

Prof. Dr.-Ing. Dr. h. c. Theo Härder
AG Datenbanken und Informationssysteme
Zi. 36/330, Tel.: 0631-205-4030
E-Mail: haerder@informatik.uni-kl.de
<http://www.dvs.informatik.uni-kl.de/>

Foliensammlung
zur
Vorlesung

**Datenbanksysteme
für Hörer
anderer Fachrichtungen**

Wintersemester 2003/2004

TU Kaiserslautern
Fachbereich Informatik
Postfach 3049
67653 Kaiserslautern

Vorlesung:

Ort: 46–260

Zeit: Fr., 10.00–11.30 Uhr

Beginn: 31.10. 2003

Vorläufiges Inhaltsverzeichnis

1. Motivation und Grundbegriffe

- Rechnergestützte Informationssysteme
- Dateisysteme, Anforderungen an DBS
- Beschreibungsmodelle

2. Informationsmodellierung

- Konzepte des Entity-Relationship-Modells
- Abstraktionskonzepte

3. Grundlagen des Relationalen Datenmodells

- Datenstrukturen und Integritätsbedingungen
- Operationen
- Abbildung von Beziehungen

4. Die Standardsprache SQL

- Datendefinition und Datenmanipulation
- Sichtenkonzept
- Einbettung in eine Wirtssprache

5. Integritätssicherung und Transaktionskonzept

- Erhaltung der logischen Integrität
- Transaktionsverarbeitung
- Datenschutz und Zugriffskontrolle

6. Wichtige Datenbankanwendungen

LITERATUR

*Date, C.J., Darwen, H.: **A Guide to the SQL Standard**,
Addison-Wesley, 4th Edition, 1997*

*Dreßler, H.: **Datenstrukturentwurf – Vom Faktenchaos zur Anwender-
datenbank**, Oldenbourg-Verlag, 1995*

*Elmasri, R., Navathe, S. B.: **Grundlagen von Datenbanksystemen**,
3. Auflage, Pearson Studium, 2002*

*Kemper, A., Eickler, A.: **Datenbanksysteme – Eine Einführung**,
4. Auflage, Oldenbourg-Verlag, 2001*

*Meier, A.: **Relationale Datenbanken – Eine Einführung für
die Praxis**, 3. Auflage, Springer-Verlag, 1997*

*Sauer, H.: **Relationale Datenbanken – Theorie und Praxis**,
4. Auflage, Addison-Wesley, 1998*

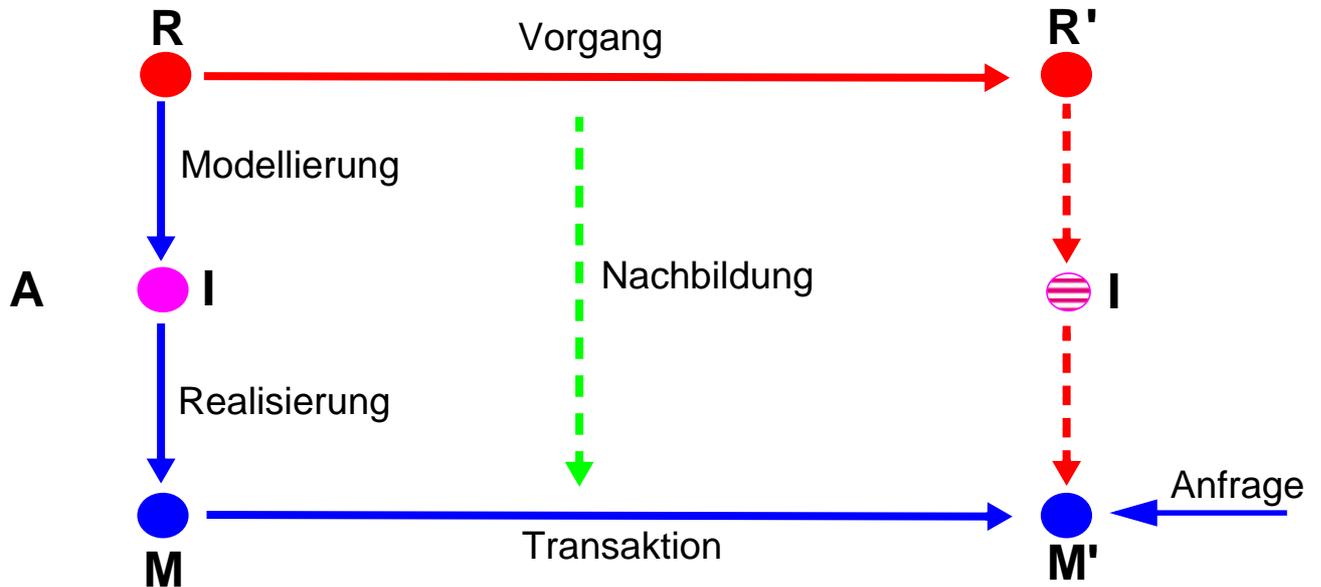
1. Motivation und Grundbegriffe

- **Modellhafte Abbildung einer Miniwelt¹**
- **Was ist ein Informationssystem?**
 - Information – Was ist das?
 - Begriffe, Komponenten eines rechnergestützten IS (KIS: kooperatives Informationssystem)
 - Beispiele
- **Nutzung von Dateisystemen**
- **Anforderungen an DBS**
 - Kontrolle über die operationalen Daten
 - Leichte Handhabbarkeit der Daten
 - Kontrolle der Datenintegrität
 - Leistung und Skalierbarkeit
 - Hoher Grad an Datenunabhängigkeit
- **Datenbanksysteme (DBS) – erste Annäherung**
 - Wichtige Eigenschaften
 - Beispiel: Relationenmodell
 - Künftige Datenbanksysteme
- **Ebenen beim Entwurf eines DBS**
 - Historische Entwicklung
 - Rolle der systematischen Abstraktion
- **Drei-Schema-Architektur**

1. Ein Datenbanksystem verwaltet Daten einer realen oder gedanklichen Anwendungswelt. Diese Daten gehen aus Informationen hervor, die stets aus den Sachverhalten und Vorgängen dieser Anwendungswelt durch gedankliche Abstraktionen (Abbilder, Modelle) gewonnen werden. Sie beziehen sich nur auf solche Aspekte des betrachteten Weltausschnitts, die für den Zweck der Anwendung relevant sind. Ein solcher Weltausschnitt wird auch als Miniwelt bezeichnet.

Miniwelt – modellhafte Abbildung

- Grobe Zusammenhänge



- R:** Realitätsausschnitt (Miniwelt)
- I:** Informationsmodell
(zur Analyse und Dokumentation der Miniwelt)
- M:** DB-Modell der Miniwelt
(beschrieben durch Objekt- und Beziehungstypen, ihre Ausprägungen sowie Integritätsbedingungen usw.)
- A:** Abbildung aller relevanten Objekte und Beziehungen
↳ Abstraktionsvorgang

- Transaktion:**

- bildet Vorgang in **R** im DBS nach und
- garantiert ununterbrechbaren Übergang von **M** nach **M'**
↳ implementiert durch Folge von DB-Operationen
- DB-Anfragen beziehen sich auf **M** bzw. **M'**

- Integritätsbedingungen:**

- Zusicherungen über **A**, **I** und **M**: $A_1: R \rightarrow I$, $A_2: I \rightarrow M$
↳ Ziel: möglichst gute Übereinstimmung von **R** und **M**
- Idealfall: Die DB ist zu jeder Zeit ein Abbild (Modell) der gegebenen Miniwelt

Miniwelt – modellhafte Abbildung (2)

- **Transaktionskonzept**

- führt ein neues Verarbeitungsparadigma ein
- ist Voraussetzung für die Abwicklung betrieblicher Anwendungen (*mission-critical applications*)
- erlaubt „Vertragsrecht“ in rechnergestützten IS zu implementieren

- **Welche Eigenschaften von Transaktionen sind zu garantieren? (ACID-Paradigma)**

- **Atomicity (Atomarität)**

- TA ist kleinste, nicht mehr weiter zerlegbare Einheit
- Entweder werden alle Änderungen der TA festgeschrieben oder gar keine („alles-oder-nichts“-Prinzip)

- **Consistency**

- TA hinterläßt einen konsistenten DB-Zustand, sonst wird sie komplett (siehe Atomarität) zurückgesetzt
- Zwischenzustände während der TA-Bearbeitung dürfen inkonsistent sein
- Endzustand muß die Integritätsbedingungen des DB-Modells erfüllen

- **Isolation**

- Nebenläufig (parallel, gleichzeitig) ausgeführte TA dürfen sich nicht gegenseitig beeinflussen
- Alle anderen parallel ausgeführten TA bzw. deren Effekte dürfen nicht sichtbar sein

