Einführung in Datenbanksysteme

Prof. Dr. Ralf Möller
Prof. Dr. Joachim W. Schmidt
Technische Universität Hamburg-Harburg
Arbeitsbereich Softwaresysteme (STS)

Donnerstag 12:00-13:30, HS20
Freitag 10:00-10:45, HS20

Übung

Atila Kaya, Michael Wessel
14:00 - 15:30, HS20

t

Kapitel 1: Einführung

Motivation: Warum Datenbanken?

„Keine größere Informatikanwendung ist ohne DB-Unterstützung denkbar."

"DB-Systeme sind heute ein selbstverständliches Hilfsmittel der betrieblichen Organisation und Verwaltung."

"Datenbanken als Schlüsseltechnologie für effiziente, komplexe Informationssysteme.“

„Kein multimediales online Content-Management ohne Datenbanktechnologie.“

Kennzeichen der Daten

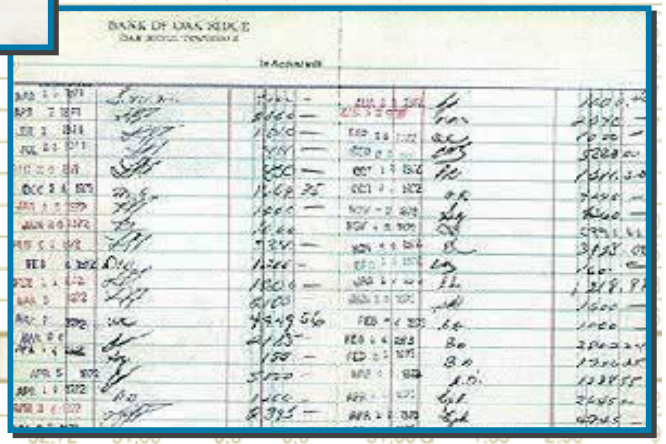
- ❑ lange Lebensdauer (Jahre, Jahrzehnte)
- ❑ reguläre Strukturen
- ❑ große Datenobjekte, große Datenmengen
- ❑ stetig anwachsende integrierte Bestände (Giga-, Terabyte an Informationen)
- ❑ immer wiederkehrende Muster von Objektbeziehungen

Motivation: Warum Datenbanken? (2)

Einsatzbeispiele für Datenbanken

- Traditionell: Für kaufmännische informationsverarbeitende Aktivitäten in Verwaltungsabteilungen großer Organisationen wie Versicherungen, Banken, Telekommunikations- oder Versandunternehmen etc.
- Heute:
  - Adreßdatenbank, Stadtkarten ... auf jedem PC
  - Datenbank wissenschaftl. Veröffentlichungen, elektronischer Forschungsbericht der TUHH
  - Kinoprogramm, Telefonbuch, Kleinanzeigenmarkt, ...
  - Multimediales Content-Management im Internet

