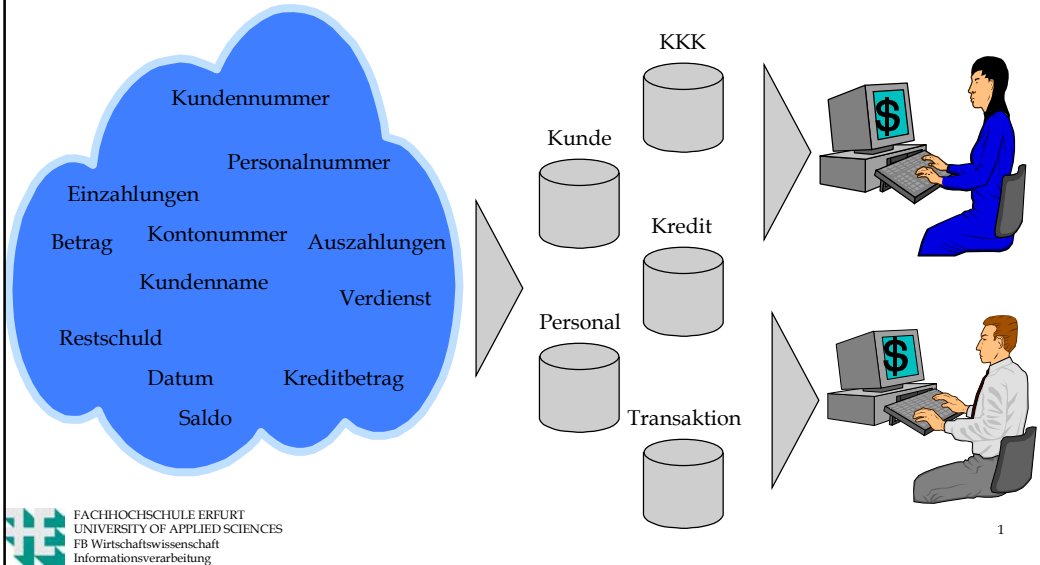


Datenorganisation



2.2. Datenorganisation

Ziel:

Grundbegriffe der Datei- und Datenbankorganisation

Gliederung:

1. Datenorganisation
2. Dateiorganisation
3. Datenbankorganisation

Literatur:

Stahlknecht S. 156 ff.

1. Datenorganisation

Warum das alles?

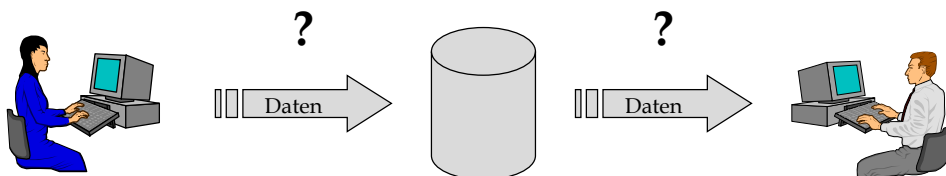
Grundsätzlich geht es darum,

- bestimmte Daten, die irgendwo und irgendwann abgespeichert wurden, wiederzufinden und
- Daten einfach zu aktualisieren und
- beliebige Auswertungen und Verknüpfungen zu erstellen

Um diesen Anforderungen zu genügen, werden Methodiken benötigt, um Daten vernünftig abzuspeichern, auf diese zuzugreifen und sie zu verarbeiten, sie auszuwerten und zu verknüpfen.

1. Datenorganisation

Grundproblem der Datenhaltung



Grundproblem der Datenhaltung ist es, eine **Verbindung** zwischen dem logischen Begriff (Kunde xyz) und dem physischen Speicherplatz (Datei, Spur, Sektor) herzustellen. Diese Verbindung wird i.d.R. durch das Betriebssystem (Speicherformen) oder eine spezielle Software (Datenbank) hergestellt.

Zielvorstellung ist ein **wahlfreier** Zugriff auf die Daten, um z.B. direkt auf die Daten eines Kunden xyz zugreifen zu können, wenn dieser die Geschäftsräume betritt.

1. Datenorganisation

Grundbegriffe

Die Basis einer guten Informationsversorgung ist eine vernünftige Datenhaltung. Ziel der Datenorganisation ist es, Verfahren zur Verfügung zu stellen, die Daten bzw. Datenbestände eines Unternehmens

- zu strukturieren, d. h., hinsichtlich ihrer logischen Zusammenhänge zu analysieren und zu ordnen (**logische Datenorganisation**) und
- auf peripheren Speichergeräten zu speichern und zur Verfügung zu halten (**physische Datenorganisation**)

Ziele der logischen Datenorganisation

- einfache Aktualisierung
- beliebige Auswertung
- flexible Verknüpfung

Ziele der physischen Datenorganisation

- geringe Zugriffszeiten
- Ausnutzung der verfügbaren Speicherkapazitäten
- Schutz von Verlust, Zerstörung und unbefugtem Zugriff



1. Datenorganisation

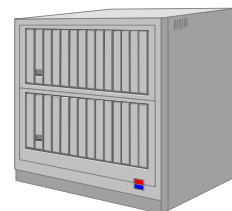
Logische Datenorganisation

Die logische Datenorganisation befasst sich mit Datenobjekten, die durch bestimmte Eigenschaften/ Attribute beschrieben werden:

- Personen (Kunden, Lieferanten, Mitarbeiter)
- Gegenstände (Rohstoffe, Maschinen, Gebäude)
- Abstrakte Begriffe: Konten, Buchungen, Rechnungen

Beispiele:

KUNDE	(Kd. Nr. ; Kd. Name; Kd. Vorname)
PERSONAL	(Per. Nr. ; Per. Name; Per. Verdienst)
KK-KONTO	(Kto. Nr. , Saldo, Datum)
KREDIT	(Kto. Nr. , Kreditbetrag, Restschuld, Datum, Kd. Nr.)
TRANSAKTION	(Tran. Nr. , Kto. Nr., Betrag, Datum)



1. Datenorganisation

Hierarchischer Aufbau

Alle Datenelemente desselben Objektes werden bilden einen logischen Datensatz:

Kredit (4711, 100.000 €, 90.000 €, 05.10.2002, 007)

Alle gleichartigen und logisch zusammengehörigen Datensätze werden zu einer Datei zusammengefasst:

Kredit (4711, 100.000 €, 90.000 €, 05.10.2002, 007)

(5318, 200.000 €, 55.000 €, 03.10.2002, 012)

(6717, 365.700 €, 265.000 €, 30.09.2002, 113)

(.....)

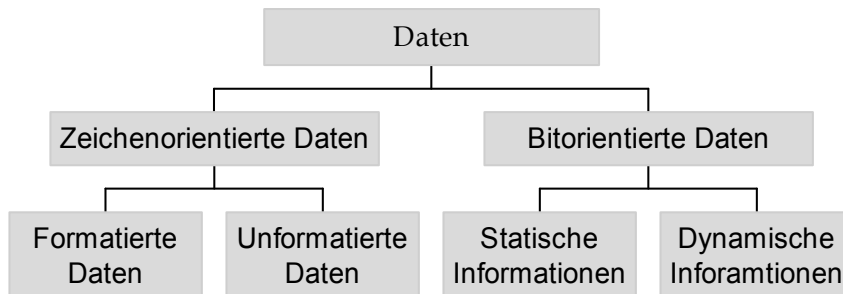
Eine Datenbank besteht aus mehreren Dateien, zwischen denen logische Abhängigkeiten bestehen:

Kredit	(4711, 100.000 €, 90.000 €, 05.10.2002, 007)	Kunde	(001, Müller, Peter)
	(5318, 200.000 €, 55.000 €, 03.10.2002, 012)		(007, Schweizer, Urs)
	(6717, 365.700 €, 265.000 €, 30.09.2002, 113)		(009, Gasser, Dörte)
	(.....)		(012, Schulte, Ines)
			(.....)

1. Datenorganisation

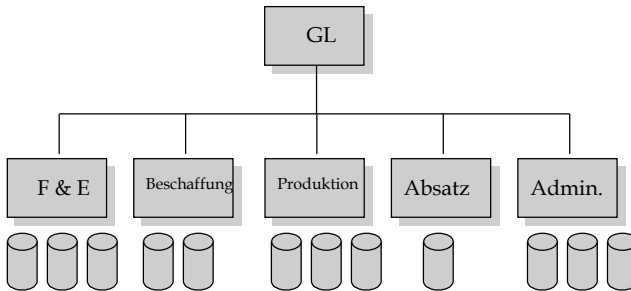
Erscheinungsformen von Daten

Die klassische Datenorganisation befasst sich nur mit zeichenorientierten Daten. Neuere Systeme allerdings integrieren auch bitorientierte Daten.



1. Datenorganisation Funktionsorientierte Vorgehensweise

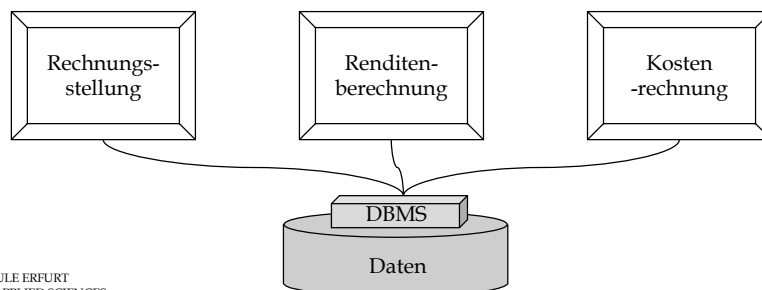
Die funktionsorientierte Vorgehensweise geht von den einzelnen betrieblichen Funktionen aus, für die entsprechende Programme benötigt und entwickelt werden. Jedes dieser Programme arbeitet mit bestimmten Daten, die in Dateien abgespeichert sind.



