



Frank  
Geisler

4. Auflage

A 3D visualization of binary data (0s and 1s) rendered as blue and grey cubes on a blue gradient background. The cubes are arranged in a grid that recedes into the distance, creating a sense of depth. The lighting is dramatic, with strong highlights and deep shadows.

# Datenbanken

## Grundlagen und Design

# Teil I

## Grundlagen

### In diesem Teil:

- **Kapitel 1**  
Einführung in das Thema Datenbanken . . . . . 19
- **Kapitel 2**  
Datenbanksysteme, Datenbankanwendungen und  
Middleware . . . . . 43
- **Kapitel 3**  
Das relationale Datenbankmodell. . . . . 93



# Einführung in das Thema Datenbanken

Die Umwelt, in der wir leben, wird immer komplexer und vielfältiger. Oft wird der Begriff des *Information-Overkills* bemüht, wenn es darum geht, die Informationsflut zu beschreiben, die aus den unterschiedlichsten Quellen Tag für Tag auf uns einprasselt. Um gute und richtige Entscheidungen treffen zu können, müssen immer mehr Informationen bedacht, ausgewertet und in Korrelation zueinander gestellt werden. Bei dieser schwierigen Aufgabe, die für uns relevanten Informationen aus dem Datenwust herauszufiltern und zum richtigen Zeitpunkt zur Verfügung zu stellen, sind wir auf die Hilfe von computergestützten Systemen angewiesen.

Bevor ich mich näher mit der Thematik Datenbanken an sich beschäftige, sollten Sie einen Blick auf das werfen, was Sie verwalten möchten, die Daten. *Daten* selbst repräsentieren Fakten. Ein mögliches Datum ist z.B. die Rechnungsnummer 32532, die eine Rechnung trägt, die ich zugestellt bekommen habe. Damit aus Daten *Informationen* werden, müssen die Daten in einen Zusammenhang gebracht werden. Stellen Sie sich vor, dass Sie einen Mitarbeiter des Unternehmens treffen, das mir die oben genannte Rechnung geschickt hat. Wenn Sie diesen Zeitgenossen außerhalb seines Büros antreffen (abgenabelt von all seinen famosen Computersystemen) und ihn mit der Rechnungsnummer 32532 konfrontieren, wird dies ziemlich wahrscheinlich mit einem Stirnrunzeln beantwortet, da sich der gute Mann unter Rechnungsnummer 32532 nichts vorstellen kann. Die Rechnungsnummer ist einfach ein Datum, das ein Faktum darstellt. Aus dem Zusammenhang gerissen hat dieses Datum für sich alleine keine Bedeutung (der Servicemitarbeiter kann noch nicht einmal sagen, ob es überhaupt eine Rechnung mit Rechnungsnummer 32532 gibt). Damit aus der Rechnungsnummer 32532 eine sinnvolle Information wird, muss diese in einen Zusammenhang gebracht werden. Das Datum muss *verarbeitet* werden.

Hat der Servicemitarbeiter wieder Zugriff auf seinen Computer, so kann er die genannte Rechnungsnummer dort eingeben (Sie werden es sicherlich schon erraten haben – hier läuft irgendwo im Hintergrund eine Datenbank) und in Windeseile erhält er weitere Daten bzw. Fakten, die in direktem Zusammenhang mit der Rechnungsnummer 32532 stehen. Weitere Daten, die in Verbindung mit der Rechnungsnummer stehen, sind z.B., dass der Kunde, auf den diese Rechnung ausge-

stellt ist, »Frank Geisler« heißt, dass der Rechnungsbetrag 145,42 € ist und dass diese Rechnung bisher noch nicht bezahlt wurde. Durch die Verknüpfung von einzelnen Daten entstehen Informationen, die wiederum Entscheidungen beeinflussen können oder Handlungen auslösen. In diesem Beispiel veranlasst die Information, dass Frank Geisler die Rechnung 32532, die einen Betrag von 145,42 € aufweist, noch nicht bezahlt hat, dass mir eine Mahnung zugestellt wird. Wir können dieses Beispiel noch ein wenig weiter spinnen. Da der Computer alle von mir getätigten Bestellungen bei der Firma kennt, kann er ohne weiteres Daten über alle von mir getätigten Bestellungen abrufen. Aus diesen Daten ergibt sich die Information, dass ich meine Rechnungen insgesamt nicht so regelmäßig bezahle und dass des Öfteren Mahnungen verschickt worden sind. Das Management dieser Beispielfirma kann nun aufgrund der aus den Daten enthaltenen Informationen Entscheidungen treffen. Eine mögliche Entscheidung ist z.B. die, dass ich bei dieser Firma keine Waren mehr auf Rechnung kaufen darf, sondern dass ich Vorkasse leisten muss, wenn ich etwas kaufen möchte. Verlassen wir das Beispiel an dieser Stelle, bevor es peinlich für mich wird...

An dem Beispiel wird nicht nur deutlich, dass erst die Informationen, die aus den Daten gewonnen werden können, das eigentlich Wertvolle und Wichtige sind, sondern dass dieselben Daten, in einen anderen Zusammenhang gebracht, andere Informationen ergeben können. Das Datum, wann eine Rechnung bezahlt wurde, ergibt, bezogen auf eine einzelne Rechnung, die Information, ob diese bereits bezahlt wurde oder nicht. Fügt man das Datum in einen anderen Zusammenhang ein, indem man z.B. alle Zahlungseingänge eines bestimmten Kunden betrachtet, so lassen sich mit denselben Daten Informationen über das Zahlungsverhalten des Kunden, ja sogar ein Zahlungsprofil erstellen.

Die Information über das Zahlungsprofil kann zu weiter reichenden Entscheidungen führen. So ist es z.B. möglich, treuen, gut zahlenden Kunden einen bestimmten Rabatt einzuräumen, wohingegen sich die Zahlungsmodalitäten von notorischen Späzahlern verschlechtern können.

Lassen Sie uns nun noch einmal die grundlegenden Aussagen der vorherigen Abschnitte zusammenfassen:

- Informationen setzen sich aus Daten zusammen.
- Durch Datenverarbeitung werden aus Daten Informationen.
- Gute Daten, die zeitnah vorliegen, helfen uns, gute Entscheidungen zu treffen.
- Der Informationsgehalt von Daten hängt vom Zusammenhang ab.

Damit aus Daten gute Informationen gewonnen werden können, müssen diese Daten sorgfältig erfasst und in einem Format vorgehalten werden, auf das man leicht zugreifen und das einfach verarbeitet werden kann. Da Daten der Ausgangspunkt aller weiteren Aktivitäten sind, ist es wichtig, dass mit den Daten sehr sorg-

fältig umgegangen wird. Datenfehler pflanzen sich durch das ganze System fort und führen zu fehlerhaften Informationen, die wiederum zu falschen Entscheidungen führen können. Der Umgang mit Daten wird als *Datenmanagement* bezeichnet. Aufgaben des Datenmanagements sind die Erzeugung, Speicherung und Wiedergabe der Daten. Da Daten eine zentrale Rolle bei der Erzeugung von Informationen spielen, ist es nicht verwunderlich, dass das Datenmanagement in vielen Firmen eine zentrale Rolle spielt.

Datenmanagement ist keine Erfindung des IT-Zeitalters. Daten wurden seit jeher in irgendeiner Form verwaltet. Sei es, dass die Daten in Stein geritzt wurden oder meterlange Aktenschränke mit Papier füllten. Die Neuerung, die das IT-Zeitalter gebracht hat, ist die Darstellung von Daten in elektronischer Form, was das Datenmanagement wesentlich vereinfacht und effizienter macht. Eine zentrale Rolle des elektronischen Datenmanagements spielt die *Datenbank*. Es gibt mindestens so viele Definitionen des Begriffs Datenbank, wie es Programmierer und Datenbankspezialisten gibt. Ich habe einmal zwei Definitionen herausgenommen, die mir am eingängigsten erscheinen und die verdeutlichen, wie der Begriff Datenbank verwendet wird:

### Wichtig

1. Eine Datenbank ist ein verteiltes, integriertes Computersystem, das Nutzdaten und Metadaten enthält. *Nutzdaten* sind die Daten, die Benutzer in der Datenbank anlegen und aus denen die Informationen gewonnen werden. *Metadaten* werden oft auch als Daten über Daten bezeichnet und helfen, die Nutzdaten der Datenbank zu strukturieren.
2. Eine Datenbank ist eine geordnete, selbstbeschreibende Sammlung von Daten, die miteinander in Beziehung stehen.

### Wichtig

Während die erste Definition eher den technischen Aspekt heraushebt und auf die Realisierung einer Datenbank als Computersystem abhebt, stellt die zweite Definition den theoretischen Aspekt in den Vordergrund und ist daher universeller verwendbar als die erste Definition.

