

Datenbanksysteme für Dummies Modellierung und Implementierung einer relationalen Datenbank

» Hier geht's
direkt
zum Buch

DIE LESEPROBE

Unterschied von Daten und Informationen

Daten als wichtige Ressource

Wozu braucht man Datenbanksysteme?

Meine erste Datenbank

Kapitel 1

Wofür Datenbanksysteme gut sind

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 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 notwendig sein; denken Sie nur an Jahresabschlüsse, Patientendaten oder Lebensversicherungen. Beispiele für Daten sind:

- ✓ Kontobewegungen,
- ✓ Telefonnummern,
- ✓ Kontakte im Smartphone,
- ✓ Außentemperatur (aktuell und stündliche Vorhersage für den Rest des Tages),

- ✓ Konzerttermine der Lieblingsband auf ihrer nächsten Deutschlandtournee,
- ✓ 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 und den Zugriff darauf zu erleichtern. Das werden wir natürlich später in Kapitel 2 noch konkretisieren. Aufgrund der rasanten technologischen Entwicklung hat die Speicherkapazität von Datenbanken in den letzten 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 oder von Nachrichten- bzw. Geheimdiensten wie beispielsweise der NSA werden riesige Datenmengen gespeichert. Diese Datenbanken haben Speicherkapazitäten im Petabytebereich 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 damals 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).
- ✓ Beim Business-to-Business (B2B) kosten Adressen, die für das Direktmarketing erworben werden sollen, bei einem Anbieter beispielsweise 10 Cent pro Adresse (Gruenderlexikon.de 2020).
- ✓ 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. Die schon erwähnten Vorhersagewerte der Außentemperatur stellen für mich Informationen dar, wenn sie für meinen Wohnort gelten und ich entscheiden will, ob ich einen Pullover anziehen muss oder nicht.



Alle gespeicherten Einkaufsbons eines Lebensmittel-Discounters vom 01.03.2022 sind Daten. Erst die anhand der aktualisierten Bestände erzeugte Liste der aktuell nachzubestellenden Artikel sind Informationen für das Unternehmen.

Ein weiteres schönes Beispiel liefert die Corona-Warn-App des Robert-Koch-Instituts aus dem Jahr 2020. Die auf dem Smartphone lokal gespeicherte Liste von Identitätscodes mit potenziell gefährlichen Kontakten sind Daten. Auch die heruntergeladene Kontaktliste mit den Codes positiv getesteter Personen sind Daten. Erst der Abgleich liefert die Information, ob und wann man einem potenziellen Infektionsrisiko ausgesetzt war.

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 2017: 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).

Es ist also wegen der Zweckeignung situationsabhängig, ob Daten Informationen sind. Mein Einkaufsbons von gestern hat wohl heute keinen Informationsgehalt mehr. Auch die Buchungs- und Sitzplatznummer für einen schon durchgeführten Flug sind nicht mehr relevant; es sei denn, dieser Flug soll zum Sammeln von Meilen mit der Vielfliegerkarte nachgetragen werden. Persönliche Erinnerungen und Kenntnisse können bei passender Aufgabenstellung ebenfalls wieder Informationen werden.

Neben dem Begriffspaar Daten und Informationen gibt es noch einen weiteren verwandten Begriff: das Wissen (Abts/Mülder 2017: 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/Schmid/Braun 2020).

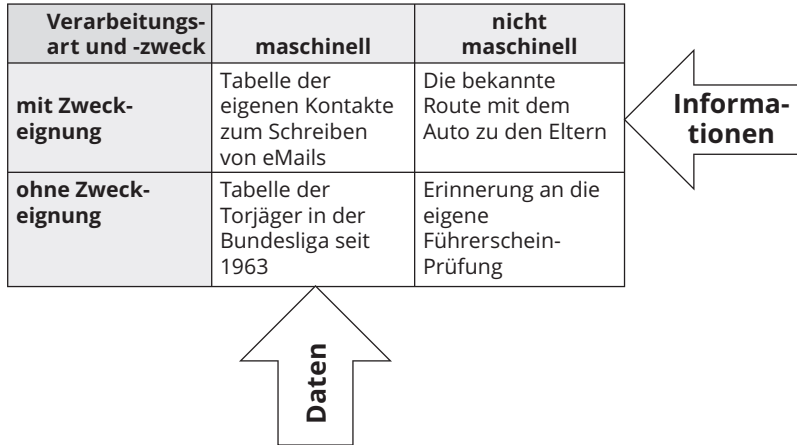


Abbildung 1.1: Beispiele für Daten und Informationen

Der Unterschied zwischen Daten, Informationen und Wissen wird am folgenden Beispiel klar; siehe auch die URL unten für eine ähnliche Aufgabe eines Mathe-Wettbewerbs aus Singapur.