- **Durability (Dauerhaftigkeit)**

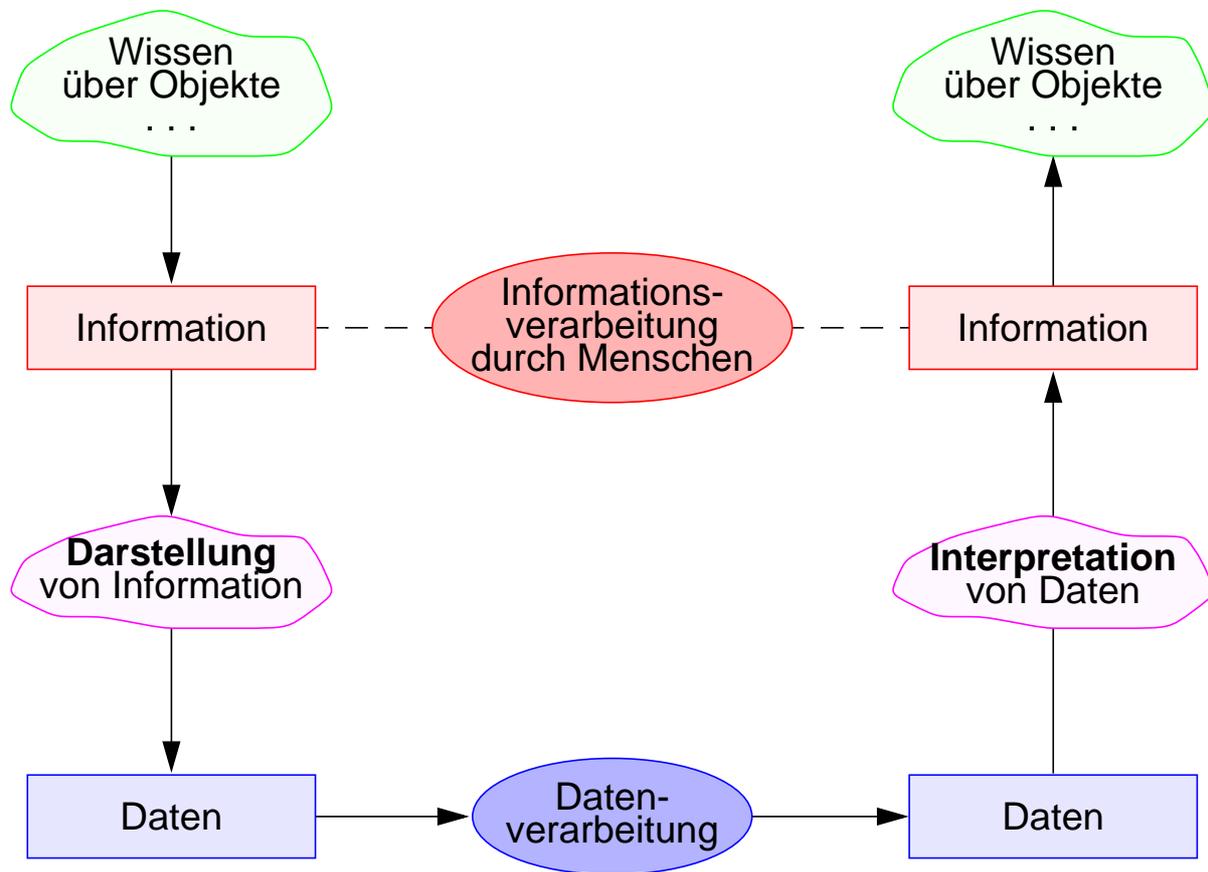
- Wirkung einer erfolgreich abgeschlossenen TA bleibt dauerhaft in der DB erhalten
- TA-Verwaltung muß sicherstellen, daß dies auch nach einem Systemfehler (HW- oder System-SW) gewährleistet ist
- Wirkungen einer erfolgreich abgeschlossenen TA kann nur durch eine sog. kompensierende TA aufgehoben werden

Information – Was ist das?

- Erklärungsversuche (pragmatisch):

1. Informationsbegriff nach DIN: **Erklärungsmodell**

(auch für Nachrichtenaustausch zwischen Sender und Empfänger)



Information: subjektive Welt der bewerteten Daten

Daten: objektive Welt der nicht-interpretierten Daten

2. Festlegung in der BWL²

Information: **Angaben über Sachverhalte und Vorgänge** (Hansen)

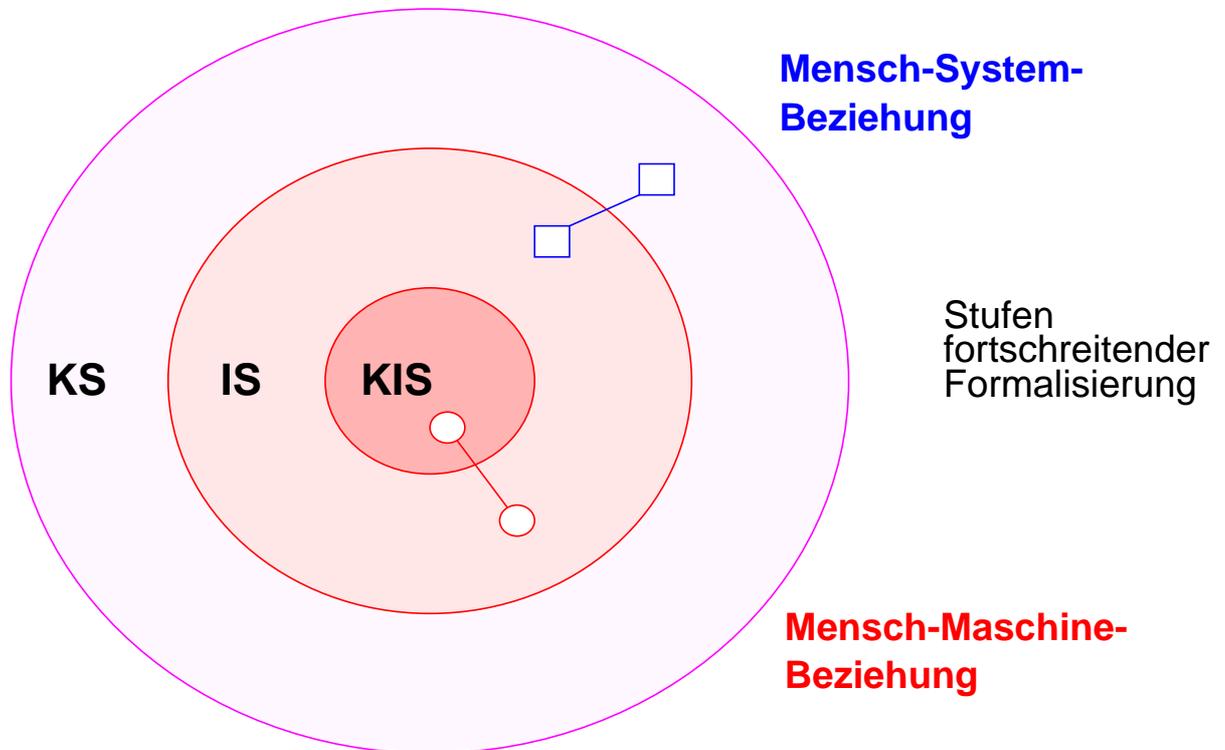
- „Dabei kann man Information im Sinne von als zweckgerichtetes Wissen zur Vorbereitung und Durchführung von Handlungen verstehen“.
- „Eine andere Sichtweise ergibt sich aus der Betrachtung der Information als Produktionsfaktor“.

2. Wirtschaftsinformatik-Lexikon, Gabler-Verlag, 1997

Was ist ein Informationssystem?

- **Charakterisierung eines IS nach Senko:**

“The purpose of an information system is to provide a *relatively exact, efficient, unambiguous* model of the significant resources of a real world enterprise.”



Formalisiertes Kommunikationsystem = Informationsystem

- **(Vage) Definitionen:**

Ein Informationssystem³ (IS) besteht aus Menschen und Maschinen, die Informationen erzeugen und/oder benutzen und die durch Kommunikationsbeziehungen miteinander verbunden sind.

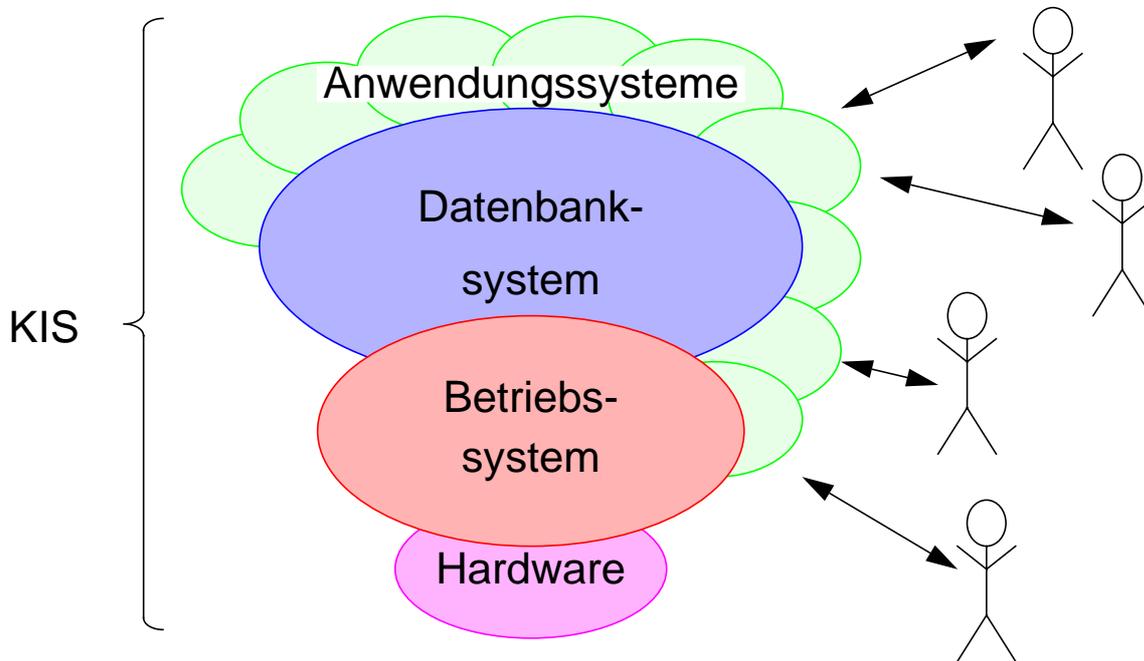
Ein betriebliches IS dient zur Abbildung der Leistungsprozesse und Austauschbeziehungen im Betrieb und zwischen dem Betrieb und seiner Umwelt.

Ein rechnergestütztes IS ist ein System, bei dem die Erfassung, Speicherung und/oder Transformation von Informationen durch den Einsatz von EDV teilweise automatisiert ist. In der betrieblichen Praxis besteht es typischerweise aus einer Menge unabhängiger Systeme, die zusammen die angestrebte Leistung erbringen (*KIS: kooperatives Informationssystem*).

3. Als „System im weiteren Sinne“ gilt (a) eine Menge von Elementen (Systembestandteilen), die (b) durch bestimmte Ordnungsbeziehungen miteinander verbunden und (c) durch klar definierte Grenzen von ihrer Umwelt geschieden sind; von „Systemen im engeren Sinne“ oder „technischem System“ spricht man, wenn sowohl die Außenwirkungen des Systems insgesamt wie auch seine Binnenstruktur (d. h. die Ordnungsbeziehungen der Systembestandteile) durch Zielfunktionen bestimmt sind (H. Wedekind).