Lassen Sie uns die zweite Definition noch einmal näher am Beispiel eines Adressbuchs betrachten, das einfach in Form einer Tabelle angelegt ist:

Name	Telefonnummer	Anschrift	Ort
Max Mustermann	0123 / 456789	Musterstraße 3	Musterhausen
Susi Sorglos	0987 / 654321	Sorglosgasse 7	Schlumpfhausen

Im dargestellten Adressbuch befindet sich zunächst eine Sammlung von Daten, nämlich die Adressen. Diese sind nach dem Alphabet geordnet. Obwohl es sich bei jeder Adresse um einen Kontakt handelt, stehen diese nicht in einer Beziehung zueinander. Es handelt sich lediglich um Instanzen des Objekts »Leute, die so interessant sind, dass sie in ein Adressbuch eingetragen wurden«. Der Ausdruck »die miteinander in Beziehung stehen« der Definition bezieht sich auf verschiedene Tabellen, die untereinander in Beziehung stehen können. Zu diesem Thema erfahren Sie im weiteren Verlauf des Buches mehr. Eine *Selbstbeschreibung* des Adressbuchs erfolgt durch die Tabellenüberschriften. Die Überschriften erklären, was der Inhalt der jeweiligen Spalte bedeutet. Diese Beschreibungen der Daten werden, wie bereits aus der ersten Definition des Begriffs Datenbank bekannt, als *Metadaten* bezeichnet. Offensichtlich ist nach der zweiten Definition ein simples Adressbuch in Form einer Tabelle eine (wenn auch nicht digitale) Datenbank. Die Metadaten werde im Datenbanksystem im so genannten *Katalog* (auch *Data Dictionary*) gespeichert. Der Katalog stellt den Teil der Datenbank dar, in dem die Metainformationen abgelegt werden.

Um eine Datenbank auf einem Computer zu verwalten, wird in der Regel ein so genanntes *Datenbankmanagement-System (DBMS)* verwendet, das sich um die Organisation der Daten kümmert und das den Zugriff auf die Daten regelt. Das Datenbankmanagement-System kann entweder aus einem einzelnen Programm bestehen, wie es oft bei Desktop-Datenbanksystemen wie z.B. Microsoft Access der Fall ist, oder es kann aus vielen Programmen bestehen, die zusammenarbeiten und so die Funktionalität eines DBMS bereitstellen. Diese Variante wird oft bei servergestützten Datenbanksystemen verwendet. In Abbildung 1.1 ist das Zusammenspiel zwischen DBMS und Datenbank zu sehen. Ein Anwender formuliert eine Abfrage an die Datenbank, die die benötigten Daten zurückliefern soll. Die Abfrage wird an das DBMS weitergereicht, das die Daten aus der eigentlichen Datenbank herausucht und diese an den Anwender zurückliefert. Ferner kann man in Abbildung 1.1 sehen, dass in der Datenbank sowohl Metainformationen als auch Nutzdaten vorhanden sind. Die Nutzdaten bestehen im Beispiel aus den Daten in den Tabellen Kunden, Berater und Projekte.

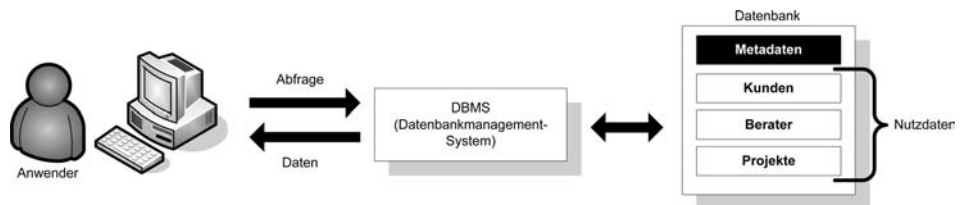


Abb. 1.1: Das DBMS verwaltet den Zugriff auf die Datenbank.

## Hinweis

Aus Bequemlichkeit wird in der Praxis oft der Begriff Datenbank anstatt Datenbankmanagement-System verwendet. Anwendungen wie z.B. Access oder Oracle werden oft als Datenbanken bezeichnet, obwohl sie in Wirklichkeit Datenbankmanagement-Systeme sind. Wenn Sie also auf den Begriff Datenbank stoßen, so müssen Sie versuchen, aus dem Kontext zu erschließen, was denn nun eigentlich gemeint ist. Geht es um das Datenbankmanagement-System, die technische Ausführung eines Datenbanksystems (also Hard- und Software) oder um das logische Konzept der Datenbank?

Datenbankmanagement-Systeme sind aus unserem modernen Leben nicht mehr wegzudenken und bilden sozusagen das Rückgrat der Informationsgesellschaft. Daten, aus denen Informationen gewonnen werden können, sind zu einem wichtigen Rohstoff geworden, der natürlich auch entsprechend behandelt werden muss. Daher stellen moderne Datenbankmanagement-Systeme viele Funktionen zur Verfügung, die für die Pflege und das Auslesen der Daten wichtig sind und den Umgang mit Daten vereinfachen. Ein wichtiger Bestandteil moderner Datenbanksysteme ist die integrierte Abfragesprache, mit der man einfach so genannte *Ad-hoc-Abfragen* an die Datenbank absetzen kann. Eine Ad-hoc-Abfrage dient dazu, Informationen abzufragen, die eine bestimmte, aktuelle Fragestellung beantworten sollen. Um auf das Beispiel des Servicemitarbeiters vom Anfang des Kapitels zurückzukommen, könnte dieser, nachdem er sich vom Schock erholt hat, einfach auf der Straße auf die Rechnung mit der Nummer 32532 angesprochen worden zu sein, und in sein Büro zurückgekehrt ist, eine Ad-hoc-Abfrage starten, die die mit der Rechnungsnummer 32532 verknüpften Daten zurückliefert und ihm Informationen zu der durch Rechnungsnummer 32532 identifizierten Rechnung gibt.

Mit Hilfe von Datenbankmanagement-Systemen wird eine Umgebung geschaffen, in der Daten besser organisiert werden können, als dies vor der Entwicklung von Datenbankmanagement-Systemen möglich war. Daten können leicht zur Datenbank hinzugefügt, geändert und gelöscht werden und es werden durch das DBMS leistungsfähige Suchfunktionen zur Verfügung gestellt, so dass bestimmte Daten schnell wieder gefunden werden können. Der Erfolg der DBMS ist so groß, dass Microsoft momentan überlegt, den nächsten Windows-Versionen (Codename Longhorn), anstelle eines normalen Dateisystems ein DBMS mitzugeben, das dann die Dateien auf der Festplatte verwaltet. Den Vorteil sieht Microsoft darin, dass nicht nur, wie bisher, Dateien, sondern auch jede andere Art von Daten (z.B. Adressen) im Dateisystem gespeichert, mit anderen Objekten (wie z.B. Dateien) in Zusammenhang gebracht und abgefragt werden kann.

Durch den schnellen Zugriff, den Datenbanksysteme auf Daten erlauben, und unter Verwendung von Tools, die die Daten in sinnvolle Informationen umwandeln, ist es dem Nutzer einer Datenbank möglich, sich schnell an die sich änder-

den Anforderungen anzupassen und aufgrund guter Daten schnelle und fundierte Entscheidungen zu treffen, was einen großen Wettbewerbsvorteil ausmacht. Eine gut organisierte Datenbank schafft Transparenz und kann einem Unternehmen so zu mehr Leistungsfähigkeit verhelfen.

## 1.1 Warum ist Datenbankdesign wichtig?

Stellen Sie sich einmal vor, dass Sie ein Haus bauen möchten. Was ist der erste Schritt, nachdem Sie die Finanzierung für Ihr Bauvorhaben unter Dach und Fach gebracht haben? Natürlich – Sie suchen sich einen fähigen Architekten, der sich zunächst einmal nach Ihren Wünschen erkundigt (wie viele Zimmer, mit oder ohne Swimmingpool, wo kommt das Arbeitszimmer hin usw.) und dann auf Basis dieser Wünsche einen Bauplan für Ihr Traumhaus entwickelt.

Komischerweise scheinen immer noch viele Menschen zu denken, dass das alles für Software-Projekte nicht gelten soll. Ich habe schon einige Projekte gesehen, in denen ein motivierter Mitarbeiter einfach sein Datenbankprogramm gestartet und angefangen hat. In unserem Architekten-Beispiel wäre das genau so, als ob der Architekt Ihrer Wahl, nachdem er erfahren hat, dass Sie ein Haus planen, sagt, »Klasse – ich fahr dann mal eben zum Baumarkt, hol ein paar Ziegelsteine und dann können wir auch schon direkt loslegen!« Ich denke, in diesem Fall werden Sie sich schnell nach einem anderen Architekten umsehen.

Genau wie für ein stabiles Haus, das allen Widrigkeiten seiner Umgebung trotzen soll, ein guter Plan vonnöten ist, der von einem Statiker abgenommen wurde, ist es für eine Datenbank wichtig, dass der eigentlichen Implementierung ein gutes Datenbankdesign vorausgegangen ist. In der Tat sollte der eigentliche Datenbankdesign-Prozess mindestens 80 bis 90% der Datenbankentwicklung ausmachen. Hierbei meine ich die reine Entwicklungszeit der Datenbank, d.h. Struktur der Tabellen, Beziehungen, Einschränkungen etc. Was in dieser Entwicklungszeit nicht berücksichtigt ist, ist die Entwicklungszeit für die Benutzeroberfläche einer Datenbankanwendung (z.B. in einer Hochsprache). Durch cleveres Datenbankdesign (das auch schon im Hinblick auf die zu entwickelnde Anwendung erstellt wurde) lässt sich aber auch die Entwicklungszeit der Datenbankanwendung drastisch verkürzen.

Ist erst einmal ein gutes Datenbankdesign vorhanden, so kann dieses leicht in einem der marktüblichen Datenbanksysteme implementiert werden. Es gibt sogar Programme, wie z.B. Microsoft Visio (um Datenbank Re- und Forward-Engineering mit Visio machen zu können, benötigen Sie die Enterprise-Architect-Version, die bei Visual Studio.NET dabei ist) oder Powerbuilder von Sybase, die es ermöglichen, das Design der Datenbank direkt am Rechner durchzuführen. Nachdem Sie auf diese Art und Weise ein Modell Ihrer Datenbank entwickelt haben, erzeugen diese Programme die Implementierung Ihres Modells (z.B. als SQL-Script) automatisch.

