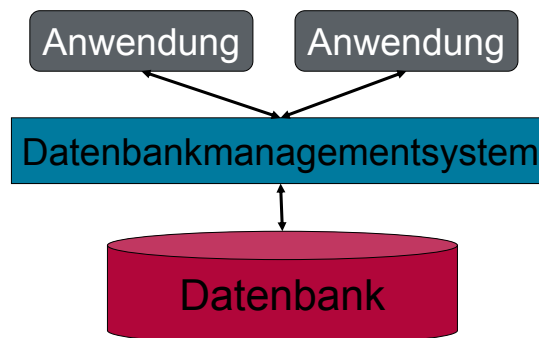


Datenbanksysteme 1
Organisatorisches und Einführung

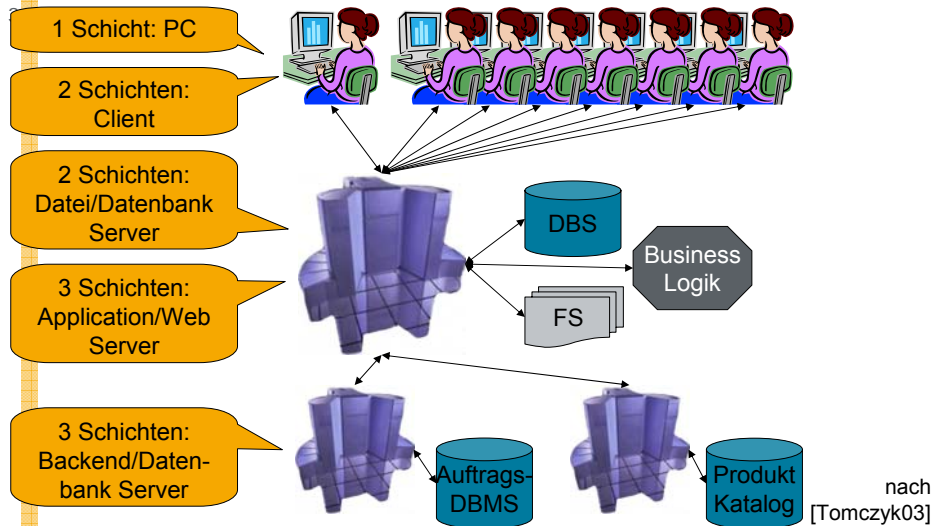
16.4.2007
Felix Naumann

Datenbanken

2

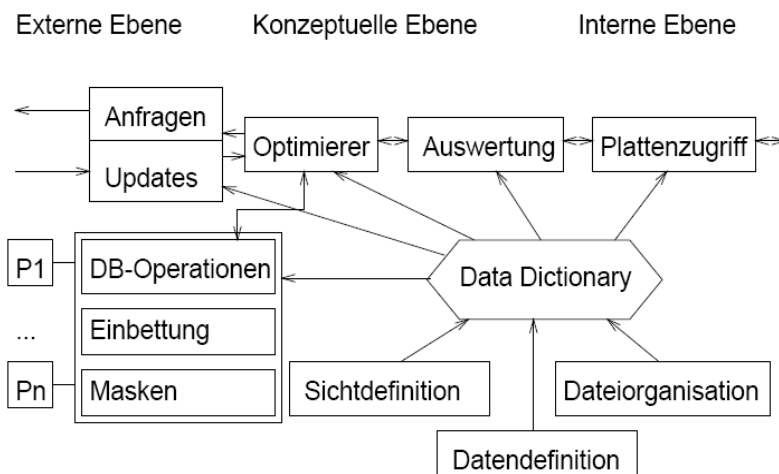


Mehr-Schichtenarchitekturen



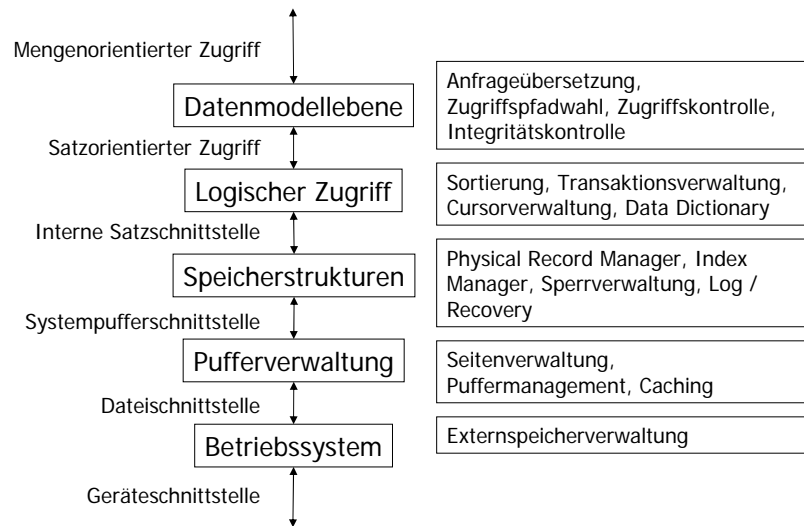
Zoom in das DBMS

4



5-Schichten Architektur

5

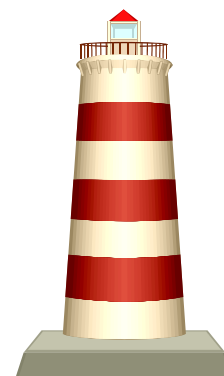


Felix Naumann | VL Datenbanksysteme II | SS 07

Überblick

6

- ➔ ■ Vorstellung der Arbeitsgruppe
- Organisatorisches
- Implementierung von Datenbanken
- Ausblick auf das Semester



Felix Naumann | VL Datenbanksysteme II | SS 07

7

Felix Naumann

- naumann@hpi.uni-potsdam.de
- Raum A-1.13 (über Frau Pamperin)
- HU, IBM Almaden, DFG/JP/HU, HPI

Universität Potsdam, Hasso-Plattner-Institut

- Fachgebiet „Informationssysteme“

Forschungsthemen:

- Datenqualität
- Informationsintegration
- Peer Data Management
- <http://www.hpi.uni-potsdam.de/~naumann/>

Felix Naumann | VL Datenbanksysteme II | SS 07

8

Leitung

- Prof. Felix Naumann
- Patricia Hobro (Sekretariat)

Wissenschaftliche Mitarbeiter

- Jens Bleiholder (Datenfusion, HumMer, FuSem)
- Armin Roth (PDMS, System P)
- Melanie Weis (Duplikaterkennung, DogmatiX, XClean)
- Jana Bauckmann (Data Profiling, Aladin)
- Alexander Albrecht (PIM)
- Frank Kaufer (Matching, Forschungskolleg)

Tutoren

- Christoph Böhm, Karsten Draba, Dustin Lange, Matthias Weidlich

Felix Naumann | VL Datenbanksysteme II | SS 07

Lehrveranstaltungen

9

Vorlesungen

- DBS I
- DBS II
- Informationsintegration
- ...

Seminare

- Bachelor: Beauty is our business
- Master: Datenreinigung
- Forschungsseminar
- ...



Extending the Database Relational Model to Capture More Meaning

E. F. COOD
IBM Research Laboratory

During the last three or four years several investigations have been exploring "semantic models" for relational databases. The intent is to represent in a form or form formalized form of the meaning of the data in that database design via known formal semantics and the database system itself can before more intelligently "know" their own data.

(1) the search for meaningful data that are useful to produce "semantic" relations.

(2) the search for meaningful data that are useful to produce "semantic" relations.

In the past we have attempted to the relational model to represent certain cases and address modeling via the introduction of new rules for domains, tables, and relations as well as new algebraic systems.

Key words and phrases: relation, relational database, relational model, relational algebra, database, data model, database system, data semantics, semantic model, knowledge representation, knowledge DB Concepts: 3.15, 3.15.4.2, 4.2, 4.2.1, 4.2.1.1, 4.2.1.2

1. INTRODUCTION

The relational model for hierarchical databases [1] was conceived ten years ago, primarily as a tool to free users from the limitations of having to deal with the complexity of change implementation. This implementation, independent of the specific hardware, was the power of the algebraic operators on a set of relations and the open relations have stimulated research in database management (see [2]).

The relational model has also provided an architectural design for the design of database and some general purpose database management systems such as IBM/DB2 [3], ORACLE [4], INGRES [5], MACRO [6], and others [7].

