

Wozu Datenbanksysteme da sind



In diesem Kapitel

- ▶ Unterschied von Daten und Informationen
 - ▶ Daten als wichtige Ressource
 - ▶ Wozu braucht man Datenbanksysteme?
 - ▶ Meine erste Datenbank
-

Sie erfahren in diesem ersten Kapitel, warum man Daten in Datenbanken speichert und wann aus Daten Informationen werden. Wenn Sie das Beispiel im zweiten Teil dieses Kapitels am Computer nachvollziehen, können Sie schon Ihre erste Datenbank aufbauen. In den folgenden Kapiteln lernen Sie dann schrittweise mehr über Datenbanksysteme.

Daten und ihre Speicherung

Daten, also gespeicherte Fakten über uns selbst, andere und unser Umfeld, begleiten uns unser ganzes Leben. Sie dienen – nach entsprechender Verarbeitung – als Grundlage von Entscheidungen und beeinflussen unser Handeln. Insbesondere für Firmen sind Daten immens wichtig und stellen eine besondere Ressource dar, die konsequent gepflegt werden muss. Ohne sie könnten, zum Beispiel, keine Kreditwürdigkeitsanalysen für Kunden gemacht werden, keine Kundenaufträge bearbeitet, keine Rechnungen geschrieben und keine Gehälter gezahlt werden. Dabei kann die Speicherung von Daten langfristig angelegt sein; denken Sie nur an Jahresabschlüsse, Patientendaten oder Lebensversicherungen. Beispiele für Daten sind:

- ✓ Kontobewegungen,
- ✓ Telefonnummern,
- ✓ Kontakte im Smartphone,
- ✓ Außentemperatur,
- ✓ Konzerttermine der Lieblingsband auf ihrer nächsten Deutschland-Tournee,
- ✓ Arbeitspläne für die Produktion,
- ✓ Bestand an Aufträgen von Kunden und für Lieferanten,
- ✓ ...

Daten werden in Dateien oder in Datenbanken gespeichert. Letztlich unterscheidet sich das darin, dass man mit Datenbanken versucht, etwas Ordnung in die gespeicherten Daten zu bringen. Das werden wir natürlich später in Kapitel 2 noch konkretisieren. Aufgrund der rasanten technologischen Entwicklung hat die Speicherkapazität von Datenbanken in den letz-

ten Jahren erheblich zugenommen. Zu den größten Datenbanken der Welt gehören sicherlich Systeme von:

- ✓ Amazon,
- ✓ AT&T,
- ✓ Facebook,
- ✓ Google und
- ✓ Youtube.

Aber auch in den Datenbanken von Forschungsinstituten wie dem Max-Planck-Institut für Klimaforschung und Meteorologie, oder von Nachrichten- bzw. Geheimdiensten wie beispielsweise der NSA, werden riesige Datenmengen gespeichert. Diese Datenbanken haben Speicherkapazitäten im Petabyte-Bereich und mehr. Zur Erinnerung: 1 Petabyte = 1.000 Terabyte = 1.000.000 Gigabyte.

Daten sind teuer. Sie müssen gekauft oder erhoben, gespeichert, verwaltet und ausgewertet werden. Man benötigt dafür qualifiziertes Personal sowie Hard- und Software.



- ✓ Facebook kaufte WhatsApp für 14 Mrd. Euro. Bei 430 Mio. Nutzern sind das 31 Euro/Nutzer (Welt 2014).
- ✓ Die erste Steuersünder-CD mit 1000 Kunden der LGT Treuhand, die 2006 vom BND erworben wurde, kostete 4,6 Mio. Euro (Leyendecker 2013).
- ✓ Ein Datensatz mit Adresse, Beruf, Alter und einem weiteren Merkmal zur Person kostet – laut Angaben des Deutschen Dialogmarketing-Verbandes – durchschnittlich 18 Cent (Brzoska/Schömann-Finck 2010).
- ✓ Die Aufgaben der nächsten Datenbankklausur im Voraus zu kennen ist fast unbezahlbar.

