

Vorlesung  
**Informations- und  
Wissensmanagement**  
(Teil 1)

Universität Karlsruhe (TH)

Klemens Böhm

klemens.boehm@ipd.uni-karlsruhe.de



**Kapitel 1:**  
**Grundlegende Konzepte  
und Architekturen**

**Literatur**

Einleitung

Welt  
ohne DB

Datenbanken

Übersicht

Terminologie

Querying

DB –  
Merkmale

Anwend.

Historie

Schema-  
Arch.

Schluß

- 1 Heuer, A., Saake, G.: *Datenbanken – Konzepte und Sprachen*. 2. Aufl., mitp-Verlag, Bonn, Januar 2000.
- 1 Kemper, A., Eickler, A.: *Datenbanksysteme*. Oldenbourg, 2004.
- 1 Heuer, A., Saake, G., Sattler, K.: *Datenbanken kompakt* mitp-Verlag, Bonn, 2001.
- 1 Elmasri, R.; Navathe, S.B.; *Fundamentals of Database Systems*. Addison-Wesley, 1999.

## Grundlegende Konzepte

### Einleitung

Welt  
ohne DB

Datenbanken  
Übersicht

Terminologie

Querying

DB –  
Merkmale

Anwend.

Historie

Schema-  
Arch.

Schluß

- 1 Motivation,
- 1 Komponenten und Funktionen,
- 1 Einsatzgebiete und Grenzen,
- 1 Entwicklungslinien,
- 1 Referenzarchitektur.

## Motivation

### Einleitung

Welt  
ohne DB

Datenbanken  
Übersicht

Terminologie

Querying

DB –  
Merkmale

Anwend.

Historie

Schema-  
Arch.

Schluß

- 1 Unser Thema: Verwaltung großer Datenmengen und Entwicklung entsprechender Anwendungen.
- 1 Warum Datenbanken?
  - 1 *Application-Development Complexity.*
    - u 50% weniger Aufwand bei Anwendungsentwicklung mit Datenbanken, gesamthaft betrachtet, bei überschaubaren Projekten.
    - u Größere Differenz bei größeren Projekten.  
⇒ Verwendung von Datenbank-Technologie ermöglicht neue Anwendungen; ihre Entwicklung ohne Datenbank-Technologie wäre zu komplex.

## Vergleich mit Bekanntem

### Einleitung

Welt  
ohne DB

Datenbanken  
Übersicht

Terminologie

Querying

DB –  
Merkmale

Anwend.

Historie

Schema-  
Arch.

Schluß

- 1 Informatik – zwei wichtige Aspekte:
  - u Erkennen von Regelmäßigkeiten/Mustern bei der Anwendungsentwicklung. Ausfaktorisieren.
  - u Schaffung von Entwicklungswerkzeugen auf abstrakterer Ebene.
- 1 Datenbanken – dasselbe für Entwicklung von Anwendungen zur Verwaltung großer Datenmengen.

## Situation ohne Datenbanken (1)

### Einleitung

Welt  
ohne DB

Datenbanken  
Übersicht

Terminologie

Querying

DB –  
Merkmale

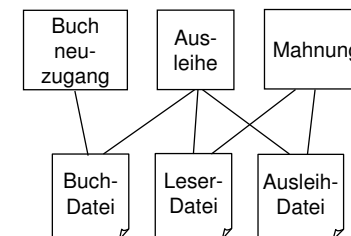
Anwend.

Historie

Schema-  
Arch.

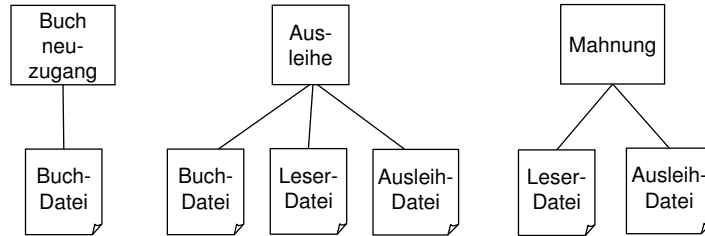
Schluß

- 1 Zugriff auf Daten, in Files abgelegt.
- 1 Funktionalität hierfür Teil der Anwendungen.



## Situation ohne Datenbanken (2)

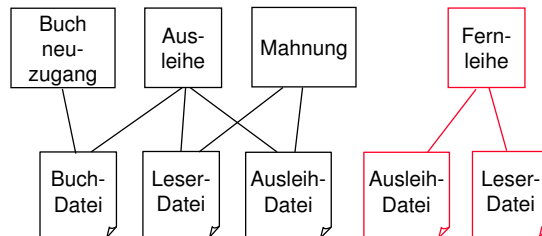
### 1 Redundanz.



## Ohne Datenbanken: Datenredundanz (1)

- 1 Daten sind *redundant*: Mehrfach gespeichert; Probleme: Verschwendung von Speicherplatz, „Vergessen“ von Änderungen, *Inkonsistenzen*; keine zentrale, „genormte“ Datenhaltung.

## Ohne Datenbanken: Datenredundanz (2)



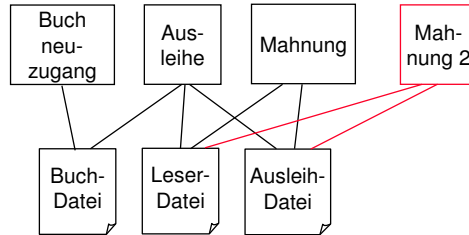
- 1 Weiteres, ähnliches Beispiel.
- 1 Benutzer i. Allg. mehrmals erfaßt.
- 1 Was passiert, wenn Benutzer umzieht?
- 1 Wenn er neue Adresse nur der Ausleihe, nicht aber der Fernleihe mitteilt?

