



Informationsmanagement

Sommersemester 2019

Kapitel 2: **Datenbank- architekturen**



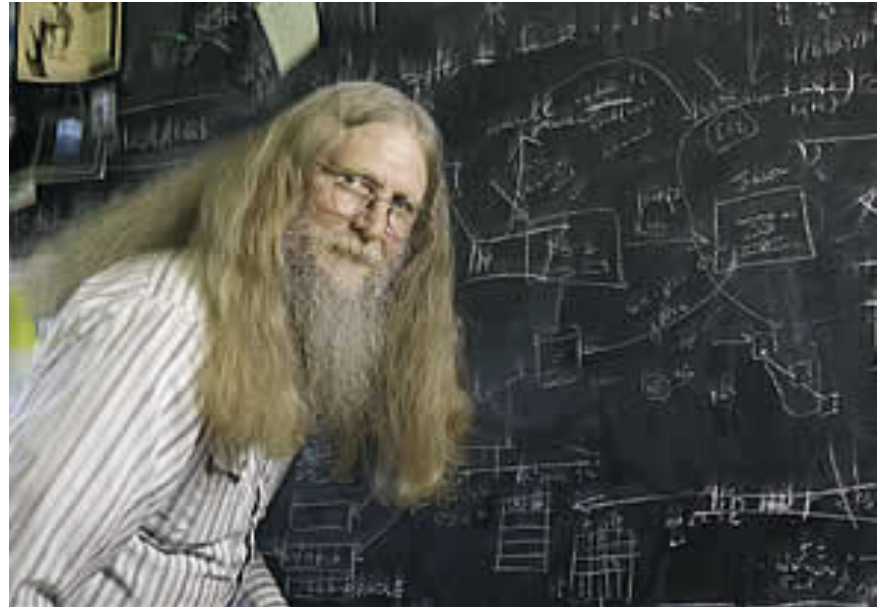
Carsten Binnig

- Since 08/2017 at TU Darmstadt (Leiter Data Management Lab)
- Research Focus: Scalable Data Management & Machine Learning
- PhD, University of Heidelberg (Databases Systems, Prof. Kossmann)
- PostDoc, ETH Zurich (Systems Group)
- Software Architect, SAP HANA (In-Memory Database System)
- Other Professor Positions
 - 2011-2014: Mannheim, Germany
 - 2014-2017: Brown University, USA

Teil 1: Relationale Datenbanken

Relationale Datenbanken spielen bei der
Informationsverarbeitung eine **zentrale Rolle**

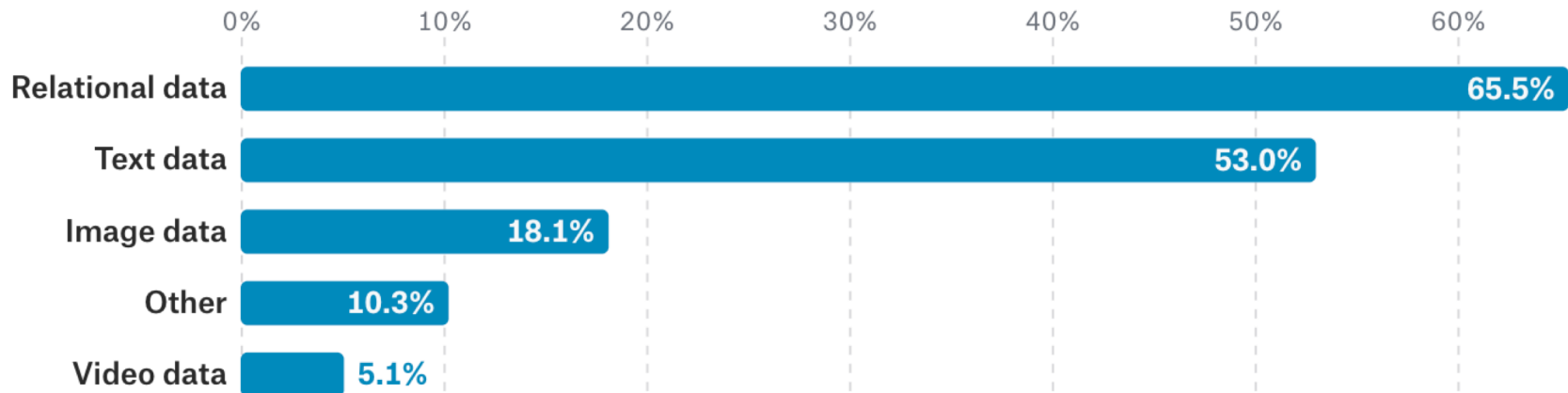
„Relational Databases are
the foundation of western
civilization“



