

Relationale Datenbanksysteme — Informationsmanagement der Zukunft



Anton STEINRINGER, Mag., Jahrgang 1959, Studium der Betriebs- und Wirtschaftsinformatik. Nach Tätigkeiten in den Bereichen Software, Marketing und Unternehmensberatung nun Organisator für Datenbankentwicklungen im Dienstleistungsrechenzentrum der österreichischen Sparkassen, der SPARDAT-Datendienst GesmbH.

Die Verwaltung großer, komplexer Datenmengen in Wirtschaft und Wissenschaft bedarf geeigneter technischer Instrumente.

»Relationale Datenbanksysteme« — von Marketingstrategen als Schlagwort benutzt, von Theoretikern der Informationswissenschaft vielgepriesen und von Betrieben vielfach genutzt, werden den hohen Ansprüchen größtenteils gerecht. Über deren theoretischen Hintergrund, mit Beispielen einfach erklärt, lesen Sie in diesem Beitrag.

Der Umbruch im wirtschaftlichen Gefüge: Information als Produktionsfaktor

Waren bislang Arbeit, Boden und Kapital die Produktionsfaktoren, wie sie von der klassischen Ökonomielehre beschrieben wurden, so muß nun die Information als weiterer wesentlicher Produktionsfaktor gesehen werden.

Doch Information kann nur so gut sein, wie sie zum richtigen Zeitpunkt, in der gewünschten Verdichtung und am richtigen Ort zur Verfügung gestellt werden kann.

Ein neuer Begriff — Informationsmanagement — entsteht und schafft ein neues Profil eines Unternehmensmitarbeiters, des Informationsmanagers. Denn welche Daten für wen zugänglich sein dürfen, ist eine Frage des Datenmanagements. Ohne ein unternehmensweites organisatorisches Konzept, das durch die Akquisition und Zurverfügungstellung der Daten geregelt ist, sollte der Schritt in eine umfassende Datenbankumgebung nicht getan werden.

Doch nicht nur das organisatorische Umfeld ändert sich, auch die eingesetzten Verfahren und Instrumente zum Management der Information erfahren eine ständige Weiterentwicklung. Sieht man von den »Expertensystemen« ab, so stellen die »Relationalen Datenbanksysteme« die jüngsten, in der Produktion nutzbaren Entwicklungen dar. Auf diese Softwaretechnologie, deren Philosophie sowie Funktionalität und Einsetzbarkeit will ich nachfolgend eingehen.

Der Begriff des Datenbanksystems

Unter einem Datenbanksystem verstehen wir ein Softwaresystem, welches das systematische Erfassen, Löschen, Ändern sowie Abfragen von Datenbeständen unter Berücksichtigung bestimmter Regeln erlaubt. Unter den wesentlichen Kriterien, an denen ein Datenbanksystem gemessen wird, sind es insbesondere die Form der Verwaltung der Daten sowie die Zugriffstechnik, welche

entscheidend für die effiziente Nutzbarkeit des Systems sind. Der Form der Datenstrukturierung und Zugriffstechnik nach unterschieden wir bislang zwischen hierarchischen und Netzwerkdatenbanksystemen.

Hierarchische und Netzwerkdatenbanksysteme

Hierarchische Datenbanksysteme strukturieren und verwalten die Informationen in Form sogenannter Baumstrukturen und legen somit auch den Zugriffsweg auf Informationen fest.

Zur Verdeutlichung von Strukturen vergegenwärtigen wir uns ein Unternehmen, welchem an der Spitze die Direktion vorsteht. Mehrere Ressorts unterstehen dieser organisatorischen Einheit. Dieses Beispiel mag man in der Weise fortführen, daß wiederum jedem einzelnen Ressort ein oder mehrere Fachabteilungen unterstehen usw. Die graphische Darstellung dieses Organigramms zeigt die Struktur in Form eines Baumes; jeweils einer organisatorischen Einheit ist (sind) eine oder mehrere Einheit(en) nachgeordnet. So ist z.B. die Kompetenz- und Berichtslinie eindeutig vorgegeben.