## Ohne Datenbanken: Datenredundanz (3)

- 1 Basis- oder Anwendungssoftware verwaltet ihre eigenen Daten in ihren eigenen (Datei-)Formaten.
  - u Textverarbeitung: Texte, Artikel und Adressen,
  - u Buchhaltung: Artikel, Adressen,
  - u Lagerverwaltung: Artikel, Aufträge,
  - u Auftragsverwaltung: Aufträge, Artikel, Adressen,
  - u CAD-System: Artikel, technische Bauteile.

## Weitere Probleme ohne Datenbanken (1)

- Mehrere Benutzer oder Anwendungen können i. a. nicht parallel auf den gleichen Daten arbeiten („Nebenläufigkeit“, *Concurrency*), ohne sich zu stören.



- Es wäre extrem aufwendig, Lösungen hierfür selbst entwickeln und realisieren zu müssen.

## Weitere Probleme ohne Datenbanken (2)

- Transaktionseigenschaften – insbesondere *Atomarität* und *Isolation*.

- Atomarität

u Beispiel, „Bank-Szenario“:

Nummer	Inhaber	Stand
	Klemens	5000
	Rudi	200

u Überweisung – zwei Elementaroperationen.

- Abbuchung(Klemens, 500),
- Einzahlung(Rudi, 500).

- Isolation – auch an diesem Beispiel erklärbar.
- Transaktionen.

## Weitere Probleme ohne Datenbanken (3)

- Keine *physische Datenunabhängigkeit*:
  - „Naives“ file-basiertes Vorgehen bei Programmierung/Benutzung von Anwendungen: Man muß interne (physische) Repräsentation der Daten kennen, z. B. bei Speicherung als Tabelle Reihenfolge der Zeilen und Spalten.

## Weitere Probleme ohne Datenbanken (4)

- Keine *physische Datenunabhängigkeit* (Forts.):

u Beispiel.

NAME	VORNAME	STRASSE	ALTER
Böhm	Klemens	Gehtsienichtsgasse	28
Buchmann	Erik	Breiter Weg	26
Duckstein	Ralf	Goethestrasse	25
Saake	Gunter	Waldweg	43

VORNAME	NAME	STRASSE	ALTER
Erik	Buchmann	Breiter Weg	26
Gunter	Saake	Waldweg	43
Klemens	Böhm	Gehtsienichtsgasse	28
Ralf	Duckstein	Goethestrasse	25

## Weitere Probleme ohne Datenbanken (5)

- Keine *physische Datenunabhängigkeit* (Forts.):
  - Physische Datenunabhängigkeit – Verstecken dieser physischen Eigenschaften vor dem Anwendungsentwickler.
  - Datenbanken geben uns physische Datenunabhängigkeit (ein herausragendes Merkmal).
  - Illustration:

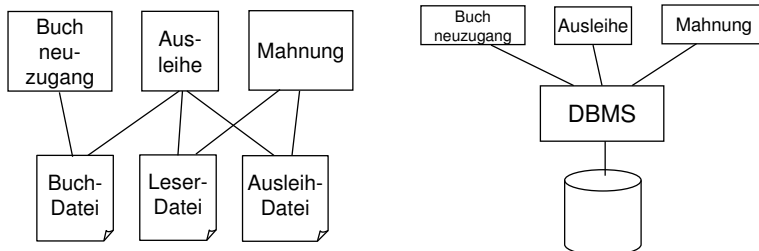
```
select NAME, VORNAME
from PERSON
where ALTER > 40
```

## Weitere Probleme ohne Datenbanken (6)

- Datenschutz und Datensicherheit sind nicht gewährleistet.
  - Datenschutz (kein unbefugter Zugriff),
  - Datensicherheit (kein ungewollter Datenverlust).

## Von Dateien zu Datenbanken

- Zugriff auf Daten, in Files abgelegt.
- Funktionalität hierfür Teil der Anwendungen (Berücksichtigung der physischen Ebene, Nebenläufigkeit, Datenschutz, Konsistenz).
- Datenbanken:  
Ausfaktorieren dieser Funktionalität.



## Mit Datenbanken: Datenintegration (1)

- Basis- und Anwendungssoftware arbeitet auf denselben Daten.  
Z. B. Adressen und Artikel nur einmal gespeichert.
  - Datenbanksysteme können große Datenmengen effizient verwalten. Kein Problem, große Datenmengen an einem Ort anzuhäufen.
  - Benutzer können parallel auf Datenbanken arbeiten (Transaktionskonzept).
- Auch ohne Datenbanken möglich, Redundanz zu vermeiden. Datenbanken machen dies aber leicht und bieten weitere Features.

## Mit Datenbanken: Datenintegration (2)

Einleitung  
Welt  
ohne DB  
[Datenbanken  
Übersicht](#)  
Terminologie  
Querying  
DB –  
Merkmale  
Anwend.  
Historie  
Schema-  
Arch.  
Schluß

- 1 Die gesamte Basis- und Anwendungssoftware arbeitet auf denselben Daten... (Fortsetzung):
  - u Datenunabhängigkeit durch 3-Ebenen-Konzept (wird gleich erklärt),
  - u Datenschutz (kein unbefugter Zugriff) und Datensicherheit (kein ungewollter Datenverlust) werden vom System gewährleistet.