Rechnergestützte Informationssysteme

- Aufbau von rechnergestützten Informationssystemen



Datenbanksystem (DBS): zentrale Komponente für KIS

Grobdefinition:

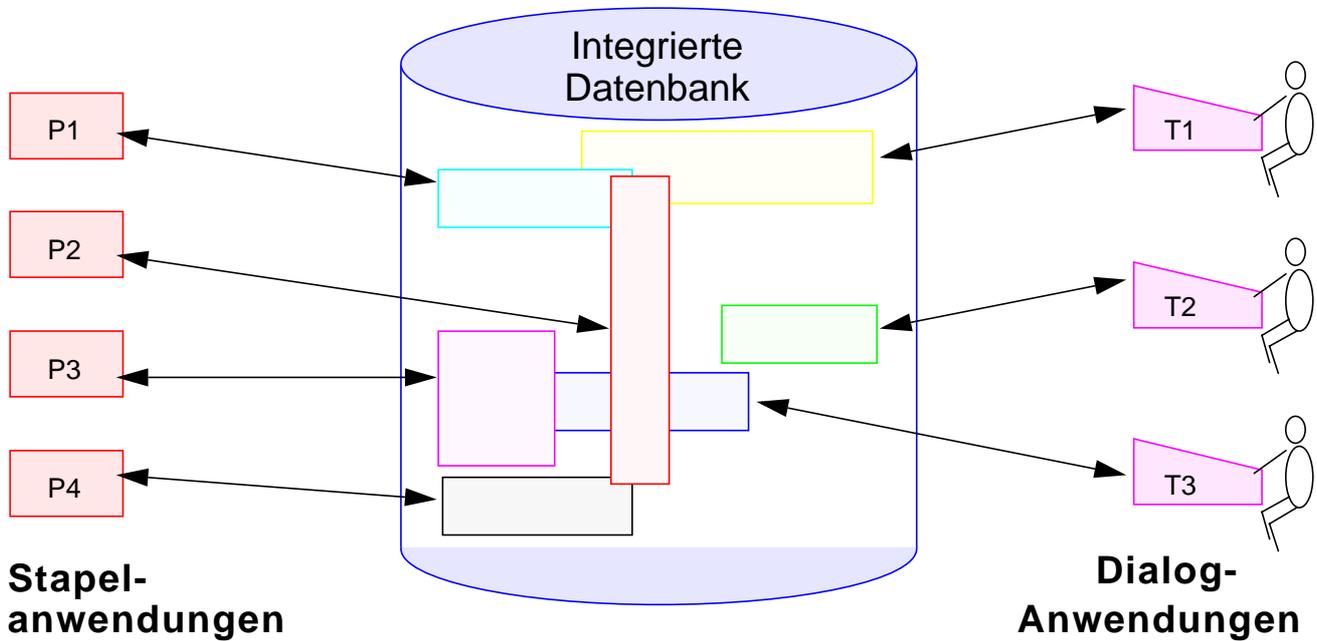
DBS = DB + Datenbankverwaltungssystem (DBVS, DBMS)

Eine **Datenbank** ist eine Sammlung gespeicherter operationaler Daten, die von den Anwendungssystemen eines bestimmten Unternehmens benötigt werden.

Ein **DBVS** ist ein standardisiertes Softwaresystem zur Definition, Verwaltung, Verarbeitung und Auswertung der DB-Daten. Es kann mittels geeigneter Parametrisierung an die speziellen Anwendungsbedürfnisse angepaßt werden.

Rechnergestützte Informationssysteme (2)

- Vereinfachte Sicht auf DB-Anwendungen (AW)



- Dialogorientierte Anwendungen:

- Transaktionsanwendungen mit Interaktion des Endbenutzers
- harte Zeitrestriktionen
- Terminal-E/A für Anforderungen und Ergebnisse

- Stapelorientierte Anwendungen:

- (Transaktions-) Anwendungen ohne Interaktion des Endbenutzers
- keine Zeitrestriktionen
- Datei-E/A für Anforderungen und Ergebnisse

Beispiele für Informationssysteme

Universitätsdatenbank

Die Universitätsdatenbank ist die Sammlung aller für die Abwicklung der an einer Universität anfallenden Verwaltungsaufgaben benötigten Daten.

Eine Universität gliedert sich i. allg. in mehrere Fachbereiche, denen sowohl die Studenten als auch die Professoren zugeordnet sind.

Die Studenten belegen verschiedene Vorlesungen von Professoren und legen bei ihnen Prüfungen ab.

Typische Anwendungen sind z. B.:

Immatrikulation der Studienanfänger, Rückmeldung der Studenten, Ausfertigen von Studentenausweisen und Studienbescheinigungen, Stundenplanerstellung und Planung der Raumbelugung, Ausstellen von (Vor)diplomzeugnissen, Exmatrikulationen, Statistiken über Hörerzahlen, Raumauslastung, Prüfungsergebnisse, etc.

Datenbank eines Produktionsbetriebes

In einem Produktionsbetrieb werden Daten über die verschiedenen Abteilungen und deren Beschäftigte mit ihren Familienangehörigen gespeichert.

Die Angestellten arbeiten an verschiedenen Projekten mit. Jedes Projekt benötigt für seine Durchführung bestimmte Teile. Jedes Teil kann von Lieferanten bezogen werden. Die Projekte werden jeweils von einem Projektmanager geleitet.

Die in einem Betrieb hergestellten Endprodukte setzen sich i.allg. aus mehreren Baugruppen und Einzelteilen zusammen.

Typische Anwendungen sind z. B.:

Einstellung und Entlassung von Personal, Lohn- und Gehaltsabrechnung, Bestellung und Lieferung von Einzelteilen, Verkauf von Fertigprodukten, Lagerhaltung, Bedarfsplanung, Stücklistenauflösung, Projektplanung.

Beispiele für Informationssysteme (2)

Datenbank einer Fluggesellschaft

Eine Fluggesellschaft fliegt verschiedene Flughäfen an. Auf diesen Flugstrecken werden Flugzeuge bestimmter Typen mit dafür ausgebildetem Personal eingesetzt. Die Piloten haben Flugscheine jeweils nur für einige wenige Flugzeugtypen. Außer den Piloten gibt es noch anderes Bord- sowie Bodenpersonal.

Die Flugbuchungen der Passagiere sowie das Anfertigen der Passagierlisten werden ebenfalls automatisiert durchgeführt.

Typische Anwendungen sind z. B.:

Flugbuchungen von Passagieren, Personaleinsatzplanung, Materialeinsatzplanung, Flugplanerstellung, Überwachung der Wartelisten, Gehaltsabrechnung.

Datenbank einer Bank

Eine Bank gliedert sich gewöhnlich in mehrere Zweigstellen auf. Die Angestellten der Bank gehören jeweils fest zu einer bestimmten Zweigstelle. Auch die Bankkunden sind immer einer Zweigstelle zugeordnet. Es sind Daten über die verschiedenartigen Konten der Bankkunden bereitzustellen, wie z. B. Girokonten, Sparkonten, Hypothekenkonten, Kleinkreditkonten, Wertpapierkonten, etc.

Typische Anwendungen sind z. B.:

Buchung von Zahlungsvorgängen auf den verschiedensten Konten, Einrichten und Auflösen von Konten, Kreditgewährung bzw. Bereitstellen von Daten über die Kreditwürdigkeit eines Kunden, Zinsberechnung und -verbuchung, sowie alle Vorgänge der Personalverwaltung wie z. B. Gehaltsabrechnung.

Zur Rolle rechnergestützter Informationssysteme im Bankenbereich:

“In banking, by contrast, the data actually is the inventory – the two are synonymous. In increasingly many cases, the DB transaction is the financial transaction. There are no real, tangible tokens (greenbacks) moved as a result of the monetary transfer transaction. If the data is bad, money is lost or created. There is no possibility of counting the money (bits) in order to verify the status. Fiscal responsibility dictates that creating or destroying money – even temporarily – is unacceptable.”
(Mike Burman, Bank of America)

Daten in Informationssystemen – strukturierte Daten

- **Strukturierte oder formatierte Daten:**

(NAME = „Müller“, TAETIGKEIT = „Kalligraph“, GEBDAT = „780623“, . . .)

- maximale Länge (= endlicher Wertevorrat)
- Werte von Variablen, Feldern, Attributen; durch Namen beschrieben
- Bedeutung weitgehend vorgegeben, relativ geringer Informationsgehalt

- **Klassische Datenbanktechnik**

- formatierte Datenstrukturen, feste Satzstruktur
- Beschreibung der Objekte durch Satztyp, Attribute und Attributwerte ($S_i/A_j/AW_k$)
- jeder Attributwert AW_k wird durch Beschreibungsinformation (Metadaten) A_j und S_i in seiner Bedeutung festgelegt
- **Beispiel:** Tabelle (Relation in Tabellendarstellung)

Schema

ANGESTELLTER

Ausprägungen

Satztyp (Relation)

PNR	NAME	TAETIGKEIT	GEHALT	ALTER
496	PEINL	PFOERTNER	2100	63
497	KINZINGER	KOPIST	2800	25
498	MEYWEG	KALLIGRAPH	4500	56

➔ **DB-Schema:** vollständige Strukturbeschreibung (Metadaten) ist vor der Speicherung von Objekten zu spezifizieren und dem DBS bekannt zu machen

- **Art der Anfrage und Aktualisierung**

- deskriptiv (nicht-prozedural)
- mengenorientiert

Daten in Informationssystemen

– unstrukturierte Daten

- **Unstrukturierte oder unformatierte Daten:**

„Er heißt Müller. Er wird in seiner Arbeitszeit überwiegend als Kalligraph eingesetzt. Geboren ist er am 23. Juni des Jahres 1978 . . .“

- beliebige Länge
- teilweise selbstbeschreibend
- Bedeutung nur schwach vorgegeben
- hoher Informationsgehalt

- **Dokumente in Information-Retrieval-Systemen**

- unformatierte Daten, keine dem IRS bekannte Dokumentstruktur
- Beschreibung der Objekte durch Dokumenttyp und Wert (D_i/W_k)
- Es gibt keine nähere Beschreibung oder Spezifikation von Struktur und Semantik, die W_k in seiner Bedeutung festlegt

➔ IRS verwaltet „lange“ Werte (z. B. Texte eines Abschnitts, Kapitels oder Buches) und stellt dafür Container (verschiedenen Typs) zur Verfügung

- **Aufgaben/Eigenschaften von IRS**

- Verwaltung von Dokumenten, Büchern, Abstracts usw.
- effiziente Suche in großen Datenmengen
- typischerweise nur Retrieval im Mehrbenutzerbetrieb
- Anfragesprache für Retrieval
(➔ Annäherung an natürliche Sprache erwünscht)

Daten in Informationssystemen

– semi-strukturierte Daten

- **HTML-Dokumente im WWW**

- Aufbau des Dokumentes (syntaktische Struktur) ist festgelegt
- Formatierungsanweisungen (Tags) lassen grobe Rückschlüsse auf den Inhalt des Dokumentes zu: TITLE, HREF, ADDRESS, ...
- Inhalt des Dokumentes ist jedoch nicht weiter beschrieben; es gibt keine Metadaten, die die Bedeutung genauer festlegen