- Nachteile:
- Redundante Datenhaltung
 - Auswertungen nach mehreren Merkmalen aufwendig
 - Flexible Verknüpfungen schwierig

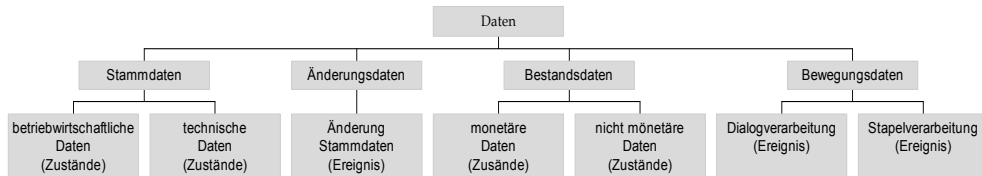
1. Datenorganisation Datenorientierte Vorgehensweise

Die datenorientierte Vorgehensweise fokussiert auf die Daten und deren logische Zusammenhänge. Alle benötigten Daten für eine Unternehmung oder einen Teilbereich werden definiert, in einem nächsten Schritt die Zusammenhänge abgeleitet und letztendlich erfolgt die Speicherung mittels einer Datenbank. Die Anwendungsprogramme werden anschließend entwickelt und greifen auf die Datenbank zu.



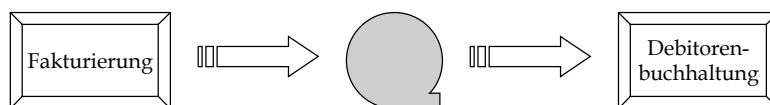
2. Dateiorganisation Verwendungszweck der Daten

Die funktionsorientierte Vorgehensweise führt zu Datenbeständen der jeweiligen Administrations- und Dispositionssysteme. Diese Daten unterscheidet man hinsichtlich ihres Verwendungszwecks wie folgt:

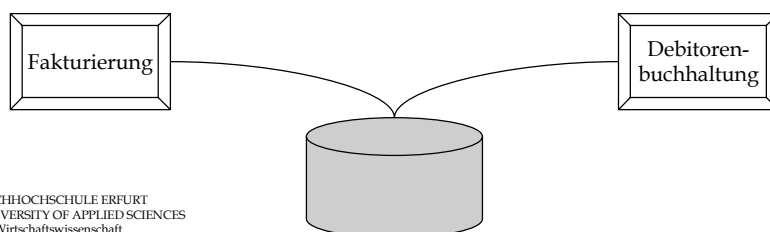


2. Dateiorganisation Einsatzformen

Programmintegrierte Verarbeitung



Dateiintegrierte Verarbeitung



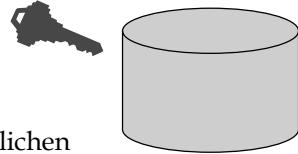
2. Dateiorganisation Nummerung

Jedes Datenobjekt muss eindeutig identifiziert werden können, um einen Datensatz zu extrahieren. Dies geschieht über eine Nummer (umgangssprachlich Schlüssel), die als

- alphabetischen Nummer
- numerischen Nummer oder
- alphanumerische Nummer

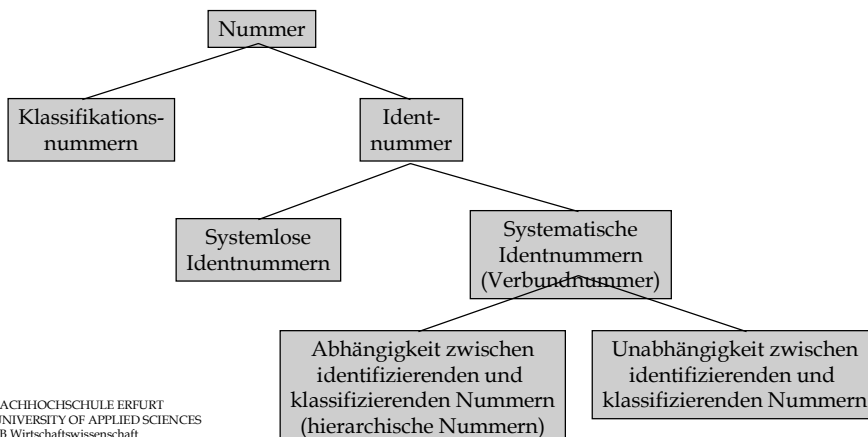
auftreten kann.

Die Nummerung ist ein zentrales Problem der betrieblichen Organisation, da berücksichtigt werden muss, welche Funktion die Nummer erfüllen und wie sie aufgebaut sein soll.



2. Dateiorganisation Ordnungsbegriffe

Jedes Datenobjekt in einer Unternehmung wird, sofern eine Speicherung von Daten hierüber vorgesehen ist, mit einem Ordnungsbegriff versehen (Kundennummer, etc.). Der Sinn einer solchen Nummer ist die eindeutige Identifikation und möglicherweise eine Klassifizierung.



2. Dateiorganisation

Primärschlüssel, Sekundärschlüssel, Match-Code

Die Nummer, die ein Datenobjekt eindeutig identifiziert, heißt Primärschlüssel.

Sekundärschlüssel sind Attribute oder Attributskombinationen, die i.d.R. nicht eindeutig identifizieren. Sie klassifizieren jedoch die Datenobjekte.

Match-Codes sind Kombinationen, die das Datenobjekt nicht genau aber fast genau identifizieren. Sie stellen den Versuch dar, aus bekannten Daten das richtige Datenobjekt zu identifizieren.

So kann bzw. ein Kunde über seinen Namen, Wohnort, Strasse identifizierbar sein, muss es aber nicht.