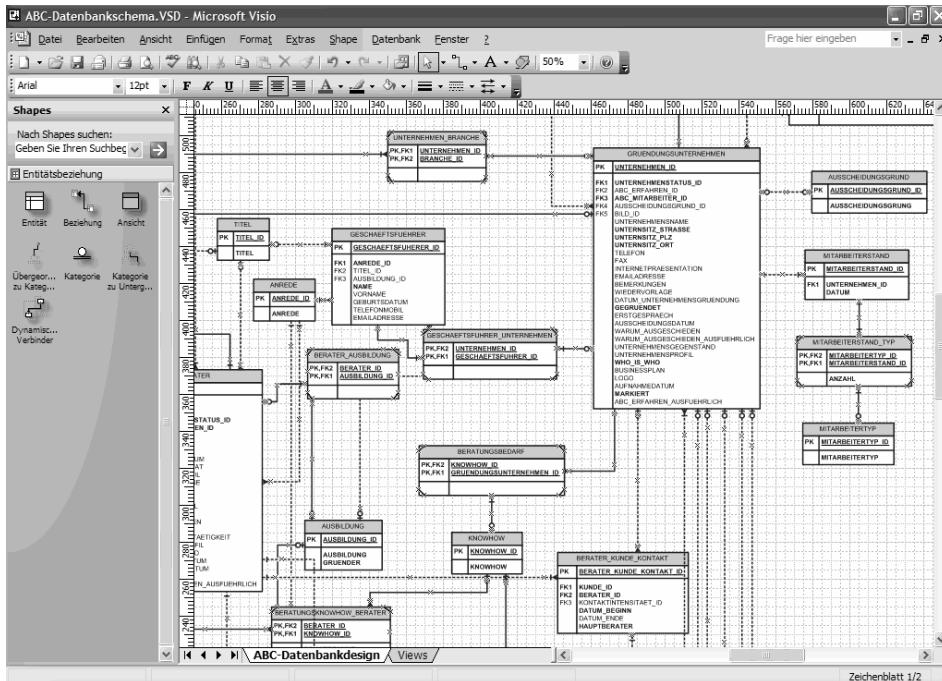


Abb. 1.2: Visio beim Datenbankdesign

Weil das gute Design einer Datenbank einer der zentralen Punkte bei der Erstellung einer Datenbankanwendung ist, beschäftigt sich der größere Teil dieses Buches mit dem Design von Datenbankanwendungen. Das tollste DBMS nützt nichts, wenn Ihr Datenbankdesign schlecht ist.

Weiter oben haben Sie festgestellt, dass Daten für Unternehmen eine wichtige Ressource darstellen, da aus Daten Informationen gewonnen werden können, die wiederum zu Entscheidungen führen. Da die Qualität der Entscheidungen von der Qualität der zugrunde liegenden Informationen und diese wiederum von der Qualität der zugrunde liegenden Daten abhängt, kann nur eine gut entworfene Datenbank die Qualität der in ihr gespeicherten Daten gewährleisten.

Ist das Datenbankdesign schlecht, so können in Ihrer Datenbank *redundante Daten* auftreten. Unter redundanten Daten versteht man Daten, die unnötigerweise mehrfach in der Datenbank vorkommen. Um zu verdeutlichen, warum das zu einem Problem werden kann, stellen Sie sich folgende Situation in dem Beispiel mit der Rechnung vor: Stellen Sie sich vor, dass die Kundendaten in jedem Datensatz der Tabelle gespeichert werden, in dem auch die Rechnungen gespeichert werden (und zwar nur dort). Um im Beispiel zu bleiben, überlegen wir nun, was passiert, wenn ich umziehe. Da meine Kontaktinformationen in jedem Rechnungsdatensatz gespeichert sind, müssen sie auch in jedem Rechnungsdatensatz geändert werden. Da ich ein guter Kunde bin und schon viel bei der Firma gekauft habe,

müssen viele Datensätze geändert werden. Bei der Änderung dieser Datensätze macht der zuständige Mitarbeiter einen Fehler und übersieht einen oder mehrere Datensätze. Arbeitet nun ein anderer Mitarbeiter mit den Daten der Datenbank, wird ihm mit Sicherheit auffallen, dass meine Rechnungen an zwei verschiedene Adressen ausgestellt sind. Da der erste Mitarbeiter (wie üblicherweise in solchen Fällen) nicht verfügbar ist, kann der zweite Mitarbeiter nicht direkt entscheiden, welche der beiden möglichen Adressen die gültige ist. Es beginnt ein aufwändiger Fehlersuchprozess, um meine aktuelle Adresse zu ermitteln.

### Wichtig

Man spricht von redundanten Daten, wenn Daten über ein und dieselbe Entität (eine *Entität* ist ein Objekt der realen Welt, das in der Datenbank verwaltet werden soll, also z.B. eine Person oder eine Rechnung) mehrfach in der Datenbank gespeichert sind. Solche unerwünschten Redundanzen sind das Ergebnis eines schlechten Datenbankdesigns.

Es gibt allerdings auch Fälle, in denen innerhalb von Datenbanken bewusst Datenredundanzen erzeugt werden. Dies ist z.B. bei Datawarehouse-Anwendungen der Fall. Hier wird zusätzliche Geschwindigkeit durch Bereitstellung von redundanten Daten erzeugt. An dieser Stelle ist aber eine unbeabsichtigte Datenredundanz gemeint – und die ist immer schlecht.

Da das Datenbankdesign so wichtig für eine stabile, robuste Datenbank ist, die erweiterbar ist und so auch zukünftigen Anforderungen noch genügt, beschäftigt sich dieses Buch ausführlich mit diesem wichtigen Thema. Wenn Sie in die Tiefen der Implementierung von Datenbankmanagement-Systemen eintauchen möchten, so ist dieses Buch mit Sicherheit nicht das Richtige. Für diesen Fall empfehle ich Ihnen die Bücher von Heuer und Saake, die auch im mitp-Verlag erschienen sind.

Um den nötigen Praxisbezug herzustellen und die in diesem Buch vorgestellten Konzepte zu verdeutlichen, wird der logische Entwurf eines kompletten Beispiels mittlerer Komplexität durchgeführt. Nähere Informationen zu dem Fallbeispiel finden Sie weiter unten im Kapitel unter Abschnitt 1.3, Seite 37.

## 1.2 Dateisystem und Datenbanken

Damit Sie die großen Vereinfachungen und Vorteile verstehen können, die Datenbanksysteme gegenüber einfachen Dateien haben, die im Dateisystem abgespeichert sind, müssen Sie sich zunächst ein wenig mit der Vergangenheit beschäftigen, und sich ansehen, welche Probleme es damals gab.

Die Vorteile einer Datenbank gegenüber dem Dateisystem sind in der Tat so groß, dass Microsoft momentan überlegt, das Dateisystem von Windows Longhorn komplett durch eine Datenbank zu ersetzen. WinFS, das neue Dateisystem, ist im Prin-

zip nichts anderes als ein im Betriebssystem integrierter SQL-Server. Unter WinFS wird der Anwender die Möglichkeit haben, herkömmliche Dateien, relationale Daten und XML-Dateien zu verknüpfen. Es ist sogar geplant, dass Daten, die in WinFS abgelegt werden, über mehrere Rechner synchronisiert werden können. Nähere Informationen zu WinFS finden Sie unter <http://msdn.microsoft.com/Longhorn/understanding/pillars/WinFS/default.aspx>.

Erst wenn Sie diese Vorteile verstanden haben, werden Sie sehen, warum die Entwicklung zu den Datenbanksystemen geführt hat, die wir heute kennen und schätzen gelernt haben. Die Informationen, die Sie in diesem Abschnitt erhalten, sind auch wichtig, wenn Sie planen, eine bestehende Anwendung, die ihre Daten im Dateisystem ablegt, in eine Anwendung zu transformieren, die die Daten in einer Datenbank speichert.

### 1.2.1 Historische Wurzeln

Die Ablage von Dateien in einem Dateisystem ist sehr ähnlich zu der Art, wie wir Daten aufbewahren würden, wenn es gar keine Computer gäbe. Stellen Sie sich vor, dass Sie einen gewaltigen Berg an Schriftstücken haben, die geordnet und sortiert werden müssen, damit man ein bestimmtes Schriftstück, das man gerade benötigt, schnell wieder finden kann. Eine mögliche Art, dem unausweichlichen Chaos Paroli zu bieten, das bei der Ablage all dieser Schriftstücke auf Ihrem Schreibtisch entstünde, ist es, sich einen Schrank mit einer Hängeregistratur zu kaufen. Sie können die verschiedenen Schubladen des Schrankes unterschiedlich beschriften, z.B. »Rechnungen« und »Schriftverkehr«. In die »Rechnungen«-Schublade können Sie dann einfach Hängeordner hängen, die mit den Namen Ihrer Kunden beschriftet sind. In diesen Hängeordnern befinden sich dann alle Rechnungen zum jeweiligen Kunden. Suchen Sie nun eine bestimmte Rechnung zu einem bestimmten Kunden, so müssen Sie lediglich die Schublade aufziehen, in der die Hängeordner mit Rechnungen hängen, die Akte des gewünschten Kunden herausuchen und dann innerhalb der Akte nach der gewünschten Rechnung suchen. Das ist recht einfach. Komplizierter wird es, wenn Sie nun alle Rechnungen herausuchen möchten, die ein bestimmter Sachbearbeiter verfasst hat. Die Namen der Sachbearbeiter stehen zwar jeweils auf den einzelnen Rechnungen, da es aber kein Ordnungskriterium gibt, das die Rechnungen den Sachbearbeitern zuweist, bleibt Ihnen in diesem Fall nichts anderes übrig, als alle Rechnungen aus der Hängeregistratur zu nehmen und einzeln daraufhin zu untersuchen, welcher Sachbearbeiter sich mit der Rechnung befasst hat. Dann müssen Sie die Rechnungen, die der gewünschte Sachbearbeiter bearbeitet hat, an die Seite legen. Eine ziemlich mühsame Aufgabe, wenn Sie mich fragen.

