

Kap. 1: Einführung / DBS-Architektur

- Aufbau von DBS
 - 3-Schichtenmodell
 - 5-Schichtenmodell
- Datenbankadministration / Tuning

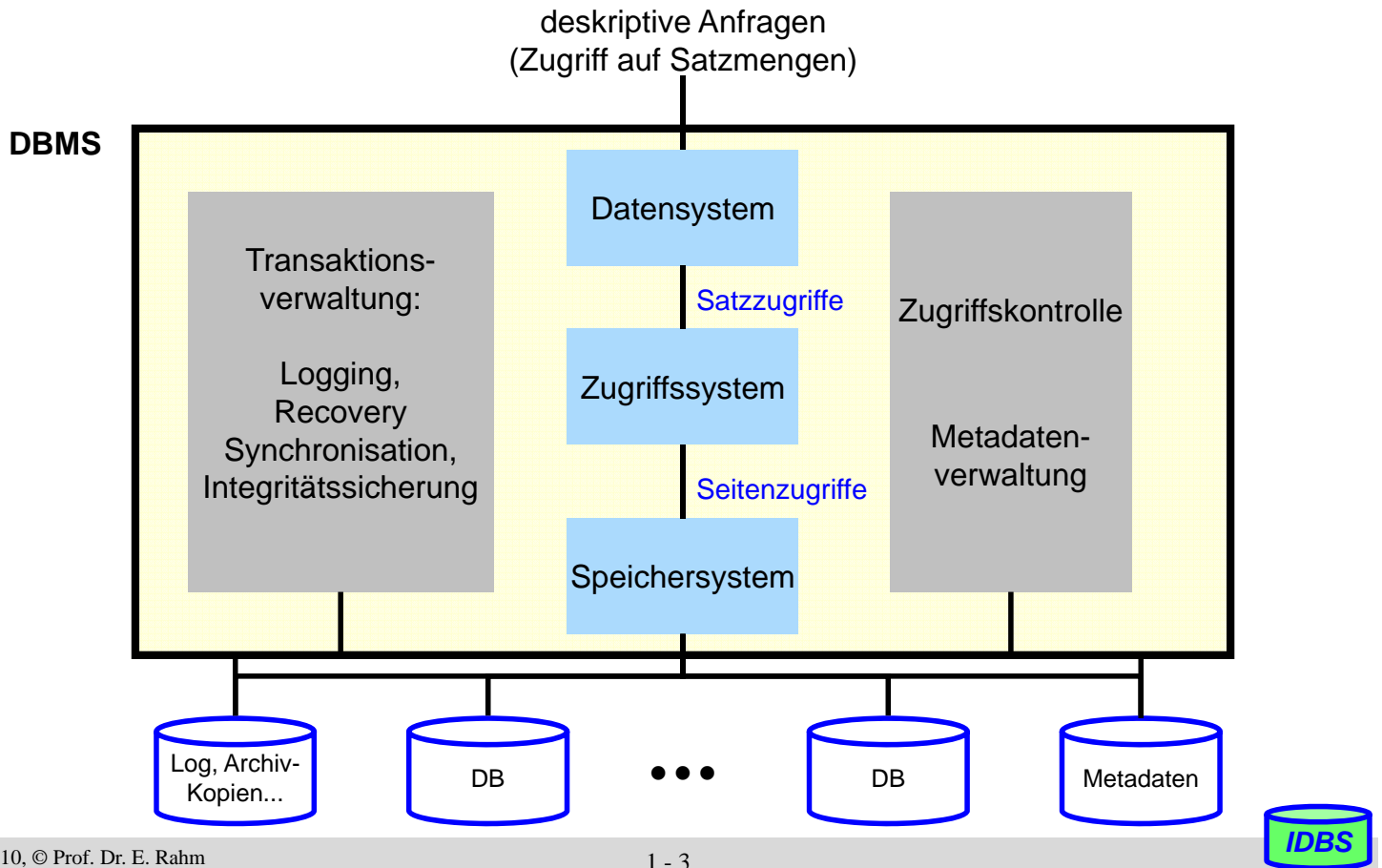


Einsatz von Datenbanksystemen

- Merkmale
 - effiziente Verwaltung großer Datenmengen
 - hohe Datenunabhängigkeit, Datenstrukturierung, mächtige Operationen
 - Transaktionskonzept mit Mehrbenutzerfähigkeit und automatischer Fehlerbehandlung ...
- Einsatzformen
 - "kommerzielle" Anwendungen im Unternehmensbereich: **Online-Shops** für Auskünfte und Buchungen / Bestellungen , Produktionsplanung und -steuerung, Personalverwaltung, Datenerfassung ...
