

Client-Server-Architekturen

# Schema

JDBC

ORDBMS ER-Modell Aktive Datenbanken

# Relational

ER Daten SQL DML

NoSQL Recovery Big Data

OODBMS DDL EER-Modell

Transaktion Datenbank Normalisierung

Architektur

# Modell

Relationale Algebra

## Informationssysteme 1: Grundlagen von Datenbanken **Grundbegriffe und Überblick**

Wintersemester 2019/2020

---

**Marco Grawunder**

Department für Informatik  
Abteilung Informationssysteme

- **DB-Konzepte und –Architektur**

- **Modellierung von Datenbanken**

- Das ER- und EER-Modell

- **Wichtige Grundlagen**

- Das relationale Modell
- Vom ER-Modell zum rel. Modell
- Relationale Algebra und Kalküle

- **Abfrage und Administration**

- SQL (DDL und DML)

- **Guter Entwurf**

- Normalisierung + Normalformen

- **Datenbanken im Mehrbenutzerbetrieb**

- Transaktionsverarbeitung
- Recovery

- **Weitere Themen**

- Aktive Datenbanken
- Objektorientierte und objektrelationale Datenbanken

- **Blick über den Tellerrand**

- Weiterführende Konzepte
- Big Data und NoSQL

# Beispiel: Buchung einer Reise

Kunde



Hotel



Flug



- Was ist eine Datenbank?
  - Anwendungssituationen (für Datenbanken)
  - Begriffe und Definitionen
    - DB, DBS, DBMS, ACID, ...
  - Eigenschaften von Datenbanken
- Datenabstraktion und Datenunabhängigkeit
  - Datenmodelle
  - Schema ▫ Instanz ▫ DB-Zustand
  - Drei-Schichten-Architektur

- DB-Sprachen
- DB-Systemumgebung

- **Digitale Repräsentation** von

- Dingen
- Entitäten
- Wissen
- Information

in/aus der wirklichen Welt

Es muss nicht immer ein physisches  
Gegenstück geben

- **Kernfragen:**

- Welche Daten speichere ich?
- Wie speichere ich die Daten?
- Wie frage ich Daten ab?
- Wie geht dies effizient und sicher?

- **Lösung:**

- **Datenbanksystem**

- **Datenbanken** (DBen) spielen heute eine wichtige Rolle...
- **„DB-Klassiker“** für
  - Personal-/Kundendatenverwaltung
  - Lagerhaltung/Warenwirtschaft
  - Buchhaltung/Rechnungswesen
  - Verwaltung des Buchbestandes einer Bibliothek...
- Einfache Objekte
  - Zeichenketten, Zahlen
- **„spezielle“ DB-Anwendungen**
  - Multimedia-Informationssysteme
  - Geografische Informationssysteme (GIS)
  - Data Warehouses (Data Mining, Online Analytical Processing (OLAP))...
- Komplexe Objekte
  - Bilder, Video- und Audio-Daten, Polygonzüge, Datenwürfel

# Konkrete Beispiele: Google News

Basketball: Ergebnisse und Tab. X Google News X +

https://news.google.com/?hl=de&gl=DE&ceid=DE:de

Suchen

Google News

Nach Themen, Orten und Quellen suchen

Top-Meldungen

- Für Sie
- Favoriten
- Gespeicherte Suchanfragen

Deutschland

Welt

Lokales

Wirtschaft

Wissenschaft & Technik

Unterhaltung

Sport

Gesundheit

Sprache und Region  
Deutsch | Deutschland

Einstellungen

Android-App herunterladen

iOS-App herunterladen

Feedback geben

Hilfe

Datenschutz · Nutzungsbedingungen

### Schlagzeilen

Mehr von Schlagzeilen

**Live-Ticker: Polizei verrät weitere Details zur Geiselnahme am Kölner Hauptbahnhof**  
Derwesten.de · vor 21 Minuten

- Geiselnahme am Kölner Hauptbahnhof: Täter soll Arabisch gesprochen haben  
Kölner Stadt-Anzeiger · vor einer Stunde
- Geiselnahme in Köln: Täter goss Benzin auf Frau – SEK griff zu!  
BILD · heute
- Kölner Geiselnahme im News-Ticker: Polizei gibt weitere Details bekannt  
FOCUS Online · vor 27 Minuten
- Geiselnahme in Köln: Polizei schließt Terror-Motiv nicht aus  
t-online.de · vor 5 Stunden

Mehr zum Thema

**Stimmenverlust der CSU: Seehofer will sich der Basis stellen**  
tagesschau.de · vor 2 Stunden

- Nach Aussagen zu Kanzlerin Merkel: Jetzt knöpft Seehofer sich Schäuble vor  
t-online.de · heute

Weitere ansehen

**Debakel bei der Bayern-Wahl: "Die SPD sollte alles auf eine Karte setzen"**  
t-online.de · heute

- SPD-Chefin Andrea Nahles: Frust-Auftritt nach Wahl-Klatsche  
BILD · gestern

Weitere ansehen

**Kontroverse über Abstammung: Senatorin fordert eine Million Dollar von Trump**

Oldenburg

Leicht bewölkt  
23°C

| Heute        | Mi           | Do          | Fr          | Sa          |
|--------------|--------------|-------------|-------------|-------------|
|              |              |             |             |             |
| 23°C<br>12°C | 23°C<br>12°C | 16°C<br>4°C | 15°C<br>4°C | 16°C<br>6°C |

C | F | K

Mehr auf weather.com

### Panorama

**Tag der Glatze: Prächtige Platten**  
SPIEGEL ONLINE · vor 2 Tagen

**Hambacher Forst: Schwänzen für den Forst | ZEIT Campus**  
ZEIT ONLINE · vor 2 Tagen

**Nach Kursrutsch an der Börse: Dax schließt fest im Plus**  
manager-magazin.de · heute

**Tel Aviv als veganes Paradies: Wo Sojamilch und Dattelhonig fließen**  
n-tv NACHRICHTEN · vor 2 Tagen

**Ironman Hawaii: "Die Begriffe Schmerz und Qual verwende ich nicht"**  
ZEIT ONLINE · vor 4 Tagen



Sales document Edit Goto Extras Environment System Help

Display Standard Order 117: Overview

Document flow Status overview Sales summary Sold-to party Orders

Sales area 0102 / 10 / 20 Sales grp Sales off.

Sold-to party 100578 HOUGHTON CHEMICAL CORP // ALLSTON MA 02134

Ship-to party 100578002 HOUGHTON CHEMICAL CORP / 30 AMOR AVENUE / CARLSTADT ...

Standard Order 117 Net value 34,463.62 USD

PO number PO date

Sales Item overview Item detail Ordering party Procurement Shipping Reason for rejection

General header data

Sales doc.type OR Standard Order Delivery prior. 2

Contact person ATTN: QUALITY Cont. tel.

Req. deliv.date D 09/24/2002 Total weight 45,311.375 KG

Payment terms L030 NET 30 DAYS INVO... Incoterms FOB Destination

Shp.Cond. 90 CPU or Arranged Title Transfer

All items

| Item   | Material | Description            | Customer material no. | Order quantity | SU  | First date | S | PO details |
|--------|----------|------------------------|-----------------------|----------------|-----|------------|---|------------|
| 10 add | 5017236  | TOLUENE BT BULK ST...  |                       | 2,500.000      | GAL | 09/24/2002 |   | otc 082    |
| 20 add | 5019003  | XYLENE BT BULK STD ... |                       | 4,000.000      | GAL | 09/24/2002 |   | otc 082    |
| 30 add | 5017236  | TOLUENE BT BULK ST...  |                       | 45,000         | LB  | 09/25/2002 |   | otc 082    |



# Konkrete Beispiele: Facebook

The screenshot shows a Facebook news feed for user Ralf Krause. The interface includes a top navigation bar with a search bar, user profile, and navigation links. The left sidebar contains navigation options like News Feed, Messenger, Marketplace, and various categories under 'Entdecken' and 'Erstellen'. The main feed area shows a post by Ralf Krause asking 'Was machst du gerade?' with options to write a post, add photos/videos, or go live. Below this is a post from a green profile picture stating 'hat ihr Titelbild aktualisiert.' with a video thumbnail of a park. The bottom of the feed shows a sponsored post from 1&1 DSL & Mobilfunk. The right sidebar features 'Stories' with a 'Happy Facts' story, a 'Gesponsert' section with a couple's photo and a 'Super-Antenne' advertisement, and a 'Werbeanzeige erstellen' link.

Facebook interface showing a news feed with posts, navigation options, and sponsored content.