➔ **WWW-Browser kann HTML-Dokumente aufbereiten und graphisch darstellen, ohne den Inhalt zu kennen**

- **Beispiel: HTML-Dokument, semi-strukturiert**

```
<!DOCTYPE HTML PUBLIC „-//W3C//DTD HTML ...“>
```

```
<HTML>
```

```
<HEAD>
```

```
  <TITLE>Publications 2002</TITLE>
```

```
  <META NAME="GENERATOR" CONTENT="Mozilla/3.01Gold (...) [Netscape]">
```

```
</HEAD>
```

```
<BODY BACKGROUND="http://wwwdbis.informatik.uni-kl.de/pics/paper.jpg">
```

```
<H1 ALIGN=CENTER>Publications 2002</H1>
```

```
<HR NOSHADE>
```

```
<H3>Last update: 2003/10/06 </H3>
```

```
<P>Hergula, K., H&auml;rder, T.:<BR>
```

```
<A HREF="papers/HH02.EDBT.html">Coupling of FDBS and WfMS for Integrating Database and Application Systems: Architecture, Complexity, Performance</A>, in: Proc. 8th Int. Conf. on Extending Database Technology (EDBT'2002), Prague, March 2002, pp. 372-389.</P>
```

```
<ADDRESS><A HREF="mailto:wwwhaerder@informatik.uni-kl.de">wwwhaerder@informatik.uni-kl.de</A>
```

```
</ADDRESS>
```

```
</BODY>
```

```
</HTML>
```

➔ **Präsentation an der Benutzerschnittstelle: für Menschen, nicht für Programme**

Daten in Informationssystemen

– semi-strukturierte Daten

- **Formatierungssprachen dienen dem Austausch von Dokumenten**

- Es gibt eine Vielzahl von Formatierungssprachen, die alle den internationalen Standard zur Textverarbeitung SGML als Meta-Sprache benutzen, um ihre Formate und Grammatik zu definieren

- **HTML**

- ist eine Sprache zur Formatierung (Strukturierung) von Dokumenten (Texten) (HyperText Markup Language, Tag Language)
- bietet eine vorgegebene Menge von Begrenzungs- und Formatierungsanweisungen (>200) mit standardisierter Bedeutung
- **Beispiel**
 - <H2>Second-Level heading </H2>
 - <P>This is a passage of text that probably belongs to the heading immediately above </P>
- vermischt **Strukturierungs- und Darstellungsaufgaben**
- kann die Suche von Dokumenten kaum unterstützen

- **DocBook**

- weitere Sprache zur Textformatierung (Software-Dokumentation)
- **Beispiel**
 - <SECT2>
 - <TITLE>Second-level heading </TITLE>
 - <PARA> This is a passage of text that certainly belongs to heading above. We know this because both are contained in the same SECT2 element.
 - </PARA>
 - </SECT2>

➔ Jede Sprache ist auf eine bestimmte Kategorie von Dokumenten zugeschnitten

Daten in Informationssystemen

– Multimedia-Daten

- **Bedarf**

Verwaltung großer Mengen von Multimedia-Datenobjekten auf verschiedenartigen Datenträgern (optischen Speichern, Videobändern, ...), so daß sie für möglichst viele Anwendungen auffindbar und zugreifbar (nutzbar) sind

- **Was sind Multimedia-Datenobjekte?**

Digitalisierte, im Rechner abgelegte Bilder und Tonaufnahmen sowie Texte und Graphiken; Videoaufzeichnungen, Hologramme, Radarsignale, ...

- **Ziel**

- einerseits Erhöhung der „Informations-Bandbreite“, bessere Benutzerschnittstellen
- aber auch mehr Information im System:

➔ **Inhalt eines Bildes (einer Tonaufnahme, ...) ist niemals vollständig in Medien wie Text oder Graphik wiederzugeben**

- **Bezeichnungen**

- **Medienobjekt** (oder Medien-Datenobjekt)
Ein Datenobjekt, das einem einzigen Medium angehört, also ein einzelnes Bild, ein Textstück
- **Multimedia-Objekt**
(Multimedia-Datenobjekt, auch „mixed-mode object“)
Aggregation (Komposition) von Medienobjekten unterschiedlichen Typs, z. B. Video (Bild + Ton)
- **Multimedia-Daten**
Sammelbegriff für Medienobjekte und Multimedia-Objekte

- **Medienobjekte**

sind aus formatierten und unformatierten Daten zusammengesetzt

Beispiel: Medienobjekt „Rasterbild“

- **Rohdaten**

- Matrix von Bildpunkten
(Pixel = Picture Element, auch „Pel“ genannt)

- **Registrierungsdaten**

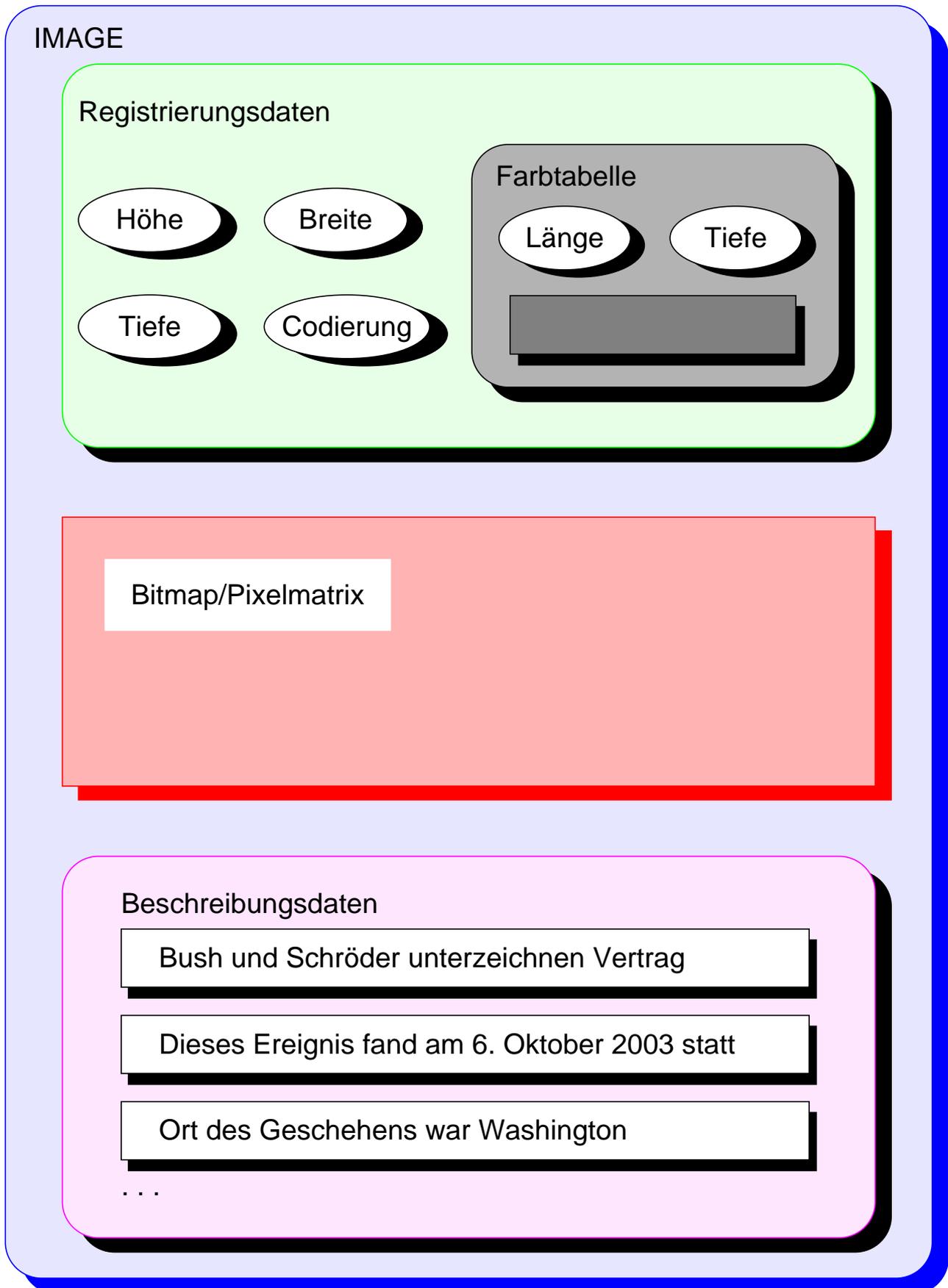
- Anzahl Bits pro Pixel („Farbtiefe“, typisch: 1, 8, 24);
- Anzahl Pixel pro Zeile (Breite des Bildes);
- Anzahl der Zeilen (Höhe des Bildes);
- Art der linearen Abspeicherung
 - zeilenweise oder spaltenweise;
- Bedeutung eines Pixels:
 - Grauwert, Farbdefinition, Index einer Farbtabelle (Palette, Colormap);
- ggf. Farbtabelle mit bestimmter Anzahl von Einträgen und Länge dieser Einträge (meist 24 Bit)
- ggf. Art der Farbdefinition: RGB, IHS, YIQ, ...
und anderes mehr

- **Beschreibungsdaten**

- Text, Schlüsselworte, Wissensrepräsentation
- Graphik: erkannte Linien und Flächen
- darauf aufbauend zweidimensionale Objekte wie Kreise, Rechtecke usw.