DAX 30, M-DAX UND TEC-DAX



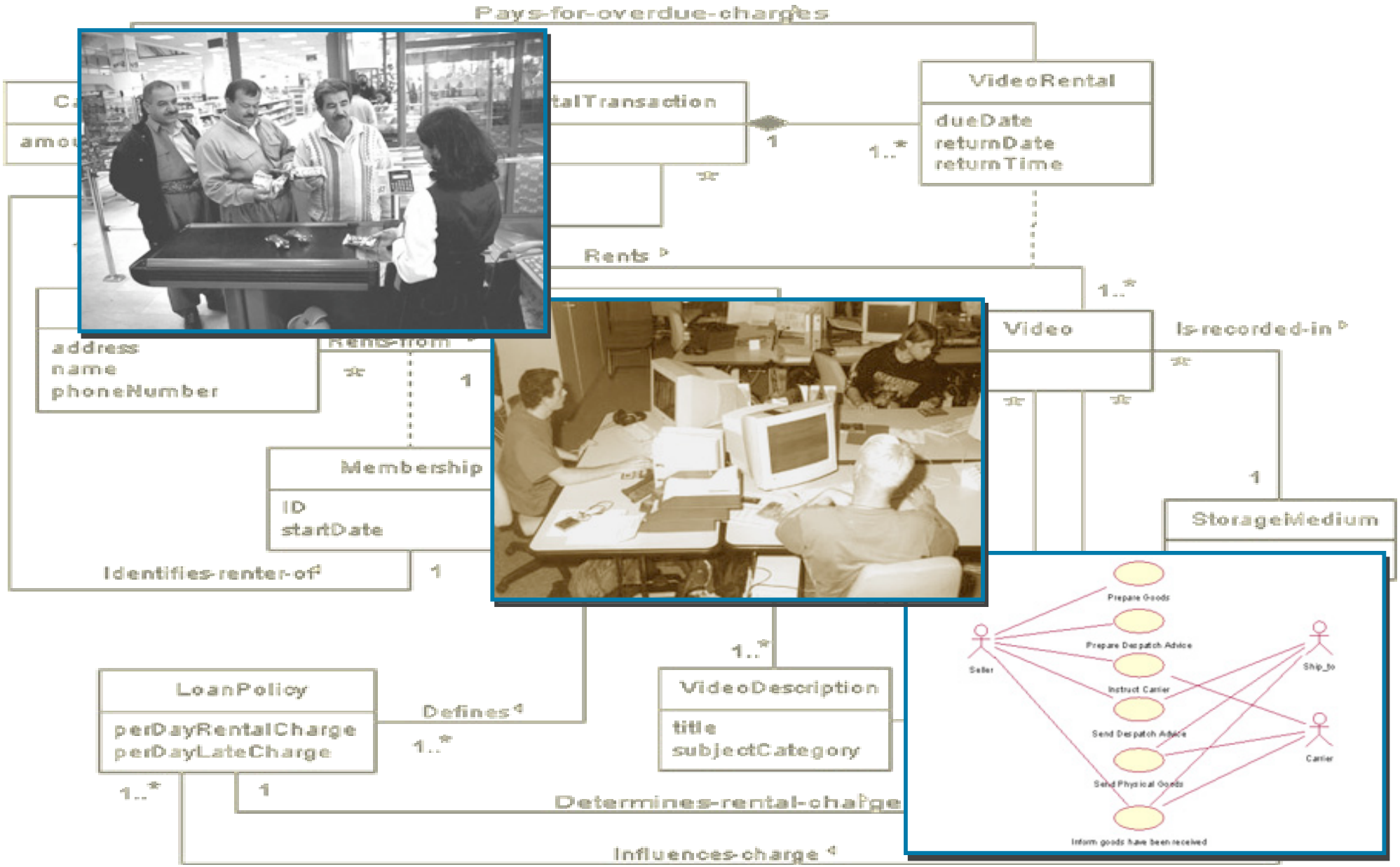
Börsenwert									ISIN		Xetra 2.10.03 Schluß	Xetra 3.10.03 18.42 Uhr	Xetra Tages Hoch	Xetra Tages Tief	Veränd. in % seit 2.10.03	Veränd. in % seit 30.12.02	Parkett Frankfurt 3.10.03 18.42 Uhr	Div.	Div.-Rend.	Ums.Ges. Tsd.St. 2.10.
									DE0008469008	DAX30	3276,64	3426,75	3429,9	3283,3	+4,6	+18,5	3426,75	68,26	2,00	
									DE0005003404	Adidas-Salomon	75,55	76,98	76,99	75,32	+1,9	-6,5	76,50	1,00	1,31	337
									DE0008404005	Allianz vink NA ▲	76,00	79,50	79,50	76,05	+4,6	-1,3	79,50	1,50	1,92	3422
									DE0007600801	Altana	53,64	54,74	54,94	53,33	+2,1	+25,8	54,70	0,75	1,39	533
									DE0005151005	BASF ▲	37,55	39,50	39,65	37,72	+5,2	+9,5	39,40	1,40	3,59	4230
									DE0005752000	Bayer ▲	17,97	18,62	18,68	17,92	+3,6	-8,9	18,65	0,90	4,90	6517
									DE0008022005	Bay. Hypo-Vereinsbk.	15,55	16,32	16,38	15,57	+5,0	+7,2	16,25	0,00	0,00	7825
									DE0005190003	BMW StA	33,00	34,75	34,86	33,07	+5,3	+20,2	34,50	0,52	1,52	3224
									[illegible]	Commerzbank	13,50	14,05	14,24	13,72	+4,1	+88,6	13,95	0,10	0,72	5114
									[illegible]		[illegible]	[illegible]	[illegible]	[illegible]	+2,5	+71,0	25,30	0,45	1,77	842
45,80	23,71	31,19	25,24	3,32	2,15	2,76	1		[illegible]		[illegible]	[illegible]	[illegible]	[illegible]	+4,6	+6,0	31,05	1,50	4,89	7682
61,80	32,97	32,15	32,15	0,64	2,48	4,37	2		[illegible]		[illegible]	[illegible]	[illegible]	[illegible]	+4,3	+26,8	55,55	1,30	2,37	4031
49,20	31,53	5,11	5,11	2,82	3,25	3,86	1		[illegible]		[illegible]	[illegible]	[illegible]	[illegible]	+0,8	+21,0	45,88	0,44	0,96	655
15,82	7,62	17,36	5,51	0,92	1,42	1,65	1		[illegible]		[illegible]	[illegible]	[illegible]	[illegible]	+0,5	+56,1	15,55	0,40	2,57	4713
14,49	8,22	53,48	30,61	-0,39	0,76	0,85	1		[illegible]		[illegible]	[illegible]	[illegible]	[illegible]	+3,2	+4,1	12,74	0,00	0,00	22365
52,90	34,11	30,02	28,30	2,90	3,38	3,77	1		[illegible]		[illegible]	[illegible]	[illegible]	[illegible]	+4,8	+13,7	43,70	1,75	4,07	5256
53,77	20,76	5,00	1,79	3,24	3,40	3,87	1		[illegible]		[illegible]	[illegible]	[illegible]	[illegible]	+4,9	+31,9	52,00	0,94	1,83	202
69,20	49,56	8,48	3,45	3,98	4,24	4,53	1		[illegible]		[illegible]	[illegible]	[illegible]	[illegible]	+1,6	-3,0	58,30	1,12	1,93	405
14,18	5,03	8,61	5,19	-0,98	-0,56	0,58	k		[illegible]		[illegible]	[illegible]	[illegible]	[illegible]	+7,1	+73,7	12,12	0,00	0,00	10059
46,75	22,27	4,45	3,01	2,63	2,51	3,05	1		[illegible]		[illegible]	[illegible]	[illegible]	[illegible]	+2,0	+6,8	37,30	1,13	3,04	539
13,45	6,80	4,46	4,02	0,14	-1,03	0,42	k		[illegible]		[illegible]	[illegible]	[illegible]	[illegible]	+6,1	+34,2	11,73	0,60	5,15	2660
21,90	10,02	2,82	1,94	0,70	1,70	2,36	1		[illegible]		[illegible]	[illegible]	[illegible]	[illegible]						
34,83	15,22	10,69	4,70	2,14	2,32	2,46	14,1	13,3	DE0007257503	Metro StA	31,79	32,89								
201,30	50,37	16,06	11,47	6,10	0,80	11,30	112,4	8,0	DE0008430026	Münchener R. v. NA ▲	86,00	90,50								
36,47	17,60	13,16	9,44	2,06	3,20	4,02	7,3	5,8	DE0007037129	RWE StA ▲	22,66	23,59								
127,15	39,80	36,15	23,64	3,31	3,51	3,95	32,6	29,0	DE0007164600	SAP ▲	105,95	115,44								
60,60	31,60	7,41	6,47	2,61	2,59	2,70	14,6	14,0	DE0007172009	Schering	37,45	37,95								
58,50	30,87	48,35	45,21	1,88	2,50	2,88	21,7	18,9	DE0007236101	Siemens ▲	52,02	54,60								
13,85	6,85	6,24	4,99	0,42	0,78	1,17	15,6	10,4	DE0007500001	Thyssen Krupp	11,55	12,19								
21,33	7,56	2,63	1,81	0,34	0,66	1,29	22,4	11,5	DE0006952005	TUI	14,49	14,79								
48,20	28,06	17,19	8,87	6,73	3,06	4,78	13,2	8,5	DE0007664005	VW StA ▲	38,59	40,60								
4200,18	2646,89			223,58	267,89	340,19	15,0	11,8	DE0008467416	M-Dax	3982,02	4026,51								
22,15	8,76	0,72	0,43	2,46	2,69	3,17	7,6	6,4	DE0005408116	Aareal Bank	20,29	20,51								
85,96	30,40	2,74	0,69	-4,40	-2,80	5,00	k.A.	10,2	DE0008400029	AMB Generali	52,30	52,00								
22,23	9,00	0,78	0,35	0,45	0,95	1,15	21,6	17,8	DE0005085906	AWD Holding	19,70	20,54	20,54	19,96	+4,3	+60,7	20,59	0,50	2,48	116

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} = \frac{Y(z)}{X(z)}$$

$$Y(z)(1+a_1z^{-1}+a_2z^{-2}) = X(z)(b_0+b_1z^{-1}+b_2z^{-2})$$

$$Y(z) = -a_1z^{-1}Y(z)-a_2z^{-2}Y(z) + b_0X(z)+b_1z^{-1}X(z)+b_2z^{-2}X(z)$$

$$y[n] = -a_1y[n-1] - a_2y[n-2] + b_0x[n] + b_1x[n-1] + b_2x[n-2]$$


Pays-for-overdue-charges
VideoRental
dueDate
returnDate
returnTime
1..*
1
Rents
1..*
Video
Is-recorded-in
1
StorageMedium
address
name
phoneNumber
Membership
ID
startDate
Identifies-renter-of
1
1..*
LoanPolicy
perDayRentalCharge
perDayLateCharge
Defines
1..*
1..*
1
VideoDescription
title
subjectCategory
Determines-rental-charge
Influences-charge
Prepare Goods
Prepare Despatch Advice
Instruct Carrier
Send Despatch Advice
Send Physical Goods
Inform goods have been received
Seller
Ship_to
Carrier