 - wissenschaftliche Anwendungen (e-Science)
 - Verwaltung von Dokumenten und Multimedia-Objekten
 - relationale DBS, objekt-relationale DBS, XML-DBS
- Architekturausprägungen
 - Zentraler Server, z.B. mit Web-Anbindung über Applikations-Server
 - Parallele DBS im lokalen Cluster
 - Heterogene verteilte DBS (z.B. Mediator-gekoppelt)

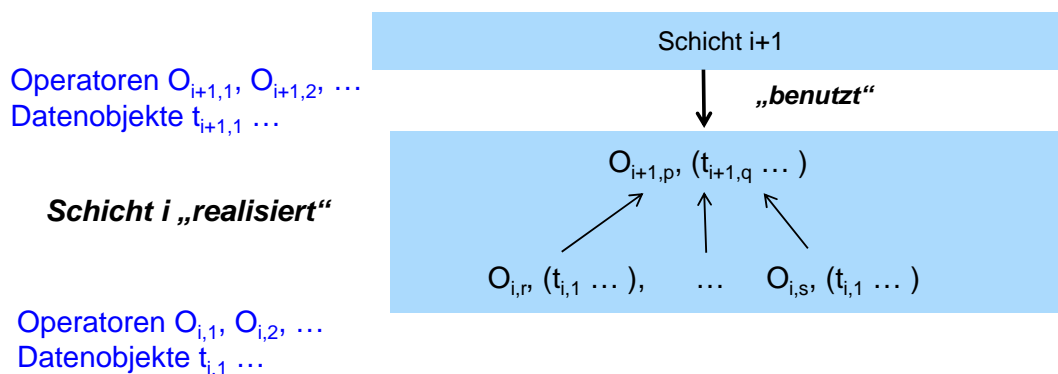


Grobaufbau eines DBS



Schichtenmodelle

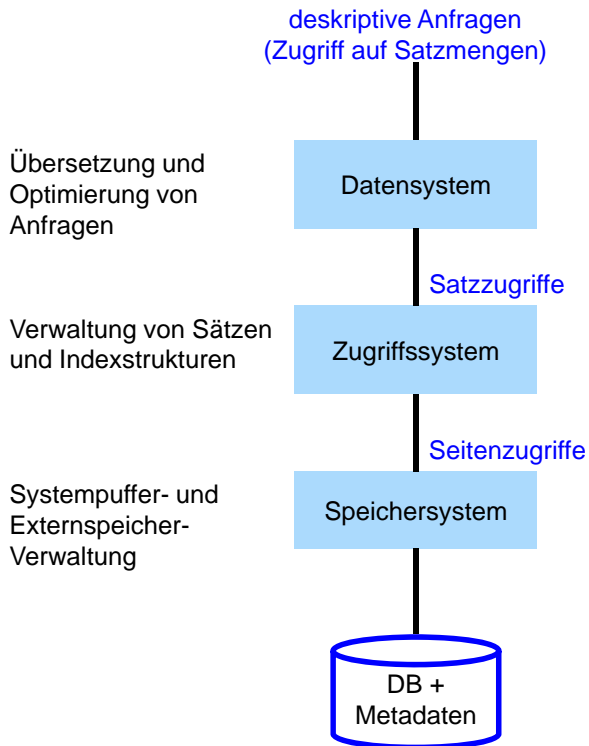
- Ziel: Architektur eines datenunabhängigen DBS
- Bildung von konzeptionellen Schichten zur hierarchischen Strukturierung
- Jede Hierarchieebene kann als virtuelle Maschine aufgefasst werden