Die beiden Freunde Jan und Tom fragen Charlotte auf einer Party, wann sie Geburtstag hat. Sie will das zunächst nicht verraten, nennt aber folgende Tage zur Auswahl, von denen einer ihr Geburtstag ist:

- 12. und 27. Februar
- 15., 27. und 28. Mai
- 17. und 18. Juni
- 12., 15. und 17. August

Anschließend verrät Charlotte Jan den richtigen Monat und Tom den richtigen Tag. Die beiden unterhalten sich dann über das, was sie daraus ableiten können.

Jan: Nun kennt jeder von uns einen Teil des Geburtstags.
Aber keiner kennt das genaue Datum.

Tom: Zuerst kannte ich das vollständige Datum nicht, aber jetzt ja.

Jan: Nun kennen wir es beide.

Was in diesem Beispiel sind Daten, Informationen und Wissen? Und vor allem: Wann hat Charlotte Geburtstag?

Die obige Tabelle (12. und 27. Februar usw.) beinhaltet die Daten. Die darauf folgende Unterhaltung enthält Informationen. Und das Wissen? Das ist der 27. Februar! Lösungshinweis: Ausschlussverfahren; den Lösungsweg zu einer vergleichbaren Aufgabe finden Sie hier: <https://www.spiegel.de/wissenschaft/mensch/so-loesen-sie-das-logik-raetsel-aus-singapur-a-1028379.html>

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 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 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 4 und 5 behandelt. Sie endet mit der Erstellung eines Modells über den Aufbau der zu speichernden Daten und deren Beziehungen untereinander. In der Praxis verwendet man dafür das Entity-Relationship-Modell (ER-Modell), das in Kapitel 5 ausführlich erläutert wird.

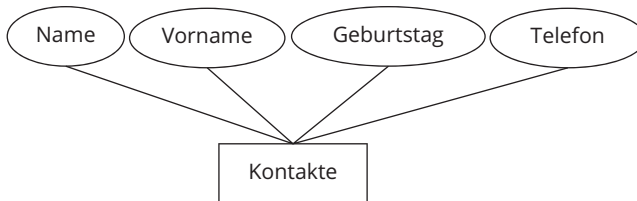


Abbildung 1.2: ER-Diagramm für die Kontakte

Der rechteckige Kasten »Kontakte« in Abbildung 1.2 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 »Telefon« 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	01.01.1970	0401231231
Mustermann	Anna	02.02.1970	0401231231
Mustermann	Lena	24.12.2000	0401231232
Meier	Johann	24.12.1960	030998877

Tabelle 1.1: Tabelle mit den Kontakten

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 (bis auf ein Programmierbeispiel in Kapitel 16) nur noch mit relationalen und einer Erweiterung davon, den objektrelationalen Datenbanksystemen, beschäftigen.

Es sollte bei der Speicherung der Daten in Tabelle 1.1 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 als DBMS Microsoft Access® betrachten, das Teil des Office-Pakets von Microsoft ist.

Die folgenden Schritte sind bei der Arbeit mit einem DBMS grundsätzlich durchzuführen. Wer Access installiert hat, kann das direkt ausprobieren. Große Programmierkünste sind dabei nicht erforderlich. Auch Laien können die Software nach einer kleinen Einführung zumindest in den Grundzügen bedienen. Es ist bei allen DBMS so oder ähnlich:

1. Datenbank einrichten,
2. Tabelle definieren,
3. Daten speichern,
4. Daten anzeigen lassen, gegebenenfalls bearbeiten.

In althergebrachter Arbeitsweise, d. h. ohne EDV, entspricht das Einrichten der Datenbank der Auswahl eines (noch leeren) Aktenschanks, in dem Aktenordner gelagert werden sollen. Die Tabellendefinition ist mit der Beschriftung eines leeren Ordners und dem Anlegen eines Registers zu vergleichen. Die Datenspeicherung entspricht dann dem Einsortieren von Dokumenten in einen Ordner; das Anzeigen ist natürlich mit dem Suchen in einem Ordner vergleichbar.