Z

## Mögliche Prüfungsfragen

Einleitung  
Welt  
ohne DB  
[Datenbanken  
Übersicht](#)  
Terminologie  
Querying  
DB –  
Merkmale  
Anwend.  
Historie  
Schema-  
Arch.  
Schluß

- 1 Was für Probleme bringt Datenhaltung ohne Datenbanken mit sich?
- 1 Wie profitiert ein Anwendungsentwickler von der Verwendung von Datenbank-Technologie?
- 1 Erklären Sie die folgenden Begriffe:
  - u Redundanz, Inkonsistenz,
  - u Atomarität, Isolation, Transaktion,
  - u physische Datenunabhängigkeit (weitere Erläuterungen hierzu folgen).

## Konzeptuelle Ebene: Relationenmodell (1)

Einleitung  
Welt  
ohne DB  
[Datenbanken  
Übersicht](#)  
Terminologie  
Querying  
DB –  
Merkmale  
Anwend.  
Historie  
Schema-  
Arch.  
Schluß

- 1 Konzeptuell ist Datenbank üblicherweise eine Menge von Tabellen.

AUSLEIH	INV.NR	NAME
	4711	Meyer
	1201	Schulz
	0007	Müller
	4712	Meyer

BUCH	INV.NR	TITEL	ISBN	AUTOR
	0007	Dr. No	3-125	James Bond
	1201	Objektbanken	3-111	Heuer
	4711	Datenbanken	3-765	Vossen
	4712	Datenbanken	3-891	Ullman
	4717	PASCAL	3-999	Wirth

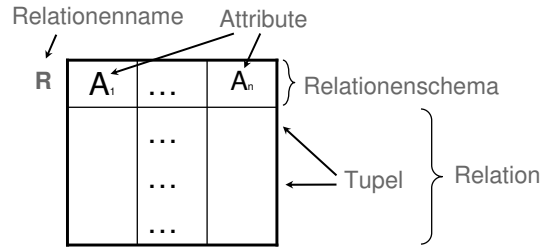
## Konzeptuelle Ebene: Relationenmodell (2)

Einleitung  
Welt  
ohne DB  
[Datenbanken  
Übersicht](#)  
Terminologie  
Querying  
DB –  
Merkmale  
Anwend.  
Historie  
Schema-  
Arch.  
Schluß

- 1 Tabellen = „Relationen“
- 1 Relationen sind Mengen von Tupeln.

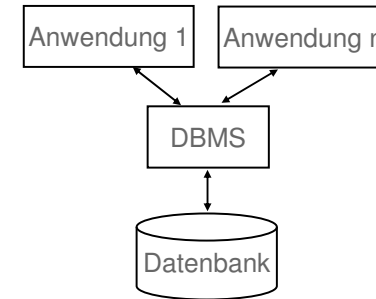
## Konzeptuelle Ebene: Relationenmodell (2)

- 1 **Fett** geschriebene Zeilen: *Relationenschema*
- 1 Weitere Einträge in der Tabelle: *Relation*
- 1 Eine Zeile der Tabelle: *Tupel*
- 1 Eine Spaltenüberschrift: *Attribut*



## DBMS vs. DBS

- 1 DBMS: Datenbank-Management-System
- 1 DBS: Datenbanksystem (DBMS + Datenbank)



- 1 Unterscheidung:
  - u DBMS ist Software zur Datenverwaltung, Datenbank sind die eigentlichen Daten.
  - u Ein DBMS, aber mehrere (viele) Datenbanken.

## Konzeptuelle Ebene: Relationenmodell

- 1 Konzeptuell ist Datenbank üblicherweise eine Menge von Tabellen.

AUSLEIH	INV.NR	NAME
	4711	Meyer
	1201	Schulz
	0007	Müller
	4712	Meyer

BUCH	INV.NR	TITEL	ISBN	AUTOR
	0007	Dr. No	3-125	James Bond
	1201	Objektbanken	3-111	Heuer
	4711	Datenbanken	3-765	Vossen
	4712	Datenbanken	3-891	Ullman
	4717	PASCAL	3-999	Wirth

## Integritätsbedingungen

- 1 *Relationenschema* + lokale Integritätsbedingungen:
  - u INV.NR ist *Schlüssel* für BUCH,
  - u d. h. INV.NR darf nicht doppelt vergeben werden.
- 1 *Datenbankschema* ist Menge von Relationenschemata + globale Integritätsbedingungen:
  - u INV.NR in AUSLEIH ist *Fremdschlüssel* bezüglich BUCH,
  - u d. h.: INV.NR taucht in einem anderen Relationenschema als Schlüssel auf.

## Mögliche Prüfungsfragen

Einleitung  
Welt ohne DB  
Datenbanken Übersicht  
Terminologie  
Querying  
DB – Merkmale  
Anwend.  
Historie  
Schema-Arch.  
Schluß

- 1 Was versteht man unter Datenbank-Schema?
- 1 Erklären Sie den Unterschied zwischen ‚Datenbank‘ und ‚DBMS‘.
- 1 Was für Arten von Integritätsbedingungen, die DBMS unterstützen, kennen Sie? (Kommt noch einmal ausführlicher.)

Klemens Böhm

IWM: Einleitung – 29

## Sprachen und Sichten (1)

Einleitung  
Welt ohne DB  
Datenbanken Übersicht  
Terminologie  
Querying  
DB – Merkmale  
Anwend.  
Historie  
Schema-Arch.  
Schluß

- 1 Wie greife ich auf Informationen in meiner Datenbank zu?
- 1 SQL als Standard-Anfragesprache.
- 1 *Anfragen, Queries* – dasselbe.
- 1 Query-Ergebnis – auch wieder Relation.