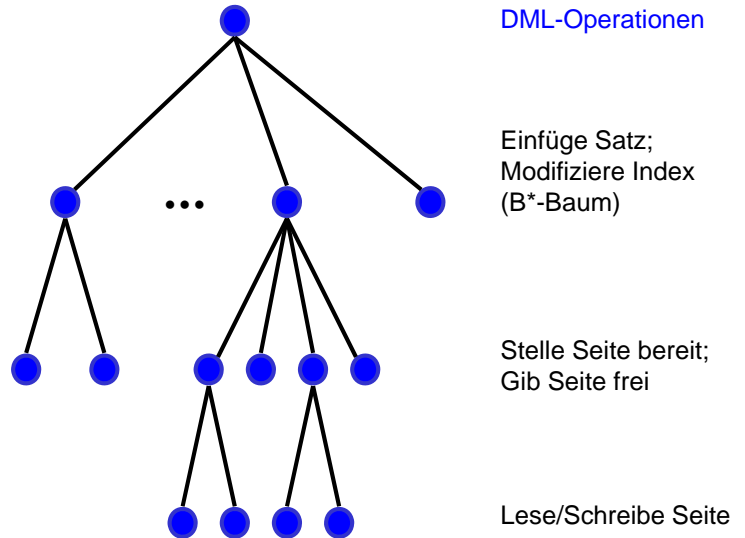
- Vorteile hierarchischer Schichtenbildung
 - höhere Ebenen (Systemkomponenten) werden einfacher, weil sie tiefere Ebenen (Systemkomponenten) benutzen können
 - tiefere Ebenen können getestet werden, bevor die höheren Ebenen lauffähig sind
 - Änderungen in Ebenen werden gekapselt

Grobaufbau eines DBS (2)

Schichtenmodell

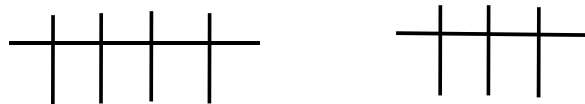


Dynamischer Kontrollfluss einer DB-Operation

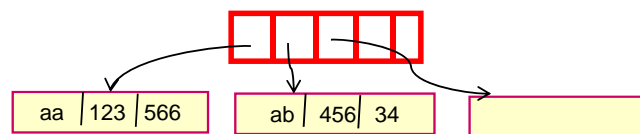


Beispiel für Daten-Abbildung

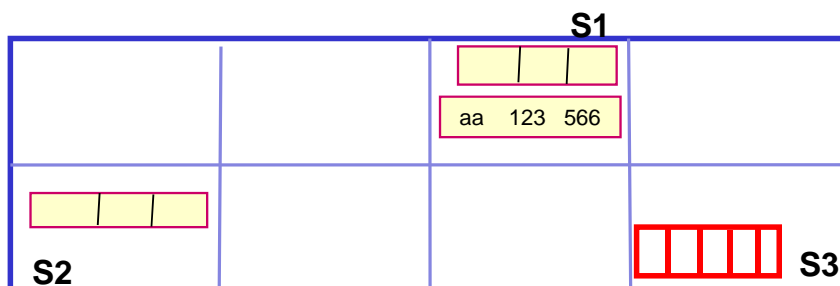
Relationen,
Sichten, Tupel



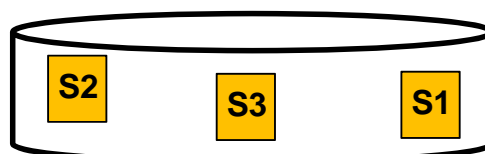
Interne Sätze



DB-Puffer
mit Seiten



Externspeicher
mit Blöcken



Datenunabhängigkeit im Überblick

Ebene	Was wird verborgen?
Datensystem	Zahl und Art der physischen Zugriffspfade/ Indexstrukturen; interne Satzdarstellung
Zugriffssystem	Pufferverwaltung; Logging
Speichersystem	Dateiabbildung, technische Eigenschaften und Betriebsdetails der externen Speichermedien



5-Schichten-Modell

Mengenorientierte Schnittstelle (i.a. SQL)

Adressierungseinheiten: Relationen, Sichten, Tupel

5

Logische Datenstrukturen

Hilfsstrukturen: ext. Schemabeschreibung, Integritätsregeln

Query-Übersetzung und -Optimierung

Satzorientierte Schnittstelle

ext. Sätze, Schlüssel, Zugriffspfade

4

Logische Zugriffspfadstrukturen

Hilfsstrukturen: Zugriffspfaddaten, interne Schemabeschreibung

Cursor-Verwaltung, Sortierkomponente

Interne Satzchnittstelle

interne Sätze, B*-Bäume, Hash-Tabellen ...

3

Speicherungsstrukturen

Hilfsstrukturen: Freispeicher-Infos, Adresstabellen ...

Record- und Index-Manager

DB-Pufferschnittstelle

Seiten, Segmente

2

Seitenzuordnung

Hilfsstrukturen: Seitentabellen, Blocktabellen ...