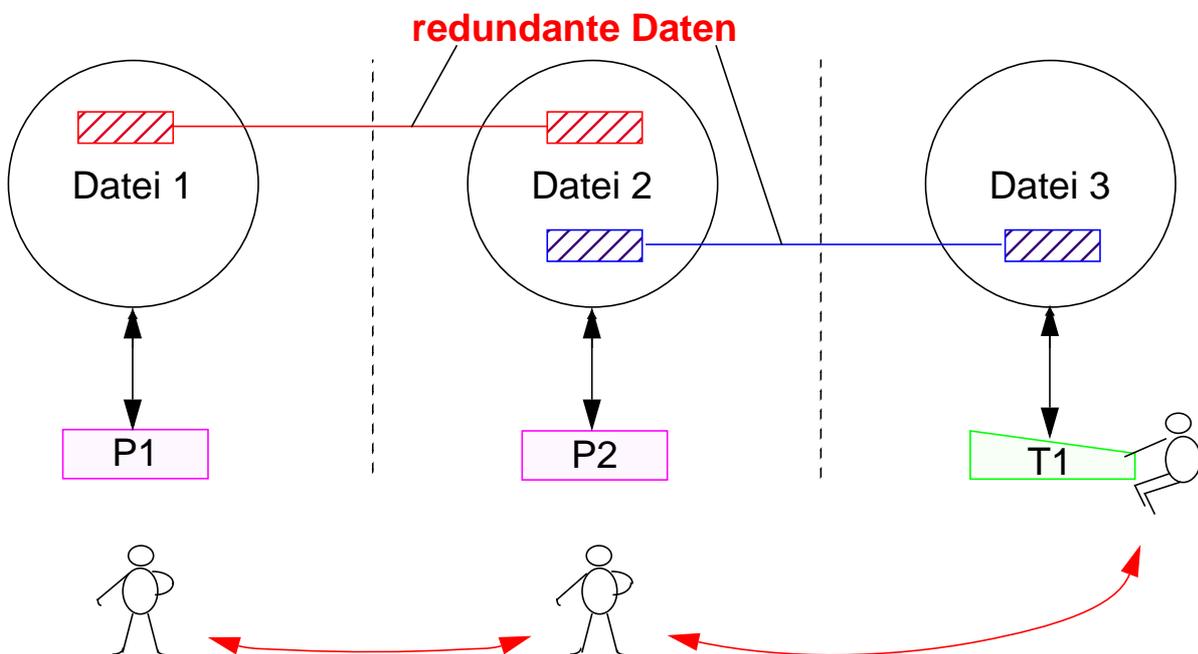
Beispiel: Medienobjekt „Rasterbild“

Abstrakte Sicht auf eine Instanz vom Datentyp IMAGE:



Nutzung von Dateisystemen

- permanente Datenhaltung innerhalb von Betriebssystem-Dateien
- Betriebssystem/Dateisystem bietet Funktionen für
 - Erzeugen / Löschen von Dateien
 - Zugriffsmöglichkeiten auf Blöcke/Sätze der Datei
 - einfache Operationen zum Lesen/Ändern/Einfügen/Löschen von Sätzen (dynamisches Wachstum)
 - verschiedene Dateitypen für sequentiellen, indexsequentiellen und direkten Zugriff
 - Verwaltung von Dateiverzeichnissen
 - Vergabe von Zugriffsrechten
 - Komprimierung, Verschlüsselung, ...



Kommunikation notwendig für Änderungen

Probleme des Dateikonzeptes

- **Beispiel**

Eine Bank hält Informationen über ihre Kunden und deren Sparkonten in Dateien *Kunde* und *Sparkonto*. Sie enthalten jeweils Sätze eines entsprechenden Typs:

TYPE

Kunde = RECORD

<i>Ku#</i>	: 1000. .9999;
<i>K-Name</i>	: char (29);
<i>K-Adr</i>	: char (40);
<i>Einkommen</i>	: integer;
<i>Gebu-Dat</i>	: integer;
<i>Fam-Stand</i>	: (vh, nv);

END;

Sparkonto = RECORD

<i>Sp#</i>	: 1. .9999;
<i>Betrag</i>	: integer;
<i>Ku#</i>	: 1000. .9999;

END;

Regel: Jeder Kunde hat maximal zwei Sparkonten

- **Probleme**

- Datenredundanz und Inkonsistenz
- Inflexibilität
- Mehrbenutzerbetrieb
- Fehlerfall
- Integritätssicherung
- Mißbrauch der Daten

➔ **Wer ist verantwortlich?**

Probleme des Dateikonzeptes (2)

- Dateien sind auf die Bedürfnisse einzelner Anwendungen zugeschnitten

↳ Datenabhängigkeit, Inflexibilität

Eine neue Anwendung Kreditwerbung benötigt zu jedem Kunden seine Sparkonten:

```
TYPE KuSpako= RECORD
    Ku#           : 1000. .9999;
    K-Name        : char (20);
    K-Adr         : char (40);
    Hauptkto      : RECORD
                    Sp#       : 1. .9999;
                    Betrag    : integer;
    END;
    Nebenkto      : RECORD
                    Sp#       : 1. .9999;
                    Betrag    : integer;
    END;
END;
```

↳ Redundanz

- Wie wird eine sortierte Liste der Kunden, die mehr als 5000,- DM auf dem Konto haben, erstellt?
- Lösung gleicher Aufgaben in allen Anwendungsprogrammen
 - Speicherverwaltung
 - Retrieval und Änderungsdienst
 - Schutzfunktionen, . . .
- Annahmen:

Alles bleibt stabil !

Alles geht gut !

Anforderungen an ein DBS

1. Kontrolle über die operationalen Daten

- **Alle Daten können/müssen gemeinsam benutzt werden**
 - keine verstreuten privaten Dateien
 - Querauswertungen aufgrund inhaltlicher Zusammenhänge
 - Entwicklung neuer Anwendungen auf der existierenden DB
 - Erweiterung/Anpassung der DB (Änderung des Informationsbedarfs)
- **Eliminierung der Redundanz**
 - Vermeidung von Inkonsistenzen
 - zeitgerechter Änderungsdienst/keine unterschiedlichen Änderungsstände
- **Datenbankadministrator (DBA) hat zentrale Verantwortung für die operationalen Daten**

Anforderungen an ein DBS (2)

2. Leichte Handhabbarkeit der Daten

- **Einfache Datenmodelle**

- Beschreibung der logischen Aspekte der Daten
- neutral gegenüber allen Anwendungen

- **Leicht erlernbare Sprachen**

- deskriptive Problemformulierung
- hohe Auswahlmächtigkeit
- Unterstützung der Problemlösung des Anwenders im Dialog

- **Logische Sicht der Anwendung**

- zugeschnitten auf ihren Bedarf
- lokale Sicht auf die DB

- **Durchsetzung von Standards (SQL)**

- unterschiedliche DBS bieten einheitliche Schnittstelle
- Portierbarkeit von Anwendungen
- erleichterter Datenaustausch

- **Erweiterung der Benutzerklassen**

- Systempersonal
- Anwendungsprogrammierer
- Anspruchsvolle Laien
- Parametrische Benutzer
- Gelegentliche Benutzer

Beispiel – Relationenmodell

Schema

FB	<u>FBNR</u>	FBNAME	DEKAN		
STUDENT	<u>MATNR</u>	SNAME	FBNR	STUDBEG	
PRÜFUNG	<u>PNR</u>	<u>MATNR</u>	FACH	DATUM	NOTE

Ausprägungen

FB	<u>FBNR</u>	FBNAME	DEKAN		
	FB 9	WIRTSCHAFTSWISS	4711		
	FB 5	INFORMATIK	2223		

STUDENT	<u>MATNR</u>	SNAME	FBNR	STUDBEG	
	123 766	COY	FB 9	1.10.00	
	225 332	MÜLLER	FB 5	15.04.97	
	654 711	ABEL	FB 5	15.10.99	
	226 302	SCHULZE	FB 9	1.10.00	
	196 481	MAIER	FB 5	23.10.00	
	130 680	SCHMID	FB 9	1.04.02	

PRÜFUNG	<u>PNR</u>	<u>MATNR</u>	FACH	PDATUM	NOTE
	5678	123 766	BWL	22.10.03	4
	4711	123 766	OR	16.01.02	3
	1234	654 711	DV	17.04.03	2
	1234	123 766	DV	17.04.03	4
	6780	654 711	SP	19.09.03	2
	1234	196 481	DV	15.10.03	1
	6780	196 481	BS	23.10.03	3

Q1: Finde alle Studenten aus FB 5, die ihr Studium vor 2000 begonnen haben.

```

SELECT *
FROM STUDENT
WHERE FBNR = 'FB5' AND STUDBEG < '01.01.00'
    
```

Q2: Finde alle Studenten des FB 5, die im Fach Datenverwaltung eine Note 2 oder besser erhalten haben.

```

SELECT *
FROM STUDENT
WHERE FBNR = 'FB5' AND MATNR IN
    (SELECT MATNR
     FROM PRÜFUNG
     WHERE FACH = 'DV' AND NOTE ≤ '2')
    
```

Anforderungen an ein DBS (3)

3. Kontrolle der Datenintegrität (system enforced integrity)

- **Automatisierte Zugriffskontrollen (Datenschutz)**

- separat für jedes Datenobjekt
- unterschiedliche Rechte für verschiedene Arten des Zugriffs
- **Idealziel:** "least privilege principle"

- **Erhaltung der logischen Datenintegrität**

- Beschreibung der "Richtigkeit" von Daten durch Prädikate
- Qualitätskontrollen bei Änderungsoperationen

- **Erhaltung der physischen Datenintegrität**

- Führen von Änderungsprotokollen für den Fehlerfall (Logging)
- Bereitstellen von Wiederherstellungsalgorithmen im Fehlerfall (Recovery)

- **Notwendigkeit des kontrollierten Mehrbenutzer-Betriebs**

- logischer Einbenutzer-Betrieb für jeden von n parallelen Benutzern (Leser + Schreiber)
- angepaßte Sperreinheiten (Sperrgranulate) mit abgestuften Zugriffsrechten
- **Ziel:** möglichst geringe gegenseitige Behinderung

➔ **Durchsetzung durch Transaktionskonzept**

Anforderungen an ein DBS (4)

4. Leistung und Skalierbarkeit

- **DBS-Implementierung gewährleistet**

- **Effizienz** der Operatoren (möglichst geringer Ressourcenverbrauch)
- **Verfügbarkeit** der Daten (Redundanz, Verteilung usw.)

- **Effizienz des Datenzugriffs**

- Zugriffsoptimierung durch das DBS, nicht durch den Anwender
- Auswahl von Zugriffspfaden durch den DBA

↳ idealerweise durch das DBS

- **Leistungsbestimmung**

- Maßzahlen für Leistung
 - Durchsatz: Anzahl abgeschlossener TA pro Zeiteinheit (meist Sekunde)
 - Antwortzeit: Zeitbedarf für die Abwicklung einer TA
- Rolle von Benchmarks⁴: TPC-C, TPC-H, TPC-W, TPC-R, . . .