*Bruce Lindsay, IBM Fellow
IBM Almaden Research Center*

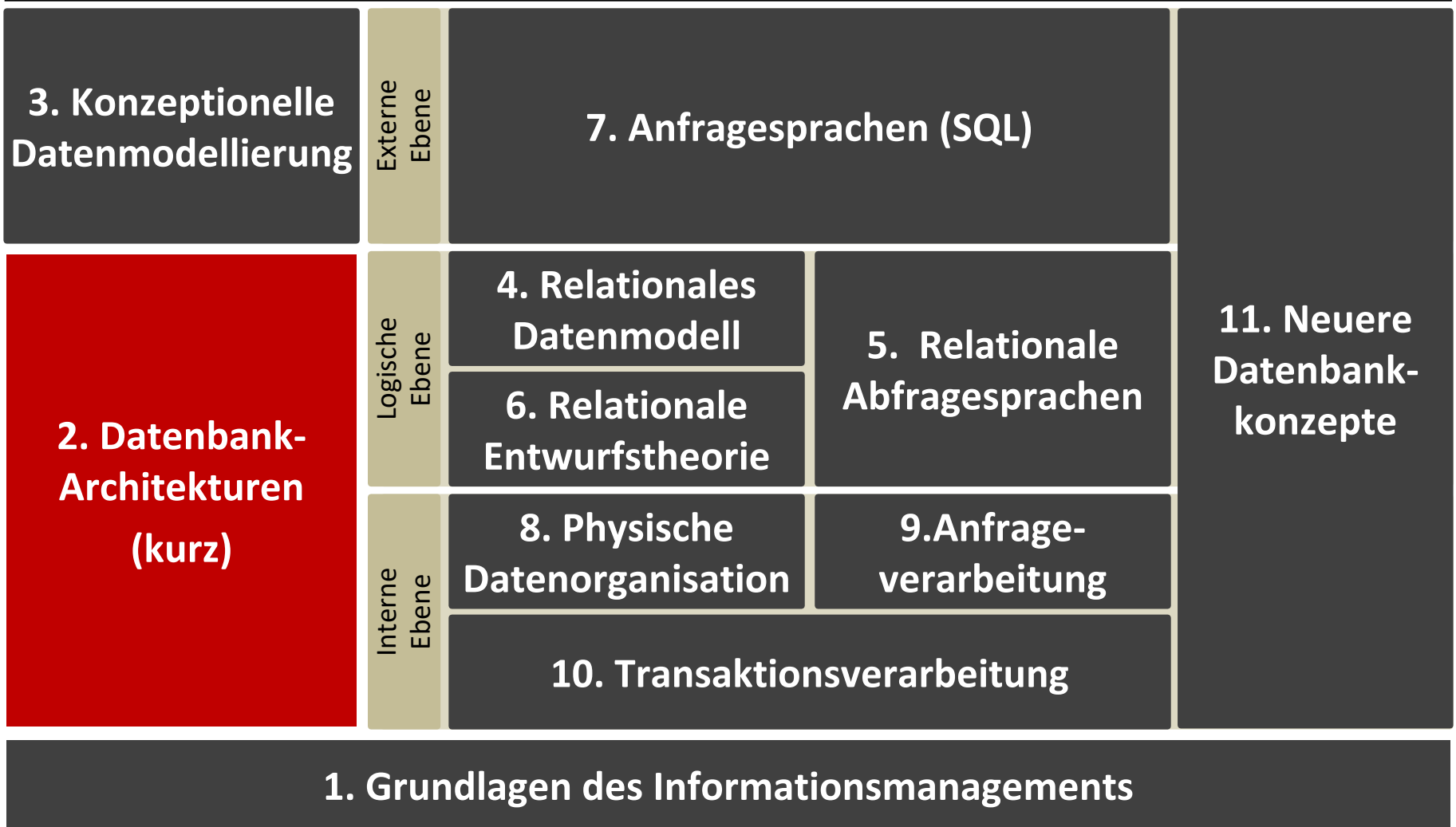
Teil 1: Relationale Datenbanken

Ein **Großteil der Daten** wird heute in relationalen Datenbanken gespeichert und verarbeitet



Source: The State of Data Science & Machine Learning 2017, Kaggle, October 2017

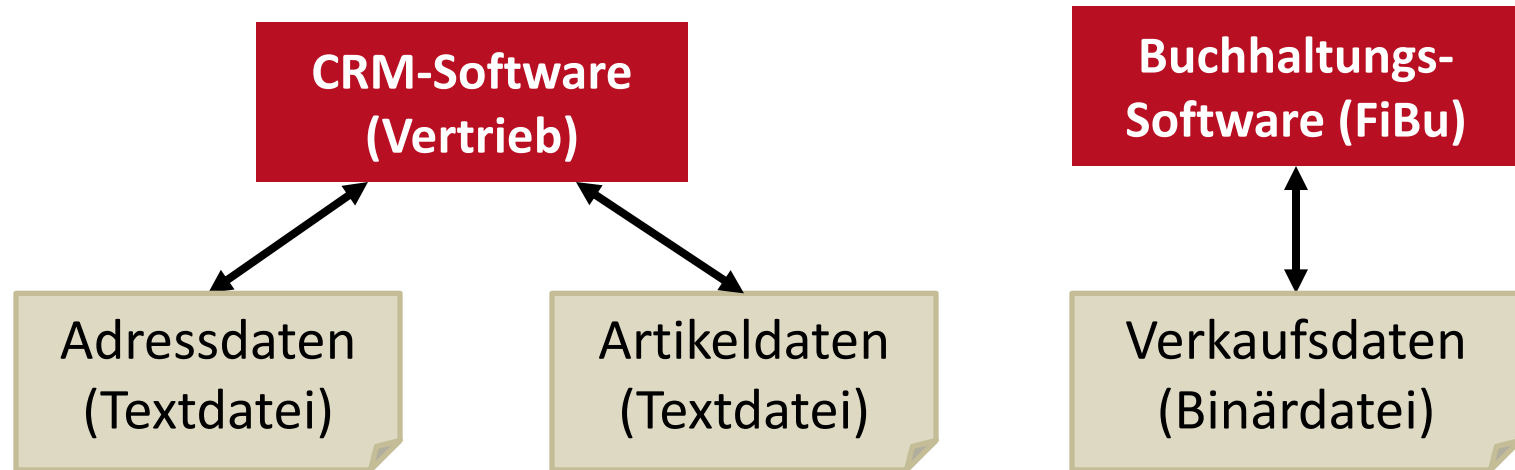
Themenüberblick (Teil 1)



