

Manfred Precht

Datenverarbeitungsstelle der TU München,
Freising-Weihenstephan

WARUM DATENBANKEN?

1. Einleitung

In den vorangehenden Vorträgen ist wiederholt und ausführlich über die Anwendung und den Einsatz von Datenbanken in landwirtschaftlichen Bereichen gesprochen worden. Ich möchte daher zum Schluß einige grundsätzliche Bemerkungen über Notwendigkeit und Vorteile einer datenbankorientierten Datenorganisation machen.

2. Herkömmliche Datenorganisation

In vielen technisch-wissenschaftlichen, kommerziellen oder administrativen Bereichen fallen große Datenmengen an. Eine wichtige Aufgabe ist, diese Daten so zu speichern, daß sie auch noch zu einem späteren Zeitpunkt ausgewertet werden können oder daß sie mehreren Personen für unterschiedliche Ziele zur Auswertung zur Verfügung gestellt werden können. Wie eine solche Datenorganisation in der Vergangenheit aussah und auch heute noch manchmal aussieht, zeigt Bild 1.

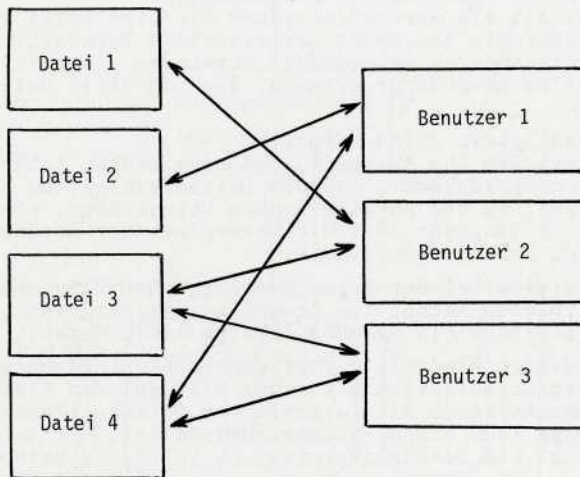


Bild 1: Herkömmliche Datenorganisation

Man stelle sich vor, die Dateien 1 bis 4 enthalten komprimierte Daten, welche aus einer Weiterverarbeitung von Rohdaten entstanden sind. Die Benutzer 1 bis 3 verarbeiten diese komprimierten Daten, indem sie diese in Anwendungsprogrammen lesen oder modifizieren. Dies bedeutet, daß sich der Bestand an Daten in diesen Dateien ändert. Es kann dann vorkommen, falls keine Absprachen zwischen den Benutzern über die Organisation der Dateien getroffen wurden, daß die Dateien teilweise die gleichen Daten enthalten, so wie es in Bild 2 dargestellt ist.

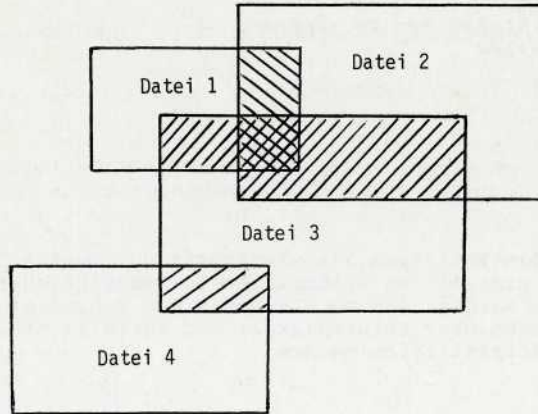


Bild 2: Redundanz der Daten

An diesem kleinen Beispiel sollten einige wichtige Probleme der Datenorganisation in größeren Bereichen kurz angerissen werden:

- Redundanz der Daten

Die von verschiedenen Anwenderprogrammen verarbeiteten Dateien enthalten teilweise dieselbe Information. Unter Umständen sind diese Dateien jedoch auf einem unterschiedlichen Aktualitätsstand.

- Komplikationen bei Änderung des Datenmodelles

Verwenden mehr als ein Anwenderprogramm dieselbe Datei und möchte z.B. ein Anwender die Anordnung der einzelnen Datensätze und -felder ändern, d.h. ein anderes Datenmodell verwenden, so erfordert dies Änderungen in allen Anwenderprogrammen, die auf diese Datensätze zugreifen.

- generelle Abhängigkeit Daten-Programme

Im vorigen Punkt kam zum Ausdruck, daß eine starke Abhängigkeit zwischen den Anwenderprogrammen und der Datenorganisation besteht. Aber auch das Austauschen der physikalischen Datenträger, wie Platten, Trommeln oder in jüngster Zeit die Massenspeicher bedingt häufig eine Änderung in den Anwenderprogrammen.

