
Einleitung

Programmiersprachen kommen und gehen. Nur wenige Sprachen, die heute im Gebrauch sind, haben Wurzeln, die mehr als zehn Jahre zurückreichen. Einige Beispiele sind COBOL, eine Sprache, die immer noch viel in Mainframe-Umgebungen genutzt wird, Java, das Mitte der 1990er-Jahre entstand und zu einer der beliebtesten Programmiersprachen geworden ist, und C, eine Sprache, die nach wie vor für die Entwicklung von Betriebssystemen, Servern und Embedded-Systemen eingesetzt wird. Das im Datenbankbereich gebräuchliche SQL geht sogar bis in die 1970er-Jahre zurück.

SQL wurde ursprünglich dazu geschaffen, Daten aus den seit mehr als 40 Jahren existierenden relationalen Datenbanken anzulegen, zu bearbeiten und abzufragen. In den letzten zehn Jahren haben aber auch andere Plattformen wie Hadoop, Spark oder NoSQL an Fahrt aufgenommen und sich aus dem Markt der relationalen Datenbanken ein Stück vom Kuchen abgeschnitten. Wie wir in den letzten Kapiteln dieses Buchs noch besprechen werden, hat sich die SQL-Sprache trotzdem weiterentwickelt, um Daten von diversen Plattformen zu verarbeiten – unabhängig davon, ob diese in Tabellen, Dokumenten oder einfachen Dateien abgelegt sind.

Wozu SQL lernen?

Egal ob Sie eine relationale Datenbank nutzen werden oder auch nicht – wenn Sie in der Data Science oder einem anderen Bereich der Datenanalyse arbeiten, werden Sie neben weiteren Sprachen und Plattformen wie Python und R sehr wahrscheinlich SQL können müssen. Daten kommen in großen Mengen und sehr schnell von überall her, und Leute, die bedeutsame Informationen aus all diesen Daten extrahieren können, werden immer gesucht.

Warum SQL mit diesem Buch lernen?

Es gibt viele Bücher, die Sie wie Idioten, Dummchen oder etwas Ähnliches behandeln, aber meist kratzen diese nur an der Oberfläche. Am anderen Ende des Spektrums finden Sie Referenzen, die jede Variante jeder Anweisung einer Sprache bis ins Detail durchgehen. Das kann nützlich sein, wenn Sie schon grob wissen, was

Sie tun wollen, und nun nur noch die Syntax dazu benötigen. Dieses Buch möchte einen guten Mittelweg finden – ein paar Hintergründe vorstellen, die Grundlagen der SQL-Sprache behandeln und dann einige der fortgeschritteneren Features behandeln, mit denen Sie glänzen können. Zudem finden Sie am Ende dieses Buchs ein Kapitel zum Abfragen von Daten aus nicht-relationalen Datenbanken, was in einführenden Büchern nur selten ein Thema ist.

Aufbau dieses Buchs

Dieses Buch ist in 18 Kapitel und 2 Anhänge unterteilt:

Kapitel 1, Der Hintergrund

geht auf die Geschichte der Computerdatenbanken ein und schildert den Aufstieg des relationalen Modells und der Sprache SQL.

Kapitel 2, Datenbanken erstellen und mit Daten füllen

zeigt, wie man eine MySQL-Datenbank anlegt, die Tabellen für die Beispiele dieses Buchs erstellt und Daten in diese Tabellen lädt.

Kapitel 3, Datenbankabfragen

führt die select-Anweisung ein und stellt die gebräuchlichsten Klauseln vor (select, from, where).

Kapitel 4, Filtern

beschreibt die verschiedenen Arten von Bedingungen, die in der where-Klausel einer select-, update- oder delete-Anweisung verwendet werden können.

Kapitel 5, Mehrere Tabellen abfragen

zeigt, wie man mehrere Tabellen mittels Tabellen-Joins benutzen kann.

Kapitel 6, Umgang mit Mengen

handelt von Datenmengen und der Frage, wie diese innerhalb von Abfragen interagieren.

Kapitel 7, Daten erzeugen, bearbeiten und konvertieren

stellt verschiedene eingebaute Funktionen vor, mit denen man Daten bearbeiten oder konvertieren kann.

Kapitel 8, Gruppieren und Aggregieren von Daten

zeigt, wie Daten zusammengefasst werden.

Kapitel 9, Unterabfragen

führt Unterabfragen ein (ich liebe Unterabfragen) und zeigt, wie und wo man sie einsetzen kann.

Kapitel 10, Weitere Joins

geht genauer auf die verschiedenen Join-Typen ein.

Kapitel 11, Bedingungslogik

erklärt, wie man Bedingungslogik (d.h. if-then-else) in select-, insert-, update- und delete-Anweisungen verwendet.

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.