During the last few years intensive investigations have been aimed at capturing

Permission to copy without fee all or part of this material is granted provided that the copies are not made for advertising or promotional purposes, for creating new collective works, or for resale. Copyright © 1978 by International Business Machines Corporation. This document is published by the International Business Machines Corporation. This document is published by the International Business Machines Corporation. This document is published by the International Business Machines Corporation.

A version of this work was presented at the 1978 International Conference on Management of Data (COMAD), Irvine, Calif., May 16-17, 1978.

Author's address: IBM Research Laboratory, 5600 Cedar Road, San Jose, CA 95128.

© 1978 by IBM Research Laboratory. 0018-0782/78/0000-0000\$01.00

IBM Research Laboratory, San Jose, CA 95128

Seminar

10

Beauty is our Business

„Wenn wir uns klarmachen, daß der Kampf gegen Chaos, Durcheinander und unbeherrschte Kompliziertheit eine der größten Herausforderungen der Informatik ist, müssen wir zugestehen: Beauty is our Business.“ Edsger W. Dijkstra, 1978

Termine und Themenvergabe

Termin	Thema
19.4.2007	Einführung
26.4.2007	Wissenschaftliche Texte Lesen
10.5.2007	Literaturkritik / Diskussion
17.5.2007	Vortragstechniken
???.2007	Vortrag 1: Mariposa (pdf)
	Vortrag 2: Fagins Algorithmus (pdf)
???.2007	Vortrag 3: Enough Already in SQL (pdf)
	Vortrag 4: Sorted Neighborhood (pdf)
???.2007	Vortrag 5: Source Capabilities (pdf)
	Vortrag 6: Data Mining (pdf)
5.7.2007	Einführung in LaTeX
12.7.2007	Vorstellung der Gliederungen & Tipps zur
23.2.2007	Abgabe der Ausarbeitungen

Vorstellung – Hörer

11

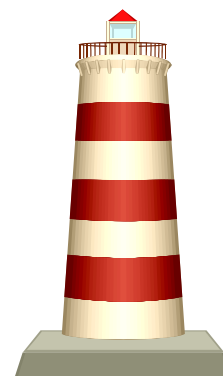
- Welches Semester?
- HPI oder IfI?
- Erasmus o.ä.?
- DBS I bei mir?

- Warum DBS II?
- Alternative: Information Systems

Überblick

12

- Vorstellung der Arbeitsgruppe
- ➔ ■ Organisatorisches
- Implementierung von Datenbanken
- Ausblick auf das Semester



Termine

13

Vorlesung

- Montags 9:15 - 10:45
 - in HS 1
- Mittwochs 17:00 - 18:30
 - in HS 2

Erste Vorlesung

- 16.4.2007

Letzte Vorlesung

- 18.7.2007

Feiertage

- 28.5. Pfingstmontag

Übungen

- Ungefähr 1 SWS
- Verteilt auf Doppelstunden im Semester
- Erste Übung
 - 2.5.2007 (Mittwoch)
- Danach: Zwei Übungsgruppen
 - Montags 13:30 in HS 1
 - Mittwochs 17:00 in HS 2

Klausur

- Voraussichtlich 25.7.
 - Eine Woche nach Semester

Feedback

14

Folien

- Vor der VL im WWW
 - ASAP
- Dated wegen updates

Anregungen zur Verbesserung:

- Gebrauch der Folien
- Infos im WWW
- ...

Fragen bitte jederzeit!

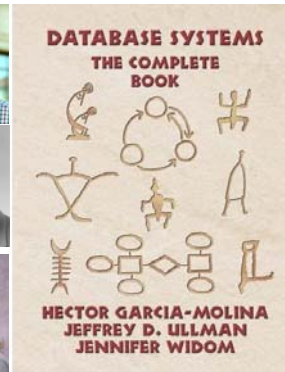
- In der Vorlesung
- Sprechstunde
 - Dienstags 15:00 – 16:00
 - Raum A-1.13
 - Am liebsten mit Anmeldung
- Email: naumann@hpi.uni-potsdam.de

Literatur

15

Database Systems - The Complete Book

- Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom: Pearson Education International, 2002.



Ebenfalls empfehlenswert

- Datenbanksysteme
Alfons Kemper, André Eickler
ISBN: 3486273922
- Datenbanken:
Implementierungstechniken
Gunter Saake, Andreas Heuer, Kai-Uwe Sattler

Und viele andere mehr...