Warum werden Daten gespeichert? Vor allem erwartet man durch den Besitz oder durch die Kenntnis von Daten Vorteile. Daten werden zu Informationen, wenn sie einen Zweckbezug haben und für Menschen lesbar aufbereitet werden. Die Wettervorhersagedaten werden zu Informationen, wenn sie meinen Wohnort betreffen und ich daran erkennen kann, ob ich morgen früh einen Schirm mitnehmen muss oder nicht. Der Unterschied zwischen Daten und Informationen ist gerade das Problem bei der Vorratsdatenspeicherung, bei der sehr viele Daten »auf Verdacht« gespeichert werden, ohne zu wissen, ob sie jemals Verwendung finden.

Daten werden nur dann zu Informationen, wenn sie in einen relevanten Kontext gestellt werden können und dort zur Lösung von Aufgaben, Problemen usw. beitragen.



Alle gespeicherten Einkaufsbons eines Lebensmittel-Discounters vom 16.03.2016 sind Daten. Erst die Liste der nachzubestellenden Artikel sind Informationen für das Unternehmen. Mein persönlicher Einkaufsbons enthält Informationen für mich, wenn ich ihn mit meiner Einkaufsliste vergleiche, oder wenn ich prüfe, ob für den Joghurt der am Regal ausgewiesene Sonderpreis berechnet worden ist.

Zusätzlich zu Daten und Informationen gibt es noch situative und prozedurale persönliche Erinnerungen und Kenntnisse, etwa

- ✓ wie kocht man eine gute Lasagne,
- ✓ die Tanzfigur »he goes she goes« beim Discofox,
- ✓ die Erinnerung an den ersten Kuss.

Nach (Abts/Mülder 2013: 9) ist die maschinelle Verarbeitung eine essenzielle Eigenschaft von Daten, während die Zweckeignung das charakteristische Merkmal von Informationen ist. Die Kombination von Zweckeignung (ja/nein) und maschineller Verarbeitung (ja/nein) gibt dann eine 4-Felder-Matrix.

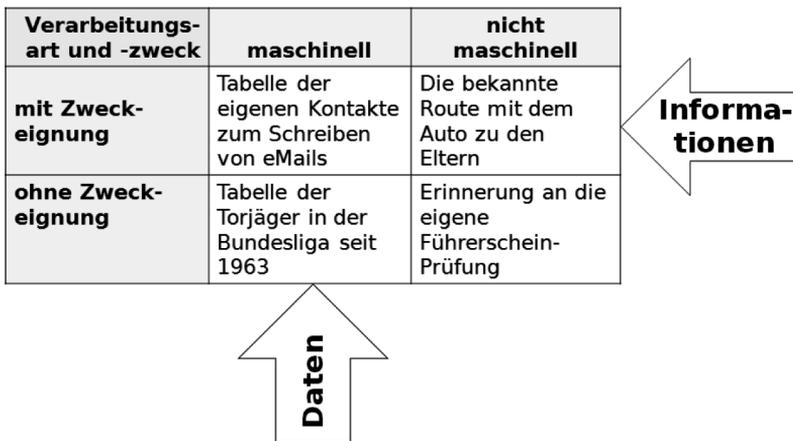


Abbildung 1.1: Beispiele für Daten und Informationen

Es ist also situationsabhängig, ob Daten Informationen sind. Mein Einkaufsbon von gestern hat wohl heute keinen Informationsgehalt mehr. Dennoch können persönliche Erinnerungen und Kenntnisse bei passender Aufgabenstellung wieder Informationen werden.



Neben dem Begriffspaar Daten und Informationen gibt es noch einen weiteren verwandten Begriff: das Wissen (Abts/Mülder 2013: 10). Wissen kann als die Summe aller persönlich gesammelten und subjektiv bewerteten Informationen betrachtet werden. Dabei unterscheidet man zwischen deklarativem Wissen (Fakten) und prozeduralem Wissen (Abläufe). Der Wissensbegriff spielt aber im Kontext der in diesem Lehrbuch behandelten Datenbanksysteme keine Rolle. Die Wissensrepräsentation und -verarbeitung ist ein eigenständiger Bereich der Informatik, siehe dazu etwa (Görz/Schneeberger/Schmidt 2014).