Pufferverwaltung, Segmentverwaltung

Dateischnittstelle

Blöcke, Dateien

1

Speicherzuordnungsstrukturen

Hilfsstrukturen: Freispeicher-Info, Extent-Tabellen, Dateikataloge ...

Datei- und Externspeicher-verwaltung

Geräteschnittstelle

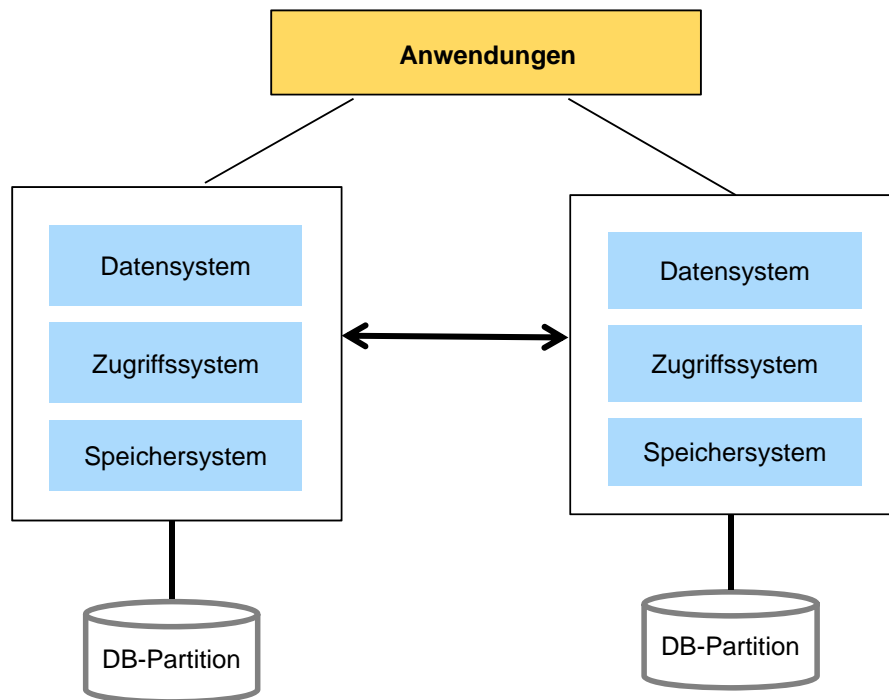
Spuren, Zylinder, Kanäle, usw.



Physische Datenträger



Architektur von Mehrrechner-DBS



- Knoten in Mehrrechner-DBS haben ähnliche Funktionen wie ein zentrales DBS
- Unterschiedliche Möglichkeiten der Kooperation zwischen DBS