Die Organisation von Dateien im Dateisystem eines Rechners ist diesem Beispiel aus der realen Welt ziemlich stark nachempfunden. Hier werden einzelne, elektronische Dateien in Verzeichnissen gespeichert, die selbst wiederum in anderen Ver-

zeichnissen gespeichert sein können. Einer der auffälligsten Unterschiede zur klassischen Hängeregistratur besteht darin, dass die Schachtelungstiefe der Verzeichnisse beliebig ist.

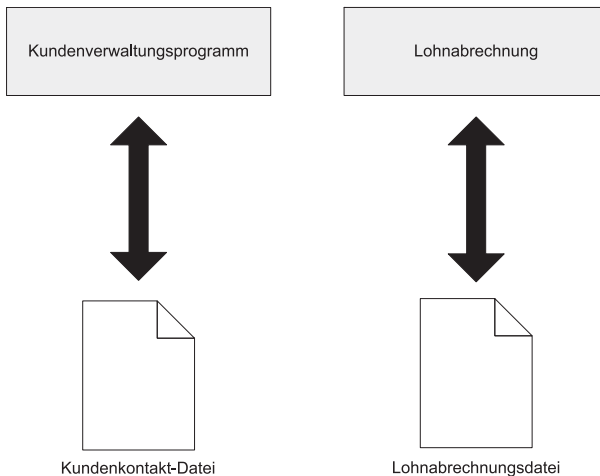
**Hinweis**

In der Realität ist die Schachtelungstiefe von Verzeichnissen nicht unbedingt beliebig – das hängt vom Betriebssystem des Rechners ab bzw. vom verwendeten Dateisystem. Gängige Dateisysteme haben im direkten Vergleich zu einer Hängeregistratur allerdings eine so tiefe Schachtelungstiefe, dass man getrost von einer beliebigen Schachtelungstiefe sprechen kann.

Die Speicherung von Daten in einer derartigen hierarchischen Struktur im Dateisystem ist legitim, so lange die Anzahl der zu verwaltenden Daten gering ist. Sobald allerdings die Datenmengen und die Anforderungen an die Verknüpfung der Daten untereinander steigen, ist die Ablage von Daten in einem hierarchischen Dateisystem ineffizient.

**1.2.2 Probleme bei der Datenhaltung im Dateisystem**

Zunächst haben Computer, die für wirtschaftliche Zwecke eingesetzt wurden, Daten, die z.B. finanzmathematische Programme benötigten, in einzelnen Dateien abgespeichert. Jedes Programm hatte seine eigenen Dateien, es gab allerdings Überschneidungen in den abgespeicherten Daten, da für verschiedene Zwecke dieselben Informationen benötigt werden. Sehen Sie sich hierzu Abbildung 1.3 an.



**Abb. 1.3:** Unterschiedliche Programme greifen auf unterschiedliche Dateien zu.

Stellen Sie sich vor, dass Sie in einer Beratungsfirma arbeiten, die dafür bezahlt wird, dass die angestellten Berater Firmenkunden in verschiedenen Bereichen des Geschäftslebens beraten (dieses Beispiel greift schon auf das Fallbeispiel vor). Die Beratungsfirma besitzt zwei Programme, einerseits ein Kundenverwaltungsprogramm (in Neu-Deutsch heißt das Customer-Relationship-Management oder kurz CRM), das Kontakte der Berater zu den Kunden verwaltet, auf der anderen Seite ein Lohnabrechnungsprogramm, das berechnet, wie viel Lohn der jeweilige Berater für seine Leistungen bekommt. Wie Sie sich sicherlich schon selbst überlegt haben, werden in beiden Dateien, sowohl in der Datei für das Kundenverwaltungsprogramm als auch in der Lohnabrechnungsdatei, zumindest die Namen der Berater doppelt vorkommen. Einerseits möchte man mit dem Kundenverwaltungsprogramm festhalten, welcher Berater welchen Kunden beraten hat, andererseits möchte man mit der Lohnabrechnung die Löhne der Berater berechnen, daher braucht man in beiden Programmen wenigstens die Namen der Berater. Wenn nun ein Berater seinen Namen ändert (z.B. durch Heirat), so muss diese Namensänderung im Beispiel an zwei Stellen durchgeführt werden. (Unser Beispiel ist natürlich sehr vereinfacht. Ein großes Beratungsunternehmen besitzt sicherlich mehr als zwei Programme, mit denen die Aktivitäten des Unternehmens und auch der für das Unternehmen tätigen Berater festgehalten werden. Sie können also davon ausgehen, dass ein Beraternamen an vielen Stellen geändert werden muss.) Dies kann besonders dann problematisch sein, wenn die beiden Programme unter der Hoheit zweier verschiedener Abteilungen stehen. Es müssen zwei Sachbearbeiter davon in Kenntnis gesetzt werden, dass sich der Name geändert hat, und diese beiden Sachbearbeiter müssen die Änderung dann auch in die jeweiligen Programme einpflegen. In der Hektik des Tagesgeschäfts kann so eine kleine Änderung schon mal schnell übersehen werden, so dass für eine Person plötzlich zwei verschiedene Namen existieren. Die Situation wird besonders dann bedenklich, wenn die Änderung schon etwas länger zurückliegt und sich niemand daran erinnern kann, wie denn nun der richtige Name des Mitarbeiters ist. In diesem Fall muss ein umständlicher und unter Umständen auch teurer Fehlerbereinigungsprozess durchgeführt werden.

In unserem Beispiel ist es natürlich einfach, den Fehler zu eliminieren. Man muss lediglich den Berater fragen, wie sein richtiger Name denn nun lautet. Viel kritischer und schwieriger wird die Fehlerbereinigung dann, wenn festgestellt wird, dass z.B. zwei oder drei von Millionen von Messwerten, die aufgezeichnet wurden, in den beiden Programmen, die sie verarbeiten, nicht übereinstimmen. Hier wird es richtig schwierig, den richtigen Messwert im Nachhinein zu ermitteln.

Es lassen sich sicherlich noch weitere Überschneidungen finden. Was passiert zum Beispiel, wenn die Berater auf Stundenbasis bezahlt werden? Im schlechtesten Fall (da keines der beiden Programme auf die Datei des anderen Programms zugreifen kann) muss ein Sachbearbeiter zuerst die in einem Monat angefallenen Stunden mit dem Kundenverwaltungsprogramm ausdrucken, um sie dann in das Lohnabrechnungsprogramm per Hand wieder einzugeben. Dies ist ein mühsamer und fehleranfälliger Prozess.

Bevor Sie sich weiter mit den Problemen beschäftigen, die die Datenhaltung in proprietären Dateien so mit sich bringt, lassen Sie uns einfach einmal einen Blick in eine solche proprietäre Datei werfen und anhand dieser Datei im Vorfeld schon einige wichtige Begriffe klären, die in Zusammenhang mit der Datenhaltung im Allgemeinen stehen. Diese Begriffe werden in Kapitel 3 wieder aufgegriffen und ausführlich am Konzept der relationalen Datenbank erklärt. Die Datei, die dem Kundenverwaltungsprogramm zugrunde liegt, könnte z.B. so aussehen, wie in Abbildung 1.4 gezeigt.

KUNDE	KUNDE_TELEFON	KUNDE_ADRESSE	KUNDE_PLZ	BERATER	KONTAKTDATUM	DAUER	STDSATZ	BERATER_KNOWHOW
Emil Schmidt	0231-1020449	Kaiserstrasse 5, Musterhausen	12345	Helena Meier	21.05.2004	1 Stunde	50,00 €	Finanzierung, IT
Hans Müller	0221-2415932	Am Weiher 3, Musterhausen	12345	Ingo Fuchs	19.04.2003	4 Stunden	45,00 €	Marketing, Strategie
Johanna Schulze	0410-1241221	Alte Poststr. 5, Musterhausen	12345	Helena Meier	04.10.2004	1 Stunde	50,00 €	Finanzierung, IT
Markus Schulte	04514-123414	Goethestr. 7, Musterburg	12354	Ingo Fuchs	18.07.2003	1 Stunde	45,00 €	Marketing, Strategie
Hans Müller	0221-2415932	Am Weiher 3, Musterhausen	12345	Helena Meier	17.04.2003	2 Stunden	50,00 €	Finanzierung, IT
Johanna Schulze	0410-1241221	Alte Poststr. 5, Musterhausen	12345	Helena Meier	04.10.2004	3 Stunden	50,00 €	Finanzierung, IT
Hans Müller	0221-2415932	Am Weiher 3, Musterhausen	12345	Ingo Fuchs	16.04.2004	1 Stunde	45,00 €	Marketing, Strategie

Abb. 1.4: Inhalt der Datei des Kundenverwaltungsprogramms

Als *Daten* bezeichnen wir alle Fakten, die in dieser Datei gespeichert sind. Beispiele für Daten sind z.B. bestimmte Telefonnummern oder Postleitzahlen. Daten selbst besitzen einen geringen Informationsgehalt. Damit Sie aus den Daten sinnvolle Informationen gewinnen können, müssen Sie sie in einen Zusammenhang bringen.

