
Datenbanksysteme II

Sommersemester 2012

Vorlesung: PD Dr. Peer Kröger

Dieses Skript basiert auf den Skripten zur Vorlesung Datenbanksysteme II an der LMU München von Prof. Dr. Christian Böhm (Sommersemester 2007), PD Dr. Peer Kröger (Sommersemester 2008) und PD Dr. Matthias Schubert (Sommersemester 2009)

http://www.dbs.ifi.lmu.de/cms/Datenbanksysteme_II_12

Team

Vorlesungsteam



PD Dr. Peer Kröger
Oettingenstr. 67, Zimmer F 108
Tel. 089/2180-9327
Sprechstunde: Di, 10⁰⁰-11⁰⁰



Thomas Bernecker
Oettingenstr. 67, Zimmer F 104
Tel. 089/2180-9321

Organisation

Termine

- Vorlesung: Do, 9.30 -12.00 Uhr Raum B001 (Oettingenstr. 67)

- Übung: Mo, 14.00 - 16.00 Uhr Raum 211 (Amalienstr. 73a)
 Mo, 16.00 - 18.00 Uhr Raum 112 (Amalienstr. 73a)
 Di, 14.00 - 16.00 Uhr Raum 018 (Amalienstr. 73a)
 Di, 16.00 - 18.00 Uhr Raum A 105 (Amalienstr. 17)
 Mi, 14.00 - 16.00 Uhr Raum D 4c (Ludwigstr. 25)
 Mi, 16.00 - 18.00 Uhr Raum 018 (Amalienstr. 73a)

Anmeldung für den Übungsbetrieb auf der Homepage

http://www.dbs.ifi.lmu.de/cms/Datenbanksysteme_II_12

Organisation

- Infos
 - Vorlesungshomepage
 - http://www.dbs.ifi.lmu.de/cms/Datenbanksysteme_II_12
 - Die Informatiker
 - <http://www.die-informatiker.net/forum/DBS>
- ECTS-Punkte- bzw. Scheinerwerb
 - Zulassung: Anmeldung für den Übungsbetrieb
 - UniWorX: <https://uniworx.ifi.lmu.de/>
 - Prüfung: Klausur

Inhalt der Vorlesung (Planung)

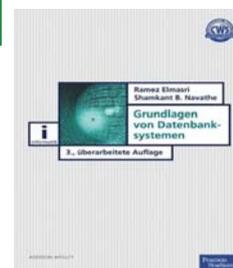
1. Einführung
2. Transaktionsverwaltung
3. Mehrbenutzersynchronisation
4. Fehlerbehandlung
5. Relationale Anfragebearbeitung
6. Data Warehouses

5

Literatur

Die Vorlesung orientiert sich nicht an einem bestimmten Lehrbuch.
Empfehlenswert sind aber u.a...

- A. Kemper, A. Eickler:
Datenbanksysteme. Eine Einführung
Oldenbourg, 6. Auflage (2006).
- R. Elmasri, S. B. Navathe:
Grundlagen von Datenbanksystemen
Pearson Studium, 3. Auflage (2004).
- G. Saake, A. Heuer , K.-U. Sattler:
**Datenbanken:
Implementierungstechniken**
mitp, 2. Auflage (2005).



6

Kapitel 1

Einführung

Vorlesung: PD Dr. Peer Kröger

Dieses Skript basiert auf den Skripten zur Vorlesung Datenbanksysteme II an der LMU München von Prof. Dr. Christian Böhm (Sommersemester 2007), PD Dr. Peer Kröger (Sommersemester 2008) und PD Dr. Matthias Schubert (Sommersemester 2009)