Ein einführendes Beispiel

In diesem Abschnitt soll an einem einführenden Beispiel gezeigt werden, wie bei der Erstellung einer Datenbank vorzugehen ist. Sie können das auch gleich selbst praktisch ausprobieren und Ihre erste eigene Datenbank erstellen. Dabei soll nur das Prinzip klar werden. Normale Datenbankanwendungen sind natürlich viel komplexer!



Bei den einzelnen Schritten sind immer Hinweise zu finden, in welchem Kapitel dieses Lehrbuches das jeweilige Thema genauer behandelt wird. So ist dieses erste Kapitel eine Art »roter Faden« für die weiteren Kapitel. Wenn Sie also noch nicht alles auf Anhieb verstehen: Das kommt noch!

Ordnung ist das halbe Leben; wir wollen deshalb eine Datenbank zur Speicherung persönlicher Kontaktdaten aufbauen. Analog zum Begriff »Bank« dienen Datenbanken zum sicheren Aufbewahren der Daten. Wie geht man bei ihrer Erstellung vor?

In der Anforderungsanalyse wird festgelegt, welche Daten zu speichern sind und wie diese strukturiert sein sollen. Wir wollen annehmen, dass die relevanten Daten aus dem Namen, Vornamen, Geburtsdatum und der Telefonnummer der einzelnen Personen bestehen. Die Anforderungsanalyse wird in den Kapiteln 5 und 6 behandelt. Sie endet mit der Erstellung eines Modells über den Aufbau der zu speichernden Daten und ihren Beziehungen untereinander. In der Praxis wird dafür das Entity-Relationship-Modell (ER-Modell) verwendet, das in Kapitel 5 ausführlich erläutert wird.

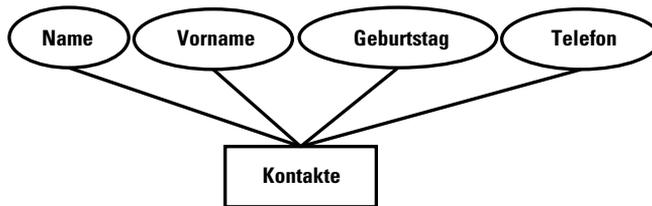


Abbildung 1.2: ER-Diagramm für die Kontakte

Der rechteckige Kasten »Kontakte« bezeichnet einen Entitätstyp, also eine Menge gleichstrukturierter Datenobjekte. Damit wird definiert, worüber Daten gespeichert werden sollen, beispielsweise Kunden, Bestellungen, Produkte. Mit den Ellipsen (Name, Vorname, Geburtstag, Telefon) kann man festlegen, durch welche Eigenschaften oder Attribute die einzelnen Objekte eines Typs beschrieben werden.

Was passiert, wenn jemand mehrere Telefonnummern hat, etwa eine Festnetz- und eine Mobilfunknummer? Dann wird das Attribut Telefonnummer ersetzt durch Festnetznummer und Mobilnummer. Das ist noch relativ einfach. Schwieriger wird es, wenn Sie beliebig viele Telefonnummern oder variable Anschlussarten wie etwa Arbeit oder Homeoffice, verwalten wollen. Dann müssen die Telefonnummern in einen eigenen Entitätstyp ausgelagert werden. Der ist dann mit den Kontakten über eine Beziehung (Relationship) zu verknüpfen, damit die einzelnen Telefonnummern der richtigen Person zugeordnet werden können. Warum das so ist, erfahren Sie in Kapitel 3, und wie man das macht, in Teil II.

Als nächstes muss festgelegt werden, in was für einer Art von Datenbanksystem die Daten gespeichert werden sollen. Es sollte klar sein, dass beispielsweise die Daten über Beziehungen zwischen Personen (Verwandtschaft, Freundes- und Kollegenkreis, ...) anders gespeichert werden müssen als reine Textdokumente. Für unsere Zwecke eignet sich am besten eine relationale Datenbank, in der alle Daten in Tabellen gespeichert werden.