Wenn Sie sich die Tabellendarstellung der Datei des Kundenverwaltungsprogramms in Abbildung 1.4 genau ansehen, so stellen Sie fest, dass in einer Spalte stets dieselben Informationen zu finden sind. In der Spalte KUNDE\_TELEFON z.B. befinden sich die Telefonnummern der einzelnen Kunden. Eine benannte Einheit, die immer dieselben Daten aufnimmt, wird im Datenbankjargon als *Feld* bezeichnet. Im Prinzip kann man sagen, dass ein Feld eine Eigenschaft einer Entität darstellt, die in der Datenbank verwaltet wird. Beispiele für Felder in Abbildung 1.4 sind KUNDE, KUNDE\_TELEFON, KUNDE\_ADRESSE usw.

Unter einem *Datensatz* versteht man eine Sammlung von verknüpften Feldern, die Daten über ein Ding des täglichen Lebens, wie z.B. eine Person oder einen Gegenstand, enthalten. In unserem Beispiel enthält ein Datensatz Daten über einen bestimmten Kundenkontakt und ist aus den Feldern KUNDE, KUNDE\_TELEFON, KUNDE\_ADRESSE, KUNDE\_PLZ, BERATER, KONTAKTDATUM, DAUER, STDSATZ und BERATER\_KNOWHOW aufgebaut. In der tabellarischen Darstellung in Abbildung 1.4 entspricht ein Datensatz einer Zeile.

Als *Datei* bezeichnet man eine Menge von Datensätzen, die zusammengehören. In unserem Beispiel bilden alle Datensätze in Abbildung 1.4 die Datei für das Kundenverwaltungsprogramm. Bitte beachten Sie, dass verschiedene Dateien nicht unbe-

dingt unterschiedlich aufgebaut sein müssen. Es ist durchaus erlaubt (und auch üblich), dass verschiedene Dateien denselben Aufbau besitzen. So ist es auch denkbar, dass man die Dateien des Kundenverwaltungsprogramms nach Beratern unterteilen kann. In diesem Fall hätten Sie zwei Dateien desselben Aufbaus, wobei die eine Datei nur Datensätze enthält, die Ingo Fuchs zugeordnet sind, und die andere Datei enthält nur Datensätze, die Helena Meier zugeordnet sind.

Eine Datei, wie sie in Abbildung 1.4 zu sehen ist, wird auch oft als *flache Datei* bezeichnet, da sie ein Minimum an Struktur besitzt. Ich habe die Tabellenform der Abbildung 1.4 lediglich aus Gründen der Übersichtlichkeit gewählt. Eine beliebige Form, in der flache Dateien gespeichert werden, ist das *CSV-Format* (*Comma Separated File*). Hierbei werden die Felder entweder, wie in Listing 1.1 zu sehen ist, mit Hilfe eines Trennzeichens (in diesem Fall das Semikolon) getrennt, oder es wird für jedes Feld eine bestimmte Zahl an Zeichen definiert. Jeder Datensatz beginnt in einer neuen Zeile.

```
Emil Schmidt;0231-1020449;Kaiserstrasse 5, Musterhausen;12345;Helena  
Meier;...  
Hans Müller;0221-2415932;Am Weiher 3, Musterhausen;12345;Ingo Fuchs;...  
Johanna Schulze;0410-1241221;Alte Poststr. 5, Musterhausen;12345;Helena  
Meier;...  
Markus Schulte;04514-123414;Goethestr. 7, Musterburg;12354;Ingo Fuchs;...  
Hans Müller;0221-2415932;Am Weiher 3, Musterhausen;12345;Helena Meier;...  
Johanna Schulze;0410-1241221;Alte Poststr. 5, Musterhausen;12345;Helena  
Meier;... Hans Müller;0221-2415932;Am Weiher 3, Musterhausen;12345;Ingo  
Fuchs;...
```

**Listing 1.1:** Beispiel für eine flache Datei

Wie Sie sehen können, enthält die in Listing 1.1 dargestellte flache Datei wirklich ein Minimum an Metainformationen. Die einzigen enthaltenen Metainformationen sind die Semikola, die die Daten der einzelnen Felder trennen. Es ist weder eine Information darüber vorhanden, welches Feld welche Bedeutung hat, noch welcher Datentyp verwendet wird. Wenn Ihnen allein diese Datei und die Information vorliegen, dass die Datei Kundenkontakte zu Beratern Ihres Beratungsunternehmens enthält, können Sie aufgrund dieser Darstellung nicht entscheiden, ob z.B. in der ersten Zeile Emil Schmidt oder Helena Meier der Berater ist (es sei denn, Sie kennen Ihre Kollegen).

Eine Datei, wie sie in Listing 1.1 zu sehen ist, bringt viele Probleme mit sich, obwohl in der Anfangszeit des Informationszeitalters 20 Jahre lang mit solchen Dateien gearbeitet worden ist.

Wenn Sie auf eine solche Datei mit Hilfe einer Hochsprache, wie z.B. C++, Basic oder Pascal zugreifen möchten, müssen Sie komplexe Funktionen und Prozeduren

schreiben, die die Daten von der Festplatte laden, ändern und speichern können. Da Sie in diesem Fall in der Hochsprache nicht nur definieren müssen, welche Daten von der Festplatte geladen werden sollen, sondern auch, wie das geschehen soll, kann die Verwaltung flacher Dateien (insbesondere in einem komplexen System, das viele flache Dateien besitzt) sehr komplex werden. Da die Struktur der verschiedenen Dateien unterschiedlich ist – die Datei des Kundenverwaltungsprogramms ist mit Sicherheit anders als die Datei des Lohnabrechnungsprogramms –, muss für jede Datei eine eigene Dateiverwaltung programmiert werden, die folgende Aufgaben erfüllen kann:

- Datei anlegen
- Daten zur Datei hinzufügen
- Daten aus der Datei löschen
- Daten in der Datei ändern
- Daten aus der Datei laden

Aufgrund der Abhängigkeit der Dateiverwaltung von der Struktur der zugrunde liegenden Dateien ist es in einem solchen System nicht möglich, Ad-hoc-Abfragen durchzuführen, daher mussten Programme geschrieben werden, die bestimmte Berichtsanforderungen erfüllen konnten. Das Schreiben dieser Programme war natürlich auch aufwändig. Je nach Anforderungen konnte es sein, dass es eine Woche oder einen Monat dauerte, bevor ein Programm erstellt war, das einen Bericht ausgeben konnte, der einen bestimmten Sachverhalt dargestellt hat. Ganz am Anfang dieses Kapitels habe ich erwähnt, dass man, um gute Entscheidungen treffen zu können, gute Informationen benötigt. Diese guten Informationen benötigt man natürlich auch zeitnah, da die Entscheidungen natürlich ziemlich schnell getroffen werden müssen. Wenn es aber eine Woche oder einen Monat lang dauert, die notwendigen Informationen zu besorgen, so ist das Datenbanksystem äußerst ineffizient, da sich die Situation, für die die Informationen benötigt wurden, unter Umständen schon wieder verändert hat, so dass die gelieferten Informationen keinen Wert mehr besitzen, weil sie viel zu spät zur Verfügung stehen. Je mehr Daten in flachen Dateien verwaltet werden und je größer der Bedarf an bestimmten Berichten oder Auswertungen ist, desto komplexer und schwieriger wird es, ein auf flachen Dateien basierendes System zu verwalten und zu steuern. Dass die Fehleranfälligkeit eines solchen Systems mit der Komplexität steigt, versteht sich von selbst.

Ein anderer wichtiger Punkt, den Sie bei der Betrachtung der flachen Dateien nicht aus den Augen lassen dürfen, ist das Verhalten eines auf flachen Dateien basierenden Systems bei Änderungen an der Struktur der Dateien. Sehen Sie sich bitte noch einmal die Struktur der Datei unseres Kundenverwaltungsprogramms an (siehe Abbildung 1.5).

	KUNDE	KUNDE_TELEFON	KUNDE_ADRESSE	KUNDE_PLZ	BERATER	KONTAKTDATUM	DAUER	STDSATZ	BERATER_KNOWLEDGE
	Emil Schmidt	0231-1020449	Kaiserstrasse 5, Musterhausen	12345	Helena Meier	21.05.2004	1 Stunde	50,00 €	Finanzierung, IT
	Hans Müller	0221-2415932	Am Weiher 3, Musterhausen	12345	Ingo Fuchs	19.04.2003	4 Stunden	45,00 €	Marketing, Strategie
	Johanna Schulze	0410-1241221	Alte Poststr. 5, Musterhausen	12345	Helena Meier	04.10.2004	1 Stunde	50,00 €	Finanzierung, IT
	Markus Schulte	04514-123414	Goethestr. 7, Musterburg	12354	Ingo Fuchs	18.07.2003	1 Stunde	45,00 €	Marketing, Strategie
	Hans Müller	0221-2415932	Am Weiher 3, Musterhausen	12345	Helena Meier	17.04.2003	2 Stunden	50,00 €	Finanzierung, IT
	Johanna Schulze	0410-1241221	Alte Poststr. 5, Musterhausen	12345	Helena Meier	04.10.2004	3 Stunden	50,00 €	Finanzierung, IT
▶	Hans Müller	0221-2415932	Am Weiher 3, Musterhausen	12345	Ingo Fuchs	16.04.2004	1 Stunde	45,00 €	Marketing, Strategie

Abb. 1.5: Struktur der Kundenverwaltungsdatei

Wenn Sie sich das Feld KUNDE\_ADRESSE ansehen, sehen Sie, dass hier zwei Informationen enthalten sind, die Straße, in der der Kunde wohnt, und die Stadt. Da Sie gerne eine Auswertung der Daten bezogen auf die Stadt hätten, wäre es sinnvoll, die Straße und den Ort in verschiedenen Feldern zu speichern. Natürlich könnten Sie die Postleitzahl als Kriterium verwenden, da diese in einem Feld gespeichert ist. Das ist aber gerade bei größeren Städten problematisch, da diese über mehrere Postleitzahlen verfügen. Würden Sie die Postleitzahl als Kriterium nehmen, so könnten Sie die Kunden nach Postleitzahlen auswerten, was aber der ursprünglichen Frage, der Auswertung nach Städten, nicht hundertprozentig entspricht. In diesem Fall müssen Sie die Struktur Ihrer flachen Datei ändern, was zu Problemen führen kann, da sämtliche Programme, die auf diese Datei zugreifen, auch geändert werden müssen.