http://www.dbs.ifi.lmu.de/cms/Datenbanksysteme_II_12

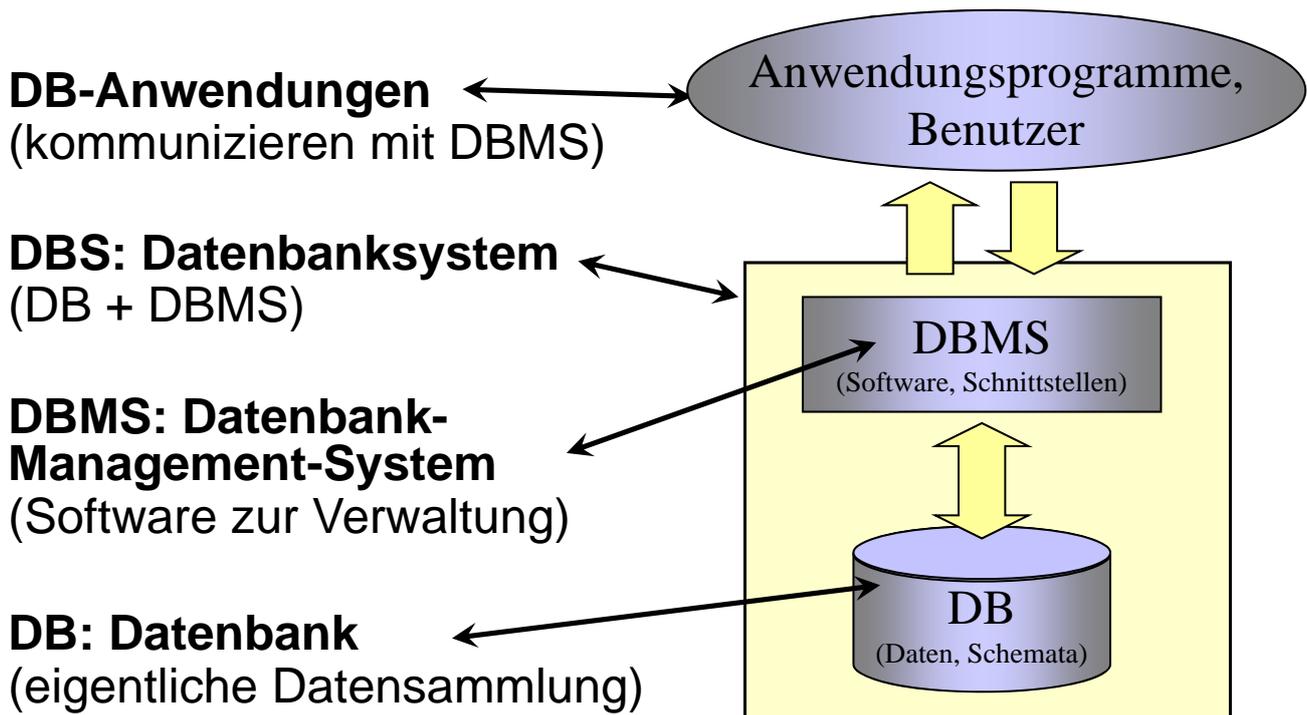
1 Einführung

1.1 Grundbegriffe

1.2 Architekturen von DBMS

1.1 Grundbegriffe

Ein Datenbanksystems (DBS) besteht aus...



9

1.1 Grundbegriffe

Liste von 9 Anforderungen (Edgar F. Codd, `82)

1. Integration

Einheitliche Verwaltung aller von Anwendungen benötigten Daten.
Redundanzfreie Datenhaltung des gesamten Datenbestandes

2. Operationen

Operationen zur Speicherung, zur Recherche und zur Manipulation der Daten müssen vorhanden sein

3. Data Dictionary

Ein Katalog erlaubt Zugriffe auf die Beschreibung der Daten

4. Benutzersichten

Für unterschiedliche Anwendungen unterschiedliche Sicht auf den Bestand

5. Konsistenzüberwachung

Überwachung der Korrektheit der Daten bei Änderungen

10

1.1 Grundbegriffe

6. Zugriffskontrolle

Ausschluss unautorisierter Zugriffe

7. Transaktionen

Zusammenfassung einer Folge von Änderungsoperationen zu einer Einheit, deren Effekt bei Erfolg permanent in DB gespeichert wird