Motivation: Warum Datenbanken? (3)

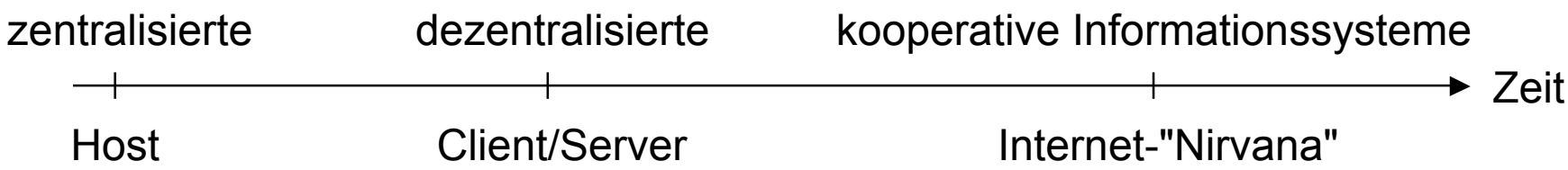
Anforderungen an die Datenverwaltung:

- ❑ Adäquate Repräsentation der Informationsstrukturen (betriebswirtschaftliche, statistische, textuelle, multimediale, grafische etc.)
- ❑ Flexible Zugriffsmodalitäten auf Informationsbestände (Hintergrundverarbeitung, interaktive Recherche, entfernter Zugriff etc.)
- ❑ Zugriff für unterschiedliche Benutzer, Sprachen und Anwendungsprogramme
- ❑ Konsistenz der Daten
  - Synchronisation
  - Fehlererholung
  - "konzeptuelle" Integrität
- ❑ Zugriffskontrolle, Datenschutz
- ❑ Effizienz beim Element- und Massendatenzugriff
- ❑ Unterstützung für Evolution der Daten und Programme

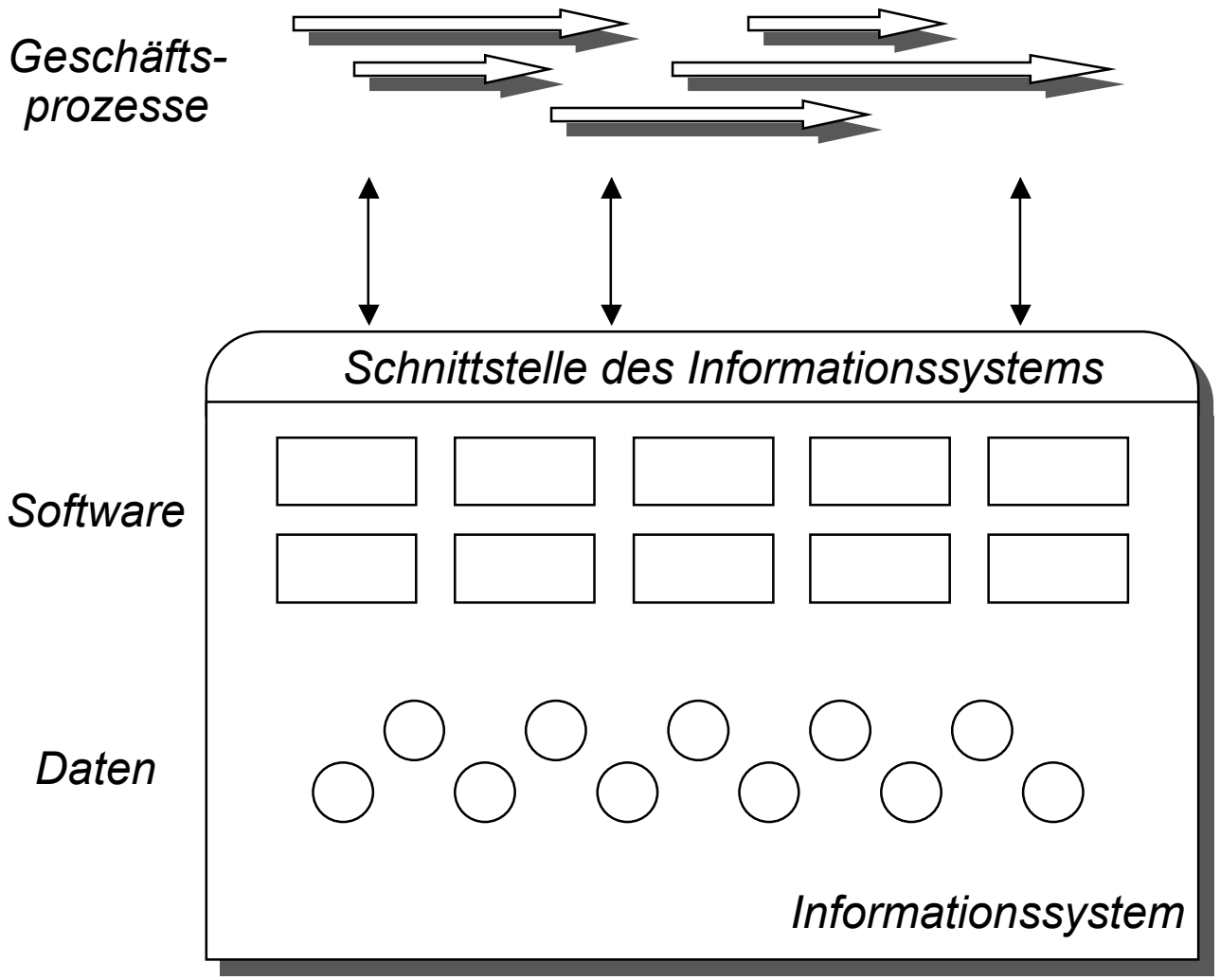
Informationssysteme

Wandel der Softwaresysteme:

- ❑ Ursprünglich: Computer zur Lösung numerischer Berechnungsaufgaben
- ❑ Später Informationssysteme: Systeme zur Repräsentation und Verarbeitung von Informationen über „Geschäftsprozesse“