- 1. Datenbank einrichten.** Um die Tabelle mit den Kontaktdaten erstellen zu können, müssen Sie zunächst festlegen, wo die Daten gespeichert werden sollen. Am besten wählen Sie bei Access dafür eine leere Desktopdatenbank. Die anderen Optionen sind hier nicht von Bedeutung. Dabei können Sie den Namen der Datenbank und den Speicherort festlegen.
- 2. Tabelle definieren.** Als nächstes müssen Sie die Tabelle »Kontakte« erzeugen. Klicken Sie dafür zunächst oben links auf »Ansicht« und im sich aufklappenden Drop-down-Menü auf »Entwurfsansicht«. Als nächstes werden Sie nach dem Tabellennamen gefragt; geben Sie dort »Kontakte« ein und bestätigen Sie Ihre Eingabe mit »ok«.

Anschließend tragen Sie in dem Bereich mit den drei Überschriften »Feldname«, »Feldtyp« und »Beschreibung (optional)« die Bezeichnungen der vier Spalten »Nachname« (»Name« ist ein reserviertes Wort und sollte nicht verwendet werden), »Vorname«, »Geburtstag« und »Telefon« Ihrer Kontakttabelle ein. Als Datentyp wählen Sie, bis auf den Geburtstag, der vom Typ DATUM/UHRZEIT ist, immer »kurzer Text«. Das sind Texte bis maximal 255 Zeichen. Die Telefonnummer sollte ebenfalls als Text definiert sein, um führende Nullen bei der Vorwahl speichern zu können (Abbildung 1.3).

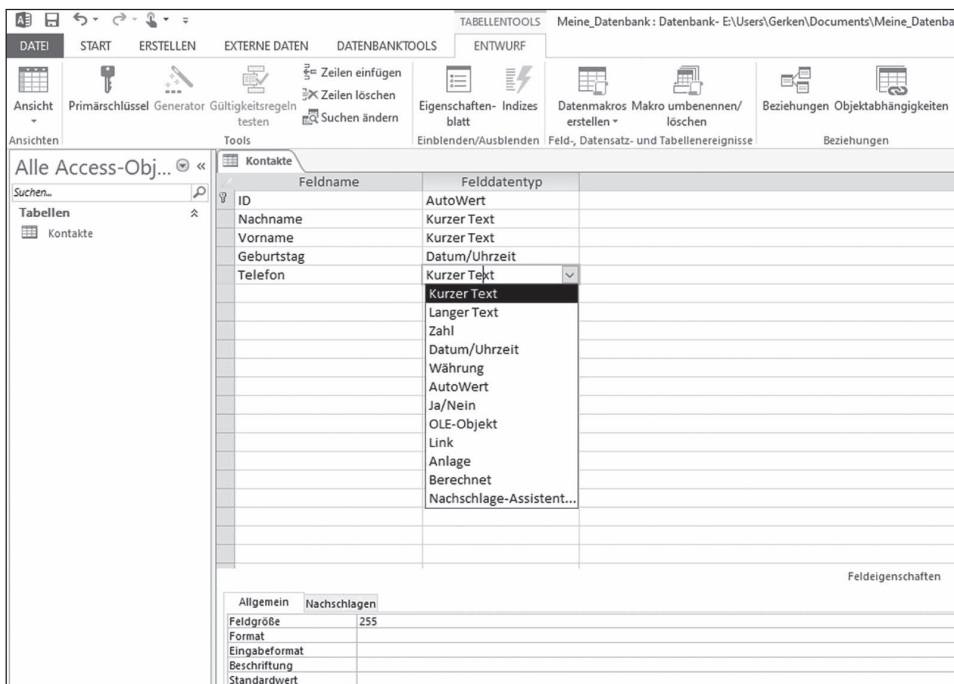


Abbildung 1.3: Erzeugen der Tabelle »Kontakte« © Microsoft Corp.

Alle Spalten in einer Datenbanktabelle müssen einen der vorgegebenen Datentypen haben. Bei Microsoft Access stehen dafür beispielweise kurzer Text, langer Text, Zahl, Datum, Währung und AutoWert zur Verfügung. Die erste Spalte jeder Tabelle ist bei Access vordefiniert, heißt ID, ist vom Typ AutoWert und enthält einen vom DBMS beim Speichern eines Datenbanksatzes automatisch vergebenen Primärschlüssel.