Im Unterschied dazu erlaubt die Netzwerkstruktur Querverbindungen zwischen (Informations-)Einheiten. Betrachten wir wiederum unser Unternehmensbeispiel: Wir müssen uns eine Unternehmensstruktur vorstellen, in der z.B. die Fachabteilungen durch abteilungsübergreifende Projektgruppen ersetzt wurden. Ein Projektleiter kann nun ein oder mehrere Projekte leiten, und somit auch an mehrere Abteilungsleiter und/oder andere Projektleiter berichten; ein Netz an Berichtslinien kann entwickelt werden.

Organisatorisches Problem der Datenverwaltung

Die organisatorische Trennung von Aufgaben und Kompetenzen (z.B. durch Gliederung in Fachbereiche) führt auch zu einer

entsprechend getrennten Datenverwaltung; jeder Fachbereich hat »seine eigenen« Daten.

Dazu ein Beispiel aus dem Bankbereich: Herr Bauer unterhält neben seinem Sparbuch auch ein Gehaltskonto bei »seinem« Institut. Im Zuge des Aufwindes an der Wiener Börse investiert er sein Kapital auch in Aktien und anderen Wertpapieren. Jede der jeweils für ein Bankprodukt zuständigen Abteilungen der Bank speichert nun die Personaldaten, Konditionen etc. des Herrn Bauer (im Falle der Mehrfachspeicherung gleicher Informationen spricht man von Redundanz).

Ohne ein unternehmensweites organisatorisches Konzept, durch das die Akquisition und Zurverfügungstellung der Daten geregelt ist, sollte der Schritt in eine umfassende Datenbankumgebung nicht getan werden.

Falls nun Herr Bauer einer dieser Abteilungen seine Adreßänderung bekannt gibt, so konnte bislang nicht sichergestellt werden, daß alle betroffenen Fachabteilungen von dieser Änderung erfahren. Unterschiedliche Aktualität der Daten innerhalb der Bank sind die Folge (man spricht von Inkonsistenz oder fehlender Integrität in den Datenbeständen).

Die Lösung: gleiche Datenbestände für alle zugänglich

Stellen wir folgende Überlegung an: Wenn alle Ressorts auf ein und denselben Datenbestand zugreifen könnten, so würde Redundanz und Inkonsistenz vermieden werden.

Daraus folgt:

- 1) Alle Unternehmensdaten müssen in einer Datenbank gespeichert sein, auf die alle zugreifen können. Datenbanktheoretiker formulieren: »Die Daten der Datenbank sind ein Abbild der Unternehmensrealität«.
- 2) Interessiert die Personalabteilung mehr die Überstundenstatistik oder Gehaltsentwicklung der Mitarbeiter, so ist die Kreditabteilung an den mit dem Mitarbeiter vereinbarten Konditionen interessiert.



Somit zeigen verschiedene Fachabteilungen Interesse an unterschiedlichen Informationen — »Sie sehen einen unterschiedlichen Ausschnitt der Realität«. Daraus folgt, daß die Daten über einen Mitarbeiter (Name, Geburtsdatum, Adresse, geleistete Überstunden, Konditionen etc.) »entkoppelbar« sein müssen, um beliebige Abfragekombinationen ermöglichen zu können. Mit der Einrichtung eines Relationalen Datenbanksystems allein kann diesem Bedürfnis nicht entsprochen werden. Eine umfassende Datenanalyse muß dem Einsatz des Relationalen Datenbankmanagementsystems vorausgehen.

Datenmodellierung / Relationenmodell — Exkurs

Die umfassende Darstellung aller Unternehmensdaten und deren Beziehungen untereinander geschieht im Datenmodellierungsprozeß. Ziel ist es, ein kanonisches, also für das gesamte Unternehmen gültiges, konzeptionelles Datenmodell zu entwickeln.