**Navigation Bar:** Sucher, Ralf, Startseite, Freunde finden, 92.

**Left Sidebar:**

- News Feed
- Messenger
- Marketplace
- Favoriten
- Pet Rescue Saga
- Entdecken
- Seiten
- Veranstaltungen
- Freundeslisten
- Spendenaktionen
- Erinnerungen
- Mehr anzeigen ...
- Erstellen
- Werbeanzeige · Seite · Gruppe
- Veranstaltung
- Spendenaktion

**Main Feed:**

- Beitrag schreiben | Foto-/Videoalbum | Live-Video
- Was machst du gerade?
- Liste | Foto/Video | Freunde mar...
- hat ihr Titelbild aktualisiert. 18 Std.
- 2
- Gefällt mir | Kommentieren | Teilen
- Kommentieren ...
- 1&1 DSL & Mobilfunk (Gesponsert)
- Jetzt 1&1 All-Net-Flat und das Huawei Mate10 Pro sichern! Neu: 1&1 liefert

**Right Sidebar:**

- Stories
- Archivieren · Einstellungen
- Zu deiner Story hinzufügen
- Teile ein Foto oder Video, oder schreib etwas
- Happy Facts (vor 20 Stunden)
- Gesponsert
- Werbeanzeige erstellen
- Ja, ich bin dabei! lebensfreunde.de Herzklopfen mit 50+
- Super-Antenne empfängt 100 Kanäle völlig ... upnewsy.co
- Das Geheimnis dahinter ist ein Gesetz, von dem kein Kabelunternehmen in der Welt will, das...

# Konkrete Beispiele: Amazon

Suchergebnis auf Amazon.de

https://www.amazon.de/s/ref=nb\_sb\_noss\_2?\_\_mk\_de\_DE=ÅÄŽŮ&url=search-alias%3Daps&field-keywords=Datenbanken&rh=i%3Aaps%2Ck%3ADatenbanken

amazon.de Prime testen

Alle Kategorien

Mein Amazon.de Angebote Gutscheine Verkaufen Hilfe

DE Hallo! Anmelden Mein Konto Testen Sie Prime Meine Listen Einkaufswagen

1-16 von 31.948 Ergebnissen oder Vorschlägen für "Datenbanken"

Sortieren nach Beste Ergebnisse

Ergebnisse anzeigen für

Software

Datenbanken

Bücher

Fachbücher

Datenbanken

Programmierung & Webdesign

Business & Karriere

MySQL

Weitere

Kindle-Shop

Programmierung & Webdesign

Datenbanken

Weitere Programmiersprachen

Business & Karriere

Fachbücher für Informatik

Weitere

Fremdsprachige Bücher

Datenbanken

Datenbankensoftware

Informationssysteme

Weitere

Alle 31 Kategorien

Filtern nach

Versandoption (Was ist das?)

☒ prime

☐ Kostenlose Lieferung ab EUR 29 Bestellwert

Buchformat

☐ Taschenbuch

☐ Kindle eBook

☐ Broschiert

☐ Gebundene Ausgabe

Autor

GESPONSERT VON RHEINWERK VERLAG

Bücher zu Datenbanken (Access, PHP, MySQL, VBA)

Hier klicken

Einstieg in PHP 7 und MySQL 5.6: Fä...

MySQL: Das umfassende Handbuch

Microsoft Access für Einsteiger: Date...

Anzeige - Ihr feedback

Datenbanken & SQL für Einsteiger: Datenbankdesign und MySQL in der Praxis 9. Oktober 2013 von Marco Emrich

Taschenbuch

EUR 24,90 ✓prime

Lieferung morgen, 26. Juli

Kostenlose Lieferung möglich.

Andere Angebote

EUR 17,73 (4 gebrauchte und neue Artikel)

Kindle Edition

EUR 0,00 kindleunlimited

Lesen Sie dieses und über 1 Million Bücher mit Kindle Unlimited.

EUR 9,99 kaufen

Grundlagen von Datenbanksystemen: Bachelorausgabe (Pearson Studium - IT) 1. Januar 2009 von Ramez A. Elmasri und Shamkant B. Navathe

Taschenbuch

EUR 34,95 ✓prime

Kostenlose Lieferung möglich.

Nur noch 15 Stück auf Lager - jetzt bestellen.

Andere Angebote

EUR 19,99 (24 gebrauchte und neue Artikel)

Datenbanksysteme für Dummies 12. Oktober 2016 von Wolfgang Gerken

Taschenbuch

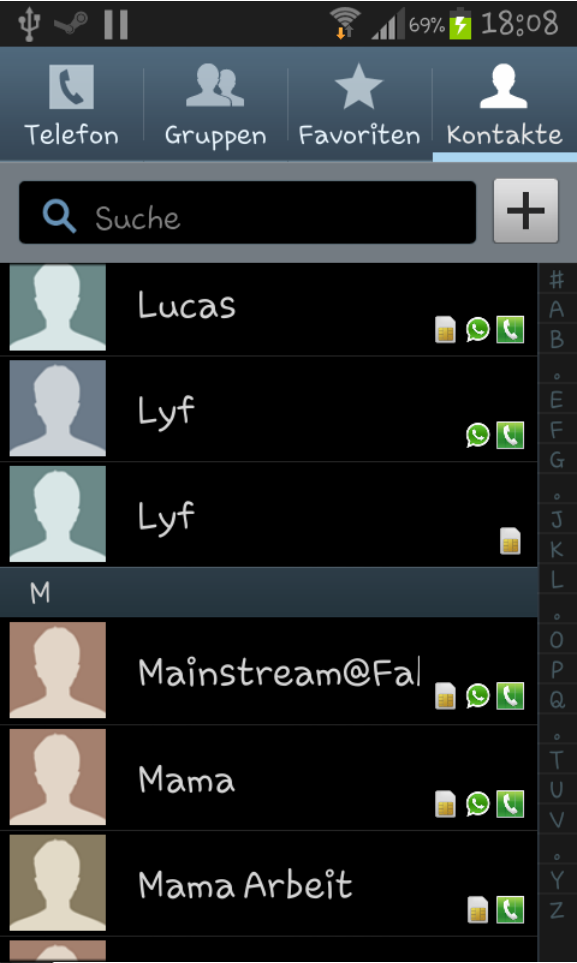
EUR 22,99 ✓prime

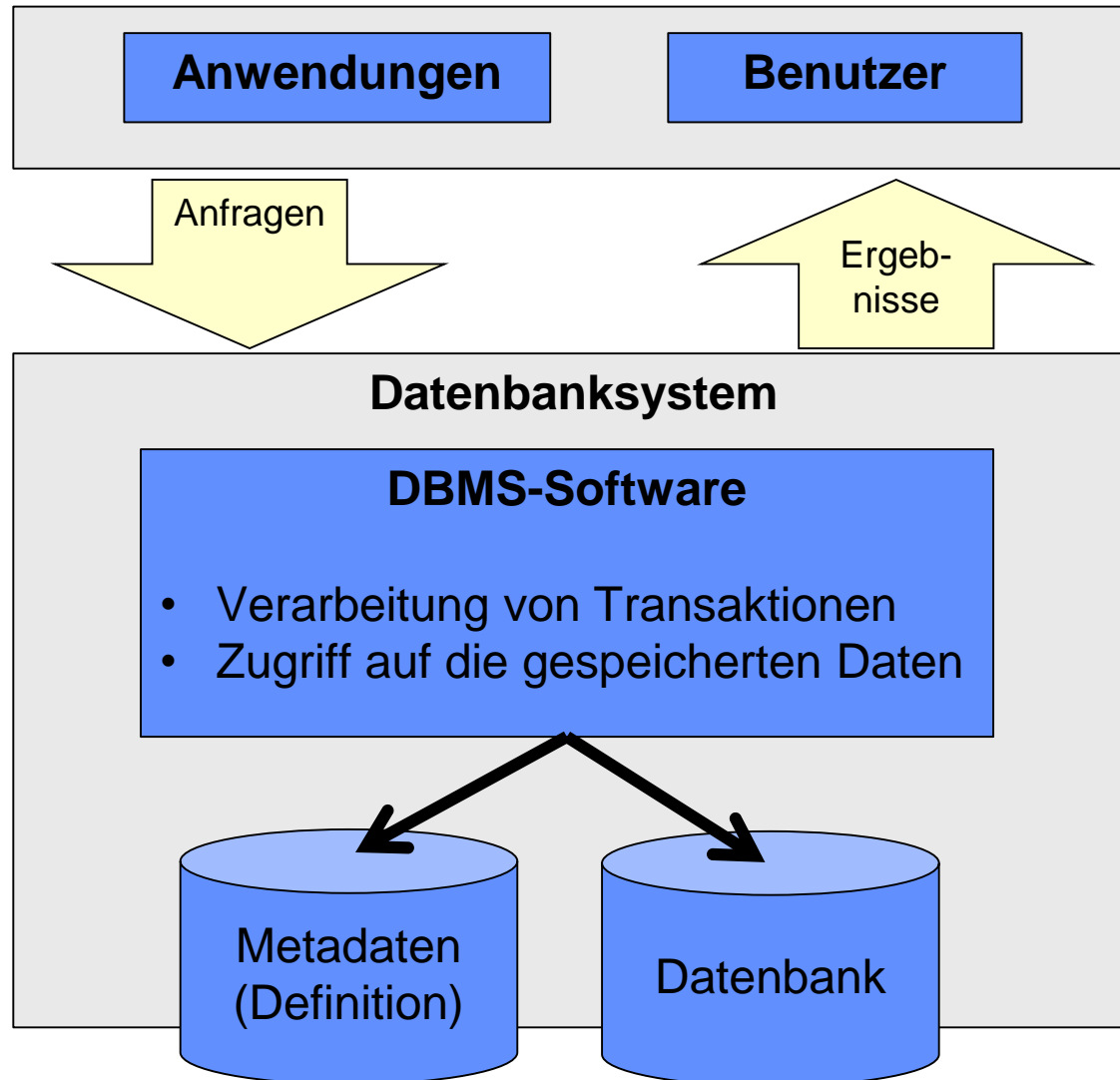
Kostenlose Lieferung möglich.

