

Kapitel 12, Transaktionen

führt Transaktionen ein und zeigt, wie man sie nutzt.

Kapitel 13, Indizes und Constraints

erklärt Indizes und Constraints.

Kapitel 14, Views

zeigt, wie man eine Schnittstelle aufbaut, die Datenkomplexität vor Benutzern verbirgt.

Kapitel 15, Metadaten

zeigt den Nutzen des Data Dictionary.

Kapitel 16, Analytische Funktionen

behandelt Funktionalität zum Erzeugen von Rankings, Untersummen und anderen Werten, die im Reporting und in der Analyse oft zum Einsatz kommen.

Kapitel 17, Mit großen Datenbanken arbeiten

zeigt Techniken zum einfacheren Managen und Arbeiten mit sehr großen Datenbanken.

Kapitel 18, SQL und Big Data

untersucht die Anpassungen der SQL-Sprache, mit denen Daten aus nicht-relationalen Datenplattformen abgefragt werden können.

Anhang A, ER-Diagramm der Musterdatenbank

zeigt das Datenbankschema, das für alle Beispiele dieses Buchs verwendet wird.

Anhang B, Lösungen zu den Übungen

enthält die Lösungen der Übungsaufgaben.

Verwendete Konventionen

In diesem Buch gelten folgende typografische Konventionen:

Kursiv

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

Nichtproportionalschrift

Wird für Codebeispiele genutzt, aber auch im Text, um Programmelemente wie Variablen- oder Funktionsnamen, Datenbanken, Datentypen, Umgebungsvariablen, Anweisungen und Schlüsselwörter zu kennzeichnen.

Nichtproportionalschrift kursiv

Wird für Text verwendet, den der Benutzer individuell eingeben muss oder der sich aus dem Kontext ergibt.

Nichtproportionalschrift fett

Befehle oder anderer Text, der vom Benutzer so eingegeben werden soll, oder Stellen im Code, die besonders hervorgehoben werden sollen.



Ein Tipp, Vorschlag oder allgemeiner Hinweis. Ich weise in solchen Einschüben beispielsweise auf neue Features von Datenbanken hin.



Eine Warnung. So werde ich zum Beispiel darauf aufmerksam machen, dass eine bestimmte SQL-Klausel unerwünschte Folgen haben kann, wenn man sie nicht vorsichtig genug einsetzt.

Verwendung der Codebeispiele

Um mit den Daten zu experimentieren, die für die Beispiele in diesem Buch genutzt wurden, haben Sie zwei Möglichkeiten:

- Laden Sie die MySQL-Server-Version 8.0 (oder neuer) herunter und installieren Sie sie. Holen Sie sich dann die Sakila-Beispieldatenbank, die Sie unter <https://dev.mysql.com/doc/index-other.html> finden.
- Rufen Sie <https://www.katacoda.com/mysql-db-sandbox/scenarios/mysql-sandbox> auf, um auf die MySQL-Sandbox zuzugreifen, in der die Sakila-Beispieldatenbank in einer MySQL-Instanz geladen ist. Sie müssen dafür einen (kostenlosen) Katacoda-Account einrichten. Dann klicken Sie auf den Button *Start Scenario*.

Haben Sie die zweite Möglichkeit gewählt, wird nach dem Starten des Szenarios ein MySQL-Server installiert und gestartet, und anschließend wird das Sakila-Schema mit seinen Daten geladen. Ist alles erledigt, erscheint ein Standardprompt `mysql>`, und Sie können damit beginnen, die Beispieldatenbank abzufragen. Das ist die einfachste Option, und ich vermute, die meisten Leser werden sie wählen – wenn das für Sie gut genug klingt, können Sie einfach zum nächsten Abschnitt springen.

Möchten Sie lieber Ihre eigene Kopie der Daten nutzen und Änderungen daran dauerhaft machen, oder sind Sie einfach daran interessiert, den MySQL-Server auf Ihrem eigenen Rechner zu installieren, bevorzugen Sie vielleicht die erste Option. Sie können sich auch dazu entscheiden, einen MySQL-Server zu nutzen, der in der Cloud gehostet wird, wie zum Beispiel bei Amazon Web Services oder Google Cloud. Sie müssen dann auf jeden Fall die Installation und die Konfiguration selbst vornehmen, was aber nicht Bestandteil dieses Buchs ist. Ist Ihre Datenbank bereit, werden Sie noch ein paar weitere Schritte ausführen müssen, um die Sakila-Beispieldatenbank zu laden.