Datenbanktabellen und ihre Definition sind Gegenstand von Kapitel 7. Dort werden Sie ebenfalls erfahren, wie Sie gut strukturierte Tabellen anlegen. Die Bedeutung von »Primärschlüssel«

und »AutoWert« wird ebenfalls in späteren Kapiteln erläutert; ebenso die weiteren Optionen wie beispielsweise »Standardwert«, »Eingabe erforderlich« oder »Indiziert«. Erinnern Sie sich an die schon beispielhaft erwähnten Integritäts- und Konsistenzbedingungen?

Jetzt sollten Sie Ihre Tabellendefinition abspeichern und zur Datenblattansicht wechseln; alternativ geht auch ein Doppelklick auf den Tabellennamen links.

- 3. Daten speichern.** Anschließend können die ersten Daten in Ihre Tabelle eingegeben werden. Die Eingabe kann immer nur zeilenweise in die Zeile erfolgen, die mit »(neu)« in der ersten Spalte gekennzeichnet ist.

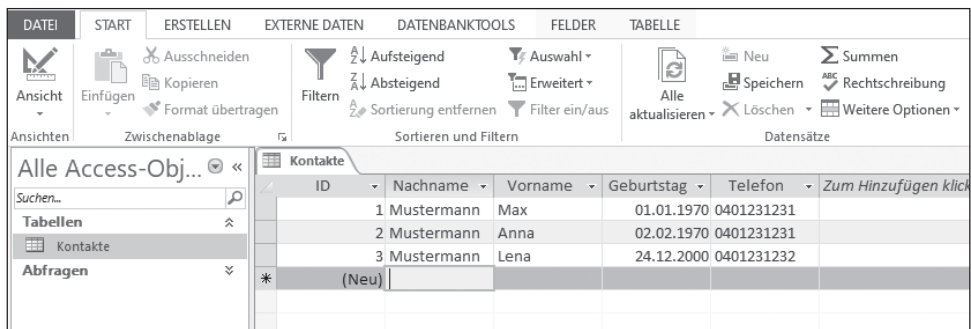


Abbildung 1.4: Eingabe von Daten in die Tabelle »Kontakte« © Microsoft Corp.

Dies können Sie mehrmals wiederholen, am besten mit den Einträgen von Tabelle 1.1, damit auch die unter 4. folgende Abfrage ein Ergebnis liefert. Nach der Eingabe kann es nicht schaden, die Tabelle zu speichern.



Wenn Sie bei der Dateneingabe etwas herumgespielt haben, ist Ihnen wahrscheinlich aufgefallen, dass Sie für den Geburtstag nur Eingaben machen können, die einem gültigen Datum entsprechen. Das ist der große Vorteil der von einem DBMS erzwungenen Typisierung der einzelnen Spalten.

- 4. Daten anzeigen lassen.** Wenn Sie sich dann die Einträge wieder ansehen wollen, stehen verschiedene Möglichkeiten zur Verfügung. Sie können nach bestimmten Daten einer Zeile filtern oder auch Daten suchen. Insbesondere steht unter dem Menüpunkt ERSTELLEN → Abfrageentwurf ein mächtiges Werkzeug zur Abfrageerstellung zur Verfügung.

Wenn Sie beispielsweise den Namen und Geburtstag aller Personen anzeigen wollen, die Mustermann heißen, sieht das in der Entwurfsansicht folgendermaßen aus (Abbildung 1.5).

Sie können aber auch, und das ist im Kontext dieses Buches viel interessanter, die Abfragesprache SQL – Structured Query Language – verwenden. Dazu brauchen Sie nur in der Abbildung 1.5 mit der rechten Maustaste auf den Reiter »Abfrage1« zu klicken, dann auf SQL-Ansicht, und schon erhalten Sie den SQL-Befehl dieser Datenbankabfrage, den Sie dann auch modifizieren können.

```
SELECT Kontakte.Nachname, Kontakte.Vorname, Kontakte.Geburtstag
FROM Kontakte
WHERE Kontakte.Nachname = 'Mustermann';
```

