
mitp
Frank
Geisler
5. Auflage
Datenbanken
Grundlagen und Design

Frank Geisler

Datenbanken
Grundlagen und Design

5., aktualisierte und erweiterte Auflage

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

ISBN 978-3-8266-8719-8
5., aktualisierte und erweiterte Auflage 2014

E-Mail: kundenbetreuung@hjr-verlag.de

Telefon: +49 6221 / 489 -555
Telefax: +49 6221 / 489 -410

www.mitp.de

Lektorat: Sabine Schulz
Sprachkorrektorat: Petra Heubach-Erdmann
Satz: III-satz, Husby, www.drei-satz.de
Coverbild: © Sebastian Kaulitzki

In Liebe für Aurelia.

Inhaltsverzeichnis

Vorwort zur 5. Auflage

Und schon wieder sind seit dem Erscheinen der 4. Auflage drei Jahre ins Land gegangen. Neu in dieser Auflage ist das Kapitel 14 *Big Data*, das sich mit den neuesten Entwicklungen in diesem Bereich beschäftigt. Dieses Thema ist in jüngster Zeit besonders durch den NSA-Datenskandal in das Bewusstsein der Öffentlichkeit geraten.

Wie in jedem Buch möchte ich mich hier natürlich bei allen Menschen bedanken, die dieses Werk erst möglich gemacht haben. Zuallererst ist da meine Freundin bzw. Verlobte Aurelia zu nennen. Vielen Dank für Dein Verständnis und Deine stetige Unterstützung bei allem, was ich so mache. Ich weiß, dass das nicht selbstverständlich ist, und möchte das an dieser Stelle auf jeden Fall noch einmal unterstreichen. Außerdem möchte ich mich bei meinen Mitarbeitern für ihre gute Arbeit bedanken. Danke an Frau Izabela Lemke dafür, dass sie es geschafft hat, aus einem Chaos einen vernünftigen Bürobetrieb zu schaffen, und auch vielen Dank an Karsten Himmel für seine unermüdliche Arbeit in unserer Infrastruktur. Eure Arbeit hält mir den Rücken für Projekte wie dieses hier frei. Besonderer Dank geht auch an Volker Hinz von Microsoft, der mich immer mit Neuigkeiten im Bereich SQL Server versorgt, und an Klaus Höltgen, der mir hilft, die PASS Regionalgruppe Ruhrgebiet zu leiten. Vielen Dank auch an die Kollegen von der oh22, ganz besonders an Oliver Engels, Tillmann Eitelberg und Kostja Klein (nicht oh22) (ihr wisst schon wofür ☺).

Natürlich möchte ich auch Sabine Schulz von mitp für die gute Zusammenarbeit danken; ohne dich wäre das Projekt sicherlich nicht denkbar. Zum Schluss geht noch ein Gruß an Kika, Alex, Helen, mein Patenkind Greta, Sylvia, Damian und Simon. Natürlich möchte ich mich auch bei Ihnen, werte Leser, bedanken – Sie sorgen mit Ihren Käufen dafür, dass ein solches Projekt überhaupt umgesetzt werden kann.

Frank Geisler

Lüdinghausen, im April 2014

Über den Autor

Frank Geisler hat an der University of Liverpool Information Technology studiert und den Studiengang mit dem Titel Master Of Science in IT abgeschlossen – er beschäftigt sich seit 1995 mit den unterschiedlichsten Datenbank- und Business-Intelligence-Lösungen.

Frank Geisler ist geschäftsführender Gesellschafter der Geisler Datensysteme GmbH & Co. KG, eines der führenden Microsoft-IT-Beratungshäuser im Ruhrgebiet. Er ist für die technische Ausrichtung des Unternehmens verantwortlich und führt, zusammen mit seinem Team, viele erfolgreiche BI-Projekte für namhafte Kunden durch. Seine Expertise hat er durch zahlreiche Microsoft-Zertifizierungen unter Beweis gestellt und hat einen Titel als Microsoft-Trainer.

Neben seiner Haupttätigkeit hat Frank Geisler schon zahlreiche Bücher geschrieben, Artikel in bekannten Fachzeitschriften veröffentlicht und ist Gründungsmitglied der PASS e.V. Deutschland – der deutschen User Community für Microsoft SQL Server. Dort zeichnet er als Regionalgruppenleiter für das Ruhrgebiet verantwortlich. Des Weiteren ist Frank Geisler oft als Sprecher auf nationalen oder internationalen IT-Konferenzen mit seinen Vorträgen zu Gast.

Teil I

Grundlagen

In diesem Teil:

Kapitel 1

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 Spätzahlern 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 heraussucht 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.

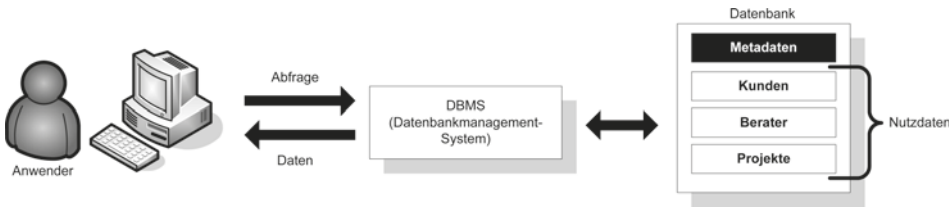


Abb. 1.1: Das DBMS verwaltet den Zugriff auf die Datenbank.