Nur noch 11 Stück auf Lager - jetzt bestellen.

https://aax-eu.amazon-adsystem.com/x/c/QIKdI3qJT-ZCUEPugq5hlqMAAAfde05UjwMAAAH2AVnw4l8/https://www.amazon.de/HSA/pages/default?pagel=55F6103D-5091-4A44-A9CC-63938F26E9EE

# Auch das sind Datenbankanwendungen...





- **Datenbanksystem (DBS) = Datenbank**
  - Die Daten selbst
  - Metadaten (Beschreibung der Daten)
- **+ Datenbankmanagement-system (DBMS)**
  - Softwarekomponenten zum Zugriff auf eine oder mehrere Datenbanken
  - Server-basiert
- Anwendungen sind kein Bestandteil des DBS

- **Daten**

- Fakten, die beobachtet / aufgezeichnet werden können
- Besitzen implizite Bedeutung („Semantik“)
- Werden erst durch Interpretation zu **Information**

- **Datenbank (DB)**

- Sammlung von Daten
  - Logisch zusammenhängend mit inhärenter Bedeutung
- Charakterisieren einen realen Weltausschnitt (**Miniwelt** oder **Universe of Discourse** (UoD))

## Hinweis

Unsystematische bzw. zufällige Datensammlungen werden i.Allg. nicht als DB bezeichnet!



- Die Struktur einer DB wird i.d.R.
  - Für einen bestimmten Zweck entwickelt (**DB-Definition**)
  - Auf ein geeignetes Speichersystem abgebildet (**DB-Implementierung**)
  - Von Benutzern und Anwendungen verwendet (**DB-Manipulation**)
- DBen Unterliegen i.d.R. einer **Closed World Assumption**
  - Annahme: Die Miniwelt ist in sich abgeschlossen
  - D.h., aus der Abwesenheit von Daten kann geschlossen werden, dass entsprechende Objekte in der Miniwelt nicht existieren

- **Datenbankmanagementsystem (DBMS)**

- Softwaresystem für Entwurf, Implementierung und Betrieb einer DB
- Besteht meist aus mehreren Programmen / Komponenten
- Wickelt Prozesse zur **Definition / Implementierung / Manipulation** von DBen ab

- **Definiton der DB**

- Spezifizierung von Datentypen, Strukturen und Einschränkungen für die Daten (→**Metadaten**)
- Grundlage zur Prüfung von **Konsistenz/Integrität** der DB

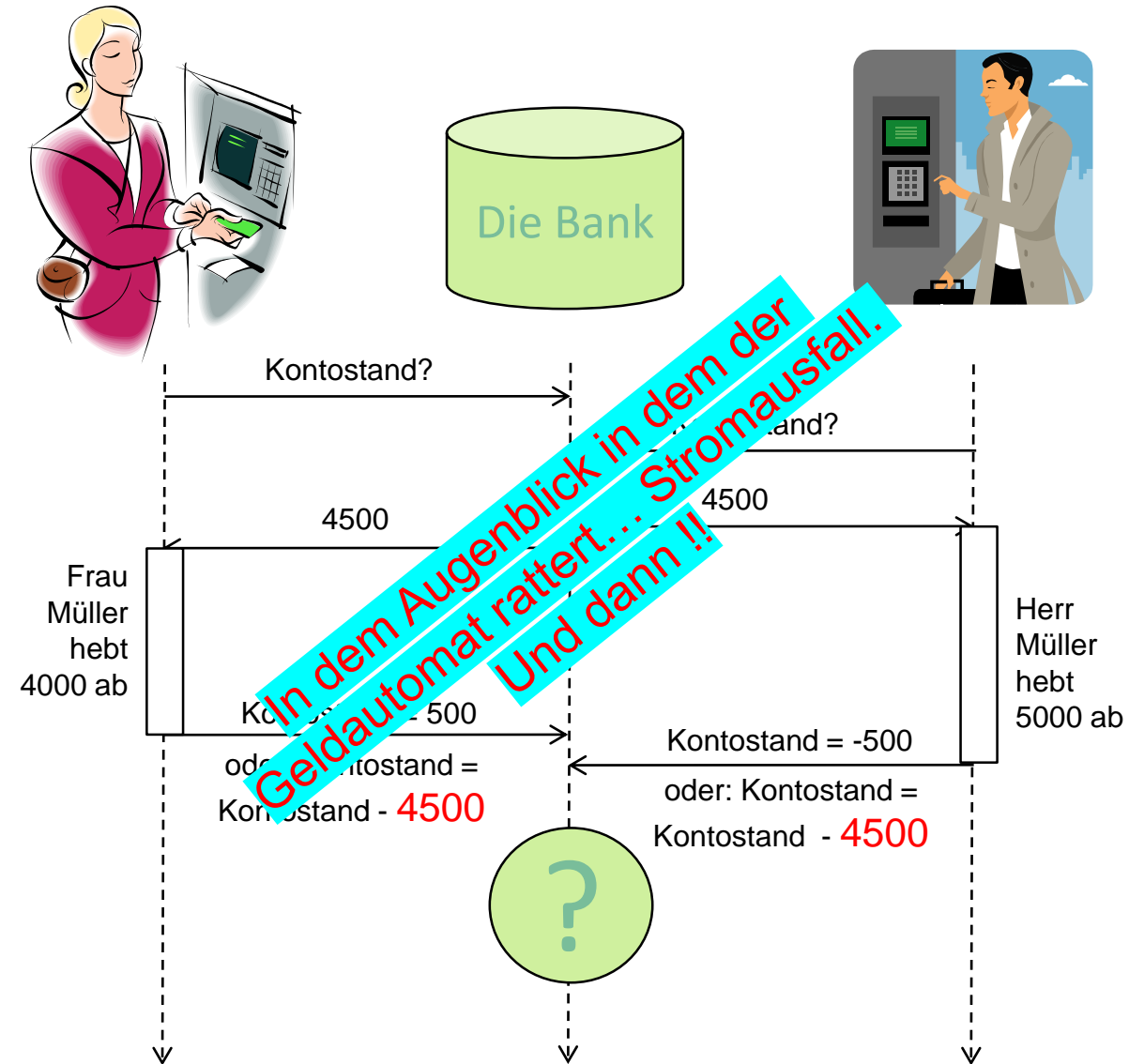
- **Implementierung der DB**

- Abbildung auf geeignetes Speichersystem
  - Durch DBMS kontrolliert
  - Unterstützt dauerhafte (persistente) Speicherung großer Datenbestände



- **Manipulation der DB**

- Einpflegen neuer Daten
- Verarbeitung von Anfragen verschiedener (zeitgleich aktiver) Nutzer und Applikationen
- **Transaktion** als zentrales Konzept
  - Zusammenhängende Abfolge von Datenbankoperationen



- **ACID-Eigenschaften von DB-Transaktionen**

- **Atomcity** (Atomarität)

- Unteilbarkeit einer DB-Transaktion
    - D.h. DB-Transaktion wird ganz oder gar nicht ausgeführt

- **Consistency** (Konsistenz)