2. Dateiorganisation

Auswahl Speicherungsform

Ausschlaggebend für die Auswahl der sinnvollsten Speicherungsform sind:

- Verarbeitungs- und Zugriffsformen
- Eigenschaften der Datei
- Aufbau des Ordnungsbegriffs

Verarbeitungs- und Zugriffs- formen

Zugriff Verarb.	fortlaufend	wahlfrei
sortiert		
unsortiert		

2. Dateiorganisation

Eigenschaften der Dateien

Benutzungshäufigkeit

Zeitliche Häufigkeit, mit der die Datei benötigt wird.

Zugriffshäufigkeit

Zeitliche Häufigkeit, mit der auf einzelne Datensätze der Datei zugegriffen wird.

Veränderungshäufigkeit

Zeitliche Häufigkeit, mit der Zu- und Abgänge von Datensätzen vorkommen.

Bewegungshäufigkeit

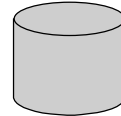
Zeitliche Häufigkeit, mit der Satzinhalte verändert werden.

Umfang

Anzahl der Datensätze und Länge der Datensätze.

Umfangsveränderungen

Anzahl der Zu- und Abgänge in einem bestimmten Zeitraum.

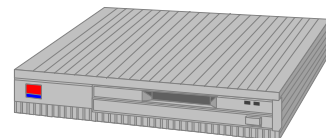
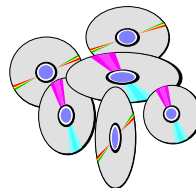


2. Dateiorganisation

Überblick und sequentielle Speicherung

Die folgenden Speicherungsformen werden unterschieden:

- sequentielle Speicherung
- verkettete Speicherung
- Indexverfahren
 - index-sequentielle Speicherung
 - index-verkettete Speicherung
 - Speicherung durch binäre Bäume
- gestreute Speicherung



Sequentielle Speicherung

- lückenlose Speicherung
- i.d.R. nach aufsteigendem Wert des Ordnungsbegriffes
- schwache Beziehung zwischen Ordnungsbegriff und physischer Speicheradresse
wg. fortlaufendem Ordnungsbegriff



2. Dateiorganisation Verkettete Speicherung

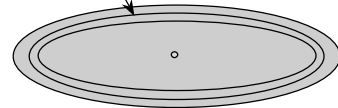
Verkettete Speicherung

- Zusätzliches Feld im Datensatz (Pointer) zeigt auf die physische Adresse des nachfolgenden logischen Datensatzes
- keine Beziehung zwischen Ordnungsbegriff und physischem Speicherplatz



Spur 2, Segment x

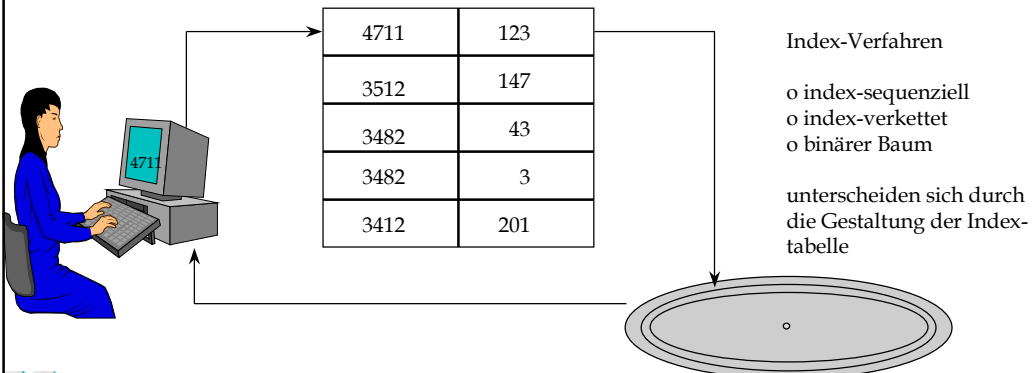
Nur noch in Verbindung mit der
Index-sequentiellen Speicherung
gebräuchlich



2. Dateiorganisation Indexverfahren

Index-Verfahren

Zusätzliche "kleine" Dateien (Indizes) werden zusätzlich zu dem eigentlichen Datenbestand abgespeichert. In diesen werden Informationen gehalten, die es ermöglichen, die Datensätze physisch zu finden.



Index-Verfahren

- o index-sequenziell
- o index-verkettet
- o binärer Baum

unterscheiden sich durch
die Gestaltung der Index-
tabelle

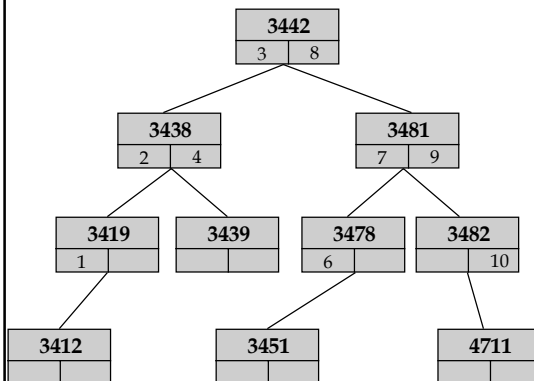


2. Dateiorganisation Index-sequentielle, Index-verkettete Speicherung

Position	Index-Tabelle		Index-Tabelle sortiert mit		Index-sequentiell		Index-verkettete		
	sortiert		sequentieller Speicherung		Speicherung		Speicherung		
NR	Kto.Nr.	Spur	Kto.Nr.	Spur	Kto.Nr.	Spur	Kto.Nr.	Spur	NF
1	3412	1,2	3412	0,1	3442	0,5	4711	3,3	7
2	3419	0,1	3419	0,2	4711	1,5	3481	1,1	8
3	3438	1,5	3438	0,3			3478	0,2	2
4	3439	0,4	3439	0,4			3451	2,4	3
5	3442	2,3	3442	0,5			3438	1,5	6
6	3451	2,4	3451	1,1			3439	0,4	10
7	3478	0,2	3478	1,2			3412	1,2	9
8	3481	1,1	3481	1,3			3482	0,5	1
9	3482	0,5	3482	1,4			3419	0,1	5
10	4711	3,3	4711	1,5			3442	2,3	4