8. Synchronisation

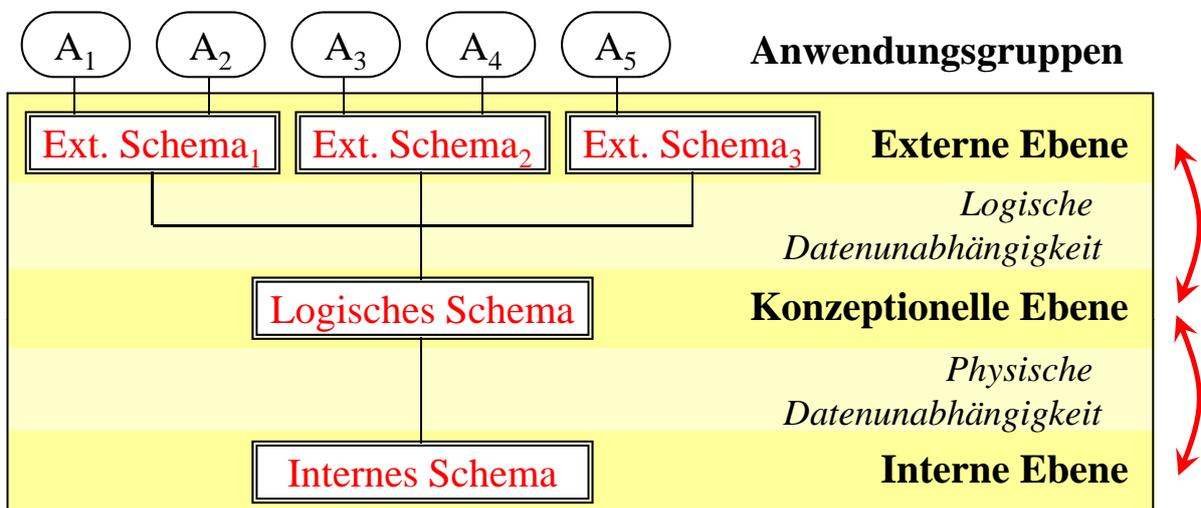
Unbeabsichtigte gegenseitige Beeinflussungen im Mehrbenutzerbetrieb werden vermieden

9. Datensicherung

Nach Systemfehlern (d.h. Absturz) oder Medienfehlern (defekte Festplatte) wird die Wiederherstellung ermöglicht (im Gegensatz zu Datei-Backup Rekonstruktion des Zustands der letzten erfolgreichen TA)

1.1 Grundbegriffe

- Drei-Ebenen-Architektur zur Realisierung von
 - physischer
 - und logischer
- Datenunabhängigkeit (nach ANSI/SPARC)



1.1 Grundbegriffe

Externe Ebene

- Sammlung der individuellen Sichten aller Benutzer- bzw. Anwendungsgruppen in mehreren externen Schemata
- Benutzer soll keine Daten sehen, die er nicht sehen will (Übersichtlichkeit) oder nicht sehen darf (Datenschutz)
 - Beispiel: Klinik-Pflegepersonal benötigt andere Aufbereitung der Daten als die Buchhaltung
- Datenbank wird damit von Änderungen und Erweiterungen der Anwenderschnittstellen abgekoppelt (*logische Datenunabhängigkeit*)

13

1.1 Grundbegriffe

Konzeptionelle Ebene

- Logische Gesamtsicht *aller* Daten der DB unabhängig von den einzelnen Applikationen
- Niedergelegt in konzeptionellem (logischem) Schema
- Ergebnis des (logischen) Datenbank-Entwurfs
- Beschreibung aller Objekttypen und Beziehungen
- Keine Details der Speicherung
- Formulierung im Datenmodell des Datenbanksystems
- Spezifikation mit Hilfe einer Daten-Definitionssprache (Data Definition Language, DDL)

14

1.1 Grundbegriffe

Interne Ebene

- Beschreibung der systemspezifischen Realisierung der DB-Objekte (physische Speicherung), z.B.
 - Aufbau der gespeicherten Datensätze
 - Indexstrukturen wie z.B. Suchbäume
- Bestimmt maßgeblich das Leistungsverhalten des gesamten DBS
- Die Anwendungen sind von Änderungen des internen Schemas nicht betroffen (*physische Datenunabhängigkeit*)

15

1 Einführung

- 1.1 Grundbegriffe
- 1.2 Architekturen von DBMS

16

1.2 Architekturen von DBMS

Schichtenmodell

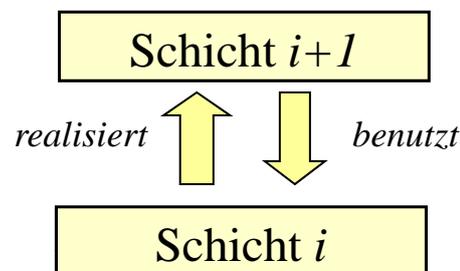
- Idee (vgl. SW-Engineering)
 - Komponenten eines komplexen Systems sind hierarchisch strukturiert
 - Keine Schicht ruft Operationen aus einer höheren Schicht auf.
- Jede Schicht definiert eine abstrakte Maschine
 - Darüber liegende Schichten setzen auf dem jeweiligen Abstraktionsgrad auf
 - Darunter liegende Schichten stellen den jeweiligen Abstraktionsgrad zur Verfügung