- Wenn eine DB vor der Ausführung einer DB-Transaktion in einem konsistenten Zustand befindet, befindet sie sich auch hinterher in einem konsistenten Zustand

- **Isolation** (Isolation)

- DB-Transaktionen beeinflussen sich gegenseitig nicht
    - D.h. DB-Transaktionen werden stets voneinander logisch unabhängig ausgeführt

- **Durability** (Dauerhaftigkeit)

- Resultate einer DB-Transaktion werden persistent gespeichert
    - D.h. Ergebnisse vollständig ausgeführter DB-Transaktionen werden dauerhaft in DB eingefügt

- Weitere Anforderungen an DBMSe in den Bereichen

- **Sicherheit**

- Sicherheitsmechanismen, um unautorisierte Nutzung von DB-Inhalten zu verhindern
- Z.B. Festlegung von Benutzergruppen, Rollen, Benutzungsrechten

- **Fehlerbehandlung**

- Mechanismen zum Erhalt der Konsistenz bei
  - Hardwarefehlern (z.B. Platten-Crash)
  - Softwarefehlern (Applikationsfehler)
- → **Recovery**

- **Integrität**

- Konzepte zur qualitätsgesicherten Datenhaltung
  - Constraints
  - Assertions
  - aktive DB-Mechanismen (z.B. Event-Condition-Action-Regeln)

- **Aufbereitung und Präsentation von Daten**

- Benutzer- oder Applikationsabhängig
- → **Views**

- Integration
  - Einheitliche, nicht-redundante Datenverwaltung
- Operation
  - Definieren, Speichern, Abfragen, Ändern
  - Deklarativ
- Benutzersichten
  - Verschiedene Anwendungen, Zugriffskontrolle, Umstrukturierung
- Integritätssicherung
  - Korrektheit und Konsistenz des Datenbankinhalts
- Transaktionen
  - Mehrere DB-Operationen als Funktionseinheit
- Synchronisation
  - Koordination paralleler Transaktionen

- Datenschutz
  - Ausschluss nicht-autorisierter Zugriffe
- Datensicherheit
  - Wiederherstellung von Daten nach Systemfehlern
  - Persistenz
  - Große Datenmengen, Effizienz
- Katalog
  - Zugriff auf Datenbankbeschreibung im Data Dictionary (Metadaten)



E.F. Codd „Relational database: a practical foundation for productivity“ in CACM 25(2)  
(<http://dl.acm.org/citation.cfm?doid=358396.358400>)

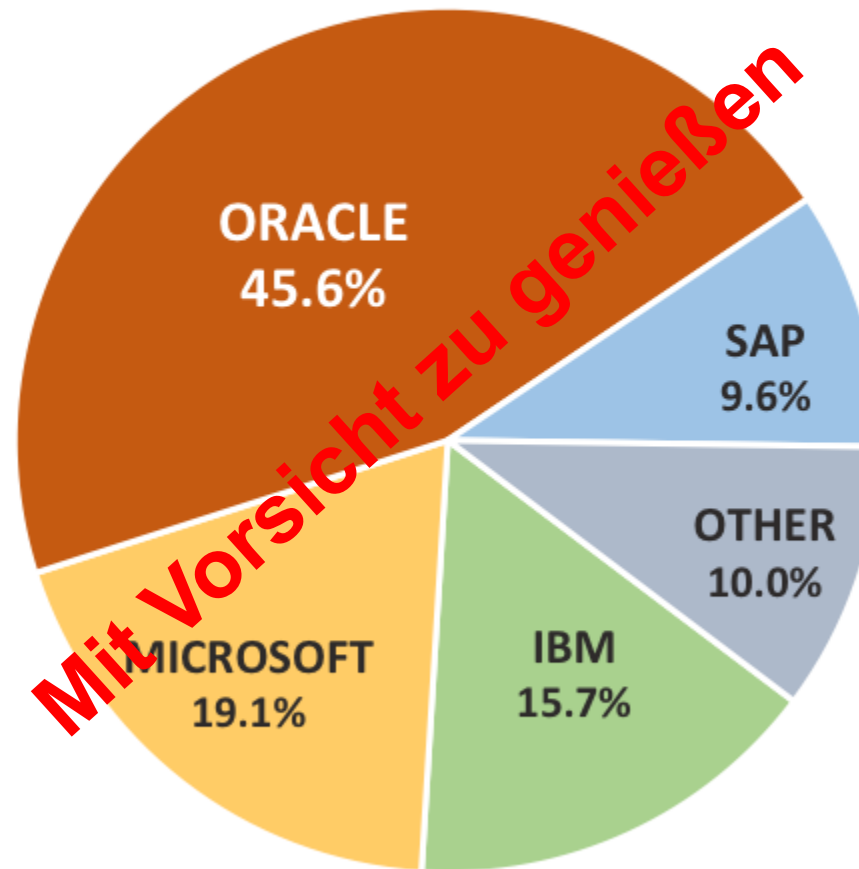


...



## 2016 COMMERCIAL DATABASE MARKET SHARE

(Source: Gartner, Inc. 2016)





- Popularität
- Basierend auf Informationen und Suchen im Web
- [https://db-engines.com/de/ranking\\_definition](https://db-engines.com/de/ranking_definition)

355 Systeme im Ranking, Oktober 2019

| Rang     |          |          | DBMS                         | Datenbankmodell             | Punkte   |          |          |
|----------|----------|----------|------------------------------|-----------------------------|----------|----------|----------|
| Okt 2019 | Sep 2019 | Okt 2018 |                              |                             | Okt 2019 | Sep 2019 | Okt 2018 |
| 1.       | 1.       | 1.       | Oracle +                     | Relational, Multi-Model i   | 1355,88  | +9,22    | +36,61   |
| 2.       | 2.       | 2.       | MySQL +                      | Relational, Multi-Model i   | 1283,06  | +3,99    | +104,94  |
| 3.       | 3.       | 3.       | Microsoft SQL Server +       | Relational, Multi-Model i   | 1094,72  | +9,66    | +36,39   |
| 4.       | 4.       | 4.       | PostgreSQL +                 | Relational, Multi-Model i   | 483,91   | +1,66    | +64,52   |
| 5.       | 5.       | 5.       | MongoDB +                    | Document, Multi-Model i     | 412,09   | +2,03    | +48,90   |
| 6.       | 6.       | 6.       | IBM Db2 +                    | Relational, Multi-Model i   | 170,77   | -0,79    | -8,91    |
| 7.       | 7.       | ↑ 8.     | Elasticsearch +              | Suchmaschine, Multi-Model i | 150,17   | +0,90    | +7,85    |
| 8.       | 8.       | ↓ 7.     | Redis +                      | Key-value, Multi-Model i    | 142,91   | +1,01    | -2,38    |
| 9.       | 9.       | 9.       | Microsoft Access             | Relational                  | 131,18   | -1,53    | -5,62    |
| 10.      | 10.      | 10.      | Cassandra +                  | Wide column                 | 123,22   | -0,18    | -0,17    |
| 11.      | 11.      | 11.      | SQLite +                     | Relational                  | 122,62   | -0,74    | +5,88    |
| 12.      | 12.      | ↑ 13.    | Splunk                       | Suchmaschine                | 86,84    | -0,17    | +9,94    |
| 13.      | 13.      | ↑ 14.    | MariaDB +                    | Relational, Multi-Model i   | 86,77    | +0,71    | +13,64   |
| 14.      | 14.      | ↑ 16.    | Hive +                       | Relational                  | 84,74    | +1,64    | +23,64   |
| 15.      | 15.      | ↓ 12.    | Teradata +                   | Relational, Multi-Model i   | 78,74    | +1,78    | +0,11    |
| 16.      | ↑ 18.    | ↑ 20.    | Amazon DynamoDB +            | Multi-Model i               | 60,18    | +2,36    | +5,71    |
| 17.      | ↓ 16.    | ↓ 15.    | Solr                         | Suchmaschine                | 57,57    | -1,40    | -3,75    |
| 18.      | ↓ 17.    | ↑ 19.    | FileMaker                    | Relational                  | 56,67    | -1,47    | +0,63    |
| 19.      | 19.      | ↓ 18.    | SAP Adaptive Server          | Relational                  | 55,84    | -0,26    | -2,73    |
| 20.      | ↑ 21.    | ↑ 21.    | SAP HANA +                   | Relational, Multi-Model i   | 55,35    | -0,04    | +0,98    |
| 21.      | ↓ 20.    | ↓ 17.    | HBase                        | Wide column                 | 54,84    | -0,88    | -5,84    |
| 22.      | 22.      | 22.      | Neo4j +                      | Graph                       | 49,46    | +1,25    | +6,81    |
| 23.      | 23.      | 23.      | Couchbase +                  | Document, Multi-Model i     | 32,21    | +0,91    | -3,71    |
| 24.      | 24.      | ↑ 28.    | Microsoft Azure Cosmos DB +  | Multi-Model i               | 31,33    | +0,46    | +11,07   |
| 25.      | 25.      | 25.      | Microsoft Azure SQL Database | Relational, Multi-Model i   | 27,51    | -0,03    | +1,24    |
| 26.      | ↑ 27.    | 26.      | Informix                     | Relational, Multi-Model i   | 26,00    | +0,60    | -0,24    |
| 27.      | ↓ 26.    | ↓ 24.    | Memcached                    | Key-value                   | 25,90    | -0,56    | -4,66    |
| 28.      | 28.      | ↑ 33.    | Google BigQuery +            | Relational                  | 25,63    | +1,08    | +8,96    |
| 29.      | 29.      | ↓ 27.    | Vertica +                    | Relational, Multi-Model i   | 22,51    | -0,04    | +1,15    |