Wenn solche Nachteile bei der Organisation großer Datenmengen für den Zugriff einer größeren Anzahl von Anwenderprogrammen ins Gewicht fallen, empfiehlt es sich, ein Datenbanksystem einzuführen.

In den letzten Jahren sind viele Anstrengungen unternommen worden, die herkömmliche Datenorganisation mit einer Vielzahl von Einzelauswertungen durch Datenbanksysteme mit formatierten Datenbeständen zu ersetzen. Es gibt heutzutage kaum einen Hardware-Hersteller, der in seinem Software-Angebot nicht ein Datenbanksystem zur Verfügung hätte.

3. Definition und Architektur einer Datenbank

In Anlehnung an [1] möchte ich eine Datenbank wie folgt definieren:

"Eine Datenbank ist eine Sammlung von strukturierten Datenbeständen, die durch ein Datenmodell beschrieben werden und über ein Datenbankmanagement-System für viele Benutzer verfügbar sind."

Dabei soll eine Datenbank folgende Forderungen erfüllen:

- Redundanzfreie Speicherung der Daten

- Datenunabhängigkeit
- Datenschutz
- Datensicherung
- Benutzerfreundlichkeit

Die Datenbank und das Datenbankmanagement-System (die entsprechende Software) bilden zusammen das Datenbanksystem.

Bild 3 zeigt die Architektur einer Datenbank (vgl.[2]). Die einzelnen Benutzer können sowohl im Stapelbetrieb wie im Dialogbetrieb arbeiten. Bei Stapelverarbeitung werden übliche Programmiersprachen wie FORTRAN, PL1 o.ä. verwendet, der Benutzer eines Bildschirmes verwendet eine interaktive Kommando-Sprache zur Verwaltung der Daten oder eine Kommunikationssprache, welche es ihm ermöglicht, Fragen an die Datenbank zu stellen.

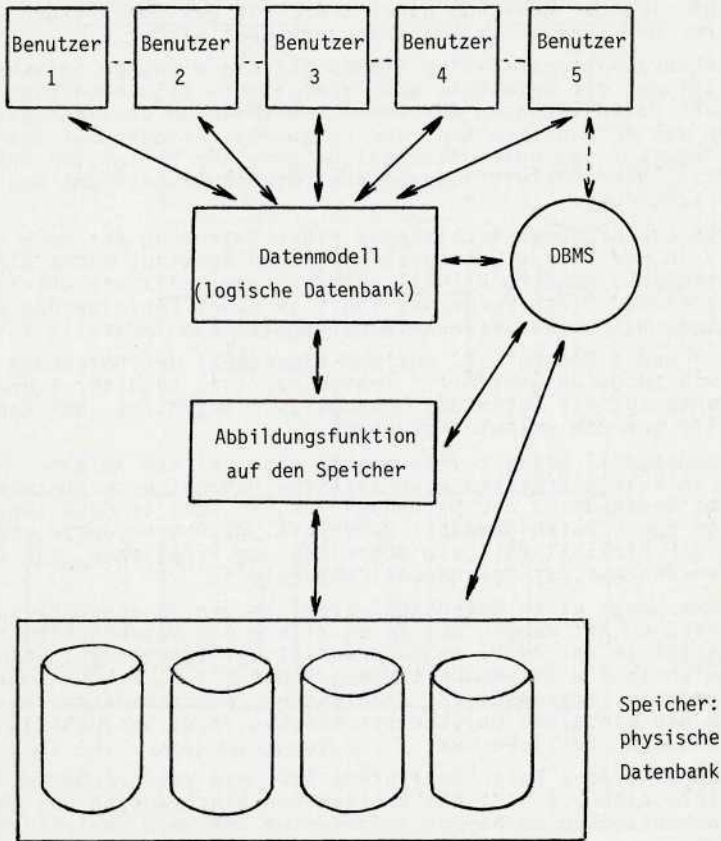


Bild 3: Vereinfachte Architektur einer Datenbank

Allen diesen Sprachen ist gemeinsam, daß sie als Untermenge eine "Datensprache" enthalten, mit welcher auf die Datenbank zugegriffen wird. In den konventionellen Programmiersprachen ist die "Datensprache" eine Reihe von Prozeduraufrufen. Bei der Kommando- und der Kommunikationssprache kann die "Datensprache" in die einzelnen Kommandos integriert sein.

Ferner wird jedem Benutzer ein Arbeitsbereich zur Verfügung gestellt für den Datentransfer zwischen Benutzer und Datenbank. Für den Anwendungsprogrammierer sind dies die I/O-Puffer der Programme, für den Benutzer am Terminal der Arbeitsbereich des Terminals.