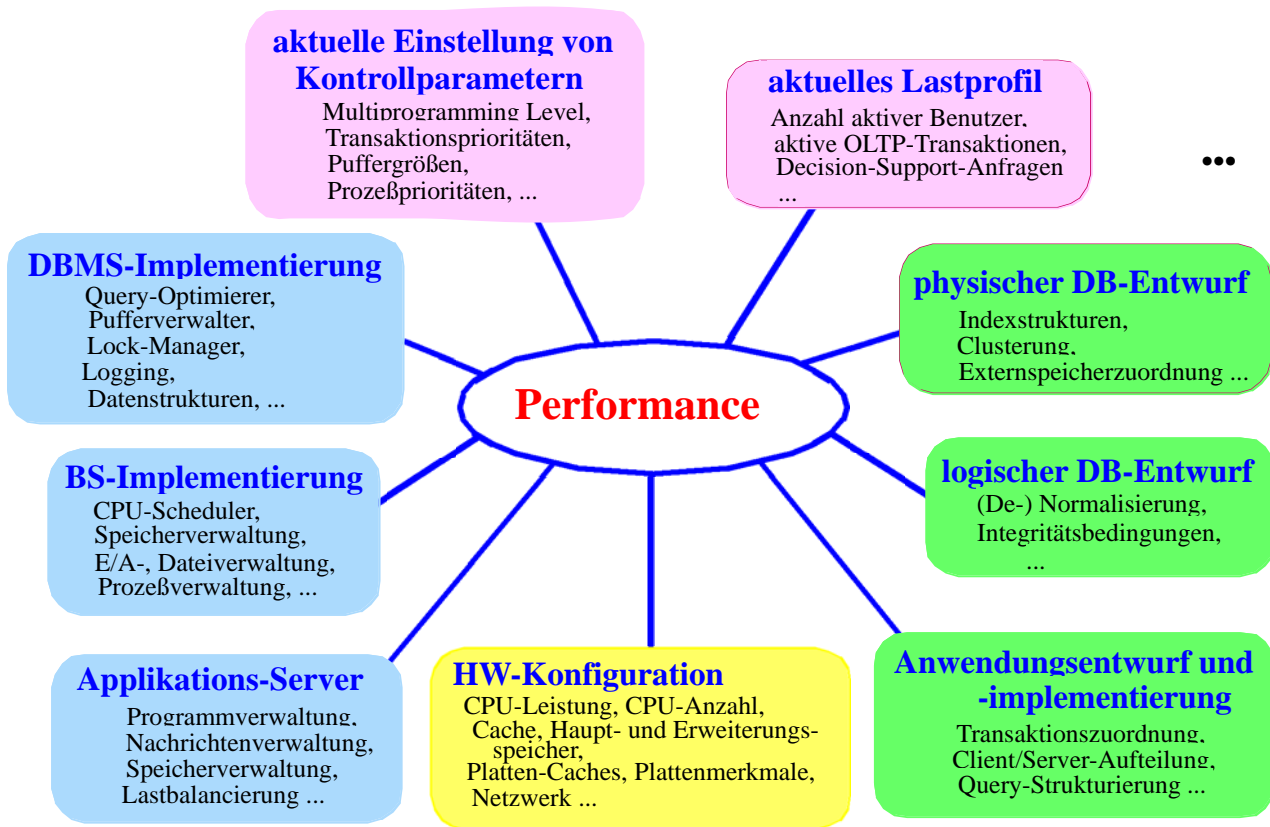
Datenbankadministrator (DBA)

■ Aufgaben

- Festlegung des internen (physischen) DB-Schemas
- ggf. Entwurf des konzeptionellen DB-Schemas mit Festlegung von Integritätsbedingungen
- Kontrolle über die externen Schemata
- Vergabe / Entzug von Zugriffsberechtigungen
- Strategie für Datensicherung und Recovery festlegen
- Installation neuer DBMS-Releases
- Überwachung des Systembetriebs
- **Performance-Tuning** (höherer Durchsatz, bessere Antwortzeiten) ••

■ Unterstützung durch Dienstprogramme, z.B. für

- Laden der DB
- Erstellen von Archivkopien
- DB-Reorganisation
- Erfassung und Analyse von Messwerten



DBS-Kontrollparameter (Bsp. Sybase)

Function	Default	Function	Default	Function	Default
max online engines	1	deadlock checking period	500	default fill factor percent	0
# user connections	25	deadlock retries	5	i/o accounting flush interval	1000
# remote connections	20	page lock spinlock ratio	100	max network packet sizem	512
total memory	-	table lock spinlock ratio	20	remote server pre-read packets	3
user log cache size	2 KB	partition spinlock ratio	10	stack size	-
procedure cache percent	20	user log cache spinlock ratio	20	time slice	100
number of sort buffers	0	number of open databases	12	event buffers per engine	100
sort page count	0	number of open objects	500	freelock transfer block size	30
housekeeper free write percent	1	number of index trips	0	max engine freelocks	10
recovery interval	5 min.	i/o polling process count	10	page utilization percent	95
additional network memory	0	disk i/o structures	256	partition groups	1024
audit queue size	100	max async i/os per engine	2 Mrd.		
number of locks	5000	max async i/os per server	2 Mrd.		
lock promotion HWM	200	# pre-allocated extents	2		
lock promotion LWM	200	number of devices	10		
lock promotion PCT	100	# extent i/o buffers	0		





Tuning-Probleme

■ Status Quo in derzeitigen Transaktions- und Datenbanksystemen:

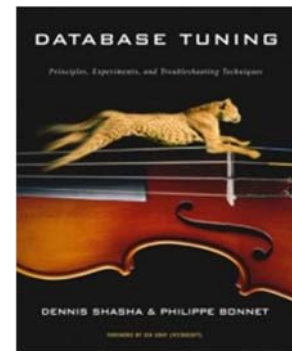
- weitgehend manuelles Tuning durch Systemverwalter (z.B. DBA)
- komplizierte Systemverwaltung durch Vielzahl von internen Parametern
- mangelnde Abstimmung zwischen Resource-Managern, insbesondere zwischen DBS, Applikations-Server und Betriebssystem
- unzureichende Unterscheidung verschiedener Lastgruppen bei der Zuteilung von Betriebsmitteln (Sperrern, Pufferplatz, etc.)