Name	Vorname	Geburtstag	Telefon
Mustermann	Max	1.1.1960	0401231231
Mustermann	Anna	2.2.1960	0401231231
Mustermann	Lena	24.12.1990	0401231232
Meier	Johann	24.12.1950	030998877

Tabelle 1.1: Tabelle Kontakte

Dieser Typ von Datenbanksystem stellt – immer noch – die Standardlösung zur Implementierung von Datenbanken dar. Die einzelnen Typen von Datenbanksystemen werden in Kapitel 3 vorgestellt. Im weiteren Verlauf dieses Buches werden wir uns dann nur noch mit relationalen und einer Erweiterung davon, den objektrelationalen Datenbanksystemen, beschäftigen.

Es sollte bei der Speicherung der Daten natürlich sichergestellt werden können, dass als Geburtstag nur ein gültiges Tagesdatum gespeichert werden kann und dass es keine Telefonnummer ohne zugehörigen Namen geben darf. Man bezeichnet das als Integritäts- und Konsistenzbedingungen. Auch davon später mehr.

Bevor man jetzt die Daten tatsächlich abspeichern, verwalten und lesen kann, muss noch die Software – das sogenannte Datenbank-Managementsystem (DBMS) – ausgewählt werden. Das DBMS dient als Schnittstelle zur Datenbank; näheres dazu in Kapitel 3. Für dieses einführende Beispiel wollen wir ein ganz einfaches DBMS verwenden, den SQLite Manager, der als Add-on von Firefox installiert wird. Benutzer von Smartphones oder Tablets mit Android-Betriebssystemen haben schon mit SQLite zu tun, weil dieses DBMS ihre Kontakte usw. speichert. Näheres zu SQLite finden Sie zum Beispiel in (Allen/Owens 2010). Im weiteren Verlauf des Buches werden wir dann auf andere relationale Datenbanksysteme wie MySQL[®] oder Oracle[®] eingehen. Es ist also nicht schlimm, wenn Sie dieses Tool nicht installieren wollen oder können.



Öffnen Sie dazu den Firefox-Browser und klicken Sie auf das Symbol für die Menüschaltfläche »Menü öffnen«, oben rechts. Im aufgeklappten Menü klicken Sie auf »Add-ons« und geben dann im mit einer Lupe gekennzeichneten Suchfeld »SQLite Manager« ein (auf die genaue Schreibweise achten!), den Sie anschließend installieren müssen. Alternativ können Sie ihn auch mit Ihrer Suchmaschine suchen.

Nach Neustart von Firefox können Sie den SQLite Manager starten. Sie finden ihn oben unter Extras. Alternativ können Sie ihn auch zu Ihrer Symbolleiste oben im Fenster hinzufügen und von dort aus starten. Dazu klicken Sie wieder auf das Menü-Symbol, und wählen Sie anschließend im Menü die Option »Anpas-

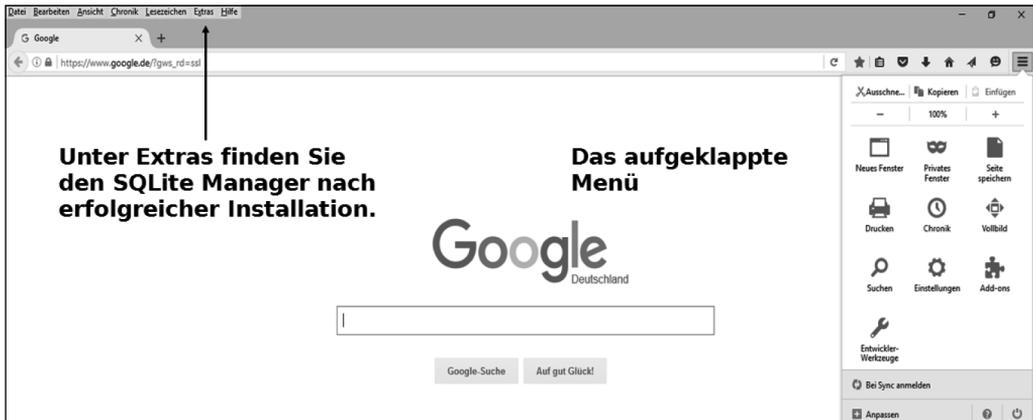


Abbildung 1.3: Aufgeklapptes Menü und »Extras« zum Start

sen«, ganz unten. Klicken Sie dann auf das Symbol des SQLite Managers, ein blaues Festplattensymbol, und verschieben Sie es mit gedrückter linker Maustaste in den Bereich für die Symbolleisten und Schaltflächen oben im Firefox-Fenster.