Tedd Codd [1] veröffentlichte bereits 1970 Regeln, nach denen das »Kanonische konzeptionelle Datenmodell« entwickelt werden sollte.

In der Girozentrale Wien wurde ein solches, für den Konzern gültiges Datenmodell derart entwickelt, daß Personen des Managements in einer sogenannten »Top-Down-Vorgangsweise« die wesentlichen »Säulen« der Unternehmensdaten aufzeigten.

Kunde, Mitarbeiter, Produkt, Geschäft, Betriebsmittel, etc. wurden als unternehmensweit genutzte Begriffe erkannt. Zwischen diesen Begriffen, in der Datenmodellierung als »Entitäten« oder »Objekte der Realität« bezeichnet, wurden deren Beziehungen zueinander als Linien dargestellt.

So entsteht im Wege der Verfeinerung des Datenmodells ein dichtes Netz an Verbindungslinien zwischen zahlreichen Entitäten (Datenmodell).

Nun besitzen die einzelnen Entitäten (z.B. Kunde) bestimmte, eben diese Entität beschreibende Merkmale (z.B. Vorname, Nachname, Titel). Codd empfiehlt die Darstellung dieser beschreibenden Merkmale (sogn. Attribute) mit ihren Ausprägungen (Vorname = Karl, Nachname = Mayer) in Tabellenform (Relation, siehe Abb. 2).

Dabei werden in den Spalten die Attribute (Merkmale) einer Entität eingetragen (VOR-

NAME, NACHNAME .. in Abb. 2) und in den Zeilen die Ausprägungen derselben. Einer dieser Spalten kommt jedoch eine besondere Bedeutung zu. Ein darin enthaltener Wert darf nur einmal vorkommen; man spricht von der »Eindeutigkeit« dieses Attributes, welches auch als »Schlüssel« bezeichnet wird (PERSID, KONRID, ADRID in Abb. 2).

Um die vorhin angesprochene Entkoppelbarkeit der Attribute untereinander gewährleisten zu können, muß eine solche Tabelle (Relation) der 3. Normalform genüge tun. Diese im Rahmen der Normalformlehre (nach Codd) entwickelte Richtlinie besagt, daß alle Ausprägungen von Attributen einer Tabelle, welche nicht Schlüssel sind, voneinander unabhängig sind. Sie stehen nur mit dem Schlüsselattribut in direkter Abhängigkeit. Über diesen Schlüsselbegriff erfolgt ja auch letztlich der direkte Zugriff auf die damit in Verbindung stehenden Attribute. Diese abstrakte Darstellung verlangt nach einem Beispiel:

Aufgabenstellung: Wir möchten die Personen, deren Adressen und Konditionen verwalten.

Wie war es bisher? In einer Datei wurden zu jeder Person nebst deren persönlichen Daten auch die Adresse sowie Konditionen abgespeichert (Abb. 1).

Situation 1:

Folge: Frau Anna Wolf ehelicht Herrn Manfred Berger und zieht zu ihm. Infolgedessen müssen lediglich zwei Änderungen im Datensatz der Frau Anna Wolf vorgenommen werden (Namens- und Adreßänderung).

Situation 2:

Aufgrund einer Unternehmensentscheidung muß der Mengenrabatt von 10% auf 5% reduziert werden. Dies hat erhöhten Änderungsaufwand zur Folge:

Obwohl nur ein Parameter geändert wird, müssen 3 Datensätze verändert werden.

Im Unterschied dazu:

Wesentlich einfacher gestalten sich die Änderungen, wenn ein Datenmodell in 3. Normalform zugrundeliegt:

In dieser Umgebung werden die Daten aus der in Abbildung 1 dargestellten Tabelle auf folgende Tabellen (Relationen) (Abb. 2) aufgeteilt.