Felix Naumann | VL Datenbanksysteme II | SS 07

Übungen und Leistungserfassung

16

Übungen – Organisatorisches

- Mehr von Jana Bauckmann bei erster Übung
- Allein oder in Zweiergruppen
- Mind. 50% der Punkte auf jedem Blatt
 - Ein Freischuss
- Tutor
 - Dustin Lange

Übungen – Inhaltliches

- „Theoretische“ Übungen
 - Auf Papier
- Praktische Übungen
 - DB2 im Studentenpool
 - Ergebnisse auf Papier

Klausur

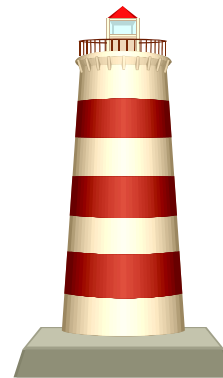
- 90 Minuten

Felix Naumann | VL Datenbanksysteme II | SS 07

Überblick

17

- Vorstellung der Arbeitsgruppe
- Organisatorisches
- Implementierung von Datenbanken
- Ausblick auf das Semester



Das Megatron 2002 DBMS

18

- Naive Sicht
 - DBMS sollte recht einfach sein
- Megatron 2002
 - Unter UNIX verfügbar
 - Relationales Datenmodell
 - SQL Anfragen



19

- Zuerst: NDA unterzeichnen!
- Relationen werden in ASCII Dateien gespeichert
 - Studenten(Name, Matrikel, Studiengang)
 - /usr/db/Studenten
- Eine Zeile pro Tupel
- Felder werden mit „#“ separiert

```
Smith # 123 # CS
Jones # 522 # EE
...
```

20

- Schema wird in gesonderter ASCII Datei gespeichert
 - /usr/db/schema
- Eine Zeile pro Relation
- Zeile beginnt mit Relationenname
- Gefolgt von Attributname/Datentyp Paaren
 - Getrennt durch „#“

```
R1 # A # INT # B # STR ...
R2 # C # STR # A # INT ...
...
```

Eine Session mit Megatron

21

- Maschine: dbhost
- Megatron prompt: &

- dbhost> MEGATRON3000
Welcome to MEGATRON 3000!
- & ...
...
- & quit
- dbhost>

Eine Session mit Megatron

22

- SQL Anfragen werden mit # beendet

- & **SELECT * FROM Studenten #**

<u>Name</u>	<u>Matrikel</u>	<u>Fach</u>
SMITH	123	Informatik
Jones	522	Mathematik
- Speicherung von Anfrageergebnissen in einer Datei
- & **SELECT * FROM Studenten WHERE Matrikel > 500 | HoheMtr #**
- Erzeugt neue Datei /usr/db/HoheMtr mit einer Zeile

- `SELECT * FROM R WHERE <Bedingung>`
 1. Schema-Datei lesen um Attribute und Datentypen von R zu bestimmen
 2. Prüfen, ob Bedingung semantisch korrekt ist.
 3. Zeige jedes Attribut als Zeilenkopf an
 4. Male eine Linie
 5. Lesen der Datei namens R. Für jede Zeile
 1. Prüfe Bedingung
 2. Zeige Zeile als Tupel an, falls Bedingung erfüllt ist.

- `SELECT * FROM R WHERE <Bedingung> | T #`
 1. Schema-Datei lesen um Attribute und Datentypen von R zu bestimmen
 2. Prüfen, ob Bedingung semantisch korrekt ist.
 3. Lesen der Datei namens R. Für jede Zeile
 1. Prüfe Bedingung
 2. Schreibe Zeile in Datei /usr/db/T, falls Bedingung erfüllt ist.
 4. Füge in Datei /usr/db/schema eine Zeile hinzu für Relation T

- & SELECT Büro FROM Studenten, Fachbereiche
WHERE Studenten.Name = „Smith“
AND Studenten.Fach = Fachbereich.Name #
 1. Schema-Datei lesen um Attribute und Datentypen von Studenten und Fachbereiche zu bestimmen.
 2. Lesen von Datei /usr/db/Studenten. Für jede Zeile:
 1. Lesen von Datei /usr/db/Fachbereiche. Für jede Zeile:
 1. Erzeuge Join-Tupel
 2. Prüfe Bedingungen
 3. Zeige Tupel an, falls Bedingung erfüllt