Lassen Sie uns einmal untersuchen, was alles gemacht werden muss, um die angesprochene simple Änderung des Feldes KUNDE\_ADRESSE durchzuführen. Zunächst einmal müssten wir ein Programm schreiben, das die Daten aus der alten Form in die neue Form konvertieren kann. Dieses Programm müsste die neue Dateistruktur auf Festplatte anlegen, Datensatz für Datensatz aus der alten Dateistruktur lesen, die Daten in die neue Dateistruktur transformieren (in unserem Fall das Feld KUNDEN\_ADRESSE beim Komma auftrennen, den ersten Teil in das neue Feld STRASSE und den Rest in das neue Feld ORT schreiben) und die neuen Datensätze dann in die neue Dateistruktur auf Festplatte schreiben. Auch wenn diese Arbeit recht aufwändig ist, ist sie sicherlich machbar. Was allerdings problematischer ist, ist der zweite Teil der Datenumstellung. Nachdem wir die Daten von der alten in die neue Struktur transformiert haben, müssen wir nun alle Programme, die mit der Datei arbeiten (also sowohl Benutzerprogramme, mit denen neue Daten zur Datei hinzugefügt oder bestehende Daten verändert werden können, als auch sämtliche Berichtsprogramme), aufspüren und umprogrammieren, so dass sie mit der neuen Dateistruktur arbeiten können. In einem großen Unternehmen kann dies zu einem sehr zeitaufwändigen und mühsamen Prozess werden. Bedenken Sie bitte, dass dieser Prozess für jede noch so kleine Änderung an der Datenstruktur durchgeführt werden muss. Anwendungsprogramme, die eine solche Abhängigkeit von der Struktur der zugrunde liegenden Dateien aufweisen, nennt man auch *strukturell abhängig* – der Zugriff auf eine Datei hängt von ihrer Struktur ab.

Eine andere wichtige Abhängigkeit, die Anwendungsprogramme aufweisen, ist die *Datenabhängigkeit*. Unter Datenabhängigkeit versteht man die Abhängigkeit einer Anwendung von der physikalischen Darstellung eines bestimmten Feldes. Stellen

Sie sich z.B. vor, dass Sie eine bestimmte Information als Integer-Wert (also als Ganzzahl ohne Nachkommastellen) abgespeichert haben. Wenn Sie mit Ihrem Programm auf diese Information zugreifen möchten, müssen Sie stets den vorgegebenen Datentyp und das physikalische Format, in dem die Daten gespeichert sind, beachten und können auf den Integer-Wert immer nur als Integer zugreifen. Nach einiger Zeit stellen Sie fest, dass die in diesem Feld gespeicherten Daten auch Nachkommastellen haben können. Sie müssen die physikalische Darstellung der Zahl ändern, so dass auch Nachkommastellen gespeichert werden können. In diesem Fall müssen Sie alle Anwendungsprogramme umprogrammieren, damit sie mit dem neuen Datentyp umgehen können.

Sie werden mir jetzt sicherlich entgegnen wollen, dass die oben geschilderte Datenabhängigkeit für Daten, die in einer Textdatei, wie wir sie bisher betrachtet haben, gespeichert sind, wohl nicht zutreffend sein kann, da sämtliche Daten als Text (also im Prinzip als Strings) gespeichert sind und dass diese nach dem Einlesen ohnehin in den notwendigen Datentyp, also z.B. eine Zahl, umgewandelt werden müssen. Ich stimme Ihnen voll und ganz zu. Dies ist sicherlich im Fall einer Textdatei richtig, aber denken Sie nun einmal daran, wie Dateien in einer Binär-Datei gespeichert werden können. Es gibt verschiedene Formate, in denen man z.B. Zahlen darstellen kann (Festkomma-Darstellung oder mit Hilfe von Exponent und Matisse). Man unterscheidet hier zwischen dem *logischen Format* der Daten, also dem Format, in dem Menschen die Daten verstehen, und dem *physikalischen Format* der Daten, das ist das Format, in dem die Daten vom Computer gespeichert werden. Hierbei ist zu beachten, dass ein logisches Format, z.B. Zahlen, mehrere physikalische Darstellungen haben kann und dass diese unter Umständen auch vom verwendeten Computer bzw. vom verwendeten Betriebssystem abhängen können.

Ein weiteres wichtiges Problem, das Systeme, die auf flachen Dateien aufbauen, nur unzureichend lösen, ist die Implementierung eines sicheren Zugriffs auf die Daten. Da flache Dateien oft nichts anderes als CSV-Dateien, also im Prinzip Textdateien, sind, ist es recht einfach, Sicherheitsmechanismen auszuhebeln, die möglicherweise in den Datenbankanwendungen implementiert sind. Eine Textdatei können Sie einfach in einen Text-Editor laden und schon haben Sie Zugriff auf alle Daten, die in der jeweiligen Datei gespeichert sind.

Der Datei-Charakter und die fehlenden Sicherheitsmaßnahmen führen dazu, dass Daten im Unternehmen nicht zentral zur Verfügung stehen, sondern dass die verschiedenen Abteilungen eines Unternehmens damit beginnen, eigene Datenbestände aufzubauen. Da diese Datenbestände voneinander isoliert gehalten werden, spricht man in diesem Zusammenhang auch von *Dateninseln* oder *Informationsinseln*. In einem solchen Szenario ist es unwahrscheinlich, dass bei Änderungen sämtliche Instanzen der Daten geändert werden. Daher kommt es hier oft vor, dass es verschiedene Versionen derselben Daten gibt, wie z.B. in unserem Beispiel oben, in dem sich ein Name geändert hat. Sie können sich sicherlich vorstellen, wie eine solche Struktur in einem größeren Unternehmen leicht außer Kontrolle geraten kann.

### 1.2.3 Datenredundanzen und Anomalien

Sie haben im vorherigen Abschnitt gesehen, dass die Verwaltung von Daten in Dateien dazu führt, dass es zur Bildung von Dateninseln kommt und dass dieselben Daten an verschiedenen Orten abgespeichert werden. Wenn Daten, die dieselbe Information über eine Person oder einen Gegenstand speichern, an verschiedenen Orten gehalten werden, spricht man von *Datenredundanz*. Datenredundanz führt zu verschiedenen Problemen, die ich in diesem Abschnitt näher beleuchten werde.

Eines der größten Probleme, das durch Datenredundanz verursacht wird und auf das Sie im Verlauf dieses Kapitels schon des Öfteren gestoßen sind, ist die *Dateninkonsistenz*. Man spricht von Dateninkonsistenz, wenn verschiedene Versionen derselben Daten existieren und diese verschiedenen Versionen im Konflikt miteinander stehen, das heißt, es ist nicht ohne weiteres möglich, anhand der Daten zu entscheiden, welches die aktuellste bzw. gültige Version der Daten ist. Stellen Sie sich vor, dass Sie in der Kundenverwaltungsdatei die Telefonnummer der Kundin Johanna Schulz ändern. In der Beispieldatei ist diese Telefonnummer in zwei Datensätzen vorhanden, Sie ändern diese aber lediglich in einem Datensatz, so wie in Abbildung 1.6 zu sehen ist.

**Dateninkonsistenz**

	KUNDE	KUNDE_TELEFON	KUNDE_ADRESSE	KUNDE_PLZ	BERATER	KONTAKTDATUM	DAUER	STDSATZ	BERATER_KNOWLEDGE
	Emil Schmidt	0231-1020449	Kaiserstrasse 5, Musterhausen	12345	Helena Meier	21.05.2004	1 Stunde	50,00 €	Finanzierung, IT
	Hans Müller	0221-2415932	Am Weiher 3, Musterhausen	12345	Ingo Fuchs	19.04.2003	4 Stunden	45,00 €	Marketing, Strategie
	Johanna Schulze	0410-1241335	Alte Poststr. 5, Musterhausen	12345	Helena Meier	04.10.2004	1 Stunde	50,00 €	Finanzierung, IT
	Markus Schulte	04514-123414	Goethestr. 7, Musterburg	12354	Ingo Fuchs	18.07.2003	1 Stunde	45,00 €	Marketing, Strategie
	Hans Müller	0221-2415932	Am Weiher 3, Musterhausen	12345	Helena Meier	17.04.2003	2 Stunden	50,00 €	Finanzierung, IT
	Johanna Schulze	0410-1241221	Alte Poststr. 5, Musterhausen	12345	Helena Meier	04.10.2004	3 Stunden	50,00 €	Finanzierung, IT
	Hans Müller	0221-2415932	Am Weiher 3, Musterhausen	12345	Ingo Fuchs	16.04.2004	1 Stunde	45,00 €	Marketing, Strategie

**Abb. 1.6:** Datenredundanz verursacht Dateninkonsistenzen.

Fragen Sie nun Datensätze aus der Datei ab, erhalten Sie unterschiedliche Informationen über die Telefonnummer der Kundin Johanna Schulz, je nachdem, ob Sie den dritten oder den sechsten Datensatz abfragen. Ihren Daten fehlt *Datenintegrität*.