Als Erstes müssen Sie das `mysql`-Kommandozeilentool starten und ein Passwort angeben, danach führen Sie die folgenden Schritte aus:

1. Öffnen Sie <https://dev.mysql.com/doc/index-other.html> und laden Sie die Dateien für *sakila database* aus dem Abschnitt *Example Databases* herunter.
2. Legen Sie sie in einem lokalen Verzeichnis wie `C:\temp\sakila-db` ab (das nutzen wir für die nächsten beiden Schritte, aber Sie können natürlich gern einen anderen Pfad wählen).
3. Geben Sie `source c:\temp\sakila-db\sakila-schema.sql`; ein und drücken Sie die Taste `Enter`.
4. Geben Sie `source c:\temp\sakila-db\sakila-data.sql`; ein und drücken Sie die Taste `Enter`.

Jetzt sollten Sie eine laufende Datenbank haben, die mit all den Daten gefüllt ist, die Sie für die Beispiele in diesem Buch benötigen.

Danksagungen

Ich möchte meinem Lektor Jeff Bleiel dafür danken, dass er mir dabei geholfen hat, diese dritte Auflage Wirklichkeit werden zu lassen. Vielen Dank auch an Thomas Nield, Ann White-Watkins und Charles Givre, die so nett waren, das Buch für mich durchzusehen. Außerdem möchte ich mich bei Deb Baker, Jess Haberman und all den anderen Leuten bei O'Reilly Media bedanken, die an diesem Buch beteiligt waren. Schließlich möchte ich meiner Frau Nancy und meinen Töchtern Michelle und Nicole für ihre Ermutigung und Inspiration danken.

Der Hintergrund

Ehe wir nun die Ärmel aufkrempleln und uns in die Arbeit stürzen, wäre es sicherlich hilfreich, die Geschichte der Datenbanktechnologie zu umreißen, um besser zu verstehen, wie sich relationale Datenbanken und die SQL-Sprache entwickelt haben. Daher möchte ich zunächst einige Grundkonzepte vorstellen und die Geschichte der computergestützten Datenspeicherung und -abfrage anschauen.



Wer möglichst schnell damit beginnen möchte, Queries zu schreiben, kann gerne direkt zu Kapitel 3 springen, aber ich empfehle, später noch einmal die ersten beiden Kapitel anzuschauen, um die Geschichte und den Einsatz von SQL besser zu verstehen.

Einführung in Datenbanken

Eine *Datenbank* ist im Grunde nichts weiter als eine Menge von zusammenhängenden Informationen. So ist zum Beispiel ein Telefonbuch eine Datenbank mit den Namen, Telefonnummern und Adressen aller Bewohner einer bestimmten Gegend. Doch so ein Telefonbuch ist zwar überall zu finden und wird viel genutzt, aber es hat auch einige Mängel:

- Wegen der Vielzahl der Einträge kann es zeitraubend sein, die Telefonnummer eines bestimmten Teilnehmers darin zu finden.
- Der einzige Index eines Telefonbuchs ist die alphabetische Ordnung nach Nachname und Vorname. Eine Möglichkeit, die Namen der Menschen mit einer bestimmten Adresse herauszufinden, gibt es nur in der Theorie.
- Von dem Augenblick an, da das Telefonbuch in Druck geht, fängt es bereits an zu veralten: Menschen ziehen fort, wechseln ihre Telefonnummer oder gehen an eine andere Adresse in derselben Gegend.

Diese Nachteile hat jedes manuelle System der Datenspeicherung, also beispielsweise auch Patientendaten, die in einem Aktenschrank abgelegt sind. Da eine papiergebundene Datensammlung eine derart sperrige Angelegenheit ist, gehörten *Datenbanksysteme* zu den ersten Computeranwendungen überhaupt. In ihnen werden Daten computergestützt gespeichert und abgefragt. Da ein Datenbanksystem diese Arbeiten nicht auf Papier, sondern elektronisch leistet, kann es die Daten