- **Skalierbarkeit**

- Software- und Hardware-Architektur sollen hinsichtlich des DBS-Leistungsverhaltens automatisch durch Hinzufügen von Ressourcen (CPU's, Speicher) skalieren
 - **Scaleup**: bei Wachstum der Anforderungen (DB-Größe, Transaktionslast)
 - **Speedup**: zur Verringerung der Antwortzeit

4. Transaction Processing Council: www.tpc.org

Anforderungen an ein DBS (5)

5. Hoher Grad an Datenunabhängigkeit

- **Konventionelle Anwendungsprogramme (AP) mit Dateizugriff**

- Nutzung von Kenntnissen der Datenorganisation und Zugriffstechnik
- gutes Leistungsverhalten, aber

- **Datenabhängige Anwendungen sind äußerst unerwünscht**

- verschiedene Anwendungen brauchen verschiedene Sichten auf dieselben Daten
- Speicherungsstruktur und Zugriffsstrategie müssen durch den DBA oft den geänderten Leistungsanforderungen angepaßt werden
- **sonst:** extremer Wartungsaufwand für die Anwendungsprogramme

↳ deshalb:
möglichst starke Isolation der APs von den Daten

- **Realisierung verschiedener Arten von Datenunabhängigkeit:**

- **Geräteunabhängigkeit**
- **Speicherungsstrukturunabhängigkeit**

↳ **Minimalziel: physische Datenunabhängigkeit**
(durch das Betriebssystem/Datenbanksystem)

- **Zugriffspfadunabhängigkeit**
- **Datenstrukturunabhängigkeit**

↳ **logische Datenunabhängigkeit**
(vor allem durch das Datenmodell!)

Grobcharakterisierung von DBS

• Aufgaben/Eigenschaften

- Verwaltung von persistenten Daten (lange Lebensdauer)
- effizienter Zugriff (Suche und Aktualisierung) auf große Mengen von Daten (GBytes – TBytes)
- flexibler Mehrbenutzerbetrieb
- Verknüpfung / Verwaltung von Objekten verschiedenen Typs (→ typübergreifende Operationen)

• Datenstrukturen

- Beschreibung der Objekte durch Satztyp, Attribute und Attributwerte ($S_i/A_j/AW_k$)
- jeder Attributwert AW_k wird durch Beschreibungsinformation (Metadaten) A_j und S_i in seiner Bedeutung festgelegt.

Schema

ANGESTELLTER

Satztyp (Relation)

Ausprägungen

PNR	NAME	TAETIGKEIT	GEHALT	ALTER
496	PEINL	PFOERTNER	2100	63
497	KINZINGER	KOPIST	2800	25
498	MEYWEG	KALLIGRAPH	4500	56

→ **DB-Schema:** vollständige Strukturbeschreibung (Metadaten) ist vor der Speicherung von Objekten zu spezifizieren und dem DBS bekannt zu machen

• Art der DB-Sprache

- abhängig von Datenmodell
- formale Sprache
 - navigierend oder deskriptiv
 - satz- oder mengenorientiert

Grobcharakterisierung von DBS (2)

- **Datenmodell/DBS-Schnittstelle**

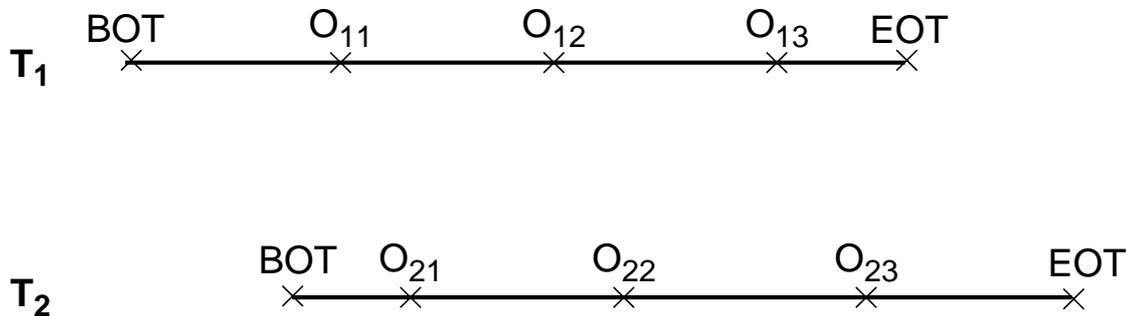
- Operationen zur Definition von Objekttypen (Beschreibung der Objekte)
 - ↳ DB-Schema: Welche Objekte sollen in die DB gespeichert werden?
- Operationen zum Aufsuchen und Verändern von Daten
 - ↳ AW-Schnittstelle: Wie erzeugt, aktualisiert und findet man DB-Objekte?
- Definition von Integritätsbedingungen (*Constraints*)
 - ↳ Sicherung der Qualität: Was ist ein akzeptabler DB-Zustand?
- Definition von Zugriffskontrollbedingungen
 - ↳ Maßnahmen zum Datenschutz: Wer darf was?

- **Art der Suche**

- Zeichen-/Wertvergleich:
(TAETIGKEIT = 'PFOERTNER') AND (ALTER > 60)
- **exakte** Fragebeantwortung:
alle Sätze mit spezifizierter Eigenschaft werden gefunden
(und nur solche)
- Synonymsuche, unscharfe Suche, usw.
werden nicht unterstützt
- Verstehen natürlicher Sprache, Erkennen von Mehrdeutigkeiten sind
nicht Thema von formatierten DBS
(↳ Information-Retrieval-Systeme)
Bsp.: "Time flies like an arrow –
fruit flies like a banana" (Groucho Marx)

Grobcharakterisierung von DBS (3)

- Ablaufkontrollstruktur: Transaktion



BOT: Begin of Transaction

EOT (Commit): End of Transaction

O_{ij} : DB-Operation; Lese- und Schreiboperationen auf DB-Daten

mit den Eigenschaften

- **A**tomicity: Alles-oder-Nichts
- **C**onsistency: Gewährleistung der Integritätsbedingungen
- **I**solated Execution: „logischer Einbenutzerbetrieb“
- **D**urability: Persistenz aller Änderungen

- Einsatzformen

- zentralisiertes DBS, Hauptspeicher-DBS
- Mehrrechner-DBS (lokal oder ortsverteilt)
- Hochleistungs-Transaktionssysteme, . . .

Beispiel – Objekt-Relationales Datenmodell

- **Integrierte Suche über Inhalt**

- SQL ermöglicht den einheitlichen Zugriff auf herkömmliche und neue Datentypen
- Eine Anfrage kann sich auf ALLE Datentypen zugleich erstrecken
- Es können dabei benutzerdefinierte Datentypen und Funktionen ausgenutzt werden

- **Intuitives Anfragebeispiel**

„Finde die ‘bärtigen’ Studenten (mit FB-Angabe), deren Heimatort mehr als 50 km vom Studienort entfernt liegt und die über Schemaentwurf von relationalen Datenbanksystemen mit der Note 2 oder besser geprüft wurden“

```
SELECT S.Sname, S.Eingeschrieben_in->FBname
FROM   STUDENT S
WHERE  CONTAINS(S.Geprüft->Protokoll, "Datenbanksystem"
                IN SAME SENTENCE AS
                ("Schema" AND "Entwurf" AND "relational"))
AND S.Geprüft->Note <= 2
AND SHOWS(S.Passbild, "mit Bart")
AND DISTANCE(S.HAdr, S.Eingeschrieben_in->Adresse) < km (50);
```

Textdaten

herkömmliche Attribute

Bilddaten

räumliche Daten

WWW-basiertes
Verarbeitungsmodell



Transaktions-
verarbeitung

Client/Server-
Verarbeitungsmodell



Objektorientiertes
Verarbeitungsmodell



**Next-Generation
DBMS**

```

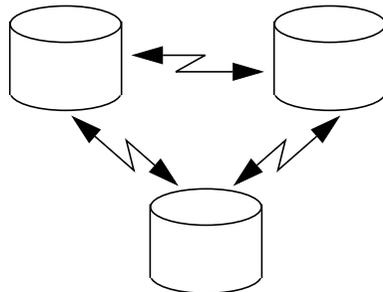
SELECT Unfall.Fahrer, Unfall.Vers-Nummer
FROM Unfall, Autobahn
WHERE CONTAINS(Unfall.Bericht, "Schaden"
               IN SAME SENTENCE AS
               ("schwer" AND "vordere" AND "Stoßstange"))
AND FARBE(Unfall.Foto, 'rot') > 0.6
AND ABSTAND(Unfall.Ort, Autobahn.Ausfahrt) < miles (0.5)
AND Autobahn.Nummer = A8;