Schematische Struktur eines Informationssystems



Informationsaustausch mit Aktivitäten der Umgebung

Algorithmen zur Informationsverarbeitung und Integritätssicherung

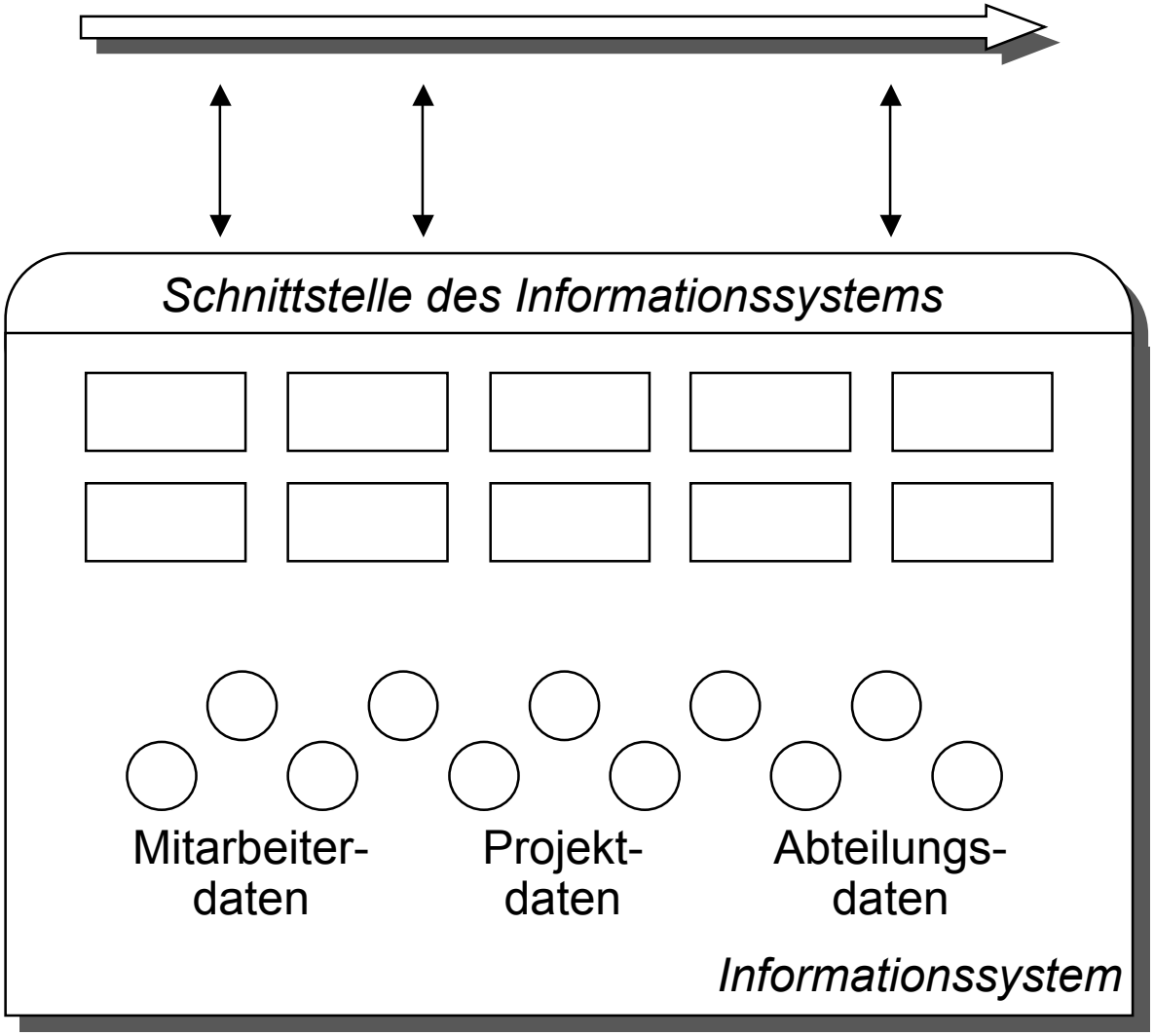
Umfangreicher, persistenter Datenbestand

Beispiel: Ein Firmeninformationssystem

Beratungsprojekt durchführen

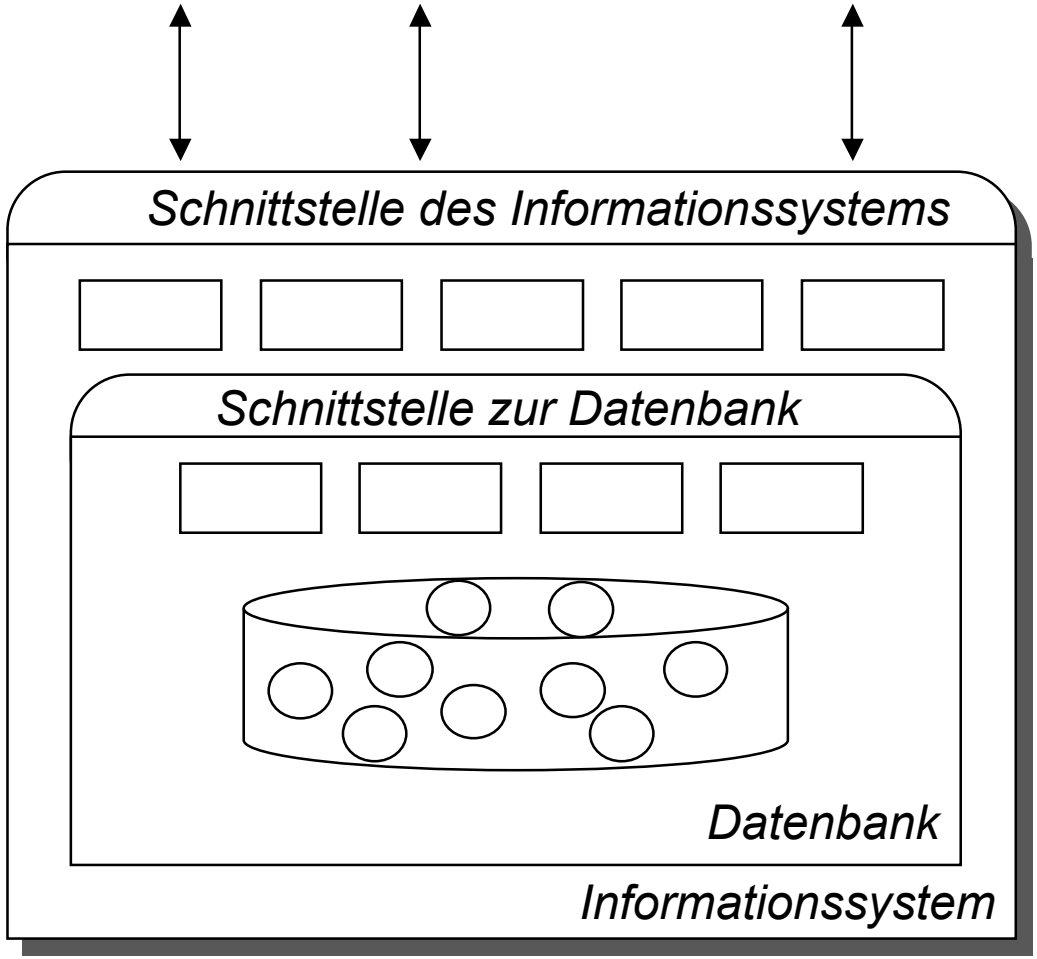
Budgetberechnung, Gehaltsabrechnung, Ressourcenberechnung etc.

Mitarbeiter, Projekt, Abteilung



Datenbanksysteme

Realisierung eines Informationssystems mit einer Datenbank:

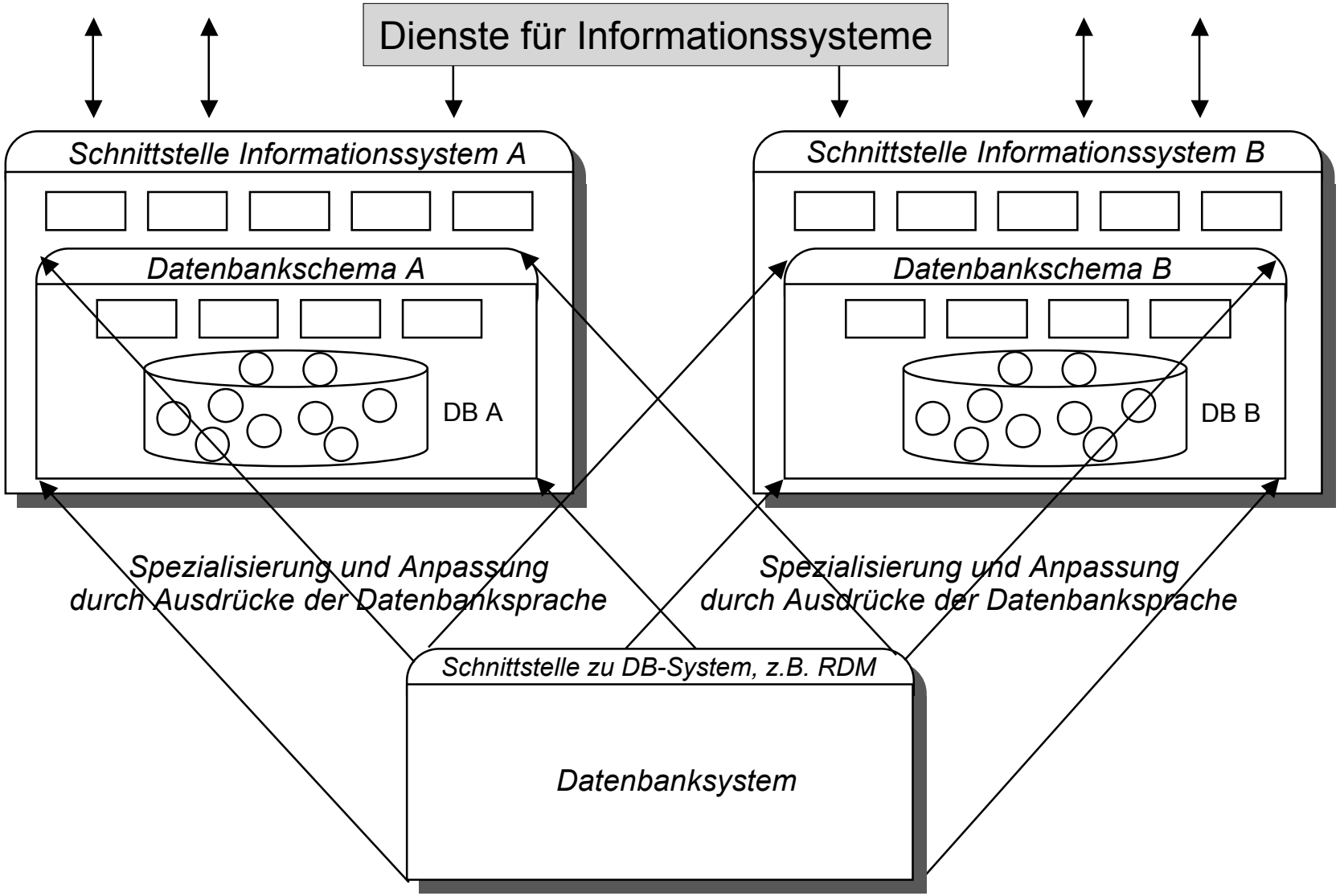


Algorithmen zur Informationsdarstellung, -verarbeitung und zur Integritätssicherung

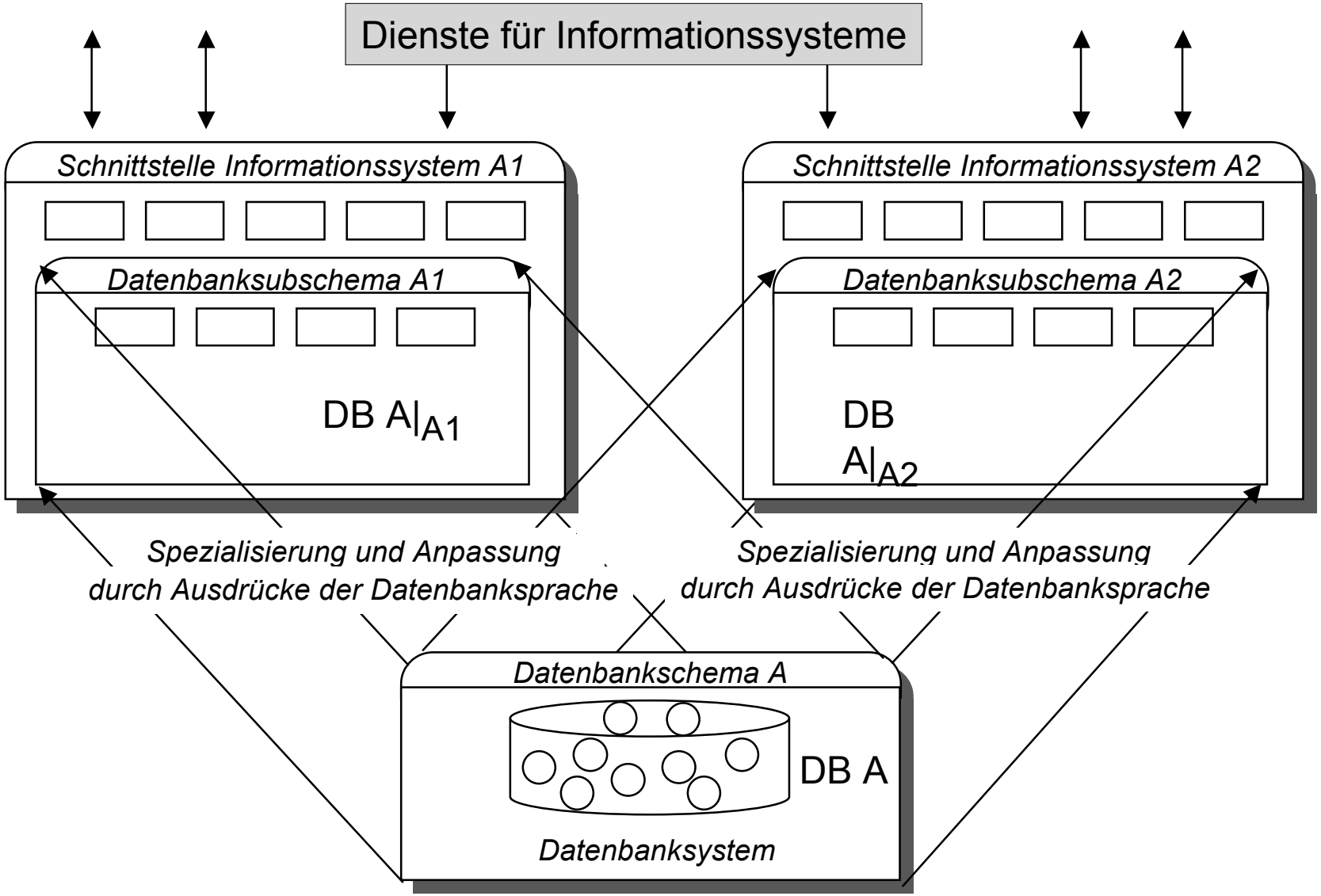
Dienste des Datenbanksystems zur Datenspeicherung, -anfrage und Integritätssicherung (Datenbankschema)

Datenbankzustand

Generisches Datenbankmodell und -system



Sichten: DB-Subschemata und Subdatenbanken



Analogie: Datenbanken ⇝ Programmiersprachen

Datenbankschema (Datenstrukturen, Tabellenstrukturen)	**Modulschnittstelle** (Klassen, Typen)
Datenbank (Datenwerte, Tabelleninhalte)	**Modulimplementierung** (Objekte, Methoden)
Datenbankmodell (Relationales Modell)	**Programmiersprachen** (C++, ANSI C)
Datenbanksprache (SQL)	**Konkrete Programmiersprache** (GNU C++)
Datenbanksystem (Oracle 9i)	**Konkreter Compiler** (GNU C++ 2.7.2 für Solaris)

Anforderungen an Datenbankmodelle und -systeme

PQRI-Anforderungen:

Charakteristika von Informationssystemen:

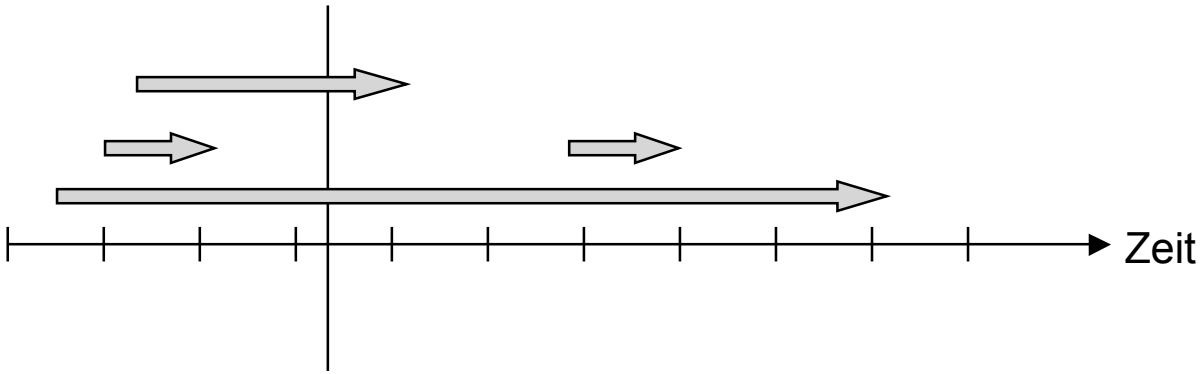
- *Persistenz* (Langlebigkeit) des Informationsbestands
- *Quantität* des regulär strukturierten Informationsbestands
- *Reaktivität* auf die Aktivitäten seiner Umgebung
- *Integrität* des Informationsbestands und der ein- und ausgehenden Information

Unterstützung der Persistenz (1)

Alle Daten, deren Lebensdauer die Lebensdauer eines einzelnen Betriebssystemprozesses überschreitet, werden als persistent bezeichnet.

Ein Informationssystem unterstützt die Speicherung unterschiedlichster Daten (mit unterschiedlicher Lebensdauer).

Ziel: Daten existieren solange, wie die Geschäftsprozesse, die sie benötigen.



akueller Datenbankzustand

Unterstützung der Persistenz (2)

Beispiele (in Klammern Dauer der Persistenz):

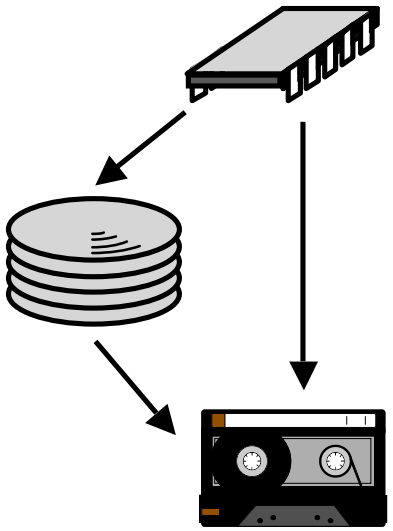
- Synchronisation paralleler Transaktionen (Sekundenbruchteile)
- Sicherung von Zwischenzuständen einer langandauernden Aktivität (Minuten)
- Aufwendige Informationsanfragen (Stunden, DB-Zustand bei Beginn der Anfrage wird für die Dauer der Anfrage scheinbar „eingefroren“)
- Zustand eines Systemdialogs über mehrere Sitzungen (Tage)
- Versionen von Datenbeständen (Wochen, Monate)
- Protokollinformation (Jahre)
- Historische Datenbankzustände (Jahrzehnte)
- Datenbankzustand (unbestimmbar, anwendungsabhängig)

Unterstützung der Persistenz (3)

- **Speicherhierarchie**: Datenspeicherung auf Sekundär- (Festplatte) und Tertiärspeichern (Band, Wechselplatte) ➠ Datenmanipulation im Primärspeicher (RAM)
- **Persistenzabstraktion**: Manipulation von Daten unabhängig von ihrer Lebensdauer
- **Meta- und Schemadaten:** Haltung zusammen mit den Nutzdaten
- **Schemaevolution**: Umstrukturierung ohne Datenverlust
- **Fehlererholung**: Persistenz über Ausfälle hinweg

Unterstützung der Persistenz (4)

Kosten	**Zugriffszeit**	**Durchsatz**	**Umfang**	**Lebensdauer**
sehr teuer	Nanosekunden	3 GB/s	begrenzt	flüchtig
teuer	Millisekunden	70 MB/s	begrenzt	persistent
billig	> Sekunden	1 MB/s	"unendlich"	persistent



Unterstützung der Quantität

Durch die Regularität der Informationsstrukturen läßt sich der Informationsbestand in Klassen zusammenfassen, zwischen denen regelhafte Beziehungen bestehen können (➠ Massendatenstrukturen) und auf denen Invarianten definiert werden können (➠ statische Typisierung, constraints).

- **Iterationsabstraktion**: stereotype imperative Programmuster werden ersetzt durch deklarative Zugriffsbeschreibungen
- **Effizienz** der Datenselektion zur Vermeidung teurer Sekundär- und Tertiärspeicherzugriffe durch Pufferung, Indizierung etc.

Beispiel:

```
firstPerson()
result:= true
while dbstatus = 0 do
  person = getPerson()
  result:= result and person.salary > 100
  nextPerson()
end
```

```
result:= ∃ p in Person: p.salary > 100
```

Unterstützung der Reaktivität

Ein Informationssystem reagiert auf eingehende Daten, antwortet auf Anfragen und kann selbständig Aktionen auslösen (Benutzeroberflächen, Dienstschnittstellen).

- **Synchronisation** nebenläufiger Aktivitäten auf dem gleichen Informationsbestand
- **Transaktionen**: zusammengehörige Aktionen werden atomar, konsistenzerhaltend, isoliert und mit dauerhaftem Effekt ausgeführt

  (➠ Integritätssicherung, Synchronisation, Fehlererholung)
- **Verhaltensmodellierung** erfolgt mit algorithmisch vollständigen Sprachen

  (➠ Programmierspracheneinbettung, zusammen mit deklarativen Zugriffsbedingungen)

Unterstützung der Integrität (1)

Aspekte der Integritätssicherung:

- **Binnenwirkung**: Konsistenz eines gekapselten langlebigen Systemzustandes, z.B. Einstellung eines Mitarbeiters, Beendigung eines Projektes, ...
- **Außenwirkung**: Informationsstrukturen und Geschäftsanforderungen für alle Aktivitäten der Umgebung, d.h., die „geschäftlichen Rahmenbedingungen“

Klassifizierung von Integritätsbedingungen:

- **Modellinhärente Integritätsbedingungen** werden implizit durch das Datenbankmodell erzwungen (z.B. Typisierung: das Alter einer Person ist ein Integer).
- **Applikationsspezifische Integritätsbedingungen** werden durch explizite Deklaration im Datenbankschema oder durch explizite Überprüfung in Algorithmen erzwungen (z.B. das Alter einer Person liegt zwischen 0 und 100).

Unterstützung der Integrität (2)

Klassifizierung nach der zeitlichen Ausdehnung:

- **Statische Integritätsbedingungen** müssen in jedem Zustand erfüllt sein (quantifizierte Prädikate, z.B. jeder Mitarbeiter arbeitet in genau einer Abteilung).
- **Dynamische Integritätsbedingungen** müssen in jeder Zustandsänderung erfüllt sein (z.B. Gehälter von Mitarbeiter nehmen nie ab).

Integritätsbedingungen können auch nach ihrem Sichtbarkeits- und Wirkungsbereich unterschieden werden (z.B. Objekte, Klassen).

Integrität wird erhalten durch:

- **Deklarative Integritätssicherung**: Klauseln oder quantifizierte boolesche Prädikate im Datenbankschema (z.B. zu jedem Student s gehört genau eine Universität u)
- **Prozedurale Integritätssicherung**: explizit programmierte Tests (z.B. `if person.age < 18 then abort else insertStudent(person)`)

Deklarative Integritätssicherung

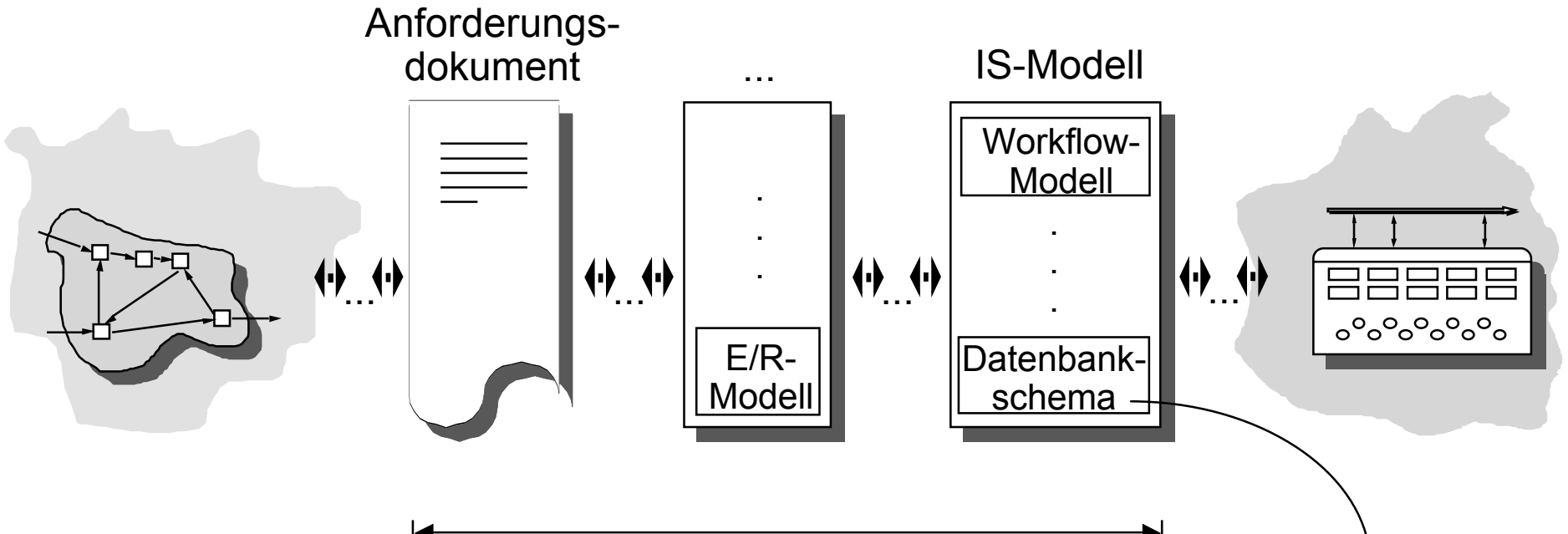
Vorteile:

- ❑ Verbesserte Systemwartbarkeit
- ❑ Verbesserte Verstehbarkeit
- ❑ Optimierbarkeit

Probleme:

- ❑ Synchronisation verschiedener Integritätsbedingungen kann zu nicht-deterministischem Verhalten führen.
- ❑ Terminierung und Konsistenz von Ausnahmebehandlungen ist nicht garantiert.
- ❑ Lokalisierung von Integritätsbedingungen
- ❑ Inadäquate Sprachmittel für die Ausnahmebehandlung

