aber im Allgemeinen nicht. Eine Quellenangabe enthält üblicherweise Titel, Autor, Verlag und ISBN, zum Beispiel: »SQL Pocket Guide by Alice Zhao (O'Reilly). Copyright 2021 Alice Zhao, 978-1-492-209040-3.«

Falls Sie das Gefühl haben, dass Ihr Einsatz der Codebeispiele nicht durch die oben genannten Erlaubnisse abgedeckt wird, schreiben Sie uns bitte an permissions@oreilly.com.

Danksagungen

Ich danke Jonathan Gennick, dem Erstautor dieses Buchs. Außerdem danke ich Andy Kwan für sein Vertrauen in mich, Jonathans Arbeit fortzuführen.

Ich hätte dieses Buch ohne die Hilfe meiner Lektorinnen und Lektoren Amelia Blevins, Jeff Bleiel und Caitlin Ghegan sowie meiner Fachkorrektorinnen und -korrektoren Alicia Nevels, Joan Wang, Scott Haines und Thomas Nield niemals fertigstellen können. Ich weiß die Zeit wirklich zu schätzen, die ihr aufgewandt habt, um jede Seite dieses Buchs gründlich zu lesen. Euer Feedback war unglaublich wertvoll.

Meinen Eltern danke ich, dass sie in mir die Liebe für das Lernen und die Kreativität genährt haben. Dank geht auch an meine Kinder Henry und Lily – eure Begeisterung für dieses Buch ist herzerwärmend. Und schließlich danke ich meinem Ehemann Ali – danke für all deine Hinweise zu diesem Buch, für deine Ermutigung und dafür, dass du mein größter Fan bist.

SQL-Crashkurs

Dieses kurze Kapitel bietet Ihnen einen schnellen Einstieg in die grundlegende SQL-Terminologie und die SQL-Konzepte.

Was ist eine Datenbank?

Beginnen wir ganz von vorn. Eine *Datenbank* ist ein Ort, an dem man auf organisierte Art und Weise Daten ablegen kann. Es gibt viele Möglichkeiten, Daten zu organisieren, und entsprechend gibt es viele Datenbanken, unter denen man wählen kann. Die zwei Kategorien, in die Datenbanken fallen, nennen sich *SQL* und *NoSQL*.

SQL

SQL ist die Abkürzung für *Structured Query Language* (strukturierte Abfragesprache). Stellen Sie sich vor, Sie hätten eine App, die sich die Geburtstage all Ihrer Freunde merkt. Mit sehr hoher Wahrscheinlichkeit würden Sie SQL verwenden, um mit dieser App zu reden.

Deutsch: »Hallo App. Wann ist der Geburtstag meines Mannes?«

```
SQL: SELECT * FROM birthdays  
WHERE person = 'husband';
```

SQL-Datenbanken werden oft als *relationale Datenbanken* bezeichnet, weil sie aus Relationen bestehen, die üblicherweise Tabellen genannt werden. Viele Tabellen, die miteinander verbunden sind, bilden eine Datenbank. [Abbildung 1-1](#) zeigt das Bild einer Relation in einer SQL-Datenbank.

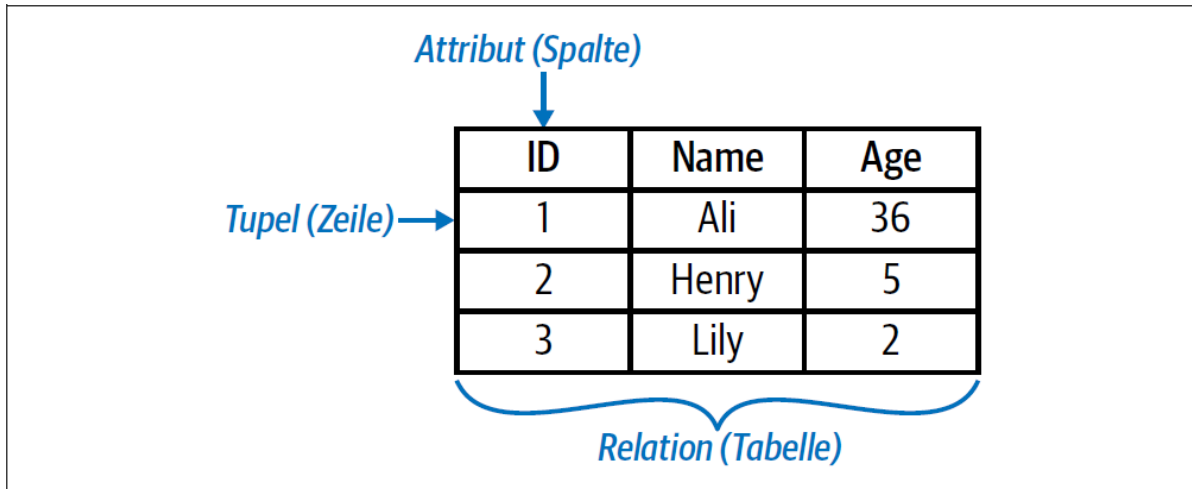


Abbildung 1-1: Eine Relation (auch bekannt als Tabelle) in einer SQL-Datenbank

Das Wichtigste, das man über SQL-Datenbanken wissen sollte, ist, dass sie vordefinierte *Schemata* erfordern. Stellen Sie sich ein Schema als die Art vor, in der die Daten in einer Datenbank organisiert oder strukturiert sind. Nehmen wir einmal an, Sie wollten eine Tabelle anlegen. Bevor Sie Daten in die Tabelle laden, muss die Struktur der Tabelle festgelegt werden, darunter solche Dinge wie die Spalten der Tabelle, ob diese Spalten Integer- oder Dezimalwerte enthalten usw.