2. Dateiorganisation Binärer Baum

Binärer Baum



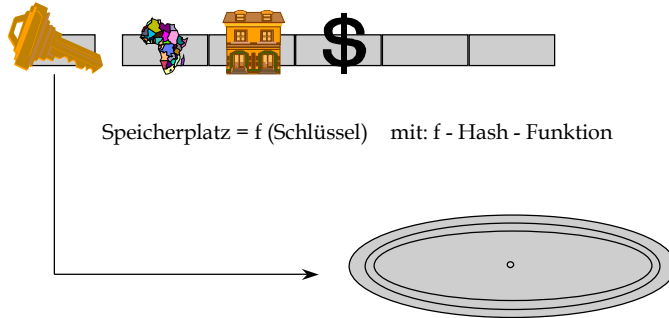
Indextabelle

Position	Geordneter				
	Baum				
NR	Kto.Nr.	Spur	LNF	RNF	
1	3412	1,2	"	"	
2	3419	0,1	1	"	
3	3438	1,5	2	4	
4	3439	0,4	"	"	
5	3442	2,3	3	8	
6	3451	2,4	"	"	
7	3478	0,2	6	"	
8	3481	1,1	7	9	
9	3482	0,5	"	10	
10	4711	3,3	"	"	

2. Dateiorganisation Gestreute Speicherung

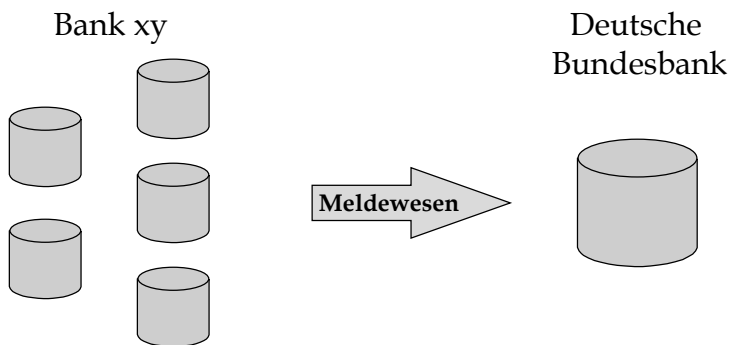
Gestreute Speicherungsformen

- Zusammenhang zwischen Ordnungsbegriff und physischem Speicherplatz



Ist die Hash - Funktion umkehrbar eindeutig, heißt das Verfahren **"Speicherung mit direkter Adressierung"**; ist sie nicht umkehrbar eindeutig, heißt es **"Speicherung mit indirekter Adressierung"**

2. Dateiorganisation Nachteile der Dateiorganisation



Informationen über

- o Kredite (Großkredite, Konsortialkredite)
- o Einlagen (Spareinlagen, Kündigungsgelder, etc)
- o Eigenkapital
- o etc.

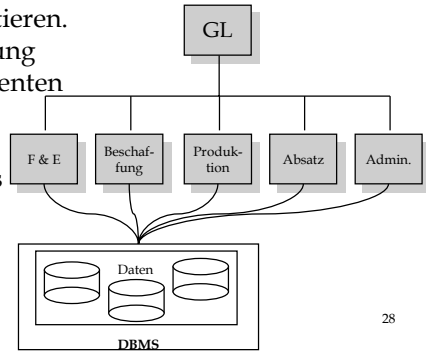
Probleme:

- o gleiche Datenelemente sind unterschiedlich benannt
- o Zusammenführen von Daten
- o flexibler Zugriff notwendig, um alle Kennzahlen bilden zu können

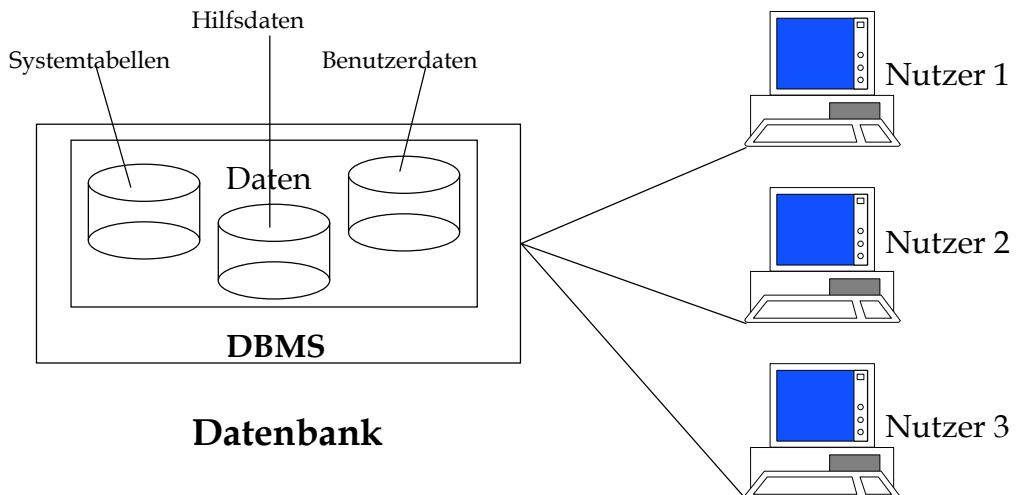
3. Datenbankorganisation Datenorientierte Vorgehensweise