■ Verschärfung der Situation in verteilten Systemen / Mehrrechner-DBS

- stark zunehmende Anzahl von Kontrollparametern
- Festlegung der Verteilung von Daten, Programmen und Lasteinheiten
- komplexere Lastsituationen
- ggf. heterogene Systemstruktur

Tuning-Leitlinien*

- Think globally, fix locally (does it matter?)
 - Slow query. Tune it? (Check: Is query frequent enough?)
 - Disk is saturated. Buy new disk ? (Check: scans instead of index usage? log and data mixed?)
- Partitioning breaks bottlenecks (spatial and temporal)
 - dividing work across several resources to avoid saturation
 - e.g. data partitioning; multiple data structures; splitting of transactions;
- Start-up costs are high; running costs are low
 - disk access (number of IOs more significant than data volume)
 - client/server communication (number of messages more critical than message size)
 - high cost of query parsing (save compiled query)
- Be prepared for trade-offs
 - indexes and inserts

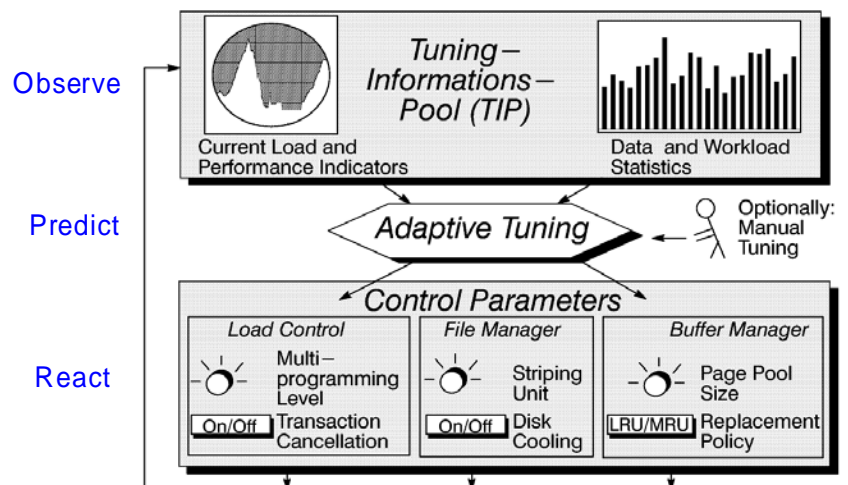


* Shasha, D./Bonnet, P.: *Database Tuning*. Morgan Kaufmann, 2003



Automatisches Tuning?

- zunehmende Unterstützung durch „Advisors“ für Indexierungsvorschläge, materialisierte Sichten etc. (Microsoft SQL-Server, DB2, Oracle ...)
- weitergehende Zielvorstellung: **Autonomic Computing** (Self-Configuring, Self-Tuning, Self-Repairing, ...)
- erfordert u.a. automatische Überwachung und Steuerung der Datenbankverarbeitung
 - Online-Monitoring
 - automatische Erkennung von Leistungsproblemen
 - automatische Bestimmung der Engpässe
 - automatische Linderung der Probleme durch geeignete Anpassung von Kontrollparametern
- Probleme:
 - Stabilität, hohe Dynamik und große wechselseitige Abhängigkeiten
 - Effizienz vs. Effektivität



Zusammenfassung

- Schichtenmodell ist allgemeines Erklärungsmodell für die Realisierung von DBS
 - unterschiedliche Schichtenbildungen möglich (z.B. 3-Schichten-Ansatz)
 - Schichtenbildung unterstützt Datenunabhängigkeit
- Implementierungskonzepte zentralisierter DBS finden sich auch in Mehrrechner-DBS
 - jeder Knoten verfügt über alle Funktionen eines zentralisierten DBS
- DBA: verantwortlich für sicheren Betrieb und effektive Nutzung großer Datenbanken
- DB-Tuning (Performance-Optimierung)
 - hohe Komplexität aufgrund zahlreicher Parameter, wechselseitigen Abhängigkeiten und hoher Dynamik
 - zunehmende Tool-Unterstützung / Automatisierung für Teilaufgaben