Client-Server-Architekturen

# Schema

JDBC

ORDBMS ER-Modell Aktive Datenbanken

# Relational

ER Daten SQL DML  
NoSQL Recovery Big Data  
OODBMS DDL EER-Modell  
Transaktion Datenbank Normalisierung  
Architektur

# Modell

Relationale Algebra

## Abstraktionskonzepte und Datenmodelle

- **Programmunabhängigkeit**
  - DBen i.d.R. unabhängig von Programmen entwickelt
  - Daher aus unterschiedlichen Anwendungen zugreifbar
- **Datenunabhängigkeit**
  - Datennutzung erfolgt ausschließlich über abstrakte Darstellung der DB
  - D.h. Erweiterungen der DB zur Speicherung neuer Fakten wirken sich nicht auf Anwendungen aus
- DBMSe bieten **konzeptuelle Sicht** auf DBen
  - anwendungsnah, implementierungsneutral
  - Setzt keine Details über Speicherung der Daten / Realisierung der Programm-Zugriffsfunktionen voraus
- **Datenmodelle (DMe)** als Basis der Abstraktion
  - Bilden Grundlage der DB-Beschreibungssprachen

- **Datenmodelle**

- (syntaktische) **Sammlung von Elementen**
- Dienen der **Definition von DB-Strukturen**
- Bieten **Basisoperationen** zur Formulierung von Anfragen und zur Änderung (Update) von DBen
- Erweiterung um **benutzerdefinierte Operationen** ermöglicht Formulierung komplexer Anfragen und Transaktionen mit Updates

- **DB-Struktur (Schema)**

- Sammlung von
  - **Datentypen** (z.B. String, Integer, ...)
  - **Beziehungen** (z.B. „jeder Mitarbeiter hat einen Vorgesetzten“)
  - **Einschränkungen** (z.B. das Geburtsdatum muss in der Vergangenheit liegen)über den Daten
- Populäres Beispiel
  - Das **relationale Datenmodell**

- **Konzeptuelle Datenmodelle**

- Konzepte zur Beschreibung einer DB-Anwendung („Miniwelt“)
- Zur Diskussion mit Auftraggebern bzw. zukünftigen Benutzern
- Zur Dokumentation

- **Logische Datenmodelle**

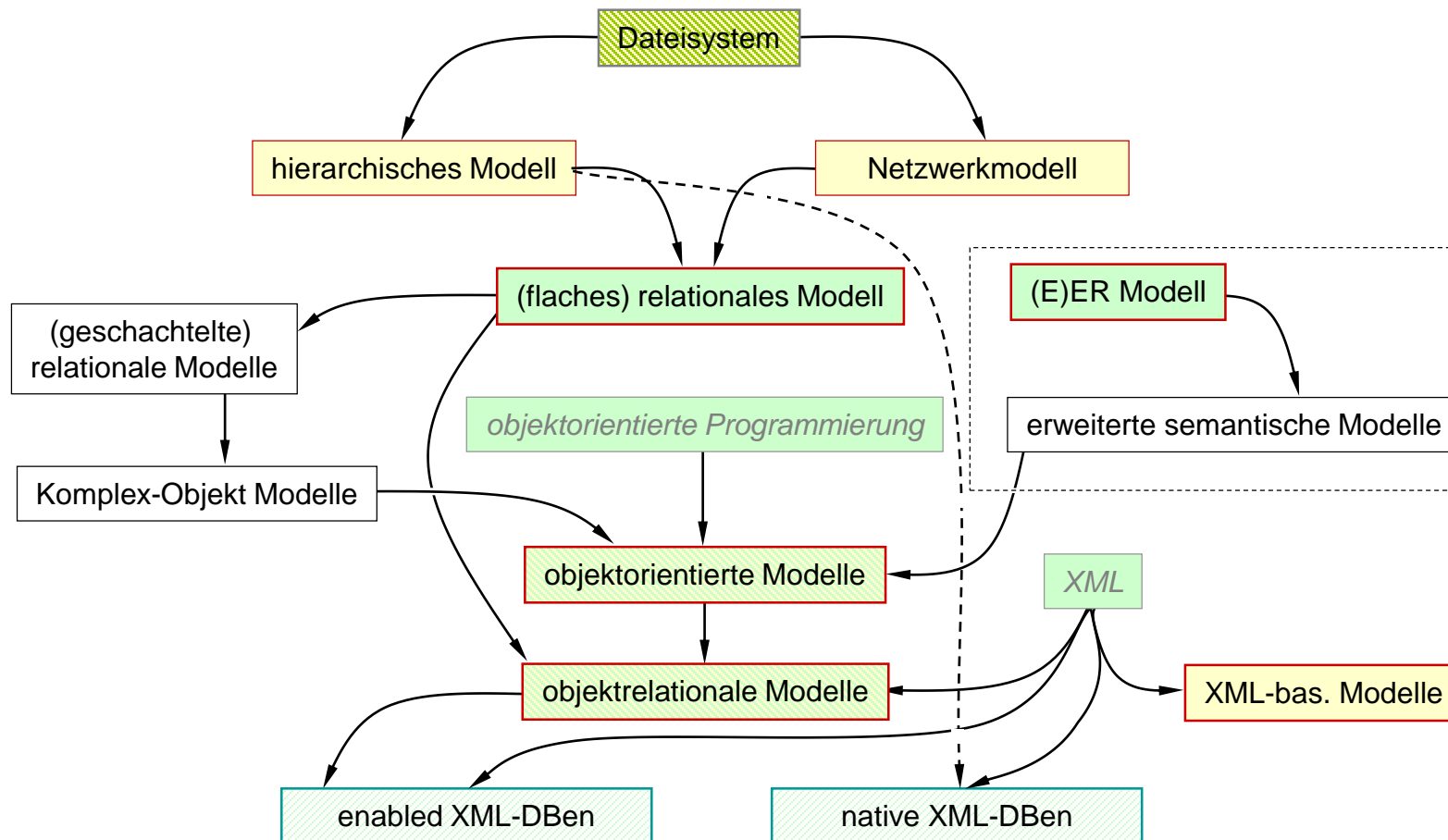
- (Darstellungs- und Implementierungsdatenmodelle)
- Konzepte, in die konzeptuelle Beschreibungen relativ einfach übersetzt werden können

- Nicht zu weit von der physischen Art der Speicherung entfernt
- Können direkt implementiert werden