Der gestartete SQLite Manager erscheint dann in einem neuen Fenster.

Die folgenden Schritte sind bei der Arbeit mit einem DBMS grundsätzlich durchzuführen. Wer die Installation des SQLite Managers gemacht hat, kann das direkt ausprobieren. Es ist bei allen DBMS so oder ähnlich:

- ✓ Datenbank einrichten,
- ✓ Tabelle definieren,
- ✓ Daten speichern,
- ✓ Daten anzeigen lassen.

1. Datenbank einrichten

Um die Tabelle mit den Kontaktdaten erstellen zu können, müssen Sie festlegen, wo die Daten gespeichert werden sollen. Dies geschieht über den Menüpunkt Datenbank  Neue Datenbank. Zunächst legen Sie fest, wie Ihre Datenbankdatei heißen und dann, wo sie gespeichert werden soll.



Wenn das Fenster des SQLite Managers dabei flackert, sollten Sie es auf den gesamten Bildschirm vergrößern bzw. in einem neuen Tab starten (Extras  Einstellungen).

Danach sieht Ihr Bildschirm etwa wie folgt aus:

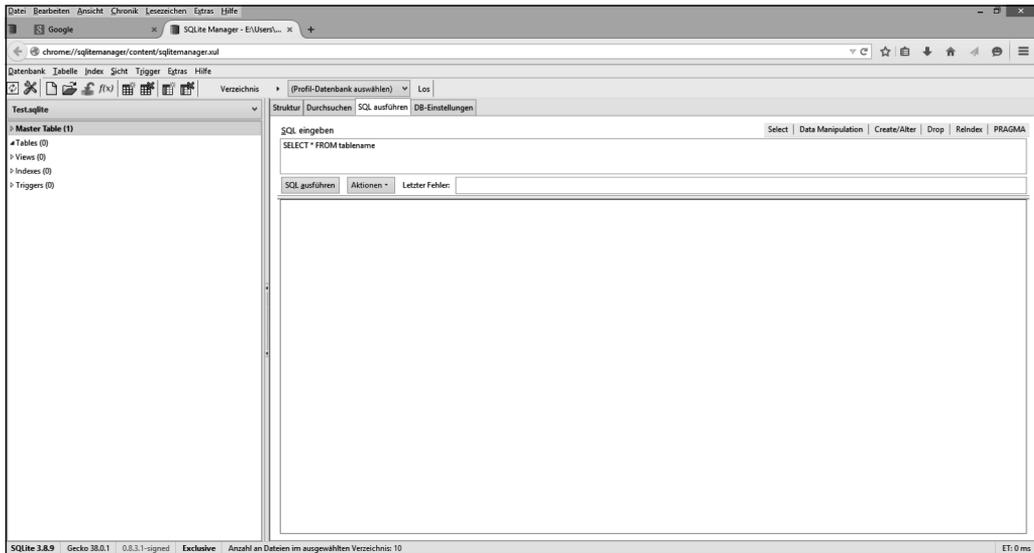


Abbildung 1.4: Das Fenster des SQLite Managers nach Neustart

2. Tabelle definieren

Über den Menüpunkt **Tabelle**  **Tabelle** erstellen oder über das zugehörige Icon können Sie nun die Tabelle »Kontakte« erzeugen. Dazu müssen Sie als Namen der Tabelle »Kontakte« eingeben und die vier Spalten Name, Vorname, Geburtstag und Telefon festlegen. Als Datentyp wählen Sie, bis auf den Geburtstag, der vom Typ DATETIME ist, immer TEXT oder VARCHAR. Die Telefonnummer sollte ebenfalls als Text definiert sein, um führende Nullen bei der Vorwahl speichern zu können.

Datenbanktabellen und ihre Definition sind Gegenstand von Kapitel 8. Dort werden Sie ebenfalls erfahren, wie Sie gut strukturierte Tabellen anlegen. Die Bedeutung von »Primärer Schlüssel«, »Autoincrement« oder »Unique« wird ebenfalls in späteren Kapiteln erläutert. Zur Option »NULL erlauben« sei nur gesagt, dass diese Spalte, wenn das Häkchen nicht gesetzt ist, bei der Dateneingabe immer ausgefüllt werden muss, um einen Datensatz speichern zu können. Erinnern Sie sich an die schon beispielhaft erwähnten Integritäts- und Konsistenzbedingungen?