Die Entwicklung von Datenbanken ermöglicht eine Trennung der Datenhaltung von der Bearbeitung der Daten. Damit ist die datenorientierte Vorgehensweise anwendbar.

Der erste Schritt bei der Entwicklung von Anwendungssystemen ist es hierbei, die Datenstrukturen zu entwickeln und sie entsprechend auf der Datenbank zu implementieren. Geschieht das für Teilbereiche der Unternehmung können hierdurch Redundanzen und Inkonsistenzen in den Daten weitgehend vermieden werden. Im Idealfall werden alle Daten einer Unternehmung in einer Datenbank gehalten, so dass ein unternehmensweiter Datenpool existiert.



3. Datenbankorganisation Schema eines Datenbanksystems



3. Datenbankorganisation Datenbanken

Entscheidet sich eine Unternehmung, ihre Daten auf einer Datenbank zu halten, ist es sinnvoll, sich als erstes mit der logischen Struktur der Daten zu befassen.

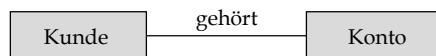
Die entwickelte Struktur kann dann auf einer Datenbank implementiert werden, wobei die folgenden logischen Datenbankmodelle unterschieden werden.

- **hierarchisches Datenbankmodell**
- **Netzwerk-Datenbankmodell**
- **relationales Datenbankmodell**

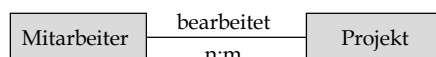
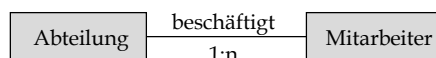
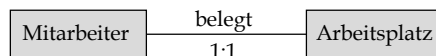
Die angeführten Datenbankmodelle unterscheiden sich hinsichtlich der Flexibilität der Zugriffe, oder anders ausgedrückt durch die Leistungsfähigkeit des DBMS. Hierbei stellt das relationale Datenbankmodell die größte Leistungsfähigkeit zur Verfügung und ist - für den Aufbau neuer Datenbanken - das meist verwendete Datenbankmodell.

3. Datenbankorganisation Beziehungen zwischen Daten

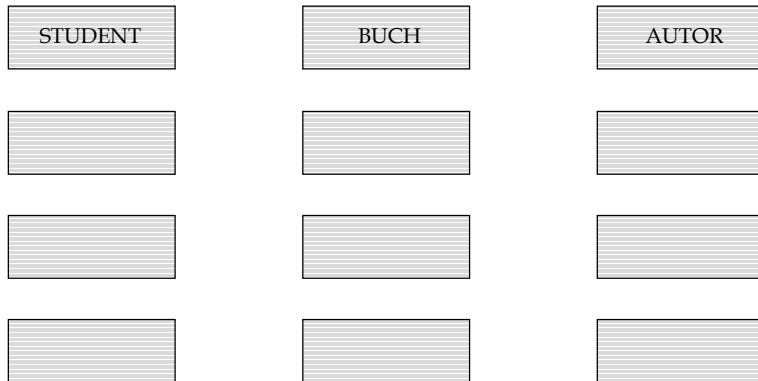
Zwischen Datenobjekten können Beziehungen existieren. Dem Kunden einer Bank gehört beispielsweise mindestens ein Konto.



Kardinalitäten von Beziehungen



3. Datenbankorganisation Bsp. Beziehungen zwischen Daten



3. Datenbankorganisation Hierarchische Datenbanksysteme

Ein hierarchisches Datenbankmodell zeichnet sich dadurch aus, daß alle Beziehungen durch eine Baumstruktur dargestellt werden. Jedes Datenobjekt hat genau einen Vorgänger (Ausnahme: oberste Stufe), kann aber mehrere Nachfolger haben.

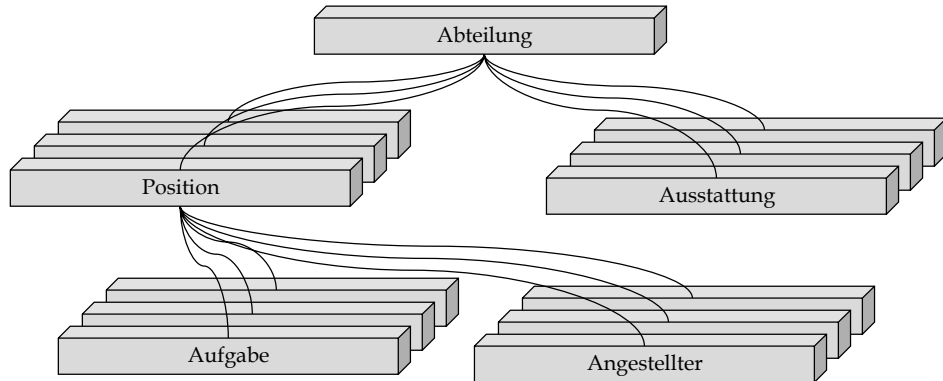
Mit einem solchen Datenbanktypen lassen sich 1:1 und 1:n Beziehungen abbilden.