- **Physische Datenmodelle**

- Konzepte zur Spezifikation der konkreten Datenspeicherung

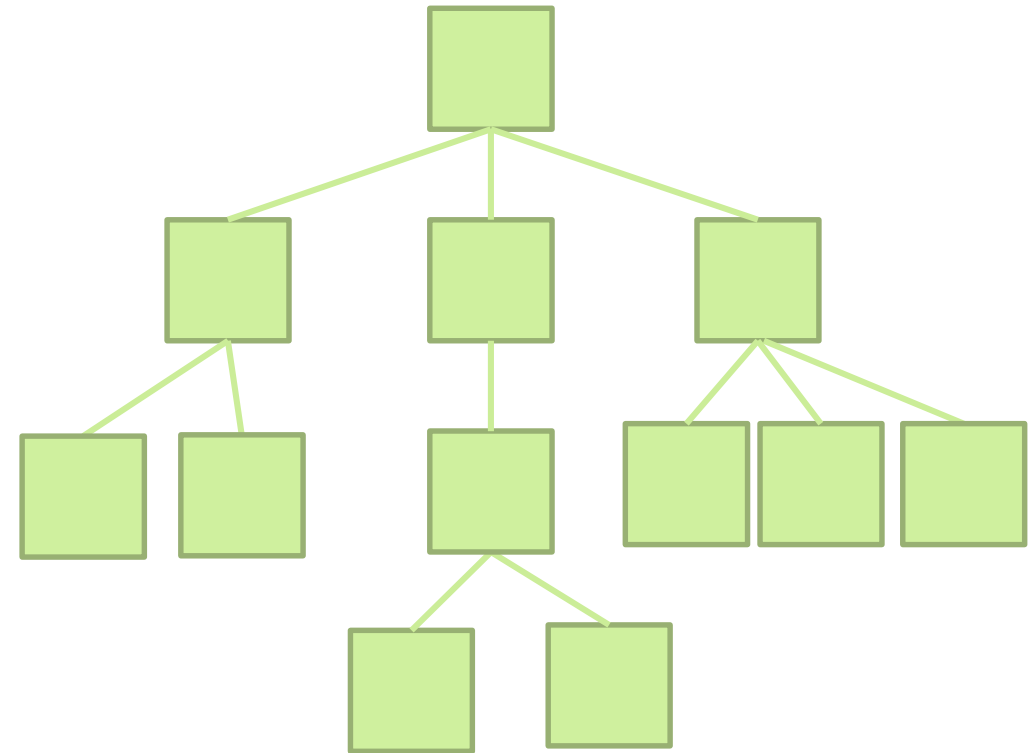
- Zusammenhang verschiedener Entwicklungen (logisch/konzeptionell)



## Warum die Historie?

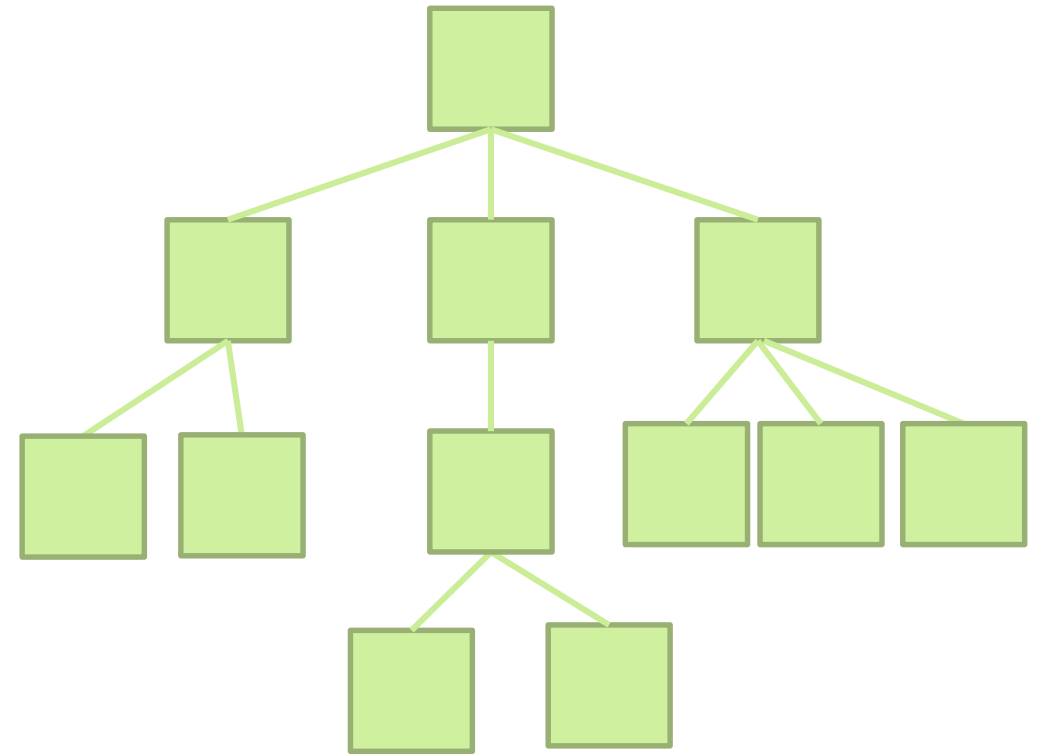
Viele Ideen finden sich in modernen NoSQL-Datenbanken wie Key-/Valuestores, Dokumentenorientierten Datenbanken, Spaltenorientierten Datenbanken etc. wieder.

- Gilt als das älteste klassische Datenmodell
- Datensätze werden hierarchisch strukturiert
  - Ein Datensatz bildet mit allen von ihm hierarchisch abhängigen Datensätzen eine Einheit
  - Beispiel: Gliederung eines Dokuments in Kapitel, Unterkapitel und Abschnitte





- **Natürliche Hierarchien** können i.d.R. direkt abgebildet werden
  - Beispiel Personaldatei (Firma → Abteilung → Angestellter)
- Andere Zusammenhänge werden durch **künstliche Hierarchien** dargestellt
  - Haben keine natürliche Entsprechung in der Miniwelt
  - Beispiel Artikeldatei (Artikel → Lieferant → Adresse)



- **Einstufige Hierarchien**

- Einem Elternelement werden ein oder mehr Kindelemente zugeordnet

- **Mehrstufige Hierarchien**

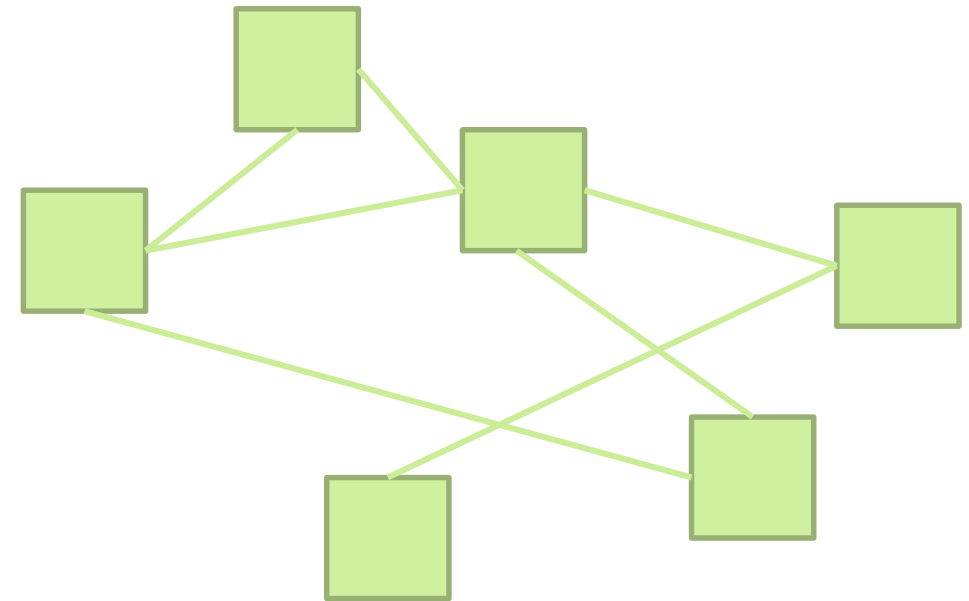
- Mehrere Gruppen (sind selbst Hierarchien) werden in Beziehung gesetzt
  - Jedes Element ist nur in einer Gruppe Kindelement
  - Das Wurzelement ist kein Kindelement → keine Zyklen

- Beziehungstypen im hierarchischen Modell

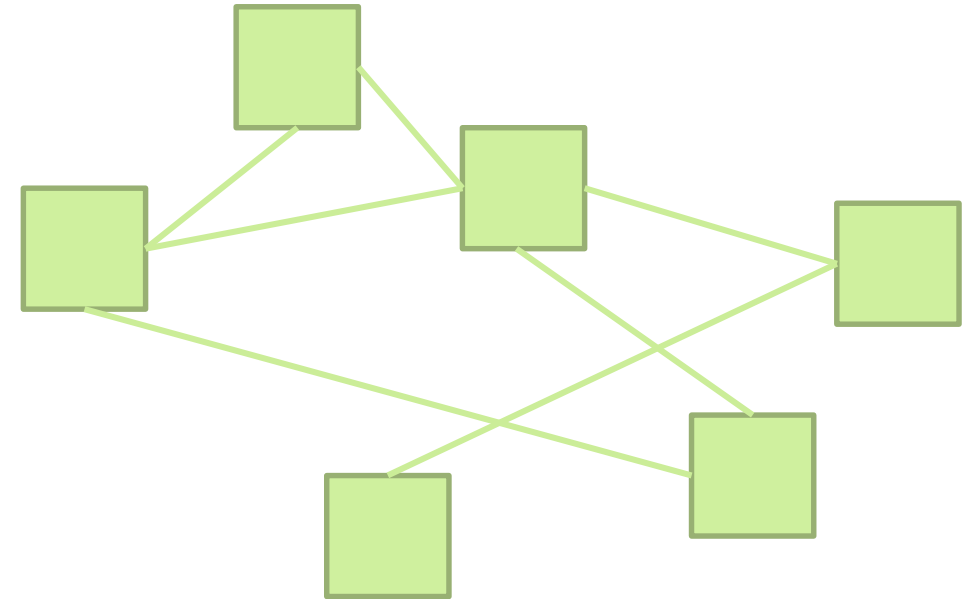
- 1 : 1
  - Elternelement wird Kindelement zugeordnet
- 1 : n
  - Elternelement wird mehreren Kindelementen zugeordnet
- m : n
  - **Nicht direkt darstellbar**

- Sequentielles Lesen möglich
  - Kinder folgen beim Lesen stets ihren Eltern
  - Kinder haben eine Ordnung (technisch gelöst über Zeiger)
  - Vorteilhaft für das Finden, Ändern, Hinzufügen, Löschen von Daten
- Einfaches und effizientes DMe
  - In speziellen Anwendungsfällen auch heute noch relevant
  - Z.B. Directory-/Web-Server, Index-Verwaltung in MySQL-DBen
- Nachteile
  - Mangelnde Flexibilität
  - Hierarchische Struktur für viele Anwendungen zu starr
  - Keine Darstellung von m:n-Beziehungen möglich

- Weiterentwicklung des hierarchischen Modells
  - Ein Element kann mehreren Gruppen zugeordnet werden
  - Mehrere Wurzelemente möglich („Polyhierarchie“)
- Mächtiger als hierarchisches Modell
  - „Natürliche“ Beschreibung von m:n-Beziehungen
  - Darstellung über zwei 1:n Beziehungen
  - Beispiel Kursbelegung
    - Student → Belegung ← Kurs



- Ermöglicht das Modellieren komplexer Anwendungszusammenhänge
- Nachteile
  - Verlust von Einfachheit und Übersichtlichkeit
  - Sequentielles Lesen ist komplizierter bzw. ineffizienter und daher langsamer



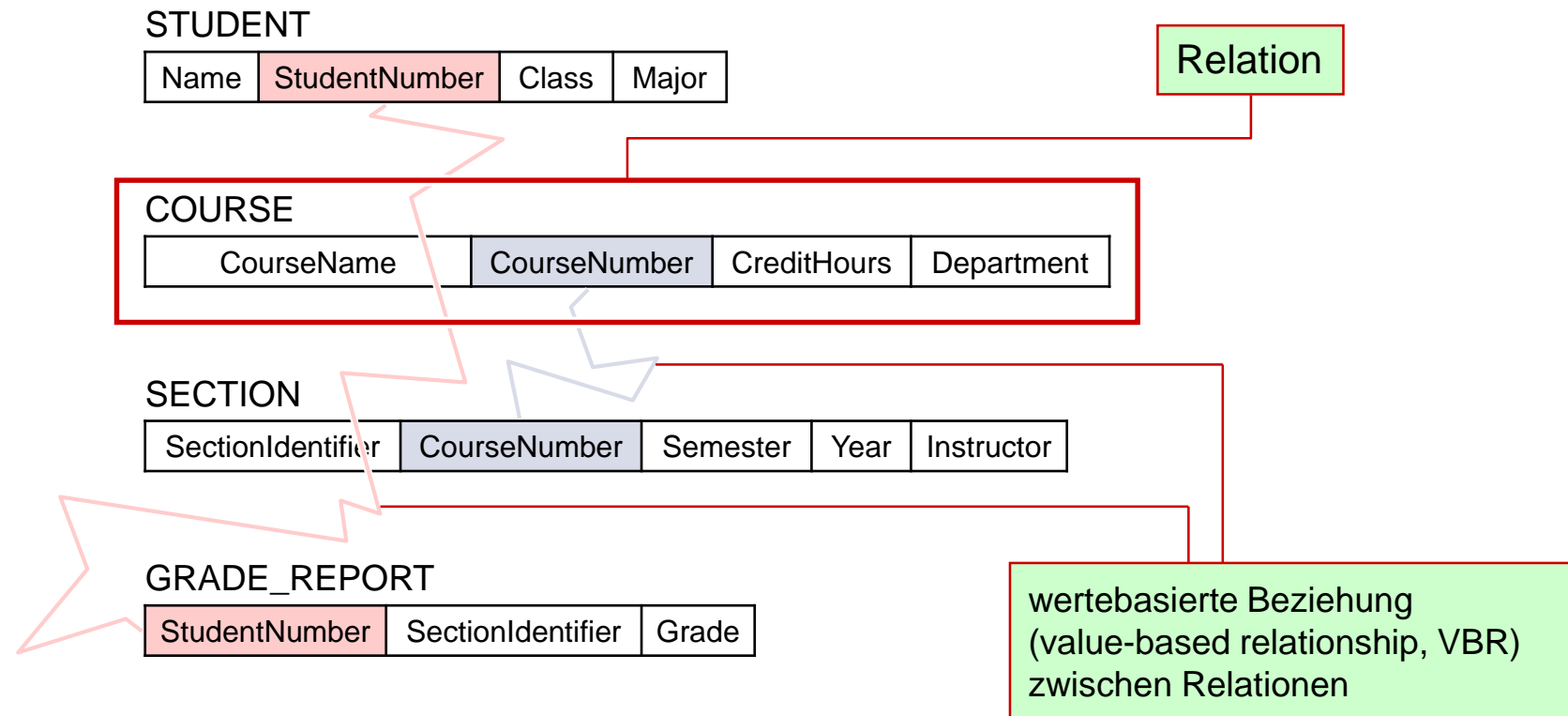
- Ist heute das **in der Praxis wichtigste** logische Datenmodell
  - Wird im Verlauf dieses Moduls vertieft behandelt
- Grundlegende Elemente
  - **Tabellen bzw. Relationen**
    - Bilden Objekte bzw. Konzepte der (realen) Anwendungswelt ab
      - Z.B. Student, Kurs, Angestellter, ...
    - Ihre Eigenschaften werden durch Spalten der Tabelle festgelegt
      - Z.B. Name, Adresse, Thema, Gehalt, ...

- Wiedergabe der Konzepte durch konkrete Wertbelegungen in den Zeilen der Tabelle

## – Beziehungen

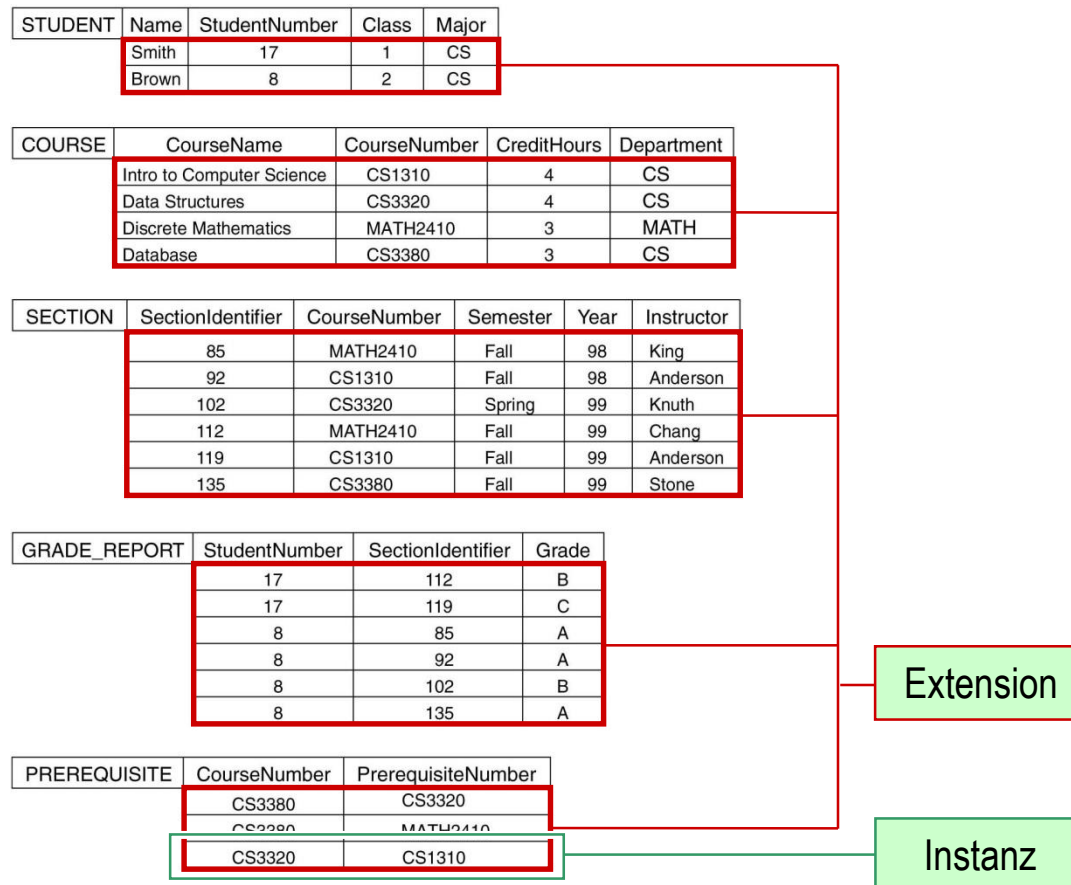
- Logische Verknüpfungen einzelner Fakten durch ihre Gruppierung in Tabellen sowie wertbasierte Zusammenhangsbeschreibungen
  - Z.B. Kursbelegung, Projektmitarbeiter
- Keine direkten „physischen“ Beziehungen wie im hierarchischen Modell oder Netzwerkmodell