Klemens Böhm

IWM: Einleitung – 30

Verbindung zum Datenbankserver erfolgreich

Bitte Anfrage eingeben:

select \* from Bücher

Zuvor wurde eingegeben:  
**INSERT INTO Bücher  
VALUES ('12345', 'Datenbanken', 'MITP');  
commit**

Abfrage starten

ISBN	TITEL	
12345	Datenbanken	MITP

time	percent	rows	query
0.001728	100%	1	select * from Bücher
0.001728			

Verbindung zum Datenbankserver erfolgreich

Bitte Anfrage eingeben:

select \* from Bücher

Zuvor wurde eingegeben:  
**INSERT INTO Bücher  
VALUES ('54321', 'Tante Julia', NULL);  
commit**

Abfrage starten

ISBN	TITEL	
12345	Datenbanken	MITP
54321	Tante Julia	

time	percent	rows	query
0.001733	100%	2	select * from Bücher
0.001733			

## Sprachen und Sichten (2)

Einleitung  
Welt  
ohne DB  
Datenbanken  
Übersicht  
Terminologie  
Querying  
DB –  
Merkmale  
Anwend.  
Historie  
Schema-  
Arch.  
Schluß

- 1 SQL als Standard-Anfragesprache –  
weiteres Beispiel:  
**select** INVENTARNR, TITEL, NAME  
**from** BUCH, AUSLEIH  
**where** NAME = 'Meyer'  
**and** Titel = 'Datenbanken'

## Sprachen und Sichten (3)

Einleitung  
Welt  
ohne DB  
Datenbanken  
Übersicht  
Terminologie  
Querying  
DB –  
Merkmale  
Anwend.  
Historie  
Schema-  
Arch.  
Schluß

- 1 Definition von Benutzersichten:  
Häufig vorkommende Datenbankabfragen  
(Queries) können unter einem „Sichtnamen“  
als „virtuelle“ Tabelle gespeichert werden.
  - 1 *Sicht* = Query (+ Name),
  - 1 englisch: View.

## Definition von Sichten in SQL – Beispiel (1)

Einleitung  
Welt  
ohne DB  
Datenbanken  
Übersicht  
Terminologie  
Querying  
DB –  
Merkmale  
Anwend.  
Historie  
Schema-  
Arch.  
Schluß

- 1 Zugrundeliegende Relation:  
MGA(Mitarbeiter, Gehalt, Abteilung)
  - 1 **create view** MG **as**  
**select** Mitarbeiter, Gehalt  
**from** MGA  
**where** Gehalt > 20
  - 1 View dann im Prinzip verwendbar  
wie ‚normale‘ Relation, z. B.:  
**select \* from** MG **where** Gehalt < 40

## Definition von Sichten in SQL – Beispiel (2)

Einleitung  
Welt  
ohne DB  
Datenbanken  
Übersicht  
Terminologie  
Querying  
DB –  
Merkmale  
Anwend.  
Historie  
Schema-  
Arch.  
Schluß

MGA	Mitarbeiter	Gehalt	Abteilung
	Klemens	60	IPD
	Erik	30	DKE
	Holger	15	DKE
	Gunter	80	DB

MG	Mitarbeiter	Gehalt
	Klemens	60
	Erik	30
	Gunter	80

- 1 **create view** MG **as**  
**select** Mitarbeiter, Gehalt  
**from** MGA  
**where** Gehalt > 20
  - 1 **select \* from** MG **where** Gehalt < 40

## Sprachen und Sichten (4)

Einleitung  
Welt  
ohne DB  
Datenbanken  
Übersicht  
Terminologie  
Querying  
DB –  
Merkmale  
Anwend.  
Historie  
Schema-  
Arch.  
Schluß

- 1 Sichten werden gern kombiniert mit Datenschutz-Mechanismen.
- 1 Unterschiedliche Benutzer sehen unterschiedlichen Ausschnitt der Datenbank.
- 1 Beispiel – Universität (Studierende, Dozenten, Prüfungsamt, Dekanat etc.):
  - u Datenschutz,
  - u Übersichtlichkeit,
  - u organisatorische Gründe (Ablenkung, Einmischung).

Z

## Konzeptuelle Ebene: Relationenmodell

Einleitung  
Welt  
ohne DB  
Datenbanken  
Übersicht  
Terminologie  
Querying  
DB –  
Merkmale  
Anwend.  
Historie  
Schema-  
Arch.  
Schluß

- 1 Konzeptuell ist Datenbank üblicherweise eine Menge von Tabellen.

AUSLEIH	INV.NR	NAME
	4711	Meyer
	1201	Schulz
	0007	Müller
	4712	Meyer

BUCH	INV.NR	TITEL	ISBN	AUTOR
	0007	Dr. No	3-125	James Bond
	1201	Objektbanken	3-111	Heuer
	4711	Datenbanken	3-765	Vossen
	4712	Datenbanken	3-891	Ullman
	4717	PASCAL	3-999	Wirth

## Anfrageoperationen (1)

Einleitung  
Welt  
ohne DB  
Datenbanken  
Übersicht  
Terminologie  
Querying  
DB –  
Merkmale  
Anwend.  
Historie  
Schema-  
Arch.  
Schluß

- 1 SELEKTION: Zeilen (Tupel) auswählen.  
 $\sigma_{\text{NAME}=\text{Meyer}}(\text{AUSLEIH})$ 

