

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, und C, eine Sprache, die nach wie vor für die Entwicklung von Betriebssystemen, Servern und Embedded-Systemen eingesetzt wird. Doch das im Datenbankbereich gebräuchliche SQL geht bis in die 1970er-Jahre zurück.

In SQL werden Daten aus relationalen Datenbanken angelegt, bearbeitet und abgefragt. Relationale Datenbanken sind vor allem populär, weil sie, wenn sie gut entworfen sind, gewaltige Datenmengen bewältigen können. Im Umgang mit großen Datenmengen hat SQL einiges mit den todschicken Digitalkameras und ihren Hochleistungszooms gemeinsam: Mit SQL kann man große Datenmengen durchforsten oder einzelne Zeilen heranzoomen (oder irgendetwas dazwischen). Andere Datenbankmanagementsysteme tendieren dazu, unter hoher Belastung zusammenzubrechen, da ihr Fokus zu eng ist (die Zoomlinse kann einen bestimmten Höchstwert nicht überschreiten). Aus diesem Grund sind alle Versuche, relationale Datenbanken und SQL vom Thron zu stoßen, gescheitert. Auch wenn SQL eine ältere Sprache ist, wird es uns noch lange erhalten bleiben und blickt in eine glänzende Zukunft.

Wozu SQL lernen?

Wenn Sie mit einer relationalen Datenbank arbeiten, müssen Sie wissen, wie Sie mit den Daten in Ihrer Datenbank umgehen können, egal ob Sie nun Anwendungen schreiben, administrative Aufgaben erfüllen oder Reports generieren. Selbst wenn Sie die SQL-Anweisungen mit einem Tool - etwa einem Reporting-Programm - generieren, müssen Sie gelegentlich die automatische Generierungsfunktion umgehen und eigene SQL-Anweisungen schreiben.

SQL hat zusätzlich den Vorteil, dass es Sie zwingt, die Datenstrukturen, in denen die Informationen Ihres Unternehmens gespeichert sind, zu betrachten und zu verstehen. In dem Maße, in dem Sie mit den Tabellen Ihrer Datenbank vertrauter werden, fallen Ihnen vielleicht auch selbst Änderungen oder Ergänzungen Ihres Datenbankschemas ein.

Warum SQL mit diesem Buch lernen?

SQL ist in mehrere Kategorien unterteilt. Anweisungen, mit denen man Datenbankobjekte anlegt (Tabellen, Indizes, Constraints usw.), bezeichnet man allgemein als SQL-Schemaanweisungen. Die Anweisungen, mit denen die in der Datenbank gespeicherten Daten erstellt, bearbeitet und abgefragt werden, heißen SQL-Datenanweisungen. Wenn Sie ein Administrator sind, verwenden Sie sowohl Schema- als auch Datenanweisungen in SQL. Sind Sie hingegen Programmierer oder Report-Verfasser, möchten (oder dürfen) Sie unter Umständen nur SQL-Datenanweisungen benutzen. Dieses Buch zeigt zwar auch viele SQL-Schemaanweisungen, aber sein Schwerpunkt liegt auf den Programmierfunktionen.

Mit ihren wenigen Befehlen sehen SQL-Datenanweisungen manchmal enttäuschend simpel aus. Dieser Eindruck wird durch einige SQL-Bücher noch dadurch verstärkt, dass sie nur gerade einmal die Oberfläche dessen ankratzen, was mit dieser Sprache möglich ist. Wenn Sie allerdings richtig mit SQL arbeiten wollen, müssen Sie die Fähigkeiten

dieser Sprache vollständig verstehen und wissen, wie man die verschiedenen Features kombinieren kann, um mächtige Ergebnisse zu erzielen. Ich denke, dieses Buch ist das einzige, das SQL wirklich detailliert behandelt, und zwar ohne den eingebauten Türstopper-Effekt (Sie kennen ja diese 1.250-Seiten-Wälzer, die sich „Vollständige Referenz“ nennen und die Tendenz haben, auf den Regalen zu verstauben).

Die Beispiele in diesem Buch laufen auf MySQL, Oracle Database und SQL Server; allerdings musste ich mich für eines dieser Systeme entscheiden, um meine Musterdatenbank zu hosten und die Ergebnismengen zu formatieren, die mir die Beispielabfragen zurücklieferten. Ich habe MySQL gewählt, weil es frei, leicht zu installieren und einfach zu administrieren ist. Lesern, die einen anderen Server benutzen, empfehle ich, MySQL dennoch herunterzuladen und die Musterdatenbank darauf zu installieren. Auf diese Weise können Sie die Ergebnisse ausführen und mit den Daten experimentieren.

Aufbau dieses Buchs

Dieses Buch ist in 1.5 Kapitel und 3 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 generieren, konvertieren und manipulieren, stellt verschiedene eingebaute Funktionen vor, mit denen man Daten manipulieren 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.

Anhang A, ER-Diagramm der Musterdatenbank, zeigt das Datenbankschema, das für alle Beispiele dieses Buchs verwendet wird.

Anhang B, MySQL-Erweiterungen für SQL, stellt einige interessante Nicht-ANSI-Features der SQL-Implementierung von MySQL vor.

Anhang C, Lösungen der Übungen, enthält die Lösungen der Übungsaufgaben.

Danksagungen

Ich möchte meiner Lektorin Mary Treseler dafür danken, dass sie mir geholfen hat, diese zweite Auflage Wirklichkeit werden zu lassen. Vielen Dank auch an Kevin Kline, Roy Owens, Richard Sonen und Matthew Russell, die so nett waren, das Buch für mich während der Weihnachtsferien durchzusehen. Außerdem möchte ich den vielen Lesern der ersten Auflage danken, die mir Fragen, Kommentare und Verbesserungsvorschläge haben zukommen lassen. Schließlich möchte ich meiner Frau Nancy und meinen Töchtern Michelle und Nicole für ihre Ermutigung und Inspiration danken.