Situation 1 (wie oben)

In der Relation PERSON muß der Name

WOLF durch BERGER ersetzt werden; in der Relation PERSON-ADRESSUORDNUNG muß PERS3 mit ADR3 verbunden werden. Dabei muß nicht der gesamte Satz (vgl. Abb. 1) gelesen werden.

Situation 2 (wie oben):

Nun zeigt sich die Stärke des relationalen Gedankens: Lediglich der Prozentsatz in der Relation »KONDITIONEN« muß geändert werden; alle anderen Daten bleiben unangetastet.

Relationale Datenbankmanagementsysteme (RDBMS)

Den Ausführungen des Exkurses über Datenmodellierung kann der Leser entnehmen, daß die Informationsdarstellung in Tabellenform sehr einfach und verständlich ist. Da den Tabellen keine hierarchische Struktur immanent ist, kann auf jede beliebig wählbare Informationseinheit (Relationen, Attribute) zugegriffen werden.

Entwickler von Datenbanksystemen erkannten bald, daß sie die relationale Idee (Sicht der Daten in Tabellenform) ihren Datenbanksystemen zugrundelegen müßten, um den Markterfordernissen entsprechen zu können.

Rasch erkannten zahlreiche Anbieter von Datenbanksystemen als Reaktion auf die Codd'schen Theoreme die relationalen Eigenschaften ihrer Systeme; — doch Vorsicht — nicht alle als »relational« bezeichneten Systeme genügen den gestellten Anforderungen:

Tabellenform und die relationale Algebra

Die Daten stellen sich für den Benutzer der Datenbank in Tabellenform dar. Auf diese Tabellen müssen Operationen der relationalen Algebra wie SELECT, PROJECT und JOIN durchgeführt werden können.

Die relationalen Operatoren

Neben den klassischen Mengenoperationen wie Vereinigung, Durchschnitt und Differenz unterstützt das Relationale Datenbanksystem die Relationenoperationen SELECT, PROJECT und JOIN.

SELECT ermöglicht die Auswahl bestimmter Zeilen einer Tabelle, welche sich aufgrund einer Bedingung qualifizieren.

(»Zeige alle männlichen Personen aus der Tabelle Person«)

PROJECT zeigt die Inhalte von ausgewählten Spalten an. (»Zeige al-

Vorname	Nachname	Geschlecht	Geburtsdatum	Ort	Straße	Hausnummer
Hubert	Steiner	männlich	14.5.1961	Wien	Taborstraße	5
Manfred	Berger	männlich	9.6.1953	Reutte	Siebenbrunnst.	6
Anna	Wolf	weiblich	28.7.1959	Innsbruck	Berggasse	12
Sigrid	Gärtner	weiblich	18.1.1965	Wien	Taborstraße	5
Typ der Kondition	% gültig von-bis					
Mengenrabatt	10 19860101-19860630					
Mengenrabatt	10 19860101-19860630					
Mengenrabatt	10 19860101-19860630					
Sonderkondit	30 19860501-19860601					

Abb. 1: Konventionelle Datenspeicherung



Relation Person				
Personen-identifikation PERSID	Vorname	Nachname	Geschlecht	Geburtsdatum
Pers1	Hubert	Steiner	männlich	14.5.1961
Pers2	Manfred	Berger	männlich	9.6.1953
Pers3	Anna	Wolf	weiblich	28.7.1959
Pers4	Sigrid	Gärtner	weiblich	18.1.1965

Relation Adresse				
Adressen-Identifikation ADRID		Ort	Straße	Hausnummer
Adr1		Wien	Taborstraße	5
Adr2		Innsbruck	Berggasse	12
Adr3		Reutte	Siebenbrunnstraße	6

Relation Konditionen			
Konditionen Identifikation KONDID	Typ der Kondition	%-Satz	gültig von-bis
Kond1	Mengenrabatt	10%	19860101-19860630
Kond2	Sonderkonditionen	30%	19860501-19860601