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 ändern-

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 Tramhaus 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.

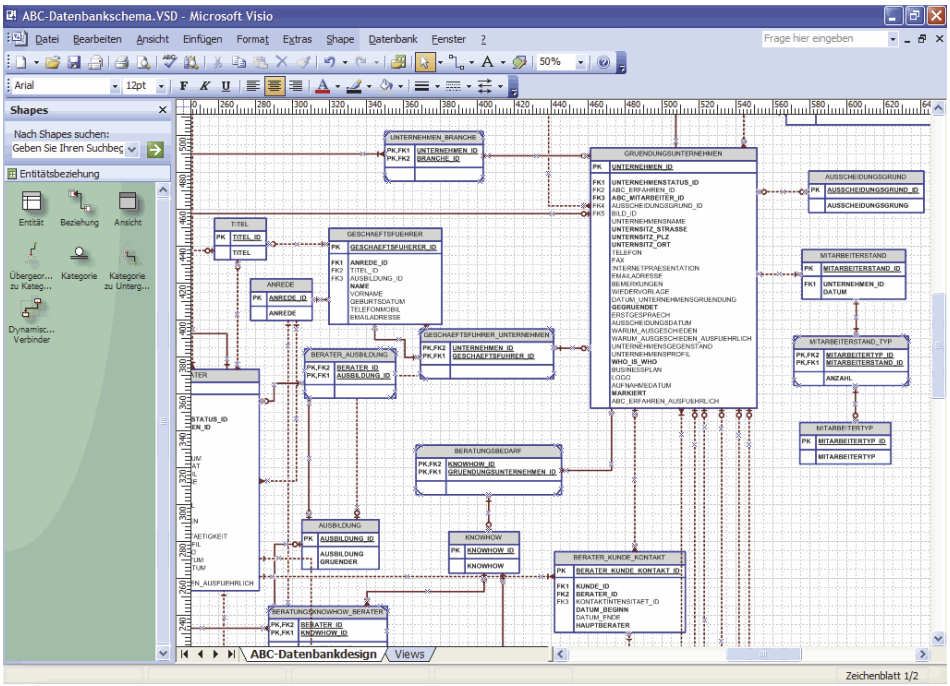


Abb. 1.2: Visio beim Datenbankdesign

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 Data-Warehouse-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 immer mehr Unternehmen auch Dateien in Datenbanken speichern. Sys-

teme, die die Speicherung von Dateien innerhalb einer Datenbank ermöglichen, werden als *Content-Management-Systeme* bezeichnet. Üblicherweise können zu den eigentlichen binären Daten, die eine Datei ausmachen, weitere Informationen, die so genannten *Metainformationen*, gespeichert werden.

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 heraussuchen und dann innerhalb der Akte nach der gewünschten Rechnung suchen. Das ist recht einfach. Komplizierter wird es, wenn Sie nun alle Rechnungen heraussuchen 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 Rechungen 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 Rechungen, 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 Verzeichnissen 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.

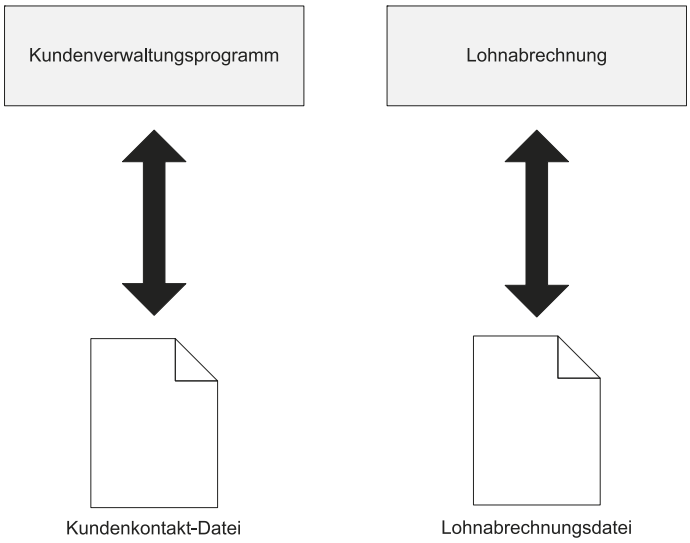


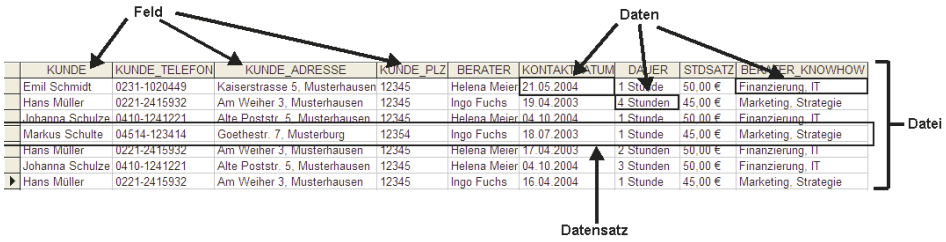
Abb. 1.3: Unterschiedliche Programme greifen auf unterschiedliche Dateien zu.

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-