Irgendwann jedoch kommt der Zeitpunkt, an dem sich die Daten nicht mehr auf diese strukturierte Weise organisieren lassen. Ihre Daten haben vielleicht unterschiedliche Felder. Oder möglicherweise brauchen Sie eine effektivere Methode, um eine große Menge an Daten zu speichern und abzurufen. Hier kommt NoSQL ins Spiel.

NoSQL

NoSQL steht für *Not only SQL* (nicht nur SQL). Ich werde in diesem Buch nicht näher darauf eingehen, wollte es aber erwähnen, da der Begriff seit den 2010er-Jahren an Popularität gewonnen hat und es wichtig ist, zu verstehen, dass es auch noch andere Wege gibt, um Daten zu speichern, als nur Tabellen.

NoSQL-Datenbanken werden oft als *nicht relationale Datenbanken* bezeichnet, und es gibt sie in allen Formen und Größen. Ihre wichtigsten Merkmale bestehen darin, dass sie dynamische Schemata haben (das heißt, ihre Schemata müssen nicht im Vorhinein starr festgelegt werden) und dass sie eine horizontale Skalierung erlauben (was bedeutet, dass die Daten sich über mehrere Rechner ausbreiten können).

Die beliebteste NoSQL-Datenbank ist *MongoDB*, die zu den dokumentenorientierten Datenbanken gehört. [Abbildung 1-2](#) zeigt, wie Daten in MongoDB gespeichert werden. Sie werden bemerken, dass die Daten nicht mehr in einer strukturierten Tabelle

vorliegen und die Anzahl der Felder (vergleichbar einer Spalte) für die einzelnen Dokumente (vergleichbar einer Zeile) variiert.

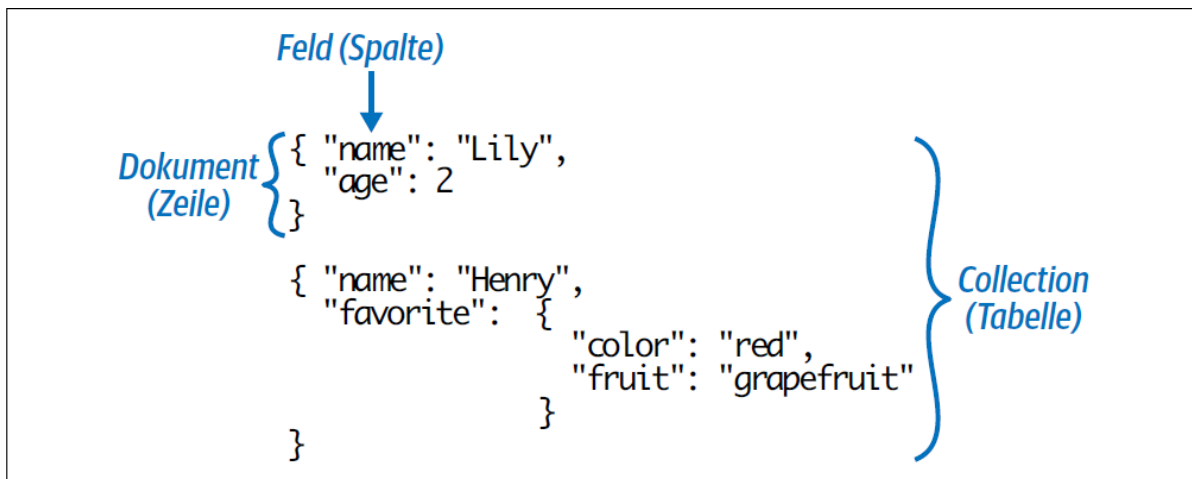


Abbildung 1-2: Eine Collection (eine Variante einer Tabelle) in MongoDB, einer NoSQL-Datenbank

Wie gesagt, der Schwerpunkt dieses Buchs liegt auf SQL-Datenbanken. Selbst mit der Einführung von NoSQL speichern die meisten Unternehmen die Mehrzahl ihrer Daten in Tabellen relationaler Datenbanken.

Datenbankmanagementsysteme (DBMS)

Sie haben vermutlich schon Begriffe wie *PostgreSQL* oder *SQLite* gehört und fragen sich vielleicht, worin diese sich von SQL unterscheiden. Es gibt zwei Typen von *Datenbankmanagementsystemen* (DBMS), also von Software, die benutzt wird, um mit einer Datenbank zu arbeiten.

Dazu gehören Dinge wie das Festlegen, wie Daten importiert und organisiert werden, aber auch Dinge wie die Handhabung des Zugriffs von Benutzern oder Programmen auf die Daten. Ein *relationales Datenbankmanagementsystem* (RDBMS) ist eine Software, die speziell für relationale Datenbanken oder Datenbanken, die aus Tabellen bestehen, gedacht ist.

Jedes RDBMS hat eine andere Implementierung von SQL, das heißt, die Syntax variiert leicht von Software zu Software. So würden Sie zum Beispiel in fünf unterschiedlichen RDBMS zehn Zeilen mit Daten ausgeben:

MySQL, PostgreSQL und SQLite

```
SELECT * FROM birthdays LIMIT 10;
```

Microsoft SQL Server

```
SELECT TOP 10 * FROM birthdays;
```