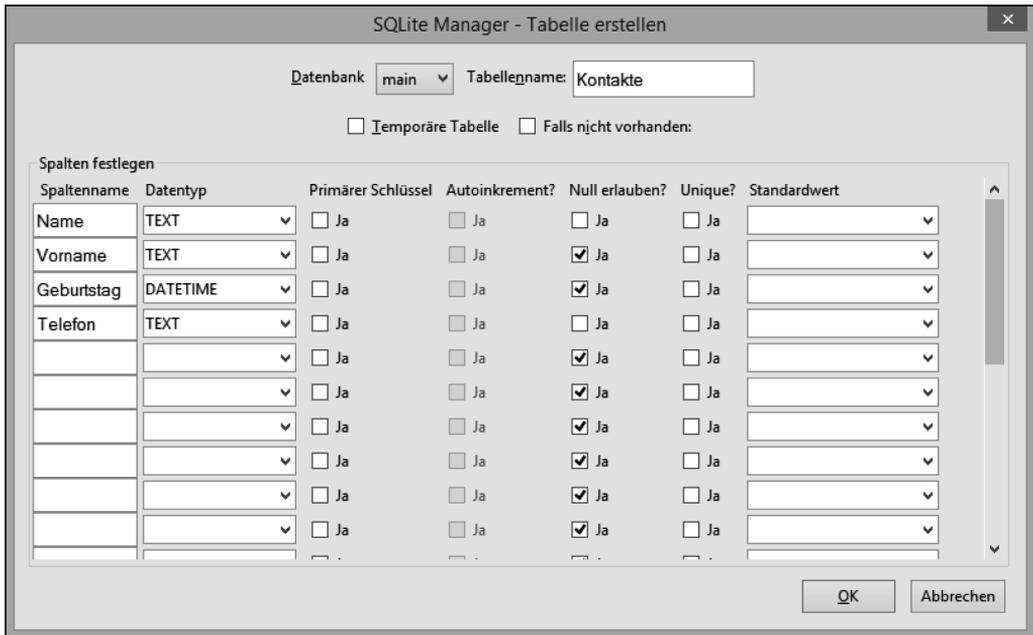


Abbildung 1.5: Erzeugen der Tabelle »Kontakte«

Zum Abschluss erscheint ein Fenster, in dem der Datenbankbefehl angezeigt wird, den Sie mit Ihren Eingaben erzeugt haben. Dort bestätigen Sie Ihre Eingaben. Ihre erstellte Tabelle wird nun im linken Menü unter »Tables« mit angezeigt.

3. Daten speichern

Anschließend können die ersten Daten in Ihre Tabelle eingegeben werden. Klicken Sie dazu links auf die Tabelle »Kontakte«, rechts auf den Reiter »Durchsuchen« und dann auf »Datensatz hinzufügen«, und geben Sie die zu speichernden Daten ein. Ihre Eingaben bestätigen Sie wieder mit OK.

Dies können Sie mehrmals wiederholen, am besten mit den Einträgen von Tabelle 1.1, damit auch die unter 4. folgende Abfrage ein Ergebnis liefert.



Wenn Sie bei der Dateneingabe etwas herumgespielt haben, ist Ihnen wahrscheinlich aufgefallen, dass Sie für den Geburtstag Eingaben machen können, die keinem gültigen Datum entsprechen. Das ist beim SQLite Manager leider so. Andere DBMS wie zum Beispiel MySQL[®] lassen das glücklicherweise nicht zu.

The dialog box 'Neuen Datensatz hinzufügen' has a title bar with a close button. Below the title bar, there is a 'Tabellenname' dropdown menu set to 'Kontakte'. Underneath, a section titled 'Feldwerte eingeben' contains four rows of input fields:

- 1. Name (TEXT) with the value 'Mustermann'
- 2. Vorname (TEXT) with the value 'Max'
- 3. Geburtstag (DATETIME) with the value '1.1.1960'
- 4. Telefon (TEXT) with the value '0401231231'

Each input field has a right-pointing arrow and a small icon. At the bottom of the dialog are two buttons: 'OK' and 'Abbrechen'.