In Unternehmen werden noch solche Datenbankmodelle betrieben, jedoch nicht mehr neu eingeführt.

Hierarchische Datenbanksysteme

IMS (Information Management System) vom IBM

3. Datenbankorganisation Bsp. Hierarchische Datenbanksysteme



Eine Abteilung speichert Datensätze über die Personalstellen und ihre Ausstattung. Jede Person steht in Beziehung zu einer Liste von Verantwortlichkeiten und zu der Person, welche die Position bekleidet.

3. Datenbankorganisation Netzwerk-Datenbanksysteme

Im Gegensatz zum hierarchischen Modell kann jedes Informationsobjekt mehrere Vorgänger haben, daraus resultiert, daß es mehrere Datenobjekte geben kann, die keinen Vorgänger aufweisen.

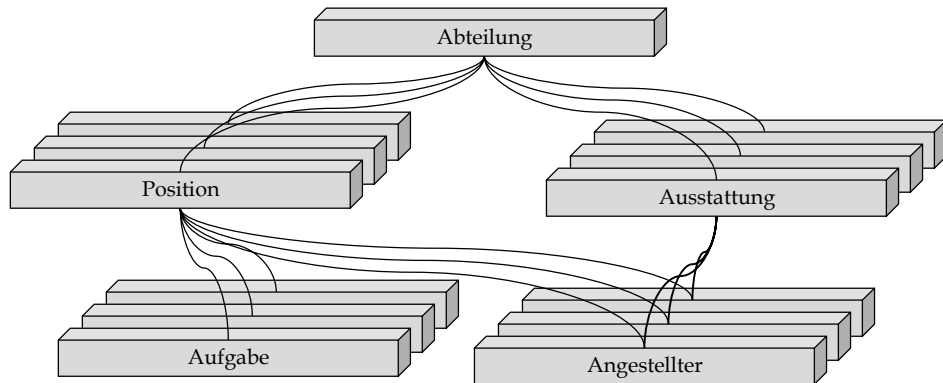
Im Netzwerkmodell lassen sich alle Kardinalitäten darstellen, d.h., es erlaubt die freie gegenseitige Verbindung der Daten, ohne dass diese vorher an eine Baumstruktur angepasst werden müssen.

Netzwerk-Datenbanksysteme

IDS (Integrated Data Base System) von Bull

UDS (Universelles Datenbanksystem) von Siemens

3. Datenbankorganisation Bsp. Netzwerk-Datenbanksysteme



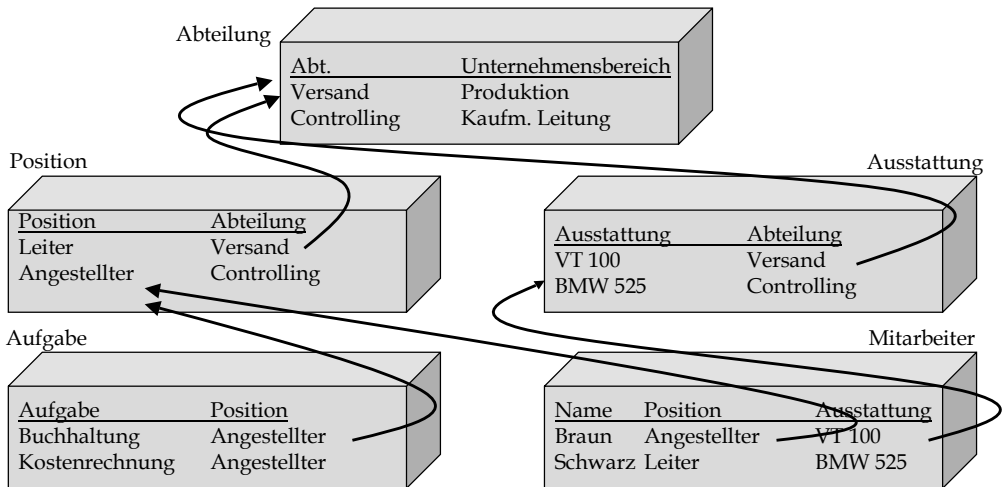
Jedes Teil in der Ausstattung kann jetzt sowohl mit der Abteilung als auch mit der Liste der für die Nutzung autorisierten Angestellten assoziiert werden.

3. Datenbankorganisation Relationale Datenbanksysteme

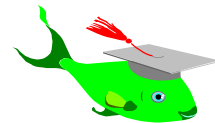
Die hierarchischen und die Netzwerk-Datenbank-Modelle machen die Darstellung komplexer Beziehungen zwischen den Datenobjekten einfach, aber dies hat seinen Preis: ein anderer Zugriff auf die Daten als durch die Beziehungen vordefiniert, ist langsam und ineffizient. Schlimmer noch, die Datenstrukturen sind schwer zu modifizieren; das Ändern der Beziehungen bedarf eines Systemadministrators, der die Datenbank schließt und neu aufbaut.