Dateninkonsistenzen treten nicht nur bei der Änderung von Daten auf, es kann auch vorkommen, dass Dateninkonsistenzen bei der Erfassung von Daten auftreten. Stellen Sie sich einmal vor, was passiert, wenn eine überlastete Sekretärin komplexe Daten (z.B. lange Telefonnummern) in mehreren verschiedenen Dateien erfassen muss, da jedes der verwendeten Programme seinen eigenen Datenbestand besitzt. Fehler bei der Eingabe sind hier unvermeidbar.

Dateninkonsistenzen sind insbesondere dann problematisch, wenn sie nach außen getragen werden und das Unternehmen verlassen. So kann es z.B. sein, dass Sie einem neuen Berater die Telefonnummer der Kundin Johanna Schulze geben (die natürlich falsch ist) und dieser die Kundin nicht erreichen kann.

Die durch Dateninkonsistenzen auftretenden Fehler nennt man auch *Datenanomalien*. Ändert sich der Wert eines Feldes, so sollte diese Änderung an einer einzigen Stelle durchgeführt werden. Sie haben in dem Beispiel oben schon gesehen, was passieren kann, wenn Daten mehrfach, also redundant gespeichert werden. Datenanomalien unterscheidet man in drei verschiedene Kategorien:

■ *Änderungs-Anomalie (Update-Anomalie)*

Die Änderungs-Anomalie haben Sie ja bereits kennen gelernt. Werden dieselben Daten an verschiedenen Stellen gespeichert, so muss gewährleistet sein, dass die Daten an allen Stellen, an denen sie gespeichert werden, auch geändert werden. In unserem Beispiel, in dem sich die Telefonnummer der Kundin Johanna Schulze ändert, muss diese Änderung nicht nur im dritten, sondern auch im sechsten Datensatz durchgeführt werden. Stellen Sie sich das Computersystem eines großen Unternehmens vor, das auf Dateien aufgebaut ist. Hier kann es vorkommen, dass dieselben Daten an über hundert verschiedenen Orten gespeichert sind. In diesem Fall ist es sehr leicht möglich, dass bei der Änderung der Daten eine Stelle übersehen wird.

■ *Einfüge-Anomalie (Insert-Anomalie)*

Eine Einfüge-Anomalie liegt dann vor, wenn man bestimmte Daten nicht erfassen kann, ohne andere Daten gleichzeitig erfassen zu müssen. In unserer Beispiel-Datei ist es z.B. nicht möglich, einen Kunden zu erfassen, ohne diesem direkt einen Berater zuzuweisen. Selbst wenn Sie die Entscheidung, welcher Berater welchen Kunden berät, erst später treffen möchten, müssen Sie trotz allem einen Berater zuweisen. Werden viele Kunden erfasst, ist auch hier wieder die Gefahr groß, dass Dateninkonsistenzen entstehen.

■ *Lösch-Anomalie (Delete-Anomalie)*

Eine Lösch-Anomalie tritt dann auf, wenn das Löschen bestimmter Daten verursacht, dass andere Daten, die eigentlich nicht gelöscht werden sollten, durch diesen Löschvorgang auch gelöscht werden. In der Beispiel-Datei zeigt sich dieses Problem, wenn Sie alle Datensätze löschen, in denen der Kunde Hans Müller vorkommt (also den zweiten, fünften und siebten Datensatz). Diesem Kunden ist der Berater Ingo Fuchs zugeordnet. Wenn Sie davon ausgehen, dass die Daten über Ingo Fuchs einzig und allein in dieser Datei stehen, so wird durch das Löschen aller Datensätze von Hans Müller automatisch jede Information über den Berater Ingo Fuchs gelöscht.

Datenanomalien treten nicht nur bei der dateibasierten Datenhaltung auf. Sie können auch in schlecht entworfenen Datenbanksystemen auftreten. Wie oben beschrieben, sind Datenanomalien das Ergebnis von Datenredundanz. Das Ziel beim Entwurf von Datenbanksystemen sollte es also sein, Datenredundanzen zu vermeiden und so das Auftreten von Datenanomalien zu verhindern.

## 1.3 Das Fallbeispiel

Im gesamten Buch wird ein durchgängiges Fallbeispiel verwendet, um die Prinzipien des Datenbankdesigns vorzustellen. Beim Fallbeispiel geht es darum, die im Verlauf des Buches erarbeiteten theoretischen Inhalte an einem greifbaren Beispiel aus der Praxis zu verdeutlichen. Sie sollen so in die Lage versetzt werden, die vermittelten Inhalte schnell in der Praxis anwenden zu können.

Lassen Sie uns nun einen Blick auf das Fallbeispiel werfen, das Unternehmen Alana Business Consult (ABC). Das Hauptgeschäft des Unternehmens Alana Business Consult ist es, anderen Firmen Berater zur Verfügung zu stellen. Diese Berater kommen aus verschiedenen Branchen und besitzen die unterschiedlichsten Qualifikationen. Die zu beratenden Firmen auf der anderen Seite haben individuelle Anforderungen, Probleme und Fragestellungen, die von den Beratern gelöst werden sollen. Wie Sie sich sicherlich vorstellen können, ist es immer eine herausfordernde Aufgabe, Berater und Firmen zusammenzubringen. Je nach Komplexität der Fragestellung werden einzelne Berater oder individuell zusammengestellte Beraterteams eingesetzt, um die in der jeweiligen Firma vorhandenen Schwierigkeiten zu lösen. Alle Berater sind fest bei Alana Business Consult eingestellt. Da niemand alles wissen kann und manche Fragestellungen über das Wissen einzelner Berater und sogar das kollektive Wissen des Unternehmens hinausgehen, arbeitet Alana Business Consult zusätzlich mit so genannten Experten zusammen. Diese Experten sind nicht fest bei ABC angestellt, es handelt sich vielmehr um freie Mitarbeiter oder Angestellte anderer Firmen, die bei besonders kniffligen Problemstellungen hinzugezogen werden können. Ist die Aufgabenstellung gelöst, die das Hinzuziehen eines Experten nötig machte, übernehmen wieder der oder die von ABC zugewiesenen Berater. Neben diesen beiden beratend tätigen Gruppen gibt es auch noch Mitarbeiter, die alle Aktivitäten von ABC verwalten und koordinieren. Jedem Kundenunternehmen ist einer dieser Mitarbeiter zugeordnet und dient dem Kunden als Ansprechpartner. Aufgabe dieser Mitarbeiter ist es, zu entscheiden, welcher Firma welcher Berater zugewiesen wird, ob eine Anfrage im Haus beantwortet werden kann oder ob ein Experte hinzugezogen werden muss und ob gegebenenfalls ein Beraterteam zusammengestellt werden muss.

Bisher wurden interne Abläufe und Prozesse bei Alana Business Consult einerseits mit Papier und Bleistift, andererseits auch mit Office-Programmen wie z.B. Microsoft Word und Excel verwaltet. Der Nachteil diese Methode liegt auf der Hand: Die Daten werden nicht strukturiert abgelegt. Mitarbeiter lassen ihre eigenen Gewohnheiten in die Daten einfließen. Es gibt unzählige Möglichkeiten, eine Telefonnummer abzuspeichern. Mögliche Darstellungsformen sind z.B.

+49 (1234) 123456

01234/123456

(01234) 123456

0049 (0)1234 / 123456

Diese verschiedenen Darstellungsformen machen es sehr schwierig, konsolidierte Datenbestände zu erzeugen. Durch die verteilte Datenhaltung in verschiedenen Formaten ist es bei Alana Business Consult leider notwendig geworden, redundante Datenbestände anzulegen und zu pflegen. Diese Pflege ist mit der Zeit und der größer werdenden Anzahl der betreuten Kunden zunehmend komplexer und zeitaufwändiger geworden. Hierdurch sind die Prozesse bei Alana Business Consult fehleranfällig geworden. Briefe mit falschen Adressen werden abgesendet oder wichtige Kundeninformationen sind nicht verfügbar. Neben diesen Problemen, die im täglichen Geschäft auftreten, ist es sehr aufwändig, auf dem verteilten Datenbestand Analysen und Auswertungen durchzuführen.

Aufgrund dieser Probleme häufen sich die Beschwerden der Kunden in letzter Zeit und auch die Mitarbeiter sind zunehmend unzufrieden mit der aktuellen Situation. Daher hat die Geschäftsführung entschieden, dass die Zeit zum Handeln gekommen ist, und ist an Sie als Datenbankentwickler herangetreten, um die unhaltbare Situation zu entschärfen. Sie sollen eine zentrale Datenbank entwickeln, die als globale (und einzige) Datenbasis für das Unternehmen dienen soll.

Aufgrund des bisher in den vorherigen Abschnitten gewonnenen Wissens entscheiden Sie sich dafür, eine Datenbankanwendung basierend auf einer relationalen Datenbank zu erstellen. Wie Sie nun, ausgehend von der geschilderten Situation und der ersten Idee, zu einer funktionsfähigen Datenbank gelangen, erfahren Sie in diesem Buch.

## 1.4 Zusammenfassung

### ■ Ad-hoc-Abfragen

Unter einer Ad-hoc-Abfrage versteht man eine spontane Abfrage an eine Datenbank. Möchte man spontan eine bestimmte Information haben, so stellt man eine Ad-hoc-Abfrage an die Datenbank. Ein modernes Datenbanksystem muss Ad-hoc-Abfragen unterstützen.