17

1.2 Architekturen von DBMS

Schnittstelle zwischen Schicht i und Schicht $i+1$ besteht aus Operationen O :

- Schicht i realisiert die Operationen O der Schnittstelle
- Schicht $i+1$ benutzt die Operationen O der Schnittstelle



18

1.2 Architekturen von DBMS

Vorteile einer Schichtenarchitektur

- Einfache Implementierung von Komponenten aus höheren Schichten: Sie können auf dem Abstraktionsgrad tiefer liegender Schichten aufbauen.
- Änderungen in höheren Ebenen wirken sich nicht auf tiefere Ebenen aus.
- Beim Entfernen höherer Ebenen bleiben tiefere Ebenen dennoch funktionsfähig.
- Tiefere Ebenen können getestet werden, bevor höhere Ebenen lauffähig sind.
- Verändert man auf einer tieferen Ebene die Implementierung, aber nicht die Schnittstelle (weder syntaktisch noch semantisch), so muss auch in höheren Schichten nichts geändert werden.

19

1.2 Architekturen von DBMS

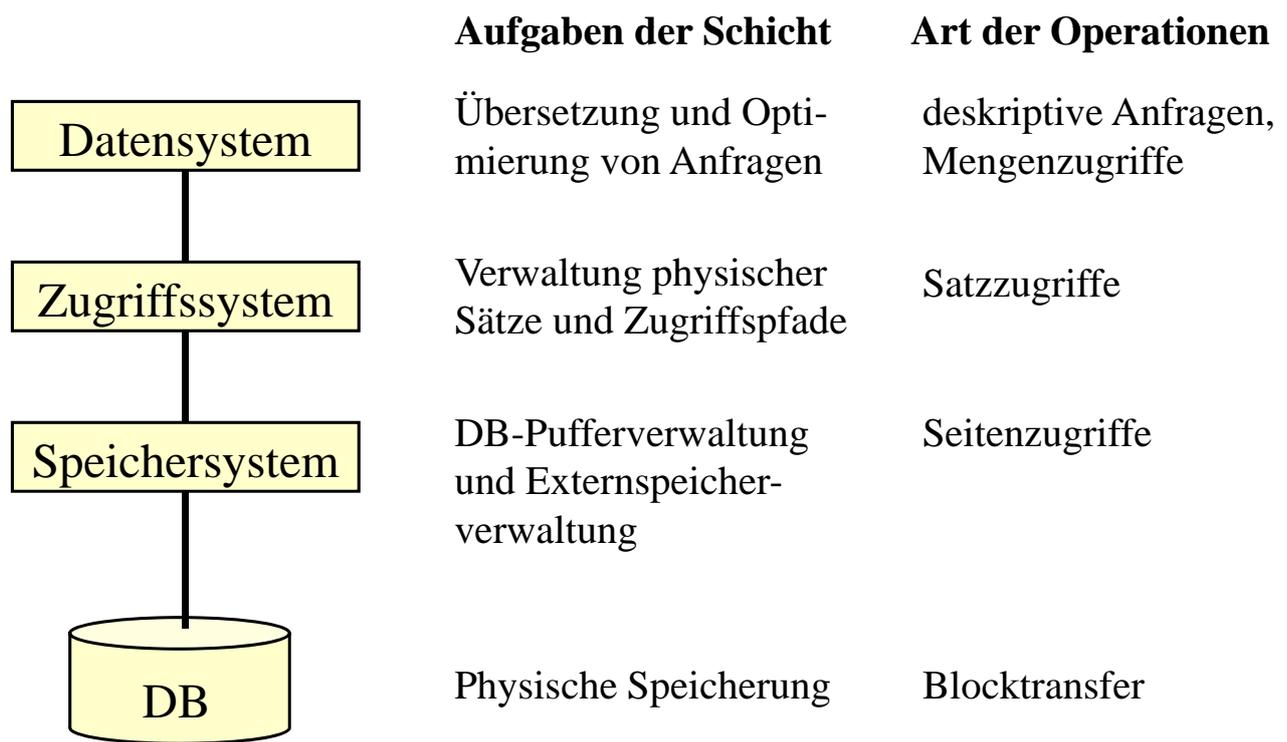
Optimale Anzahl n von Schichten

- zu wenige Schichten (z.B. $n=1$):
 - *Nachteil*: komplexe monolithische Komponenten, schwer wartbar
- zu viele Schichten (z.B. $n>10$):
 - *Vorteil*: Reduktion der Komplexität einzelner Schichten; System gut erweiterbar
 - *Nachteil*: Hohe Anzahl zu durchlaufender Schnittstellen kann zu Leistungseinbußen führen; Fehlerbehandlung kann aufwändig sein