- Tupel-Layout auf der Festplatte
 - Falls sich ein Wert ändert muss gesamte Datei neu geschrieben werden.
 - Alle Zeichen bewegen sich entsprechend in der Datei.
 - ASCII Speicherung ist teuer
 - Löschung ist ebenfalls teuer
- Teure Suche
 - Es kann nicht schnell, z.B. basierend auf einem Schlüssel, ein bestimmtes Tupel gefunden werden
 - Es wird immer gesamte Relation gelesen, selbst bei strengen Nebenbedingungen

Probleme mit Megatron 2002

27

- Anfrageausführung ist brute force
 - Joinoperationen können wesentlich schneller implementiert werden.
 - Selektionen sollten früher ausgeführt werden
- Kein Puffermanagement
 - Hauptspeicher wird nicht genutzt
 - Sämtliche Daten werden stets von Festplatte gelesen
- Keine Nebenläufigkeitskontrolle
 - Mehrere Nutzer könnten gleichzeitig Datei ändern
- Keine Zuverlässigkeit
 - Daten können bei Systemabsturz verloren gehen
 - Operationen könnten nur teilweise ausgeführt werden

Felix Naumann | VL Datenbanksysteme II | SS 07

Probleme mit Megatron 2002

28

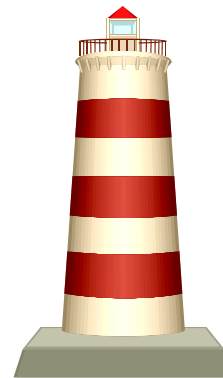
- Keine Sicherheit
 - Dateisystem ist unsicher
 - Granularität ist grob
- Kein API
 - Wie können anderen Anwendungen auf die Daten zugreifen?
- Keine Interaktion mit anderen DBMS
- Wenig Metadatenfähigkeiten
 - Statistiken, Nebenbedingungen
- Keine GUI

Felix Naumann | VL Datenbanksysteme II | SS 07

Überblick

29

- Vorstellung der Arbeitsgruppe
- Organisatorisches
- Implementierung von Datenbanken
- Ausblick auf das Semester



Kurzüberblick

30

- | | |
|--|--|
| ■ Physische Speicherstrukturen (2) | ■ Übung: Speicherung |
| ■ Physische Repräsentation von Daten (1) | |
| ■ Indexstrukturen (5) | ■ Übung Indexstrukturen |
| ■ Anfrageausführung (2) | ■ Übung: Anfrageausführung |
| ■ Optimierung (3) | ■ Übung: Optimierung |
| ■ Benchmarking (1) | ■ Übung: Benchmarking |
| ■ Recovery (2) | |
| ■ Transaktionsmanagement (3) | ■ Übung: Recovery und Transaktionsmanagement |

Physische Speicherstrukturen

31

Effizienter Umgang mit großen Datenmengen

- Teil 1: Umgang mit großen Datenmengen
- Teil 2: Effiziente Manipulation der Daten

- Speicherhierarchie
 - Cache, RAM, Sekundärspeicher, Tertiärspeicher
- Festplatten
 - Mechanik, Controller, Charakteristiken, Read/Write
- Algorithmen für Sekundärspeicher
 - Sortierung mit Two-Phase, Multiway Merge-Sort
- Zugriffsbeschleunigung
 - Zylinder, Multiple Disks, Spiegelung, Prefetching
- Diskfehler
 - Checksums, Stable Storage
- Disk Recovery (RAID)

Felix Naumann | VL Datenbanksysteme II | SS 07

Physische Repräsentation von Daten

32

- Datenelemente und Felder
 - Relationale Datenbankelemente, Datentypen, Datensätze
- Adressierung
 - Client/Server, Physische vs. Logische Adresse, Pointer Swizzling
- Datentypen variabler Länge
 - Variable Felder, Variable Datensätze, Nullwerte, BLOBs
- Schreiboperationen
 - Insert, Update, Delete