Datenspeicherung vor Datenbanken

Zu Beginn der Informationsverarbeitung wurden **Daten in Dateien** abgelegt, die anwendungsabhängig aufgebaut waren

Beispiel: Vertriebs-Software verwendet eigenes Textformat (z.B. CSV) um Daten abzuspeichern; FiBu verwendet binäres Format

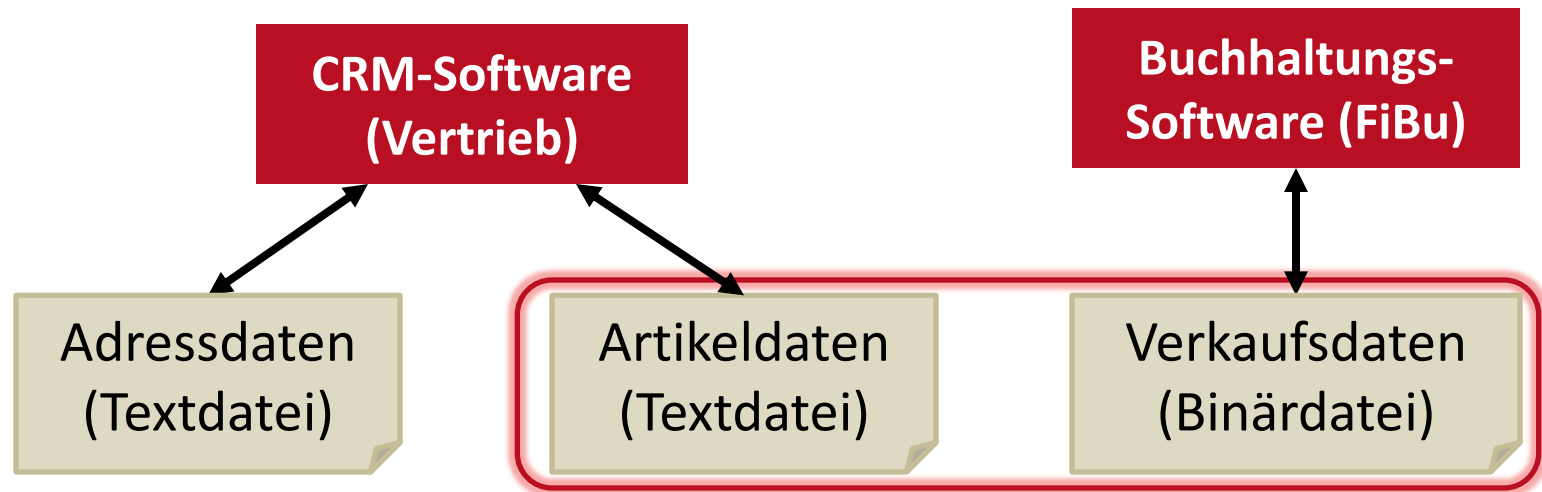


Problem: Datenredundanz

Jede Anwendung speichert **Daten redundant in eigenen Formaten**

- Verschwendung von **Speicherplatz**
- Problem bei lokalen **Veränderungen**

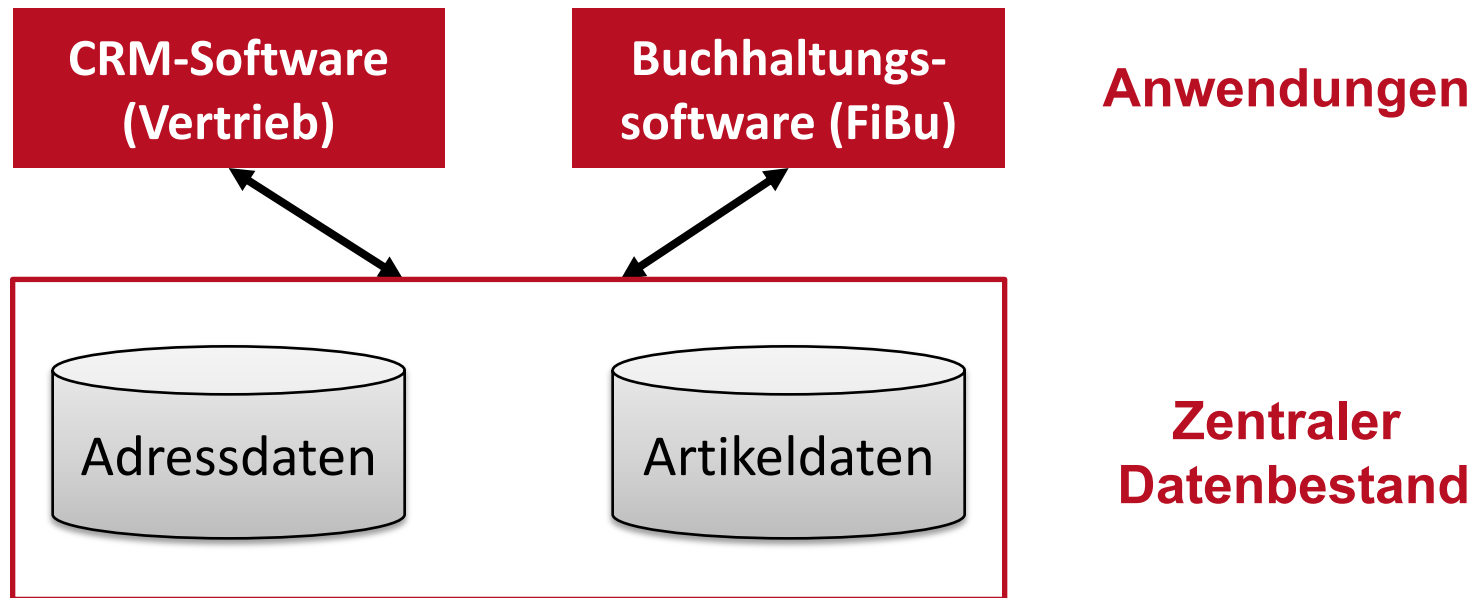
Beispiel: Artikelpreise werden für Vertrieb und FiBu benötigt und daher redundant (in beiden Anwendungen) gespeichert



- **Ineffizienter Zugriff** auf Daten (z.B. Suche in Datei)
 - Jede Anwendung muss selbst optimierte Zugriffsstrukturen schaffen!
- **Mehrbenutzerbetrieb** bei Dateien ist problematisch
 - Was passiert wenn FiBu und Vertrieb gleichzeitig Artikeldaten ändern?
- Fehlende **Zugriffskontrolle**
 - Wer darf die Umsatzzahlen aller Kunden einsehen?
 - Beispiel: Nur Vertriebsleiter oder alle Vertriebsmitarbeiter?
- Beachtung der **Integrität („Korrektheit“) der Daten** (z.B. Bestellung kann nur angelegt werden, wenn Kunde existiert)
 - Jede Anwendung muss „Korrektheit“ der Daten gewährleisten
- **Verlust von Daten** durch fehlende Backups
 - Konsistente Sicherung mehrerer Dateien problematisch in laufendem Betrieb bei gleichzeitiger Änderung der Dateien

Idee: Datenbanken

Anwendungen arbeiten auf einem **zentralen Datenbestand**



Der zentrale Bestand zusammengehöriger Daten wird als **Datenbank** bezeichnet

- Daten werden nur einmal zentral (für alle Anwendungen) gespeichert
→ **löst Problem der Datenredundanz**
- Abfragesprachen und Optimierung
→ **effizienter Zugriff**
- Transaktionen für Mehrbenutzerbetrieb
→ **gemeinsamer und paralleler Zugriff**
- Integritätsbedingungen
→ Garantieren **Korrektheit der Daten**
- Datensicherheit (z.B. Backups)
→ **Schutz vor ungewolltem Verlust**

Begriffe: Datenbanken und DBMS

Datenbank (DB)

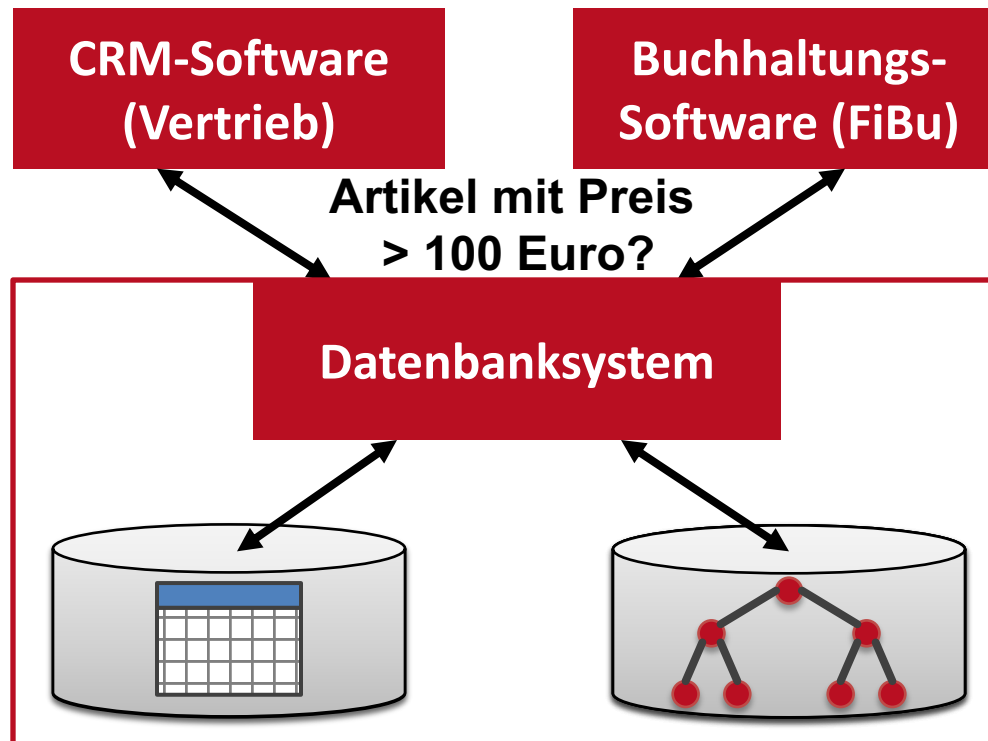
- **Darstellung eines „Weltausschnittes“** (z.B. Adressdaten, Artikeldaten, Vorlesungsdaten)

Datenbankmanagementsystem (DBMS)

- **Systemsoftware zur Verwaltung von Datenbanken**
- Daten in einer DB können nur über das DBMS **eingefügt, gelesen, geändert** oder **gelöscht** werden
- **Ein DBMS kann viele Datenbanken (DBs) verwalten**

DBMS: Datenunabhängigkeit

Anwendung ist von physischer Repräsentation der Datenbank **entkoppelt**, d.h. gleiche Schnittstelle (= logische Struktur) – egal ob Daten als Tabelle oder als Indexstruktur abgelegt sind



DBS leistet
Indirektion

- **Hierarchische DBMSs (1960s/70s):** IMS (IBM)
- **Netzwerk DBMSs (1970s):** CODASYL (Cobol), UDS (Siemens)
- **Relationales DBMSs (1980s-heute):**
 - DB2, System R, SQL/DS (IBM)
 - ORACLE
 - SQLServer (Microsoft)
 - Ingres, PostgreSQL
 - MySQL, MariaDB
 - Sybase
 - ...
 - Neuere Trends: Hauptspeicher, Cloud, ...
- **Objektorientiertes DBMSs (1990s):** z.B. Poet -> Nischendasein
- **NoSQL DBMSs (2010s):** CouchDB, MongoDB, Hadoop, ...

Schemaarchitektur

- Wie kann die Datenunabhängigkeit durch in DBMS erreicht werden?

Systemarchitektur

- Aus welchen Komponenten besteht ein DBMS?

Anwendungsarchitektur

- Wie können Anwendungen mit dem DBMS arbeiten?

Schemaarchitektur

- Wie kann die Datenunabhängigkeit durch ein DBMS erreicht werden?

Systemarchitektur

- Aus welchen Komponenten besteht ein DBMS?

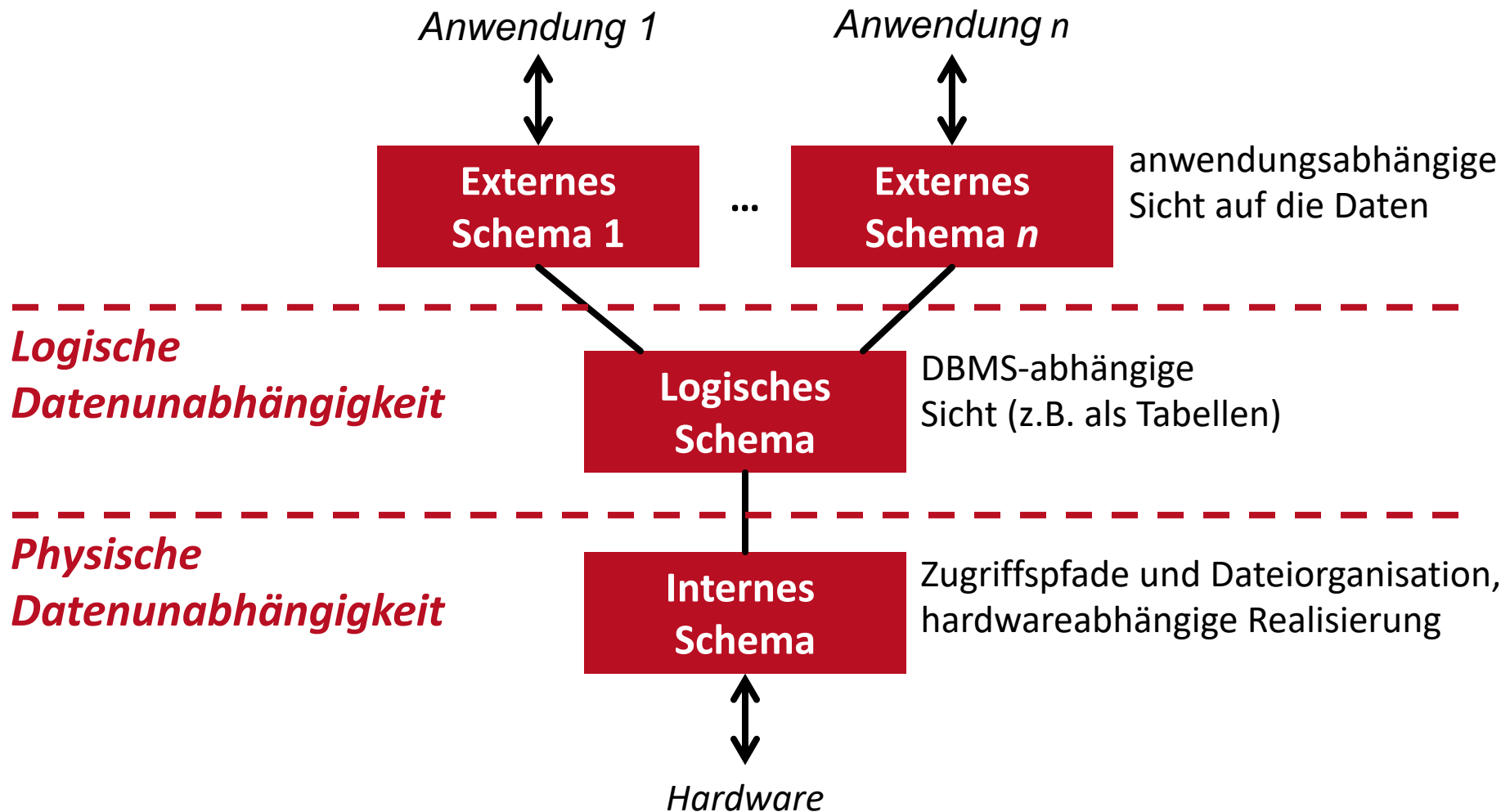
Anwendungsarchitektur

- Wie können Anwendungen und Anwendungen mit dem DBMS arbeiten?

Drei-Ebenen-Schemaarchitektur



TECHNISCHE
UNIVERSITÄT
DARMSTADT



(Jardine, 1977)

Schemaarchitektur: Datenunabhängigkeit

Physische Datenunabhängigkeit

- Änderungen an den **Speicherstrukturen** und **Zugriffspfaden (Indexe)** sind für Anwendungen unsichtbar

Logische Datenunabhängigkeit

- Änderungen an der **logischen Schicht (z.B. Änderungen einer Spalte in einer Tabelle)** sind für Anwenderprogramme und Anfragen unsichtbar
- **Jede Anwendung kann eigene Sicht** auf gemeinsame DB erhalten:
 - FiBu-Software: *Artikelnummer, Verkaufspreis, Einkaufspreis*
 - CRM-Software: *Artikelname, Beschreibung, Artikelpreis*
 - Logisch gespeichert: *ANR, ANAME3, BESCH, VK, EK, LIEFERNT*

Schemaarchitektur

- Wie kann die Datenunabhängigkeit durch ein DBMS erreicht werden?

Systemarchitektur

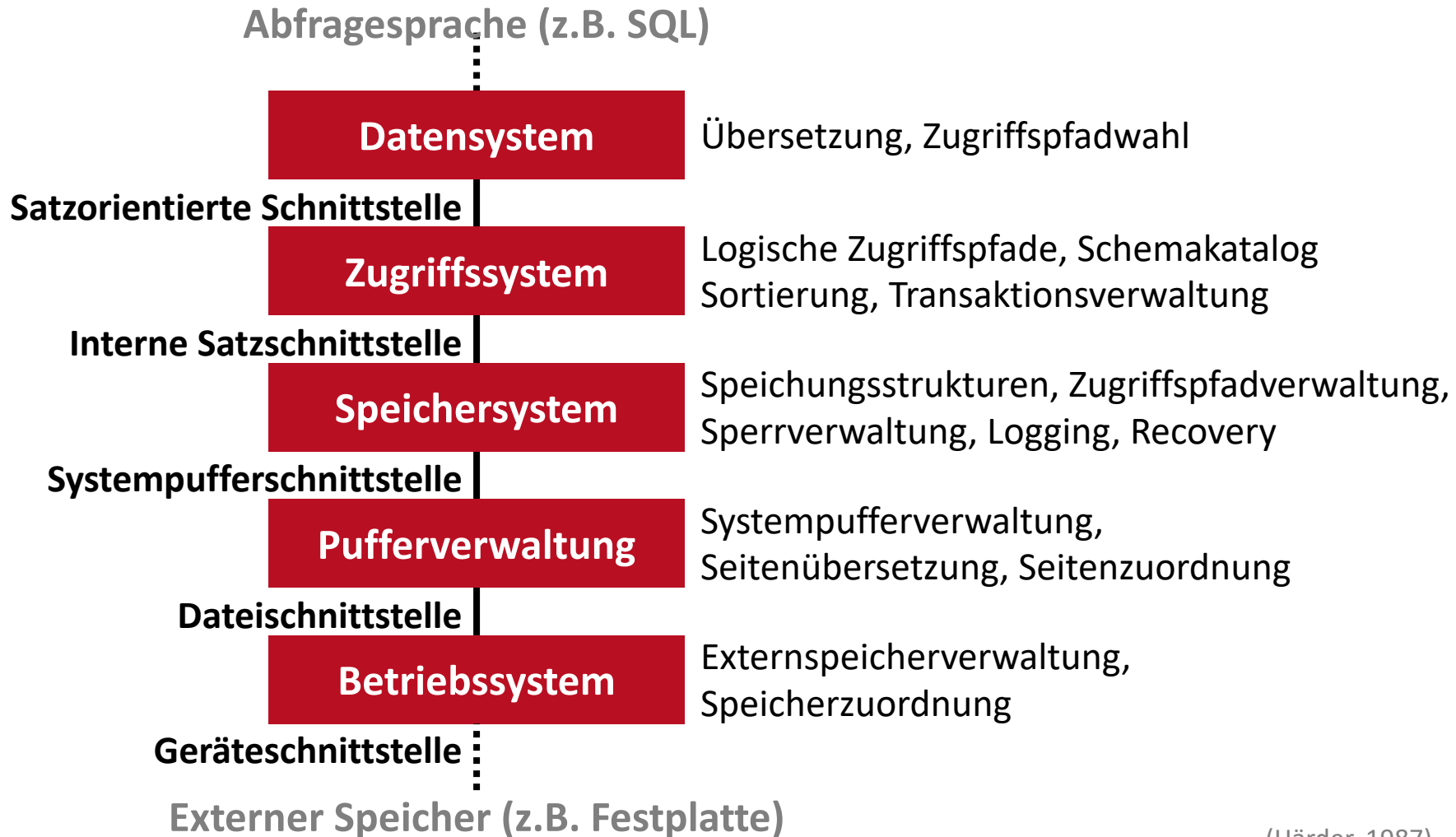
- Aus welchen Komponenten besteht ein DBMS?

Anwendungsarchitektur

- Wie können Anwendungen und Anwendungen mit dem DBMS arbeiten?

Fünf-Schichten-Systemarchitektur (nach Härder)

wird nicht
abgefragt!

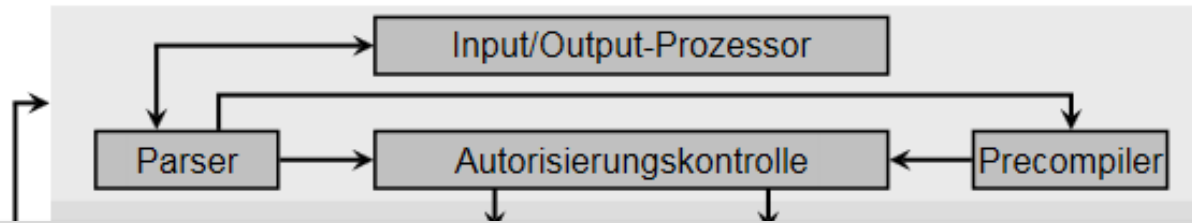


(Härder, 1987)

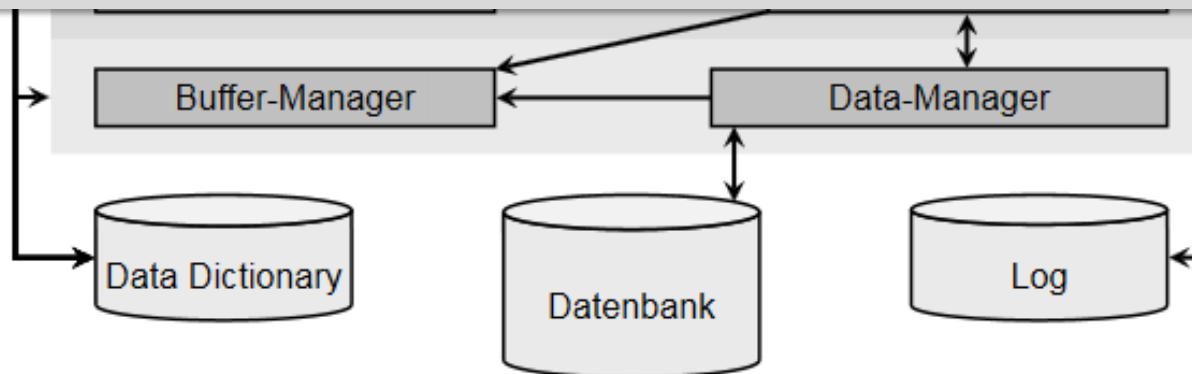
Komponenten eines DBMS

(nach Vossen)

wird nicht
abgefragt!



Weiterführende Vorlesungen an der TUDA:
Scalable Data Management Systems (Datenbanken II)
Advanced Data Management Systems (Datenbanken III)



(Vossen, 2008)

Schemaarchitektur

- Wie kann die Datenunabhängigkeit durch ein DBMS erreicht werden?

Systemarchitektur

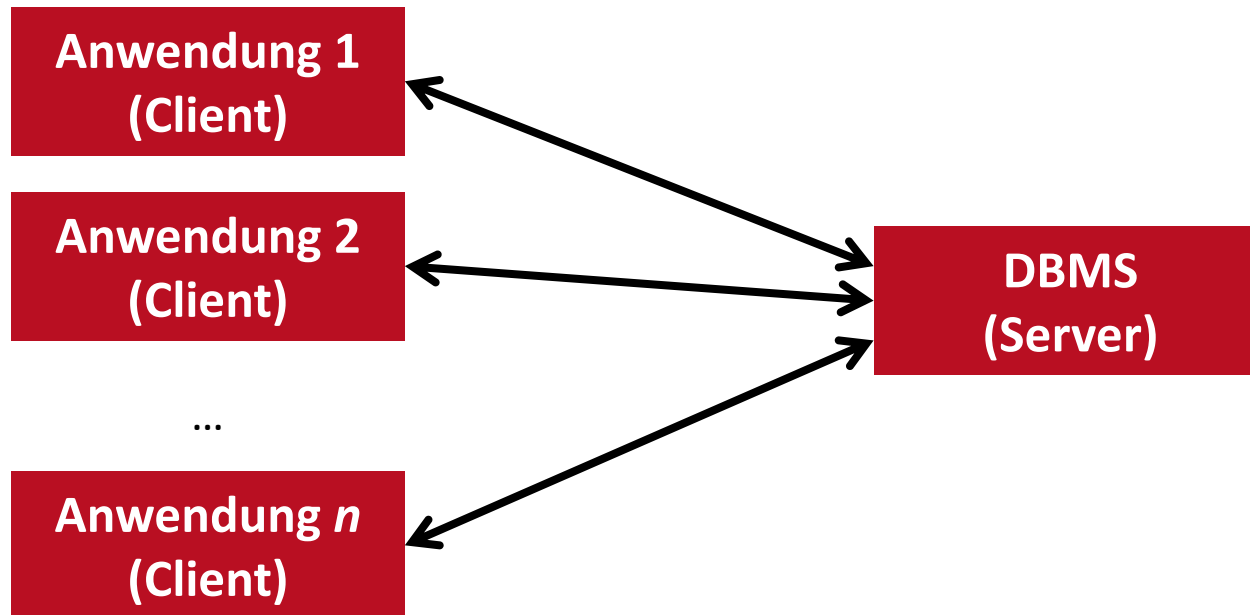
- Aus welchen Komponenten besteht ein DBMS?