20

1.2 Architekturen von DBMS

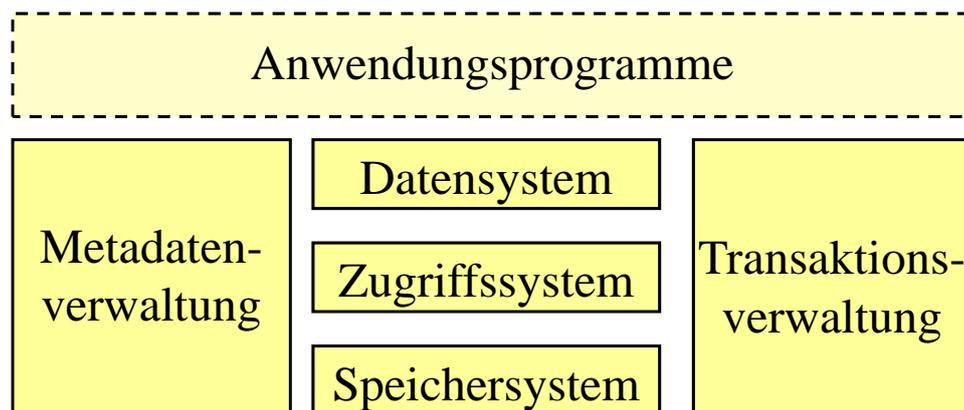
Schichten des DBMS-Kerns



21

1.2 Architekturen von DBMS

Gesamtarchitektur eines DBMS



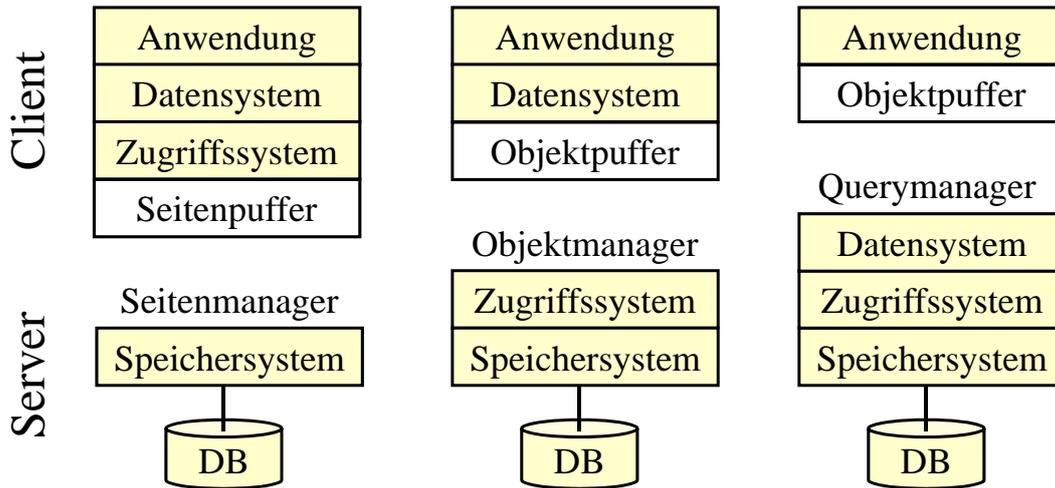
- **Daten-, Zugriffs- und Speichersystem** (wie oben) für die Grundfunktionalität
- **Metadatenverwaltung** für modellspezifische Daten (Schema, Indexe, Data Dictionary)
- **Transaktionsverwaltung** für Synchronisation und Datensicherheit

22

1.2 Architekturen von DBMS

Client-/Server- Architekturen

- Die hierarchische Schichtung der Systemkomponenten bestimmt die Aufrufstruktur, nicht aber die Prozessstruktur (Zuordnung zu physischen Recheneinheiten)
- Folgende Client/Server-Modelle sind gebräuchlich:



1.2 Architekturen von DBMS

Mehrbenutzer- und Verteilte DBS

- Verteilte Mehrbenutzer-DBS ($m:n$) vereinigen die folgenden beiden Prinzipien:

