
Datenbanksysteme II

Sommersemester 2008

Organisatorisches

Skript zur Vorlesung: Datenbanksysteme II
Sommersemester 2008, LMU München

© 2008 Dr. Peer Kröger

Dieses Skript basiert zu einem Teil auf dem Skript zur Vorlesung Datenbanksysteme II von Prof. Dr. Christian Böhm gehalten im Sommersemester 2007 an der LMU München

Vorlesungsteam



Dr. Peer Kröger
Oettingenstr. 67, Zimmer E 1.08
Tel. 089/2180-9327
Sprechstunde: Do, 15⁰⁰-16⁰⁰



Dr. Elke Achart
Oettingenstr. 67, Zimmer E 1.07
Tel. 089/2180-9326



Andreas Züfle
Oettingenstr. 67, Zimmer E 1.04
Tel. 089/2180-9321

Termine

- Vorlesung: Dienstag, 8:30-11 Uhr Raum E004 (Hauptgebäude)
- Übung: Montag, 14-16 Uhr Raum 1.27 (Oettingenstr.)
 Montag, 16-18 Uhr Raum 1.27 (Oettingenstr.)
- Mittwoch, 14-16 Uhr Raum E216 (Hauptgebäude)
- Mittwoch, 16-18 Uhr Raum A119 (Hauptgebäude)

Anmeldung für den Übungsbetrieb auf der Homepage

<http://www.dbs.informatik.uni-muenchen.de/Lehre/DBSII/>

Achtung!

- Nächste Woche entfällt die Vorlesung (am 22.04.2008)
- Die Übungen beginnen in der 4. Woche (ab dem 05.05.2008)

Infos

- Vorlesungshomepage
<http://www.dbs.informatik.uni-muenchen.de/Lehre/DBSII/>
- Die Informatiker
<http://www.die-informatiker.net/forum/DBS>

Scheinerwerb

- Zulassung: Anmeldung für den Übungsbetrieb (siehe oben)
- Scheinprüfung: voraussichtlich Klausur

Inhalt der Vorlesung (Planung)

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

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).



Kapitel 1

Einführung

Skript zur Vorlesung: Datenbanksysteme II
Sommersemester 2008, LMU München

© 2008 Dr. Peer Kröger

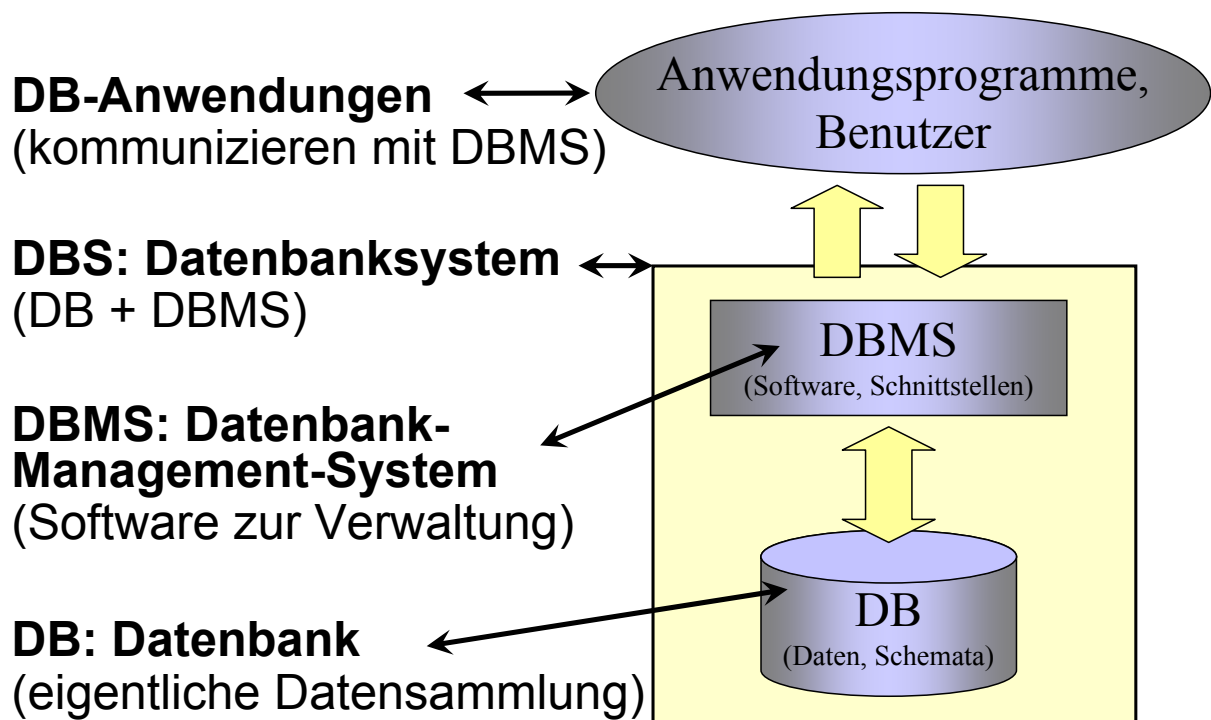
Dieses Skript basiert zu einem Teil auf dem Skript zur Vorlesung Datenbanksysteme II von Prof. Dr. Christian Böhm gehalten im Sommersemester 2007 an der LMU München

Übersicht

1.1 Grundbegriffe

1.2 Architekturen von DBMS

Ein Datenbanksystems (DBS) besteht aus...



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

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

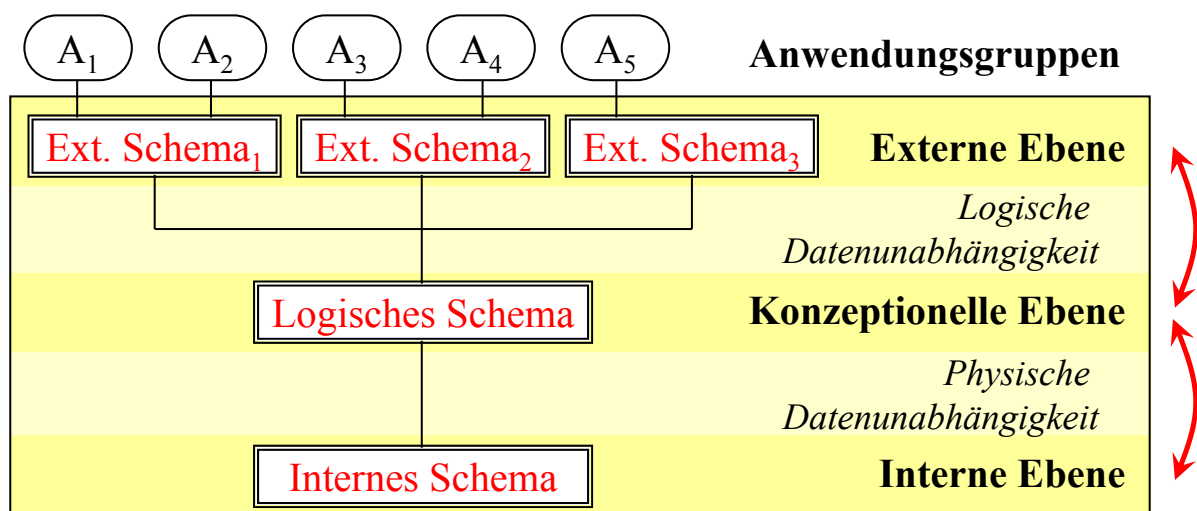
9. Datensicherung

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)

Drei-Ebenen-Architektur zur Realisierung von

- physischer
- und logischer

Datenunabhängigkeit (nach ANSI/SPARC)



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*)

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)

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*)

Übersicht

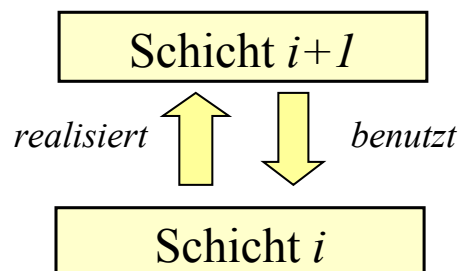
1.1 Grundbegriffe

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

- 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



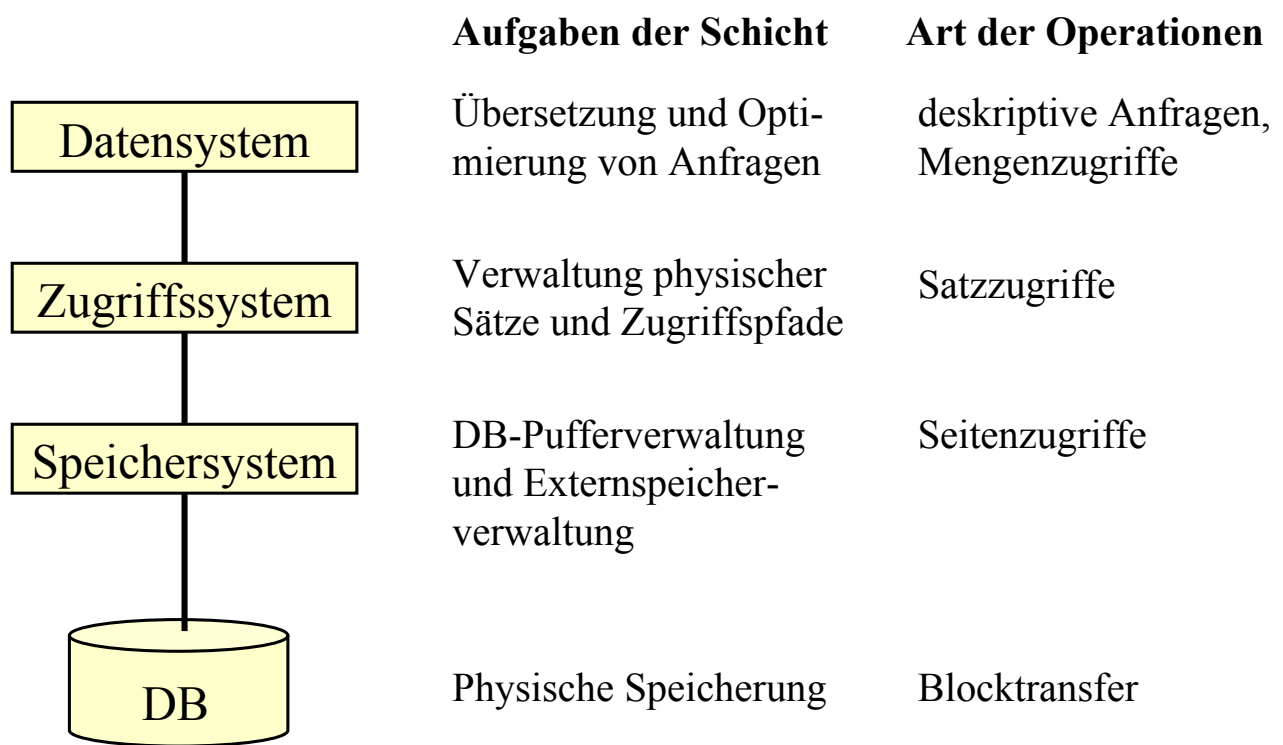
• Vorteile einer Schichtenarchitektur

- 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.

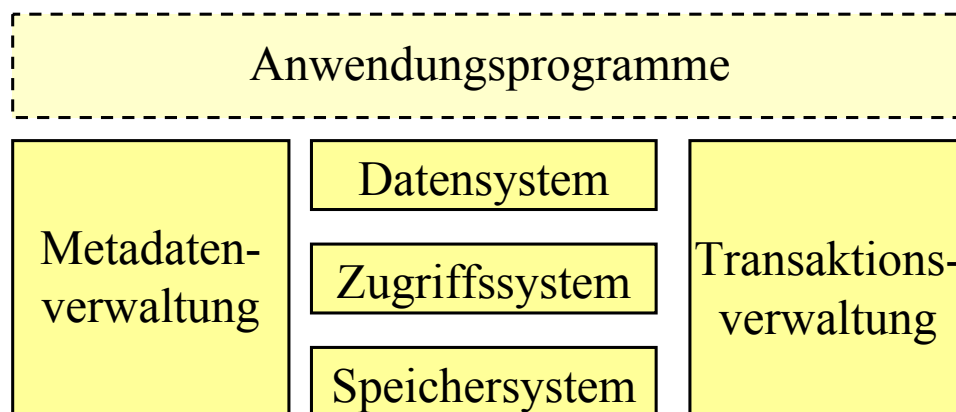
• Optimale Anzahl n von Schichten

- 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

Schichten des DBMS-Kerns



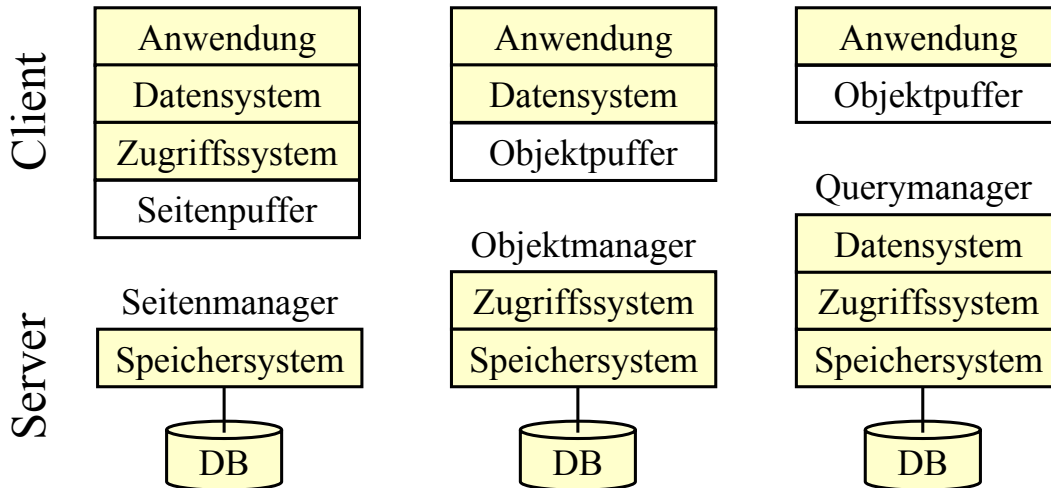
Gesamtarchitektur eines DBMS



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

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:



Mehrbenutzer- und Verteilte DBS

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