Modelle und Abstraktion



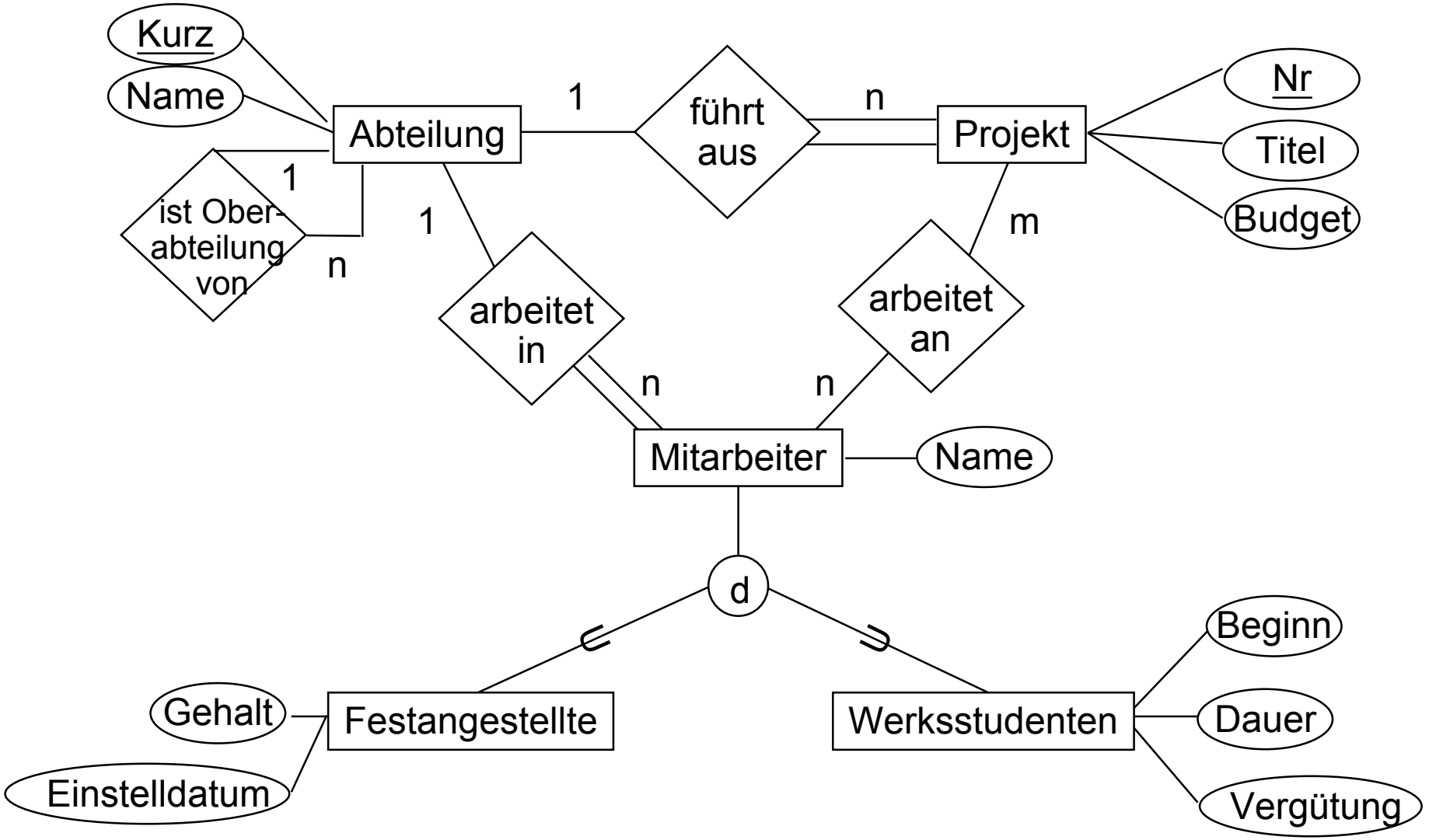
Modelle:
Abstraktionsmechanismen
- innerhalb des Modells
- bei den Abbildungen zwischen den Modellen

z.B. im relationalen Datenbankmodell

anwendungsnah

implementationsnah

Beispiel: ER-Diagramm für das Firmen-IS



Repräsentation "atomarer" Information (1)

Werte (Literale):

- Semantikunabhängig vom Datenbankzustand
- Es wird unterschieden in
  - Basiswerte (z.B. Zahlen, Zeichen, Zeichenketten)
  - `3.1415; ´Z´; "Otto"`
  - Zusammengesetzte Werte: heterogene Strukturen (Tupel, Rekords, Strukturen) und homogene Strukturen (Arrays, Listen, Mengen, Multimengen)
  - `{1,2,3,4}; record age = 31 name = "Peter" end`
- Operationen auf Werten haben Kopiersemantik (z.B. Addition, Feldzugriff)
  - `menge + {1,2,3}`
- Verlustfreie Darstellung als lineare textuelle Repräsentation

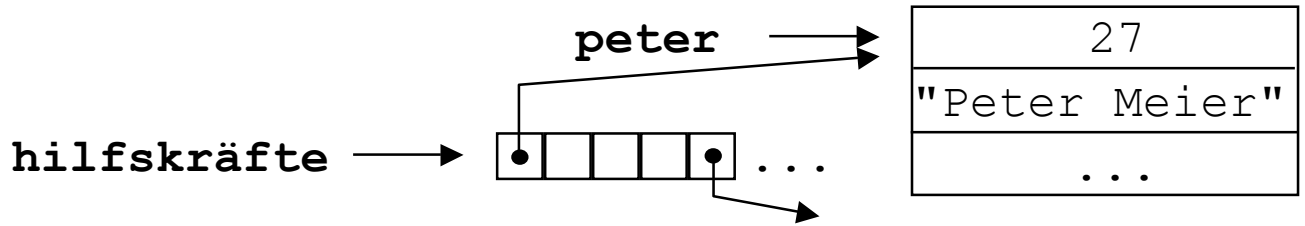
Repräsentation "atomarer" Information (2)

Objekte:

- Semantikabhängig vom Datenbankzustand (➠ Zustandsvariablen in imperativen Programmiersprachen)
- Der Zustand kann durch destruktive Zuweisung verändert werden.

```
peter.name := "Peter Meier"
```

- Objektidentität (object identity, OID) bleibt unabhängig vom Zustand erhalten
- Objekte können mehrfach über die OID referenziert werden (sharing)



- Änderungen des Objektzustands werden unmittelbar auf allen Pfaden sichtbar, z.B. die Änderung des Studentennamen (➠ Referenzsemantik)
- Spezielle Notation zur textuellen Repräsentation der (zyklischen) Graphenstruktur

Datenabstraktionskonzepte z. Informationsstrukturierung

In fast allen Datenbankmodellen findet man Konstrukte für die folgenden Abstraktionskonzepte:

- ❑ Klassifikation und Instantiierung
- ❑ Aggregation und Dekomposition
- ❑ Generalisierung und Spezialisierung
- ❑ Assoziation und Dissoziation
- ❑ Identifikation und Schlüssel

In späteren Kapiteln werden diese Konstrukte beschrieben. Nachfolgend werden die Abstraktionskonzepte anhand einer populären grafischen Notation erklärt.

Entity-Relationship-Diagramme wurden von P.P.S. Chen vorgeschlagen
(vgl. P.P.S. Chen. "The Entity Relationship Model - Toward a Unified View of Data.
In: ACM Transactions on Database Systems, Vol. 1, No. 1, März 1976, S. 9 ff.)
und mehrfach erweitert (➠ extended E/R diagram, EE/R Modell).