```

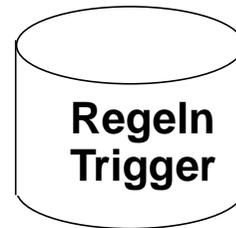
The Big Picture

1 - 33

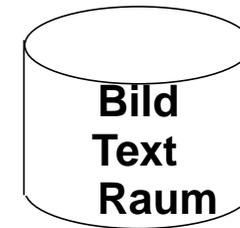
Parallele und Verteilte DBS



Aktive DBS

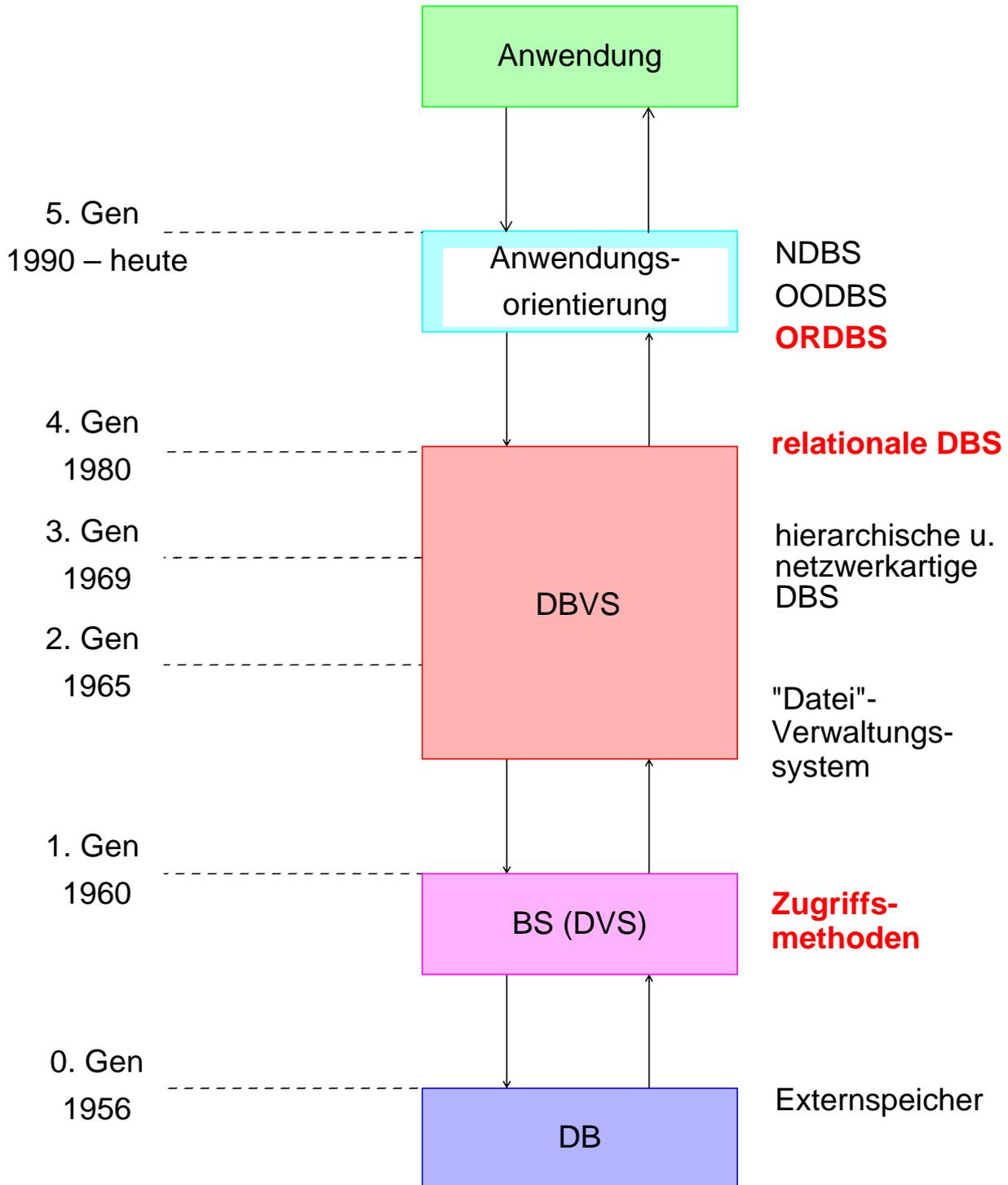


Multimedia-DBS

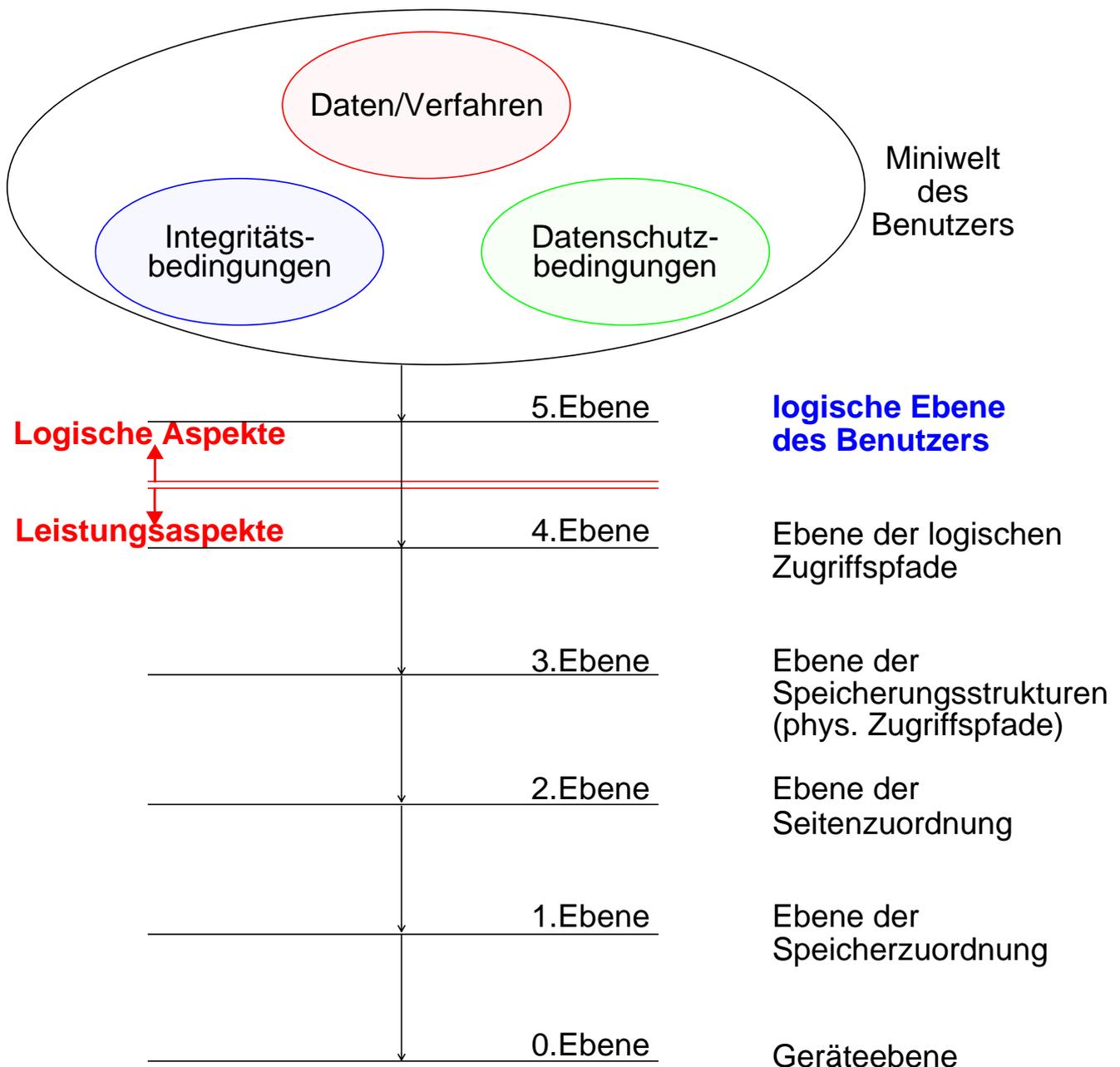


Ebenen beim Entwurf eines DBS

- Historische Entwicklung bei DBS



Ebenen beim Entwurf eines DBS⁵



• „outside-in“-Ansatz (top-down)

- Die Ausdrucksmächtigkeit des Datenmodells und seine Konzepte sowie die postulierten Betriebseigenschaften bestimmen die Anforderungen, die an das zu entwerfende DBS zu stellen sind
- Beim Entwurf erfolgt eine mehrstufige Strukturverfeinerung, bis die konkrete Implementierungsstruktur abgeleitet ist
- Idealerweise erhält man eine hierarchische Systemstruktur entsprechend der "benutzt"-Relation

5. „Eine Hauptaufgabe der Informatik ist systematische Abstraktion“ (H. Wedekind)

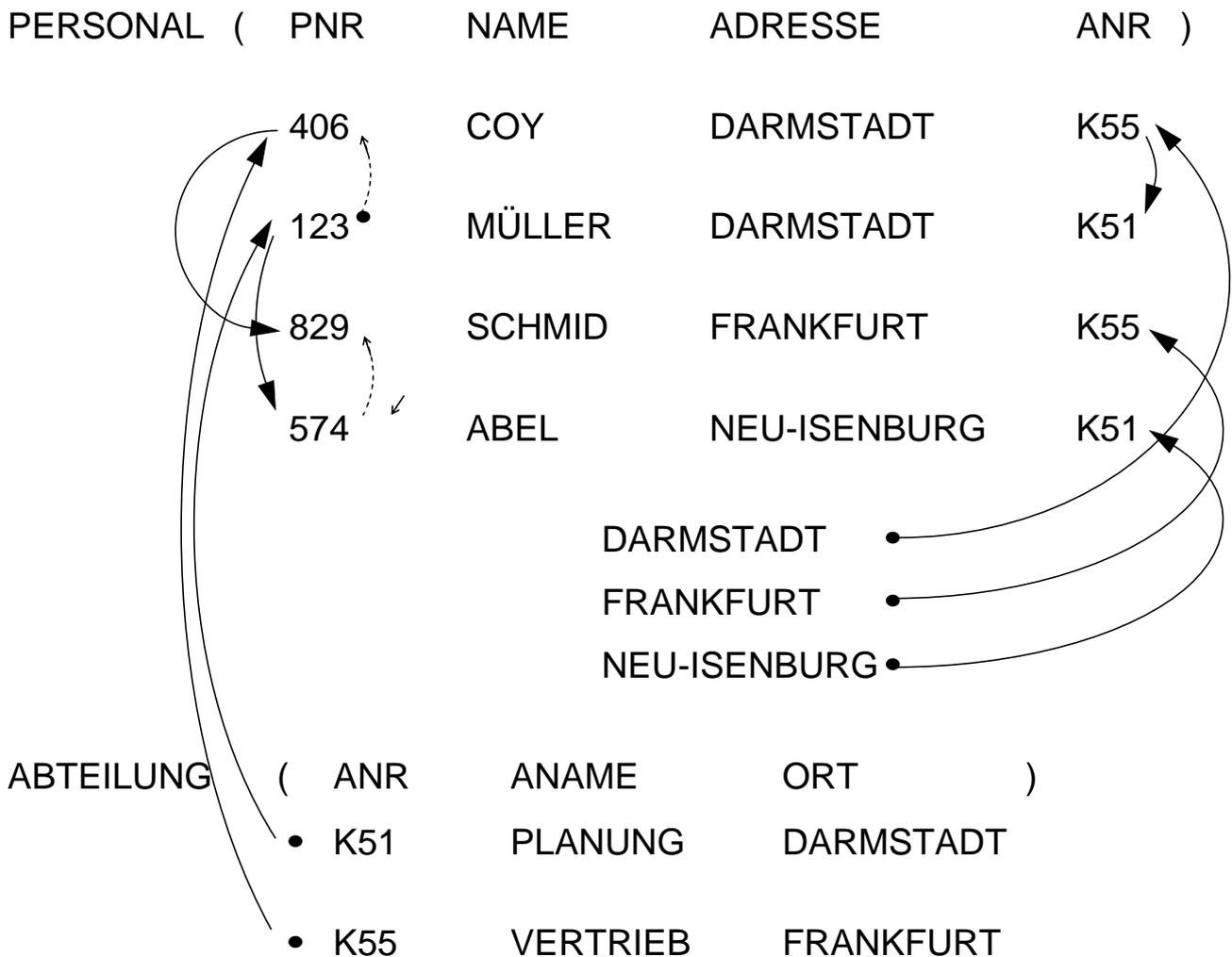
Verschiedene Sichten auf DBS-Daten

- Logischen Datenstrukturen eines Anwendungsbeispiels

PERSONAL	(PNR	NAME	ADRESSE	ANR)
		406	COY	DARMSTADT	K55	
		123	MÜLLER	DARMSTADT	K51	
		829	SCHMID	FRANKFURT	K55	
		574	ABEL	NEU-ISENBURG	K51	