Abbildung 1.6: Eingabe von Daten in die Tabelle »Kontakte«

4. Daten anzeigen lassen

Wenn Sie sich dann die Einträge wieder ansehen wollen, gehen Sie zum Reiter »SQL ausführen«. SQL ist die Standard-Abfragesprache für relationale Datenbanksysteme und wird detailliert in Teil III behandelt. Geben Sie beispielhaft `SELECT * FROM Kontakte where Name='Mustermann'` in das SQL-Eingabefeld ein, und drücken Sie dann auf »SQL ausführen«. Und schon werden Ihnen alle Datensätze angezeigt, die als Name »Mustermann« haben.

The screenshot shows the SQLite Manager interface. The 'SQL eingeben' field contains the query: `SELECT * FROM Kontakte where Name='Mustermann'`. Below the query field, the 'SQL ausführen' button is visible. The result is displayed in a table with the following data:

Name	Vorname	Geburtsdag	Telefon
Mustermann	Max	1.1.1960	0401231231
Mustermann	Anna	2.2.1960	0401231231
Mustermann	Lena	24.12.1990	0401231232

The interface also shows a sidebar with a tree view containing 'Master Table (1)', 'Tables (1)', and 'Kontakte'. The status bar at the bottom indicates 'SQLite 3.8.9', 'Gecko 38.0.1', and '0.8.3.1-signed'. The system tray shows the time as 19:48 on 12.06.2015.

Abbildung 1.7: Ergebnis der SQL Query

Sie können natürlich noch viel mehr machen; zum Beispiel, mit »Data Manipulation« Datensätze ändern oder löschen. Zur Einführung sollte das aber genügen.



Zugegeben, dieses einfache Beispiel wäre auch mit einer Tabellenkalkulation lösbar, obwohl man bei den Abfragemöglichkeiten dort sehr schnell an die Grenzen stößt. Außerdem besteht bei einer Tabellenkalkulation die große Gefahr, inkorrekte Daten zu speichern: Sie können dort, zum Beispiel, eine Spalte als Zahl definieren und trotzdem Buchstaben eintragen. Versuchen Sie das mal bei einem Datenbanksystem!

In der Praxis haben Sie es bei der Datenspeicherung mit Aufgabenstellungen zu tun, bei denen

- ✓ viele Tabellen vorhanden sind,
- ✓ umfangreiche Datenmengen abgespeichert werden müssen,
- ✓ zwischen den Daten Beziehungen bestehen,
- ✓ mehrere Benutzer über mehrere Programme auf die Daten zugreifen wollen,
- ✓ sehr unterschiedliche Abfragen gestellt werden sollen,
- ✓ aber auch Daten gespeichert, geändert oder gelöscht werden,
- ✓ wobei trotzdem die Integrität und Konsistenz der Daten gewährleistet sein soll.

Dazu braucht man eben Datenbanksysteme!

Aufgaben

Vertiefungsfragen

1. Recherchieren Sie im Internet etwas über die größten Datenbanken der Welt!
2. Recherchieren Sie im Internet, was Daten wert sind!
3. Warum ist die Telefonnummer vom Datentyp »Text« und nicht »Integer«?
4. Welche Eigenschaften haben eine Bank und eine Datenbank gemeinsam?
5. Erläutern Sie die Unterschiede zwischen Daten, Informationen und Wissen!

Übungen

1. Fügen Sie der Tabelle »Kontakte« in Ihrer SQLite Datenbank eine Adresse, bestehend aus PLZ, Straße (einzugeben als Strasse) und Ort, hinzu!
2. Tragen Sie bei Ihren bestehenden Kontakten jeweils eine Adresse ein!
3. Fügen Sie der Tabelle »Kontakte« weitere Datensätze hinzu! Beachten Sie dabei die nächste Frage.
4. Führen Sie folgende Datenbankabfragen mit SQL durch:
 - Welche Personen wohnen in Hamburg?
 - Welche Personen wohnen in Berlin und heißen Mustermann?