Felix Naumann | VL Datenbanksysteme II | SS 07

Indexstrukturen

33

- Indizes auf sequentiellen Dateien
 - Sortiert, Dicht-besetzter und dünn-besetzter Index, Mehrstufiger Index, Updates
- Sekundärindizes
 - Aufbau, Anwendungen, Indirektion, Invertierte Indizes
- B-Bäume
 - Aufbau, Anwendungen, Bereichsanfragen, Insert, Delete
- Hashverfahren (für Sekundärspeicher)
 - Insert, Delete, erweiterbares Hashen, Lineares hashen
- Eventuell: Multidimensionale Indizes
- Eventuell: Bitmap Indizes

Felix Naumann | VL Datenbanksysteme II | SS 07

Anfrageausführung

34

- Physische Operatoren
 - Scan, Sortierung, I/O Kosten, Iteratoren
- One-Pass Operationen
 - Unär: Selektion, Projektion, Duplikateliminierung, Gruppierung
 - Binär: Mengeoperationen (\cup , \cap , \rightarrow), Join
 - Nested-Loop Join
- Two-Pass Operationen (mittels Sortierung)
 - Duplikateliminierung, Gruppierung, Mengeoperationen (\cup , \cap , \rightarrow), Join
- Two-Pass Operationen (mittels Hashing)
 - ... Hash-Join, ...
- Index-basierte Algorithmen
 - Clustering Indizes, Selektion, Joins
- Puffermanagement
 - Architektur, Strategien (LRU, FIFO,...)
- Parallele Algorithmen

Felix Naumann | VL Datenbanksysteme II | SS 07

Optimierung

35

- Teils schon in DBS I
 - Parsing, Algebraische Transformationsregeln
- Kostenmodell
 - Projektion, Selektion, Verteilungen, Join
- Kostenbasierte Optimierung
 - Histogramme, Statistiken, Heuristiken, Enumerierung aller Pläne
- Joinreihenfolge
 - Links vs. rechts, Dynamische Programmierung, greedy Algorithmus
- Physische Anfragepläne
 - Selektionsmethode, Join-Methode, Pipelining

Felix Naumann | VL Datenbanksysteme II | SS 07

Benchmarking

36

- Leistungsbewertung
 - Komponenten eines Benchmarks, Ziele
- TPC Benchmarks
 - TPC A, B
 - TPC C: OLTP
 - TPC D/H: Decision Support DWH
 - TPC W: e-Commerce

Felix Naumann | VL Datenbanksysteme II | SS 07

Recovery

37

- Fehlerarten
 - Medienfehler, Systemfehler, Transaktionen en detail
- Undo Logging
 - Log-Datensätze, Log-Regeln, Wiederherstellung mit Undo, Checkpoints
- Redo Logging
 - Log_regeln, Wiederherstellung mit Redo, Checkpoints
- Undo/Redo Logging
 - Best of both worlds, Regeln, Wiederherstellung, Checkpoints
- Archivierung
 - Archive, Nebenläufige Archivierung, Wiederherstellung mit Archiven und Logs

Felix Naumann | VL Datenbanksysteme II | SS 07

Transaktionsmanagement

38

- Fortsetzung aus DBS I
 - Transaktionen, Schedules, 2PL, Sperren
- Mehrere Sperr-Modi
 - Shared und exclusive Sperren, Aufwertung von Sperren, Update-Sperren, Inkrement-Sperren
- Scheduler
 - Architektur, Sperrtabelle
- Hierarchisches Sperren
 - Sperrgranulate, warnende Sperren
- Baumprotokolle
 - Motivation, Regeln, Beweis
- Nichtsperrende Verfahren
 - Zeitmarken (Timestamps), Validierung, Optimistische Verfahren

Felix Naumann | VL Datenbanksysteme II | SS 07

Weitere Themen

39

Bei Bedarf

- Verteilte Datenbanken
- Parallele Datenbanken
- Föderierte Datenbanken

Feedback

40

Fragen bitte jederzeit!