ABTEILUNG	(ANR	ANAME	ORT)
		K51	PLANUNG	DARMSTADT	
		K55	VERTRIEB	FRANKFURT	

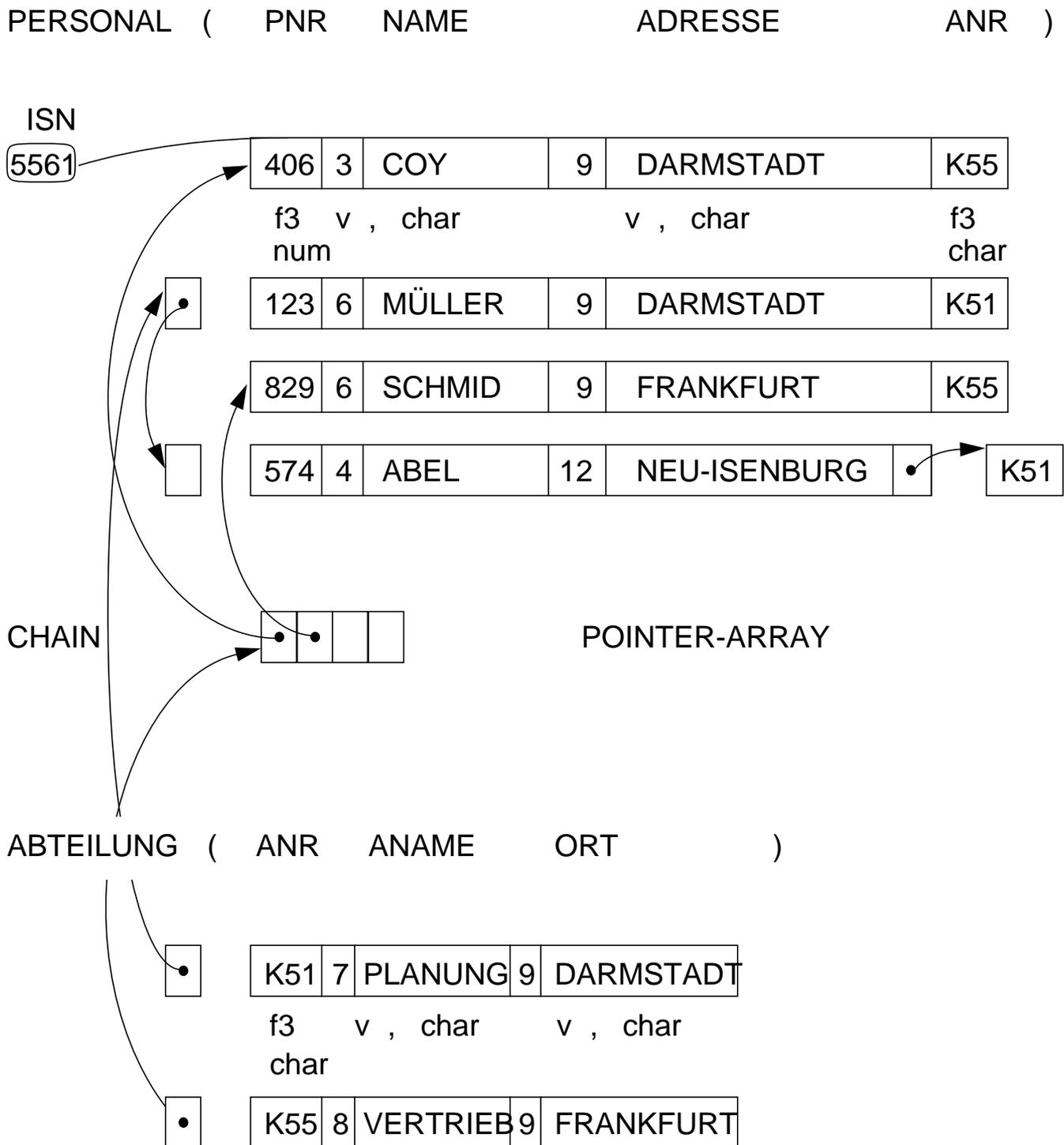
- Sicht auf die logischen Zugriffspfade



- Logische Zugriffspfade:**
1. OWNER – MEMBER
 2. Sortierreihenfolge PNR ASC
 3. Search Key (Invertierung ADRESSE)

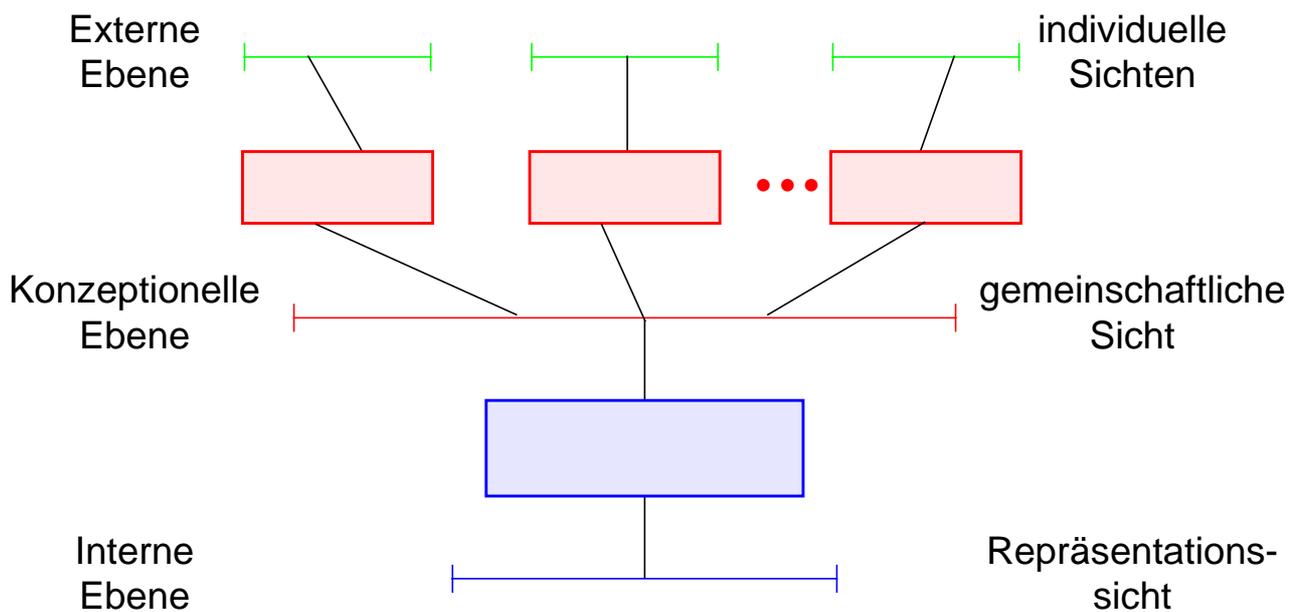
Verschiedene Sichten auf DBS-Daten (2)

- **Sicht auf die Speicherungsstrukturen**



- Speicherungsstrukturen:**
1. Formate
 2. Datentypen
 3. Implementierungstechniken

Drei-Schema-Architektur ⁶



Grobarchitektur für Schnittstellen nach ANSI/SPARC

- **Konzeptionelles Schema:**

- (zeitvariante) globale Struktur; neutrale und redundanzfreie Beschreibung in der Sprache eines spezifischen Datenmodells
- Beschreibungssprache: *DDL (Data Definition Language)*

- **Externes Schema:**

- Definition von zugeschnittenen Sichten auf Teile des konzeptionellen Schemas für spezielle Anwendungen (Benutzer)

- **Internes Schema:**

- legt physische Struktur der DB fest (physische Satzformate, Zugriffspfade etc.)
- Beschreibungssprache: *SSL (Storage Structure Language)*

6. „Die durch Abstraktion entstandenen Konstrukte der Informatik als Bedingungen möglicher Information sind zugleich die Bedingungen der möglichen Gegenstände der Information in den Anwendungen“ (H. Wedekind in Anlehnung an eine Aussage Kants aus der „Kritik der reinen Vernunft“) Vereinfacht ausgedrückt: Informatiker erfinden (konstruieren) abstrakte Konzepte; diese ermöglichen (oder begrenzen) wiederum die spezifischen Anwendungen.

Drei-Schema-Architektur (2)

- Stark vereinfachtes Beispiel für die Datenbeschreibung

Extern (PL/1) DCL 1 PERS, 2 PNR CHAR(6), 3 GEH FIXED BIN(31) 4 NAME CHAR(20)	Extern (COBOL) 01 PERSC. 02 PERSNR PIC X(6). 02 ABTNR PIC X(4).
Konzeptionelles Schema ANGESTELLTER ANG_NUMMER CHARACTER (6) NAME CHARACTER (20) ABT_NUMMER CHARACTER (4) GEHALT NUMERIC (5)	
Internes Schema SPEICHERUNG_PERS LENGTH=40 PREFIX TYPE=BYTE(6), OFFSET=0 PNR TYPE=BYTE(6), OFFSET=6, INDEX=PNR NAME TYPE=BYTE(20), OFFSET=12 ANR TYPE=BYTE(4), OFFSET=32 GEHALT TYPE=FULLWORD, OFFSET=36	

- **Sichtenbildung durch das Externe Schema**
 - **Anpassung der Datentypen** an die der Wirtssprache (DBS ist „multi-lingual“)
 - **Zugriffsschutz:** Isolation von Attributen, Relationen, ...
 - **Reduktion der Komplexität:** nur die erforderlichen Daten sind für das Anwendungsprogramm sichtbar

Zusammenfassung

- **Datenverwaltung durch Dateisysteme unzureichend**
- **DBS-Charakteristika**
 - Zentralisierte Kontrolle über die operationalen Daten (Rolle des DBA)
 - Adäquate Schnittstellen (Datenmodell und DB-Sprache)
 - Hoher Grad an Datenunabhängigkeit
 - Transaktionskonzept (ACID):
 - Atomarität
 - Konsistenzerhaltung
 - kontrollierter Mehrbenutzerbetrieb (Isolation)
 - Dauerhaftigkeit (Persistenz) erfolgreicher Änderungen
 - Effizienz
- **Drei-Ebenen-Architektur**
 - Externes Schema
 - Konzeptionelles Schema
 - Internes Schema
- **Programmierschnittstellen (DB-Sprachen)**
 - satzorientierte DB-Schnittstelle
 - ↳ hierarchische/netzwerkartige DBS
 - mengenorientierte DB-Schnittstelle
 - ↳ relationale DBS