INV.NR	NAME
4711	Meyer
4712	Meyer
- 1 PROJEKTION: Spalten (Attribute) auswählen.  
 $\pi_{\text{INV.NR, TITEL}}(\text{BUCH})$ 

INV.NR	TITEL
0007	Dr. No
1201	Objektbanken
4711	Datenbanken
4712	Datenbanken
4717	PASCAL

## Anfrageoperationen (2)

Einleitung  
Welt  
ohne DB  
Datenbanken  
Übersicht  
Terminologie  
Querying  
DB –  
Merkmale  
Anwend.  
Historie  
Schema-  
Arch.  
Schluß

- 1 Beispiel für komplexen Algebraausdruck:  
 $\pi_{\text{INV.NR, TITEL}}(\sigma_{\text{AUTOR}=\text{Heuer}}(\text{BUCH}))$

Ausgangsrelation:

INV.NR	TITEL	ISBN	AUTOR
0007	Dr. No	3-125	James Bond
1201	Objektbanken	3-111	Heuer
4711	Datenbanken	3-765	Vossen
4712	Datenbanken	3-891	Ullman
4717	PASCAL	3-999	Wirth

Ergebnis:

INV.NR	TITEL
1201	Objektbanken

## Anfrageoperationen (3)

Einleitung  
Welt  
ohne DB  
Datenbanken  
Übersicht  
Terminologie  
[Querying](#)  
DB –  
Merkmale  
Anwend.  
Historie  
Schema-  
Arch.  
Schluß

- 1 Weitere Operationen: Join, Vereinigung, Differenz, Durchschnitt, Umbenennung.
- 1 Alle Operationen beliebig kombinierbar („Algebra“, *Query-Algebra*).

## Optimierer

Einleitung  
Welt  
ohne DB  
Datenbanken  
Übersicht  
Terminologie  
[Querying](#)  
DB –  
Merkmale  
Anwend.  
Historie  
Schema-  
Arch.  
Schluß

Problem: Finde einen Relationenalgebra-Ausdruck, der äquivalent ist („das gleiche Ergebnis liefert“) zum gegebenen, aber effizienter auszuwerten ist.

## Optimierer: Algebraische Optimierung

Einleitung  
Welt  
ohne DB  
Datenbanken  
Übersicht  
Terminologie  
[Querying](#)  
DB –  
Merkmale  
Anwend.  
Historie  
Schema-  
Arch.  
Schluß

- 1 Allgemeine Regel:
  1.  $\sigma_{A1 = Konst}(\sigma_{A2 = Konst}(REL))$
  2.  $\sigma_{A2 = Konst}(\sigma_{A1 = Konst}(REL))$sind äquivalent.
- 1 Beispiel:
  1.  $\sigma_{Vorname = 'Klemens'}(\sigma_{Wohnort = 'KA'}(PERSON))$
  2.  $\sigma_{Wohnort = 'KA'}(\sigma_{Vorname = 'Klemens'}(PERSON))$
- 1 Zweite Variante bei ‚natürlicher‘ Verteilung der Daten überlegen.

## Physische Datenunabhängigkeit (1)

Einleitung  
Welt  
ohne DB  
Datenbanken  
Übersicht  
Terminologie  
[Querying](#)  
DB –  
Merkmale  
Anwend.  
Historie  
Schema-  
Arch.  
Schluß

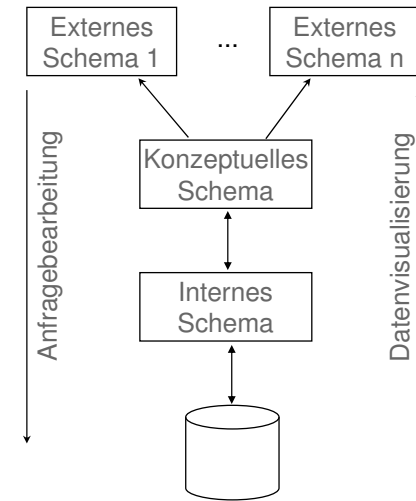
- 1 Anfragen sind *deklarativ*, Anwender sagt nur, welches Ergebnis, nicht wie es ermittelt werden soll.
- 1 Beispiel:  
**select \***  
**from** PERSON  
**where** VORNAME = 'Klemens'  
**and** WOHNORT = 'KA'
- 1 Datenbank 1:  
10 Tupel mit VORNAME = 'Klemens',  
1000 Tupel mit WOHNORT = 'KA'.
- 1 Datenbank 2: umgekehrt.

## Physische Datenunabhängigkeit (2)

- 1 Physische Datenunabhängigkeit – DBMS stellt sicher,
  - u daß Anfrage weiterhin gut funktioniert, auch wenn physische Darstellung der Daten sich geändert hat.
  - u daß Anfrage in unterschiedlichen Datenbanken (gleiches Schema, aber unterschiedliche Häufigkeiten der Daten) funktioniert.
- 1 Erlaubt höhere Komplexität bei Anwendungsentwicklung.

Z

## 3-Ebenen Architektur



## Datenunabhängigkeit (1)

- 1 Warum hat man externe Sichten?
  - 1 Unterschiedlichen Ausschnitt der Daten für unterschiedliche Benutzer (Studierende, Professor, Dekan, Raumverwaltung, Prüfungsamt)
    - u Datenschutz,
    - u Übersichtlichkeit,
    - u organisatorische Gründe (Ablenkung, Einmischung),
    - u Verstecken von Änderungen am konzeptionellen Schema
      - Benutzer wird nicht irritiert,
      - Anwendungsprogramme laufen weiterhin.
- Logische Datenunabhängigkeit.*

## Datenunabhängigkeit (2)

- 1 Mögliche Auswirkungen von Änderungen am konzeptuellen Schema:
  - u Eventuell externe Schemata betroffen (Ändern von Attributen),
  - u eventuell Anwendungsprogramme betroffen (Rekompilieren der Anwendungsprogramme, eventuell Änderungen nötig).