### ■ Änderungs-Anomalie

Werden dieselben Daten an verschiedenen Stellen gespeichert, so muss gewährleistet sein, dass die Daten an allen Stellen, an denen sie gespeichert werden, auch geändert werden. Geschieht dies nicht, so spricht man von einer Änderungs-Anomalie.

### ■ Datei

Als Datei bezeichnet man eine Menge von Datensätzen, die zusammengehören.

## ■ Daten

Als Daten bezeichnen wir alle Fakten, die in einer Datei oder einer Datenbank gespeichert sind. Daten selbst besitzen einen geringen Informationsgehalt. Damit Sie aus den Daten sinnvolle Informationen gewinnen können, müssen Sie sie in einen Zusammenhang bringen. Der Informationsgehalt von Daten hängt vom Zusammenhang ab.

## ■ Datenabhängigkeit

Unter Datenabhängigkeit versteht man die Abhängigkeit einer Anwendung von der physikalischen Darstellung eines bestimmten Feldes. Ist eine Anwendung abhängig von der physikalischen Darstellung der Daten eines Feldes, so muss die Anwendung umprogrammiert werden, sobald sich die physikalische Darstellung der Daten ändert.

## ■ Datenanomalien

Die durch Dateninkonsistenzen auftretenden Fehler nennt man auch Datenanomalien.

## ■ Datenbank

Es gibt zwei Definitionen des Begriffs Datenbank, die erste ist, dass eine Datenbank ein verteiltes, integriertes Computersystem ist, das Nutzdaten und Metadaten enthält. Die zweite Definition besagt, dass eine Datenbank eine geordnete, selbstbeschreibende Sammlung von Daten ist, die miteinander in Beziehung stehen.

## ■ Datenbankdesign

Das gute Design einer Datenbank ist einer der zentralen Punkte bei der Erstellung einer Datenbankanwendung. Ist erst einmal ein gutes Datenbankdesign vorhanden, so kann dieses leicht in einem der marktüblichen Datenbanksysteme implementiert werden.

## ■ Datenbankmanagement-System (DBMS)

Um eine Datenbank auf einem Computer zu verwalten, wird in der Regel ein so genanntes Datenbankmanagement-System (DBMS) verwendet, das sich um die Organisation der Daten kümmert und das den Zugriff auf die Daten regelt. Das Datenbankmanagement-System kann entweder aus einem einzelnen Programm bestehen oder es kann aus vielen Programmen bestehen, die zusammenarbeiten und so die Funktionalität eines DBMS bereitstellen.

## ■ Dateninkonsistenz

Unter Dateninkonsistenz versteht man den Zustand, wenn verschiedene Versionen derselben Daten existieren und diese verschiedenen Versionen im Konflikt miteinander stehen. Anhand der Daten ist es nicht möglich zu entscheiden, welches die aktuellste bzw. gültige Version der Daten ist.

**■ Datenintegrität**

Unter Datenintegrität versteht man, dass sich in der gespeicherten Datenbank Daten in einem konsistenten, widerspruchsfreien Zustand befinden.

**■ Datenmanagement**

Der Umgang mit Daten wird als Datenmanagement bezeichnet. Aufgaben des Datenmanagements sind die Erzeugung, Speicherung und Wiedergabe der Daten.

**■ Datensatz**

Unter einem Datensatz versteht man eine Sammlung von verknüpften Feldern, die Daten über ein Ding des täglichen Lebens, wie z.B. eine Person oder einen Gegenstand enthalten.

**■ Einfüge-Anomalie**

Eine Einfüge-Anomalie liegt dann vor, wenn man bestimmte Daten nicht erfassen kann, ohne andere Daten gleichzeitig erfassen zu müssen.

**■ Entität**

Eine Entität ist ein Objekt der realen Welt, das in der Datenbank verwaltet werden soll, also z.B. eine Person oder ein Gegenstand.

**■ Feld**

Ein Feld stellt eine Eigenschaft einer Entität dar, die in der Datenbank verwaltet wird.

**■ Informationen**

Informationen werden aus Daten durch Datenverarbeitung gewonnen und helfen dabei, gute Entscheidungen zu treffen. Damit anhand von Informationen gute Entscheidungen getroffen werden können, müssen Informationen zeitnah vorliegen.

**■ Informationsinseln**

Beginnen verschiedene Abteilungen eines Unternehmens damit, eigene Datenbestände aufzubauen, die voneinander isoliert gehalten werden, so spricht man in diesem Zusammenhang von Informationsinseln oder Dateninseln.

**■ Informations-Overkill**

Die ständig auf uns einprasselnde Flut von unterschiedlichsten Informationen wird als Informations-Overkill bezeichnet.

**■ Logisches Format der Daten**

Das logische Format der Daten ist das Format, in dem Menschen die Daten verstehen können.

### ■ Lösch-Anomalie

Eine Lösch-Anomalie tritt dann auf, wenn das Löschen bestimmter Daten verursacht, dass andere Daten, die eigentlich nicht gelöscht werden sollten, durch diesen Löschvorgang auch gelöscht werden.

### ■ Metadaten

Metadaten werden oft auch als Daten über Daten bezeichnet und helfen, die Nutzdaten der Datenbank zu strukturieren.

### ■ Nutzdaten

Nutzdaten sind die Daten, die Benutzer in der Datenbank anlegen und aus denen die Informationen gewonnen werden.

### ■ Physikalisches Format der Daten

Das physikalische Format der Daten ist das Format, in dem der Computer die Daten abspeichert.

### ■ Strukturelle Abhängigkeit

Ist eine Anwendung von der Struktur der zugrunde liegenden Daten abhängig, so bezeichnet man dies als strukturelle Abhängigkeit – der Zugriff auf eine Datei hängt von ihrer Struktur ab.

## 1.5 Aufgaben

Hier finden Sie Wiederholungsfragen, mit denen Sie Gelegenheit haben, sich noch einmal Gedanken über den Stoff des Kapitels zu machen. Die Lösungen zu diesen Aufgaben finden Sie in Anhang A.1. Außerdem finden Sie im Abschnitt *Zum Weiterdenken* Probleme und Aufgaben, auf die Sie Ihr frisch gewonnenes Wissen anwenden können. Hierfür werden keine Lösungen bereitgestellt, siehe Anhang A.

### 1.5.1 Wiederholung

1. Beschreiben Sie die Begriffe Daten, Feld, Datensatz und Datei!
2. Was ist ein DBMS und welche Funktion hat es?
3. Was bezeichnet man als Datenredundanz? Welche Probleme bringt die Datenredundanz mit sich?
4. Beschreiben Sie, wie es in einem Unternehmen, das seine Daten im Dateisystem verwaltet, zu Problemen kommen kann.
5. Was genau ist der Unterschied zwischen Daten und Informationen?
6. Warum brauchen wir Datenbanksysteme?
7. Warum besitzen Dateien keine Datenunabhängigkeit?

8. Warum ist Datenbankdesign wichtig?
9. Welche Arten von Datenbanksystemen gibt es? Wo liegen die Unterschiede?
10. Welche Datenanomalien kennen Sie und wie wirken sich diese aus? Geben Sie Beispiele an!

### 1.5.2 Zum Weiterdenken

Die nächsten Aufgaben beziehen sich auf die folgende Tabelle:

KUNDE	KUNDE_TELEFON	KUNDE_ADRESSE	KUNDE_PLZ	BERATER	KONTAKTDATUM	DAUER	STDSATZ	BERATER_KNOWLEDGE
Emil Schmidt	0231-1020449	Kaiserstrasse 5, Musterhausen	12345	Helena Meier	21.05.2004	1 Stunde	50.00 €	Finanzierung, IT
Hans Müller	0221-2415932	Am Weiher 3, Musterhausen	12345	Ingo Fuchs	19.04.2003	4 Stunden	45.00 €	Marketing, Strategie
Johanna Schulze	0410-1241221	Alte Poststr. 5, Musterhausen	12345	Helena Meier	04.10.2004	1 Stunde	50.00 €	Finanzierung, IT
Markus Schulte	04514-123414	Goethestr. 7, Musterburg	12354	Ingo Fuchs	18.07.2003	1 Stunde	45.00 €	Marketing, Strategie
Hans Müller	0221-2415932	Am Weiher 3, Musterhausen	12345	Helena Meier	17.04.2003	2 Stunden	50.00 €	Finanzierung, IT
Johanna Schulze	0410-1241221	Alte Poststr. 5, Musterhausen	12345	Helena Meier	04.10.2004	3 Stunden	50.00 €	Finanzierung, IT
▶ Hans Müller	0221-2415932	Am Weiher 3, Musterhausen	12345	Ingo Fuchs	16.04.2004	1 Stunde	45.00 €	Marketing, Strategie

**Abb. 1.7:** Tabelle für die Aufgaben

1. Welche Datenredundanzen gibt es in der in Abbildung 1.7 dargestellten Tabelle? Welche Auswirkungen haben diese Redundanzen?
2. Wie viele Datensätze besitzt die in Abbildung 1.7 dargestellte Tabelle und wie viele Felder besitzt ein Datensatz?
3. Welche Entitäten können Sie in der Tabelle in Abbildung 1.7 erkennen? Welche Attribute besitzen diese Entitäten?
4. In welcher Beziehung stehen die in der vorherigen Aufgabe identifizierten Entitäten zueinander?
5. Was passiert, wenn Sie in der Tabelle in Abbildung 1.7 den zweiten, den vierten und den siebten Datensatz löschen?