Eine ganz wichtige Eigenschaft einer Datenbank ist die Datenunabhängigkeit, d.h. die Definition der Datenbank oder das sog. Datenmodell sind sowohl von den Anwendungsprogrammen als auch von den physikalischen Speichern unabhängig. Man kann auch so sagen: Zwischen dem Anwendungsprogrammierer und dem physikalischen Speicher liegt das Datenmodell oder die logische Datenbank. Das Datenmodell ist der ganze Informationsinhalt der Datenbank. Der Benutzer sieht das Datenmodell als eine Sammlung von logischen Datensätzen. Spezifische EDV-Charakteristika gehen in dieses Modell nicht ein, also z.B. keinerlei Betrachtungen über die physikalische Speicherung der Datensätze und über Zugriffsmethoden. Wie die Daten physikalisch gespeichert werden, ist in der Abbildungsfunktion festgelegt. Die Abbildungsfunktion ist für den Benutzer der Datenbank nicht sichtbar und kann daher jederzeit geändert werden, wenn z.B. ein anderer physikalischer Speicher verwendet wird.

Das Datenbankmanagement-System (DBMS) ist die wichtige Software, welche den Zugriff auf die Datenbank erst ermöglicht. Alle Forderungen der Benutzer nach Datenelementen der Datenbank gehen an diesen Manager. Dieser führt die notwendigen Ein- und Ausgabeoperationen auf dem physikalischen Speicher aus unter Berücksichtigung der Definition des Datenbankmodells. Eine weitere Aufgabe des Datenbankmanagement-Systems ist die Datensicherung.

Die soeben beschriebene Architektur einer Datenbank ist noch sehr inflexibel. In der Praxis interessieren einen Benutzer nicht alle Daten einer Datenbank, sondern er will nur auf eine bestimmte Untermenge von Daten zugreifen. Diese Forderung führt zu einer Verfeinerung der Architektur eines Datenbanksystems wie es in Bild 4 dargestellt ist.

Benutzer 1 und 2 können z.B. nur auf einen Teil der Datenbank zugreifen, so wie im Datensubmodell 1 festgelegt ist; Benutzer 4 und 5 hingegen können auf die Daten des Submodells 2 zugreifen. Nur Benutzer 3 hat Zugriff auf die gesamte Datenbank.

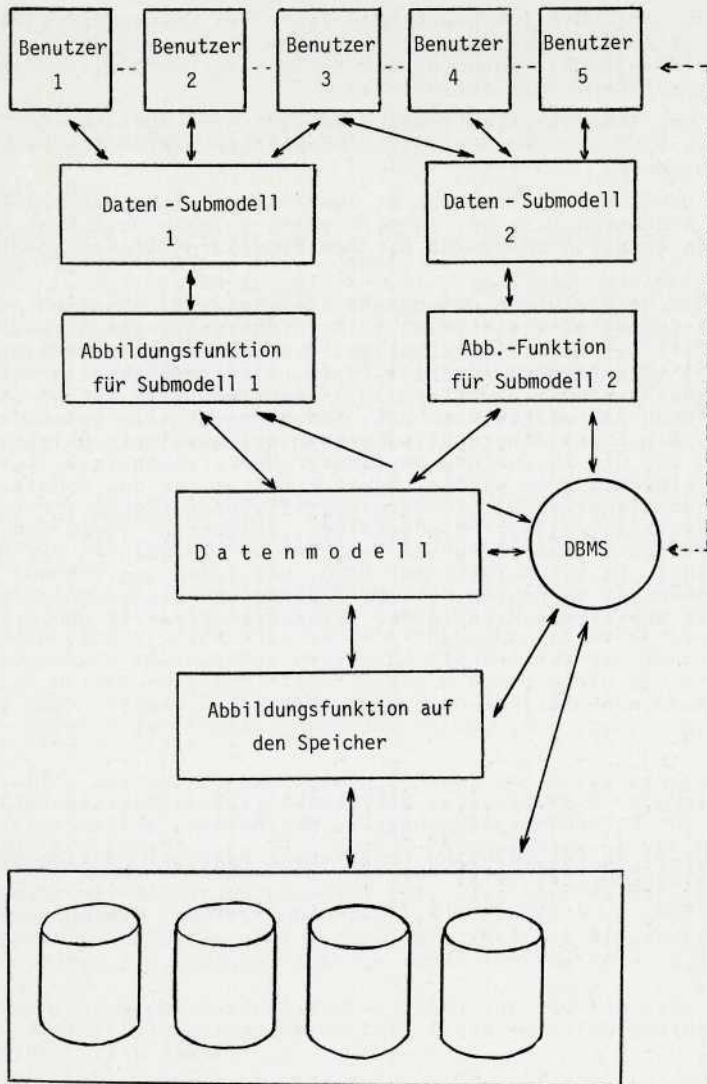
Das Datensubmodell erfüllt jedoch noch eine weitere Aufgabe. Möchte ein Benutzer in seinem Programm eine logische Datenstruktur verwenden, die noch nicht Bestandteil der Datenbank ist, so kann er dies tun, indem er selbst ein neues Datensubmodell definiert. Datensubmodelle erhöhen somit auch die Flexibilität beim Schreiben von Programmen. Die Datensubmodelle werden auf das Datenmodell abgebildet.