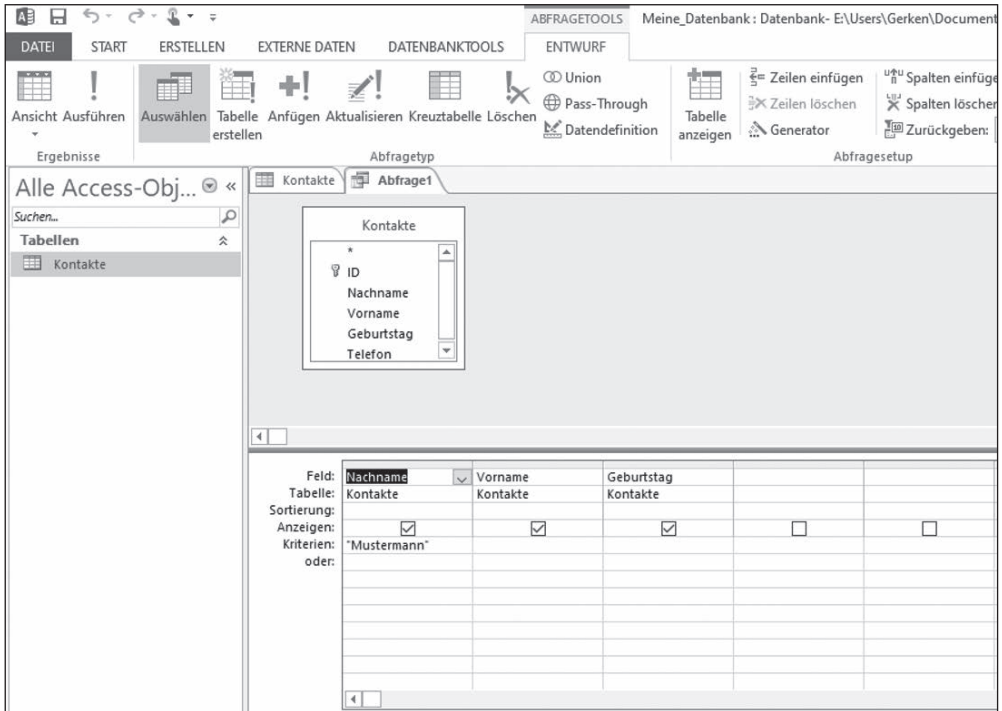


Abbildung 1.5: Definition einer Datenbankabfrage © Microsoft Corp.

SQL ist ein sehr wichtiges Thema, wenn Sie sich mit Datenbanken beschäftigen wollen. Sie lernen diese Sprache in Teil III dieses Buches ausführlich kennen.

Zugegeben, dieses einfache Beispiel wäre auch mit einer Tabellenkalkulation lösbar. Sowohl relationale Datenbanksysteme als auch Tabellenkalkulationsprogramme verwenden Tabellen. Es bestehen aber wesentlichen Unterschiede:

- ✓ Bei einem Datenbanksystem müssen die Bezeichnung und der Datentyp der einzelnen Spalten – und damit die Struktur einer Tabelle – erst vollständig definiert sein, bevor Daten eingegeben werden können.
- ✓ Sie können in eine Datenbanktabelle nur vollständige Zeilen (auch Tupel genannt) und nicht einzelne Zellen speichern. Allerdings gibt es den sogenannten NULL-Wert als Platzhalter für unbekannte oder nicht vorhandene Werte.
- ✓ In einer Datenbanktabelle können fast beliebig viele Tupel eingefügt und gespeichert werden. Bei Tabellenkalkulationen ist die Speicherkapazität erheblich eingeschränkter.
- ✓ Sie können nur vollständige Zeilen aus der Datenbank löschen. Das Löschen einer Tabellenspalte ist eine Strukturänderung.
- ✓ Es steht eine sehr mächtige und vor allem genormte Abfragesprache zur Verfügung: die schon erwähnte Structured Query Language, kurz SQL.

- ✓ Sie können Beziehungen zwischen Spalten mehrerer Tabellen herstellen. Wozu das gut ist und wie man das macht, lernen Sie später.
- ✓ Außerdem besteht bei einer Tabellenkalkulation die große Gefahr, inkorrekte Daten zu speichern: Sie können dort beispielsweise 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 überwiegend 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 keine Zahl?
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 Datenbank eine Spalte »Wohnort« hinzu und tragen Sie bei Ihren bestehenden Kontakten jeweils den Wohnort ein.
2. Fügen Sie der Tabelle »Kontakte« weitere Datensätze hinzu! Beachten Sie dabei die nächste Frage.
3. Finden Sie über eine Datenbankabfrage heraus, welche Kontakte in Hamburg wohnen.