Anwendungsarchitektur

- Wie können Anwendungen und Anwendungen mit dem DBMS arbeiten?

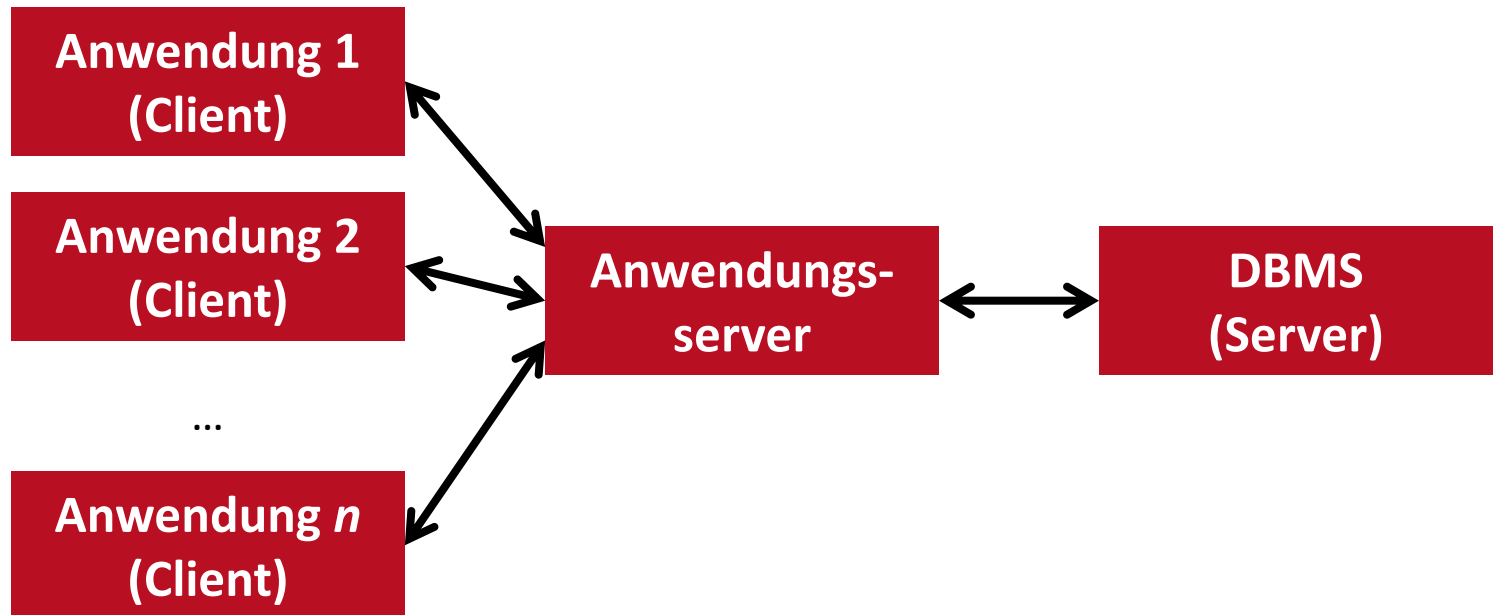
Zwei-Schichten-Anwendungsarchitektur

wird nicht
abgefragt!



Drei-Schichten-Anwendungsarchitektur

wird nicht
abgefragt!





Fragen?

- **Alfons Kemper: Datenbanksysteme - Eine Einführung, 10. Auflage. De Gruyter Studium, de Gruyter Oldenbourg 2015, ISBN 978-3-11-044375-2**
- Theo Härder: Realisierung von operationalen Schnittstellen. In: *Datenbank-Handbuch*, S. 163–335, Berlin: Springer, 1987.
- Gunter Saake, Kai-Uwe Sattler, Andreas Heuer: *Datenbanken: Konzepte und Sprachen*. 4. Auflage, Heidelberg: mitp, 2010.
- Gottfried Vossen: *Datenmodelle, Datenbanksprachen und Datenbankmanagementsysteme*. 5. Auflage. München: Oldenbourg, 2008.