Die Zusammenhänge eines Datenbanksystems sollen an einem Beispiel erläutert werden. Wir nehmen an, es existiere ein Datenbanksystem "Pflanzenproduktion in der BRD". Im Datenmodell ist festgelegt, welche relevanten Daten in die Datenbank Eingang finden, z.B. Produktionsdaten, Witterungsdaten, Ertragsdaten, Labordaten, Betriebsdaten usw.; es sind dort auch die einzelnen Objekte beschrieben, z.B. Versuchsflächen, Parzellen, Schläge, Betriebe usw.

Ein Submodell dieses Datenbanksystems bzw. ein entsprechender Benutzer interessiert sich z.B. für die Erträge von Winterweizen und möchte diejenigen Anbauflächen in Bayern aufgelistet bekommen, welche in den Jahren 1977-1979 einen Ertrag erzielt haben, der einen vorgegebenen Wert übersteigt. Der Benutzer interessiert sich also für die Objekte "Orte der Anbauflächen mit einem Ertrag größer als ein vorgegebener Wert".

Der Zugriff auf ein Exemplar eines solchen Objektes geschieht nun in folgenden Schritten:

- das DBMS erhält den Auftrag, die gewünschten Objekte zu lesen;
- das DBMS holt sich die benötigten Definitionen des entsprechenden Objekt-Typs aus dem Submodell, d.h. das DBMS muß den logischen Aufbau des Submodells kennen, um die entsprechenden Objekte aufzufinden;



Speicher: physische Datenbank

Bild 4: Verfeinerte Architektur einer Datenbank

- das DBMS besorgt die entsprechenden Teile des Datenmodells unter Anwendung der Abbildungsfunktion: $\langle \text{Submodell} \longrightarrow \text{Modell} \rangle$ und ermittelt die Beziehungen, welche benötigt werden, um die Objekte im gesamten Datenmodell aufzufinden;
- aufgrund der Abbildungsfunktion: $\langle \text{Modell} \longrightarrow \text{Speicher} \rangle$ stellt das DBMS aus den benötigten physischen Sätzen das verlangte Objekt zusammen.

Bei großen Datenbanksystemen ist es zweckmäßig, einen Datenbankadministrator zu benennen. Eine der Aufgaben eines solchen Datenbankadministrators wäre es dann, in Absprache mit den Benutzern Datensubmodelle zu definieren.

Das in Bild 4 beschriebene Datenbanksystem erfüllt noch nicht die Forderung nach dem Datenschutz. Eine erste Sicherung gegenüber unberechtigtem Zugriff ist die Identifikation, d.h. der Benutzer muß sich dem Datenbanksystem gegenüber identifizieren. Dies geschieht in der Regel durch die Angabe eines Paßwortes. Im allgemeinen soll ein Benutzer, der sich als berechtigt ausgewiesen hat, jedoch nicht alle Daten der Datenbank sehen. Die Zugriffsberechtigung kann auf bestimmte Objekte der Datenbank und auf die Ausführung bestimmter Operationen (z.B. Lesen, Schreiben) eingeschränkt werden. Damit erreicht man den Schutz von Objekten vor unerlaubtem Zugriff. Die Zugriffsberechtigung ist in der Berechtigungsmatrix (vgl. [3]) niedergelegt, in der für jeden Benutzer und jedes geschützte Objekt angegeben ist, welche Operationen der Benutzer ausführen darf. Es ist Aufgabe des DBMS, bei jedem Zugriff auf ein Objekt der Datenbank zu prüfen, ob der Zugriff und die gewünschte Operation mit den Zugriffsrechten in der Berechtigungsmatrix übereinstimmen.

LITERATUR

- [1] KRÖMER, N. u. SCHRODER, S.: Datenbanken und Informationssysteme. Verlag für Unternehmensführung Dr. Max Gehlen, 1976.
- [2] DATE, C.J.: An introduction to Database Systems. Addison-Wesley Publishing Company, 1976.
- [3] SCHLAGETER, G. u. STUCKY, W.: Datenbanksysteme: Konzepte und Modelle. Teubner, 1977.