Z

## Prinzipien

Einleitung  
Welt  
ohne DB  
Datenbanken  
Übersicht  
Terminologie  
Querying  
DB-  
Merkmale  
Anwend.  
Historie  
Schema-  
Arch.  
Schluß

- 1 Grundprinzip moderner Datenbanksysteme
  - u *3-Ebenen-Architektur*  
(physische Datenunabhängigkeit,  
logische Datenunabhängigkeit)
  - u *Trennung zwischen Schema*  
(etwa Tabellenstruktur)  
*und Instanz* (etwa Tabelleninhalt).
- 1 Angelehnt an *9 Codd'sche Regeln*.  
(Eigenschaften, die Datenbank haben muß,  
um als relational zu gelten.)

## Die neun Codd'schen Regeln

Einleitung  
Welt  
ohne DB  
Datenbanken  
Übersicht  
Terminologie  
Querying  
DB-  
Merkmale  
Anwend.  
Historie  
Schema-  
Arch.  
Schluß

1. Integration:  
einheitliche, nichtredundante Datenverwaltung,
2. Operationen: Speichern, Suchen, Ändern,
3. Katalog: Zugriffe auf Datenbankbeschreibungen  
im Data Dictionary,
4. Benutzersichten,
5. Integritätssicherung:  
Korrektheit des Datenbankinhalts,
6. Datenschutz: Ausschluß unautorisierter Zugriffe,
7. Transaktionen:  
mehrere DB-Operationen als Funktionseinheit,
8. Synchronisation:  
parallele Transaktionen koordinieren,
9. Datensicherung:  
Wiederherstellung von Daten nach Systemfehlern.

## Data Dictionary – Illustration

AUSLEIH	INV.NR	NAME
...	...	...

BUCH	INV.NR	TITEL	ISBN	AUTOR
...	...	...	...	...

REL	RNR	NAME	TYP	SCHLÜSSEL
	1	AUSLEIH		
	2	BUCH		

ATTR	RNR	NAME	TYP	SCHLÜSSEL
	1	INV.NR	INT	true
	1	NAME	STRING	false
	2	INV.NR	INT	true
	2	TITEL	STRING	false
	...	...	...	...

Z

## Mögliche Prüfungsfragen

Einleitung  
Welt  
ohne DB  
Datenbanken  
Übersicht  
Terminologie  
Querying  
DB-  
Merkmale  
Anwend.  
Historie  
Schema-  
Arch.  
Schluß

- 1 Was ist eine Sicht?
- 1 Was ist die relationale Algebra?  
Wozu braucht man sie?
- 1 Geben Sie Beispiele für Algebra-Ausdrücke,  
die nicht identisch, aber äquivalent sind?
- 1 Was leistet der Anfrageoptimierer  
einer Datenbank?
- 1 Erklären Sie: Drei-Ebenen Architektur,  
physische Datenunabhängigkeit,  
logische Datenunabhängigkeit.

## Einbindung in Programme

Einleitung  
Welt  
ohne DB  
Datenbanken  
Übersicht  
Terminologie  
Querying  
DB –  
Merkmale  
Anwend.  
Historie  
Schema-  
Arch.  
Schluß

- 1 Nahtlose Einbindung von Querysprachen in Programme.
  1. Queryergebnis ans Programm.
  2. Programmvariablen als Bestandteil der Anfrage.
- 1 Beispiel JDBC:
  - ...

```
String query = "SELECT titel, preis, "  
+ "bestand FROM buch";  
  
Statement stmt = con.createStatement ();  
ResultSet rs = stmt.executeQuery (query);
```

Klemens Böhm

IWM: Einleitung – 53

## Zugriff über WWW

Einleitung  
Welt  
ohne DB  
Datenbanken  
Übersicht  
Terminologie  
Querying  
DB –  
Merkmale  
Anwend.  
Historie  
Schema-  
Arch.  
Schluß

```
<?php  
/* Verbindung aufbauen, auswählen einer Datenbank */  
$link = mysql_connect("mysql_host", "mysql_user",  
"mysql_password")  
or die("Keine Verbindung möglich!");  
print "Verbindung zum Datenbankserver erfolgreich";  
mysql_select_db("Meine_DB") or die("Auswahl der DB  
fehlgeschlagen");  
  
// ausführen einer SQL Anfrage  
$query = "SELECT * FROM Meine_Tabelle";  
$result = mysql_query($query) or die("Anfrage  
fehlgeschlagen");  
  
// Ausgabe der Ergebnisse in HTML  
print "<table>\n";  
while ($line = mysql_fetch_array($result, MYSQL_ASSOC)) {  
    print "\t<tr>\n";  
    foreach ($line as $col_value) {  
        print "\t\t<td>$col_value</td>\n";  
    }  
    print "\t</tr>\n";  
}  
  
mysql_free_result($result); // Freigeben des Resultsets  
  
mysql_close($link); // schliessen der Verbindung  
?>
```

Klemens Böhm

IWM: Einleitung – 54

## „What about performance?“

Einleitung  
Welt  
ohne DB  
Datenbanken  
Übersicht  
Terminologie  
Querying  
DB –  
Merkmale  
Anwend.  
Historie  
Schema-  
Arch.  
Schluß

- 1 Andere Software-Systeme können große Mengen von Daten nicht effizient verarbeiten.
- 1 Schein-Gegenargument:
  - u „Generischer Code stets langsamer.“
  - u B-Baum Implementierung in MS-SQL  
→ Weltrekord TPC-C.

Klemens Böhm

IWM: Einleitung – 55

## Zahlen – Bank

Einleitung  
Welt  
ohne DB  
Datenbanken  
Übersicht  
Terminologie  
Querying  
DB –  
Merkmale  
Anwend.  
Historie  
Schema-  
Arch.  
Schluß

- Große europäische Bank; Zahlen von 05/2005:
- 1 > 70.000 Datenbanken (zumeist DB2, Oracle, IMS, SQLServer)
  - 1 Größte Datenbank:
    - u 25 Mio. Transaktionen pro Tag (30 pro Sekunde),
    - u Datenvolumen: 80 TB.
  - 1 99,7% Verfügbarkeit,
  - 1 210 Mio. gedruckte Seiten pro Jahr,
  - 1 60 Mitarbeiter nur für Datenbankadministration.

Klemens Böhm

IWM: Einleitung – 56

## Zahlen – S.A.P.

Einleitung  
Welt  
ohne DB  
Datenbanken  
Übersicht  
Terminologie  
Querying  
DB –  
Merkmale  
Anwend.  
Historie  
Schema-  
Arch.  
Schluß

### Zahlen von 05/2005)

- 1 Anzahl Tabellen in SAP ERP (früher R/3):  
ca. 20000,  
(eher wenige große Tabellen; viele kleine,  
die den Charakter von .cfg oder .ini Files  
in einem OS haben)
- 1 Anzahl Nutzer
  - u registrierte Benutzer: Tausende
  - u "concurrent users": 20-50 (i. Allg.)

## Datenbanken sind groß. (1)

Einleitung  
Welt  
ohne DB  
Datenbanken  
Übersicht  
Terminologie  
Querying  
DB –  
Merkmale  
Anwend.  
Historie  
Schema-  
Arch.  
Schluß

- 1 Warum dieser Aufwand?  
Reichen herkömmliche Techniken wirklich nicht?
- 1 Datenbanken sind groß:
  - u Größe einzelner Relationen,
  - u Anzahl der Relationen,
  - u Anfragen sind komplex  
(synthetisch generierte Anfragen haben leicht  
ein Dutzend oder mehr Joins),
  - u viele Benutzer,
  - u viele gleichzeitige Zugriffe,
  - u hohe anliegende Arbeitslast.

## Datenbanken sind groß. (2)

Einleitung  
Welt  
ohne DB  
Datenbanken  
Übersicht  
Terminologie  
Querying  
DB –  
Merkmale  
Anwend.  
Historie  
Schema-  
Arch.  
Schluß

- 1 Unter diesen Umständen
  - u spielt Reihenfolge der Joins große Rolle  
(physische Datenunabhängigkeit),
  - u sind logische Datenunabhängigkeit,
  - u ausgefeilte Nebenläufigkeits-  
Zugriffsmechanismen,
  - u Benutzerverwaltung etc. unumgänglich.

## Einsatzgebiete und Grenzen (1)

Einleitung  
Welt  
ohne DB  
Datenbanken  
Übersicht  
Terminologie  
Querying  
DB –  
Merkmale  
Anwend.  
Historie  
Schema-  
Arch.  
Schluß

- 1 Klassische Einsatzgebiete:
  - u Viele Objekte (15000 Bücher, 300 Benutzer,  
100 Ausleihvorgänge pro Woche, ...)
  - u wenige Objekttypen  
(BUCH, BENUTZER, AUSLEIHUNG),
  - u etwa Buchhaltungssysteme,  
Auftragserfassungssysteme,  
Bibliothekssysteme, Handel,  
Lagerhaltung und Produktion, Logistik,  
Finanzen (Banken, Versicherungen usw.),  
Forschung und Entwicklung.

## Einsatzgebiete und Grenzen (2)

Einleitung  
Welt  
ohne DB  
Datenbanken  
Übersicht  
Terminologie  
Querying  
DB –  
Merkmale  
Anwend.  
Historie  
Schema-  
Arch.  
Schluß

- 1 Aktuelle Anwendungen:
  - u entscheidungsunterstützende Systeme (Data Warehouses, OLAP),
  - u E-Commerce,
  - u NASA Earth Observation System (Petabyte-Datenbanken),
  - u Data Mining.

## Einsatzgebiete und Grenzen (3)

Einleitung  
Welt  
ohne DB  
Datenbanken  
Übersicht  
Terminologie  
Querying  
DB –  
Merkmale  
Anwend.  
Historie  
Schema-  
Arch.  
Schluß

- Normalerweise sind herkömmliche Datenbanksysteme überfordert mit:
- 1 Expertensysteme – wenige Objekte, viele Objekttypen, kompliziertere Operationen; ABER: Deduktive Datenbanksysteme,
  - 1 Ähnlichkeitssuche, Ranking – anderes Paradigma, keine gute interne Unterstützung, ABER: Anfragesprachen, aktuelle Forschung zu entsprechenden Erweiterungen existierender Systeme.

## Einsatzgebiete und Grenzen (4)

Einleitung  
Welt  
ohne DB  
Datenbanken  
Übersicht  
Terminologie  
Querying  
DB –  
Merkmale  
Anwend.  
Historie  
Schema-  
Arch.  
Schluß

- Normalerweise sind herkömmliche Datenbanksysteme überfordert mit (Forts.):
- 1 XML, Multimedia – zuviele Datenobjekte; einfache Informationsbedürfnisse → zuviele komplexe Anfragen, kein spezieller Multimedia-Support; ABER: XML-Datenbanken, Multimedia-Erweiterungen, ORDBMSe.

## Historie

Einleitung  
Welt  
ohne DB  
Datenbanken  
Übersicht  
Terminologie  
Querying  
DB –  
Merkmale  
Anwend.  
Historie  
Schema-  
Arch.  
Schluß

- 1 Anfang 60er Jahre: elementare Dateien, anwendungsspezifische Datenorganisation (geräteabhängig, redundant, inkonsistent)
- 1 Ende 60er Jahre: Dateiverwaltungssysteme (SAM, ISAM) mit Dienstprogrammen (Sortieren) (geräteunabhängig, aber redundant und inkonsistent)
- 1 70er Jahre: Datenbanksysteme (Geräte- und Datenunabhängigkeit, redundanzfrei, konsistent)

## System-Architekturen

Einleitung  
Welt  
ohne DB  
Datenbanken  
Übersicht  
Terminologie  
Querying  
DB –  
Merkmale  
Anwend.  
Historie  
Schema-  
Arch.  
- [Einleit.](#)  
- [ANSI-  
SPARC](#)  
Schluß

- 1 Beschreibung der Komponenten eines Datenbanksystems.
- 1 Standardisierung der Schnittstellen zwischen Komponenten.
- 1 Architekturvorschläge:
  - u ANSI-SPARC-Architektur  
~>Drei-Ebenen-Architektur.
  - u Fünf-Schichten-Architektur  
~>beschreibt Transformationskomponenten.

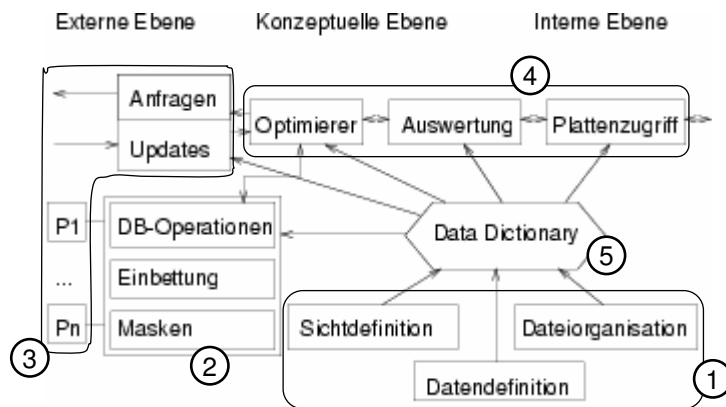
## ANSI-SPARC-Architektur (1)

Einleitung  
Welt  
ohne DB  
Datenbanken  
Übersicht  
Terminologie  
Querying  
DB –  
Merkmale  
Anwend.  
Historie  
Schema-  
Arch.  
- [Einleit.](#)  
- [ANSI-  
SPARC](#)  
Schluß

- 1 ANSI: American National Standards Institute.
- 1 SPARC: Standards Planning and Requirement Committee.
- 1 Vorschlag von 1978.
- 1 Im wesentlichen Grobarchitektur verfeinert:
  - u Interne Ebene / Betriebssystem verfeinert,
  - u mehr interaktive und Programmier-Komponenten,
  - u Schnittstellen bezeichnet und normiert.

## ANSI-SPARC-Architektur (2)

Einleitung  
Welt  
ohne DB  
Datenbanken  
Übersicht  
Terminologie  
Querying  
DB –  
Merkmale  
Anwend.  
Historie  
Schema-  
Arch.  
- [Einleit.](#)  
- [ANSI-  
SPARC](#)  
Schluß



## Klassifizierung der Komponenten

Einleitung  
Welt  
ohne DB  
Datenbanken  
Übersicht  
Terminologie  
Querying  
DB –  
Merkmale  
Anwend.  
Historie  
Schema-  
Arch.  
- [Einleit.](#)  
- [ANSI-  
SPARC](#)  
Schluß

1. **Definitionskomponenten:** Datendefinition, Dateiorganisation, Sichtdefinition,
2. **Programmierkomponenten:** DB-Programmierung mit eingebetteten DB-Operationen,
3. **Benutzerkomponenten:** Anwendungsprogramme, Anfrage und Update interaktiv,
4. **Transformationskomponenten:** Optimierer, Auswertung, Plattenzugriffssteuerung,
5. **Data Dictionary (Datenwörterbuch):** Aufnahme der Daten aus Definitionskomponenten, Versorgung der anderen Komponenten.

## Einige konkrete Systeme

- 1 (Objekt-)Relationale DBMS:
  - u Oracle9i, IBM DB2 V.7,  
Microsoft SQL Server 2000,
  - u MySQL, PostgreSQL.
- 1 Objektorientierte DBMS:  
Poet, Versant, ObjectStore,
- 1 XML-DBMS:  
Tamino (Software AG), eXcelon.

## Überblick

- ... über dieses und die nächsten Kapitel:
- 1 Grundlegende Konzepte und Architekturen,
  - 1 Anfrage- und Änderungsoperationen,
  - 1 relationale Datenbanksprachen,
  - 1 Datenbankmodelle für den Entwurf,
  - 1 Datenbankmodelle für die Realisierung,
  - 1 Datenbankentwurf und Datendefinition,
  - 1 Sichten, Datenschutz.