- In der VL
- Sprechstunde: Dienstags 15-16 Uhr
- Email: naumann@hpi.uni-potsdam.de
- Telefon: (0331) 5509 280

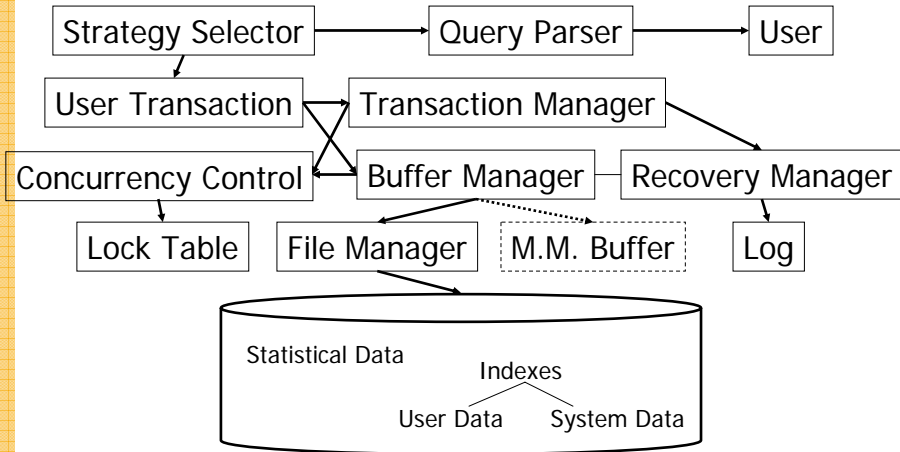
Anregungen zur Verbesserung:

- Z.B. zu
 - Gebrauch der Folien
 - Infos im WWW

The end.

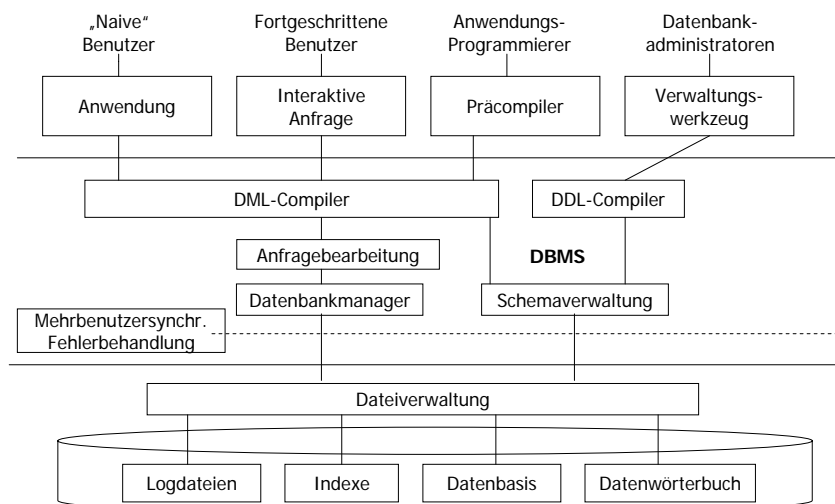
Systemarchitektur nach Garcia-Molina

41



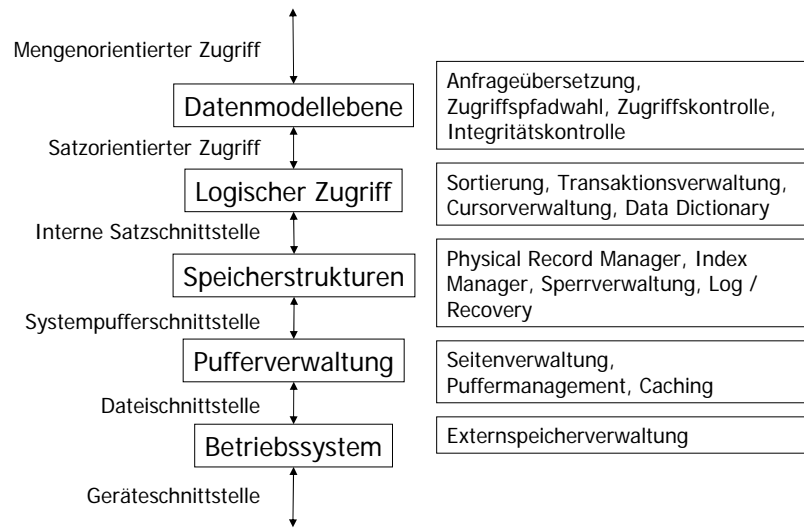
Systemarchitektur nach Leser

42



5-Schichten Architektur

43



Felix Naumann | VL Datenbanksysteme II | SS 07