Relation Person-Adressenverknüpfung		
Personenidentifikation	Adressenidentifikation	Typ
Pers1	Adr1	Hauptwohnsitz
Pers2	Adr3	Hauptwohnsitz
Pers4	Adr1	Nebenwohnsitz
Pers3	Adr2	Hauptwohnsitz

Beachte: An der Adresse selbst (Adr1) müssen keine Änderungen vorgenommen werden, sollte sich deren Charakteristikum (Haupt- oder Nebenwohnsitz) ändern.
Die Zuordnung von Adressen zu Personen kann ohne Änderungen an den »originären« Stammdaten (Person und Adresse) geschehen.

Relation Person-Kondition	
Personenidentifikation	Konditionenidentifikation
Pers1	Kond1
Pers2	Kond1
Pers4	Kond2
Pers3	Kond1

Abb. 2: Datenspeicherung nach dem Relationmodell.

le Nachnamen der Tabelle Person») stellt den logischen Verbund zweier oder mehrerer Tabelle über den Inhalt eines Schlüsselbegriffes, der allen Tabellen gemein ist, dar.

Mengenorientierte Datenmanipulation

Im Gegensatz zu konventionellen Datenbanksystemen erfolgt die Manipulation von Daten (Lesen, Ändern, Löschen, Einspeichern) nicht einzelsatz-, sondern mengenorientiert. Suchen wir z.B. in der Datenbank alle in Graz ansässigen Lieferanten für Büromöbel, so genügt ein einziger Suchbefehl, um alle ausgewählten Lieferanten zu

finden. Desgleichen können wir auch beispielsweise bestimmte Konditionen eben dieser Lieferanten mittels eines Befehles auf einmal ändern.

Navigation / Zugriff auf Daten

Im Umfeld des RDBMS muß der Anwender nichts um die internen Zugriffspfade wissen; dieses »Navigieren« ist Aufgabe des RDBMS. Vielmehr erfolgt der Zugriff durch die Angabe von Attributinhalt (im obigen Beispiel — Attribut: Wohnort; Attributinhalt: GRAZ). Die Frage »WAS will ich an Daten« steht im Vordergrund, nicht die Frage »WIE komme ich zu den Daten«.

Integrität / Konsistenz

Der Nutzen eines DBMS steht und fällt mit der Erfüllung dieser Qualitätsmerkmale.

Gemeint ist, daß jederzeit die Aktualität resp. Richtigkeit der Daten selbst, wie auch ihrer Beziehungen untereinander durch das Datenbanksystem gewährleistet werden. Beispielsweise muß das Datenbanksystem beim Erfassen einer Bestellung überprüfen, ob der gewünschte Artikel auch im Datenbanksystem erfaßt wurde. Sollte eine Artikelbezeichnung gelöscht werden, so muß sichergestellt sein, daß keine Bestellung für den betreffenden Artikel vorliegt (Referential integrity).

Eine weitere Regel besagt, daß beim Erfassungs- und Änderungsprozeß alle Schlüsselattribute unbedingt einen Wert bekommen müssen, denn über diesen logischen Schlüssel erfolgt der Zugriff auf die



damit in Verbindung stehenden Zeileninformationen (Entity integrity).

Verletzung der Integrität könnte auch dadurch zustandekommen, daß zwei oder mehrere Datenbankbenutzer auf ein und dieselbe Information zugreifen und diese »gleichzeitig«, jedoch inhaltlich unterschiedlich verändern wollen (»concurrent update«).

Ein RDBMS sieht für diesen Fall Sperrmechanismen vor; auf Daten, welche gerade von einem Benutzer geändert werden, haben andere Benutzer keinen Zugriff.

Die Freigabe dieser gesperrten Daten erfolgt nach ordnungsgemäßem Abschluß der Änderung (Transaktionslogik).

Restart / Recovery

Systemzusammenbruch, Stromausfall und dergleichen verhindern, daß eine Operation, welche gerade bei Ereigniseintritt durchgeführt werden sollte, abgeschlossen wird. Eine unterbrochene Änderung mehrerer Tabellen hinterläßt einen Datenbankzustand, welcher nicht den Integritätsanforderungen genügt. Unterstützt ein RDBMS die Eigenschaften von Restart und Recovery, so setzt das RDBMS bei Wiederanlauf die Datenbank in jenen Zustand, der die Integrität gewährleistet (das heißt, alle unvollständigen Aktionen werden zurückgesetzt).

Die Softwareumgebung eines RDBMS
Bei genauerer Betrachtungsweise der am Markt angebotenen RDBMS fällt auf, daß nebst dem Nukleus des Datenbanksystems eine Reihe von Softwarekomponenten mitangeboten wird, welche zum effizienten Betreiben der Datenbank unbedingt notwendig sind:

Das Dictionary — Beschreibung der Datenbankinformationen

Besondere Bedeutung kommt dem Dictionary zu. Darunter verstehen wir ein Verzeichnis, in dem alle Informationen über die in der Datenbank verwendeten Relationen und deren Attribute gespeichert sind. (So etwa nähere Hinweise über Bezeichnung des Attributes (z.B. Vorname), Länge (12 Zeichen), Datentyp (alphanumerisch), Schlüsselbegriff (nein) u.v.a.m.)

Als aktives Dictionary wird jene Form von Datenverzeichnis bezeichnet, welche bei der Erstellung von Programmen (in einer Sprache der 4. Generation erstellte) automatisch vermerkt, welche Attribute aus welcher Relation von eben diesen Programmen angesprochen werden.

Datenmanipulationssprache

Die Handhabbarkeit von Datenbankoperationen muß für Programmierer wie auch den sog. »Endbenutzer« gleichermaßen einfach gestaltet werden. Mit dieser Absicht wurden die in der Literatur als »Sprachen der 4. Generation« bezeichneten Instrumente entwickelt.

Im wesentlichen verfügen all diese Sprachen über Befehle zum Einspeichern, Verändern, Löschen sowie Abfragen und Darstellen von Datenbeständen. So wird in einer fast umgangssprachlichen Formulierung eine Abfrage in der Form — Finde in der Tabelle Person alle Personen mit dem

Geschlecht »männlich« — gestellt.

Zusätzliche Funktionen wie Unterstützung von Rechenoperationen, Entwicklung von Bildschirm- und Listenmasken etc. zeichnen einige dieser »4. Generationssprachen« aus.

Der wesentliche Unterschied zu bisherigen Programmiersprachen ist in der weitestgehenden Datenunabhängigkeit zu sehen. In COBOL-Programmen mußten die Datendefinitionen aller Datenbestände eingebunden werden. Wurden nun an diesen (klassisch) verwalteten Datenbeständen Strukturänderungen vorgenommen (Adressenfeld wurde verlängert), so mußten alle betroffenen COBOL-Programme adaptiert werden.

Nicht so bei Sprachen der 4. Generation in der RDBMS-Umgebung. Werden Datenbeschreibungen benötigt, so erfolgt automatisch der Zugriff auf das zentrale Dictionary, in dem der jeweils aktuelle Zustand der Datendefinition abgelegt ist.

Security-System

Gerade aus dem Titel, daß alle Unternehmensdaten in einer Datenbank abgelegt sind, resultiert die Forderung nach umfassendem Datenschutz. So sollten beispielsweise Personaldaten nicht dem Lagerverwalter, sondern nur der Personalverwaltung zugänglich sein.

Neben dem Datenschutz ist auch technisch der Funktionsschutz vorzusehen. Dieser bewirkt, daß nicht jedem Datenbankbenutzer die Funktion »Löschen« oder »Ändern« von Daten zur Verfügung gestellt wird. Funktionsschutz kann auch in der Weise realisiert werden, daß die Ausführung bestimmter Programmsysteme in Abhängigkeit vom definierten Benutzerprofil unterbunden wird.

Verteilte Datenbanksysteme

Sprachen wir bisher von **einer** Datenbank, welche auf einem Rechner installiert ist, so muß der Vollständigkeit halber auch auf folgende Neuentwicklungen hingewiesen werden:

Von verteilten Datenbanken spricht man, wenn mehrere physisch verteilte Datenbanken wie **eine** logische Datenbank verwaltet werden. Ein Beispiel mag dies veranschaulichen:

Der Verkaufsleiter eines großen Konzerns mit Hauptsitz in Wien und Filialen in Salzburg, Graz und Linz möchte an seinem Bildschirm die Umsätze der einzelnen Verkäufer aller Filialen sowie deren Überstundenstatistik sehen. Dabei sei dem Beispiel zugrundegelegt, daß jede Filiale ihre Daten vor Ort mittels eines RDBMS verwaltet; — über den Netzwerkprozessor wird jedoch die logische Sicht über alle physisch verteilten Datenbanken gewährleistet. In unserem Beispiel erhält der Verkaufsleiter nicht nur die Informationen des Wiener Hauptsitzes, sondern zugleich auch die Informationen aus den physischen Datenbanken der Filialen.

Zusammenfassung der Vor- und Nachteile von RDBMS

Vorteile:

Eine Datenbasis

Bei Einrichtung der relationalen Datenbank im Unternehmen stehen allen Benutzern die gleichen, aktuellen Daten zur Verfügung; Datenredundanz wird vermieden.

Entwicklungen in der Datenbankanwendung

- Informationen über die gespeicherten Daten sind aus den Anwendungen ausgelagert und im zentralen Dictionary verwaltet.
- Beim Einspeichern der Daten in die Datenbank muß auf nachfolgende Abfragen hinsichtlich Speicherstruktur und damit Zugriffspfade keine Rücksicht genommen werden.
- Die Frage »Was brauche ich an Informationen« und nicht »Wie komme ich zu den Informationen« steht im Vordergrund.
- Änderungen an der Datenbasis (z.B. Verlängerung eines Feldes etc.) haben keine Auswirkungen auf bereits erstellte Programmsysteme.
- Mächtige Instrumente (»Sprachen der 4. Generation«) erlauben die rasche Erstellung von operationalen Systemen und Manipulation des Informationsmaterials.

Nachteile:

Performance

Im wesentlichen konzentriert sich die Kritik an RDBMS auf das Laufzeitverhalten. Die Aufspaltung der Information in zahlreiche Tabellen führt zu enormem Verwaltungs- und Zugriffsaufwand; dafür sind die Daten weitestgehend redundanzfrei. Dies stellt **die** Herausforderung an die Hersteller von RDBMS dar, die mit ausgetüftelten Algorithmen (Zugriff auf Informationen über Indizes, Lesen zusammenhängender Informationen im voraus etc.) diesen Umstand entgegenwirken wollen.

Integrität

Der Wahrung der semantischen Integrität sowie der geordneten Beziehungen von Informationseinheiten muß größtes Augenmerk geschenkt werden. Bei der Bewertung von RDBMS kann diesbezüglich nicht genug kritische Analyse empfohlen werden.

Ausblick

Laut James Martin [2] wird sich der Anteil von DV-Kapazität zur Informationsbereitstellung von heute 20% auf über 60% steigern. Dies als Reaktion auf wachsendes Bedürfnis nach aktueller Information, welche wie eingangs erläutert, einen neuen Produktionsfaktor darstellt.

Der Schritt in die relationale Datenbanktechnologie muß als strategisch äußerst wichtig angesehen werden.

Zuvor muß jedoch das organisatorische Umfeld aufbereitet sowie die sorgfältige Datenmodellierung ausgeführt werden; geordnete, aktuelle Daten mit vielfältiger Verknüpfungsmöglichkeit lohnen diesen Aufwand.