Das relationale Modell vermeidet diese Probleme durch Entfernen der Information über komplexe Beziehungen aus der Datenbank. Alle Daten sind in einfachen Tabellen gespeichert. Die grundlegenden Beziehungen zwischen den Datenobjekten werden wie Referenzen auf Werte in anderen Tabellen dargestellt.

3. Datenbankorganisation Bsp. Relationales Datenbanksystem



3. Datenbankorganisation Darstellung relationales Datenbanksystem



Dieses Modell wurde von Codd entwickelt und 1970 erstmals veröffentlicht. Eine graphische Darstellung für dieses Modell existiert nicht, sondern eine tabellarische oder auf der Mengentheorie basierende Darstellungsform.

Tabellarische Darstellungsform

Jede Relation ist eine zweidimensionale Tabelle, jede Reihe der Tabelle entspricht einem Tupel. Die Spalten der Tabelle entsprechen den Datenelementen, die das Informationsobjekt beschreiben.

Das relationale Modell eignet sich besonders gut für ad hoc Abfragen, da die Datenstruktur für flexible Zugriffe geeignet ist.

Relationale Datenbanksysteme:

DB2 von IBM

dBASE von

Borland

ADABAS von Software AG

Oracle von

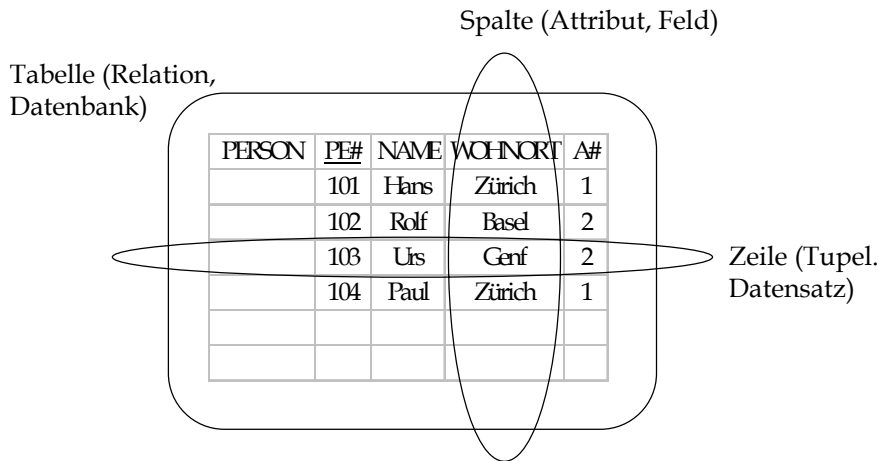
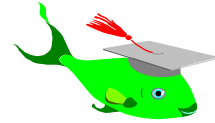
Oracle GmbH

Access von

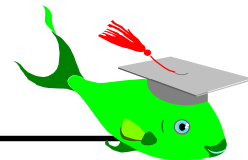
Microsoft



3. Datenbankorganisation Begriffe des Relationenmodells



3. Datenbankorganisation Regeln relationales Modell (Codd 1985)



Die wichtigsten Regeln, die Codd für das relationale Modell aufgestellt hat.

1. Jede Relation ist eine zweidimensionale Tabelle. Sie entspricht einem Datenobjekttyp.
2. Jede Zeile entspricht einem Tupel. Sie beschreibt ein bestimmtes Datenobjekt des entsprechenden Datenobjekttyps.
3. Die Spalten entsprechen den Attributen. Das Datenobjekt wird durch Attribute beschrieben.
4. Die Zeilen müssen paarweise untereinander verschieden sein.
5. Die Reihenfolge der Spalten/Reihen spielt keine Rolle.
6. Die Anzahl der Attribute heißt Grad der Relation.

3. Datenbankorganisation

Schlüssel

Ein Schlüssel ist jedes Attribut bzw. jede Attributskombination, die ein Tupel eindeutig identifiziert. Eine Attributskombination ist jedoch nur dann ein Schlüssel, wenn durch Entfernen eines beliebigen Schlüsselteils die Schlüsseleigenschaft verloren geht (**Minimaleigenschaft**).

Schlüsselkandidat

Alle Attribute oder Attributskombinationen, die geeignet sind, ein Tupel zu identifizieren, heißen Schlüsselkandidaten.

Primärschlüssel

Der Schlüssel der aus den Schlüsselkandidaten ausgewählt wird, um die Relation zu identifizieren, heißt Primärschlüssel.

Sekundärschlüssel

Attribute, die die Relation identifizieren oder klassifizieren, aber nicht Primärschlüssel sind.

Fremdschlüssel

Ein Attribut oder eine Attributskombination einer Relation, welches in einer anderen Relation den Primärschlüssel darstellt.

Matchcode

Ein i.d.R. aus mehreren Attributen zusammengesetzter Sekundärschlüssel, der zum Auffinden des zugehörigen Primärschlüssels bzw. der Daten eines bestimmten Datenobjektes dient. Der Matchcode ermöglicht das Eingrenzen der Objekte auf eine überblickbare Anzahl.



3. Datenbankorganisation

Beispieldatenbank MS-Access