- Ausschnitt einer DB zur Anwendung „Universität“





- Ausschnitt einer DB zur Anwendung „Universität“



## University-DB

- Studenten
- Kurse
- Arbeitsgruppen
- Noten
- Voraussetzungen

mit ihren jeweiligen Eigenschaften (Attributen).

## ergänzt um

- Datentypen  
(Wertebereiche)
- Beziehungen  
(Konsistenz/Integrität)
- Einschränkungen  
(von Werten/über Werte)

- Erlaubt komplexe Attributwerte
  - Attribute können selbst wieder Relationen sein
  - „Tabellen in Tabellen“
- Beispiel „Universität“:

| GRADE_REPORT | StudentNumer | SectionGrade ( SectionIdentifier, Grade ) |   |
|--------------|--------------|---|---|
|              | 17           | 112                                       | B |
|              |              | 119                                       | C |
|              | 8            | 85  | A |
|              |              | 92  | A |
|              |              | 102                                       | B |
|              |              | 135                                       | A |
| PREREQUISITE | CourseNumber | PreNr ( PrerequisiteNumber )              |   |
|              | CS3380       | CS3320                                    |   |
|              |              | MATH2410                                  |   |
|              | CS3320       | CS1310                                    |   |

- Geeignet, wenn die DB bzgl. ihrer Eigenschaften bzw. Daten
  - nicht präzise genug zu fassen ist
  - Flexibel bleiben soll
- Dies ist in den folgenden Situationen der Fall:
  - Daten besitzen eine geringe Dichte (Sparsity)
  - Struktur der Daten ist unbekannt, unpräzise oder über die Zeit einem strukturellen Wandel unterworfen
  - Daten beschreiben Kapselungshierarchien, die auch rekursiv sein können
  - Daten weisen eine inhärente Reihenfolge auf
  - Abfragen oder Aktualisierungen der Daten sollen ihre Struktur berücksichtigen

- XML-basierte Modelle bieten sich an wenn
  - Daten in unterschiedlichen Anwendungen auf verschiedenen Plattformen verarbeitet werden müssen
  - Dabei Transformation in spezielle Formate zwingend notwendig ist
- Unterscheidung in
  - **Native Ansätze**
    - Durchgängig XML-basiert
  - **Enabled XML Ansätze**
    - Integration von XML in relationale bzw. objektrelationale Datenbanken

## Hinweis

Noch flexibler: schemafreie Datenbanken, aber (noch) nicht so verbreitet

- Beschreiben konkret physische Speicherung der Daten
  - Datensatzformate
  - Datensatzanordnungen
  - Zugriffspfade
- Zugriffspfad
  - Datenstruktur
  - Unterstützt Suche von Datensätzen in einer DB
  - Beispiele für Zugriffspfade:
    - B-Bäume
    - B\*-Bäume
    - R-Bäume

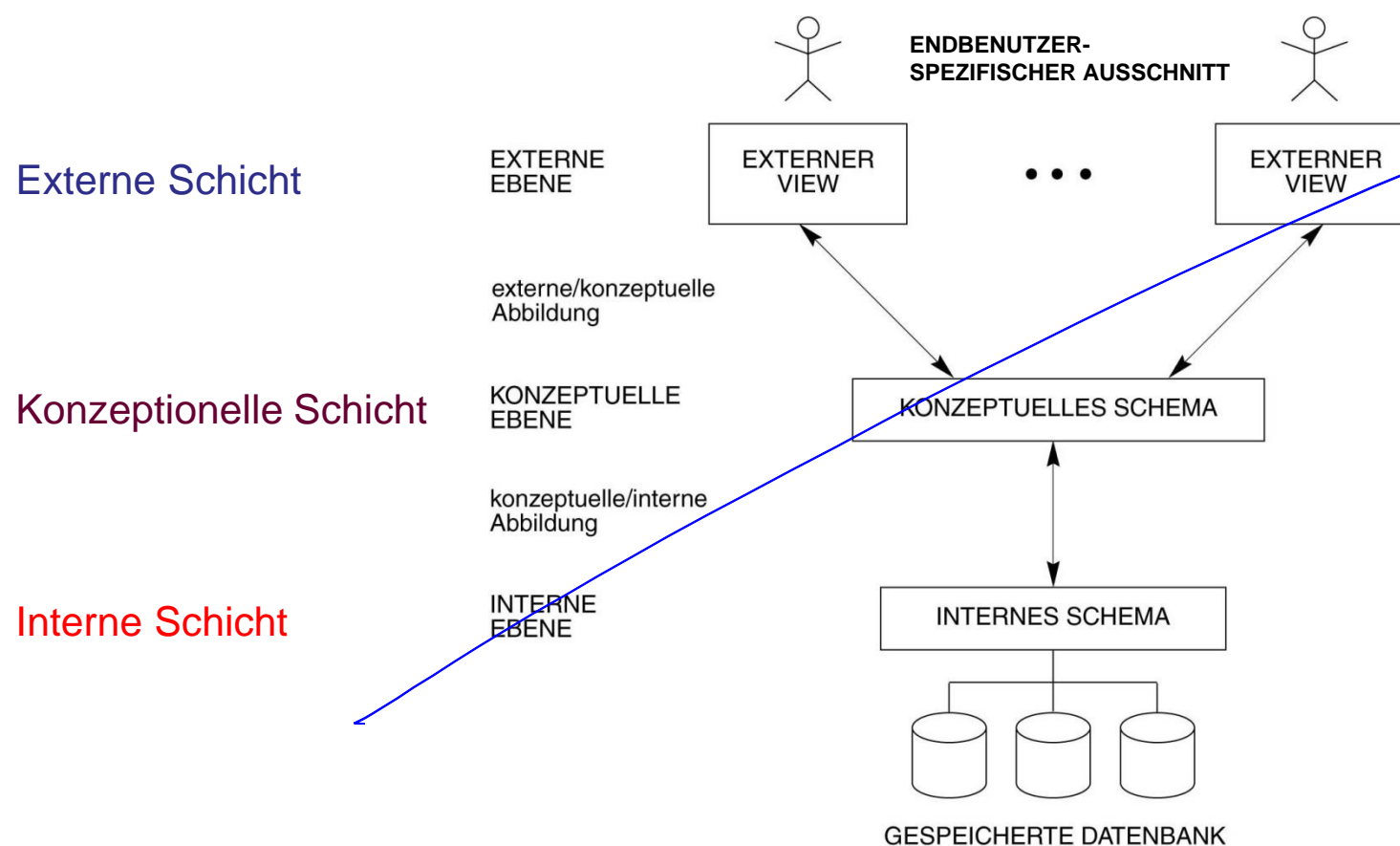
## Hinweis

Physische Datenmodelle werden in diesem Modul nicht weiter betrachtet

Zugriffspfade sind Teil von Informationssysteme 2

- **Schema** (Intension)
  - Beschreibung der kompletten Struktur einer DB
  - Sollte sich in der Praxis gar nicht oder nur selten ändern
  - Jedes Element des Schemas wird als **Schemakonstrukt** bezeichnet
- **Instanz** (elementare Extension)
  - Einzelne aus konkreten Datenelementen bestehende Datensätze
  - Entsprechen dem Schema
- **DB-Zustand** (Gesamt-Extension oder Snapshot)
  - Gesamtheit der aktuell in der DB gespeicherten Daten
  - Kann sich selten bis häufig ändern

- Erste Standardisierungsbemühungen der DBMS-Hersteller gab es bereits in den 60/70er Jahren

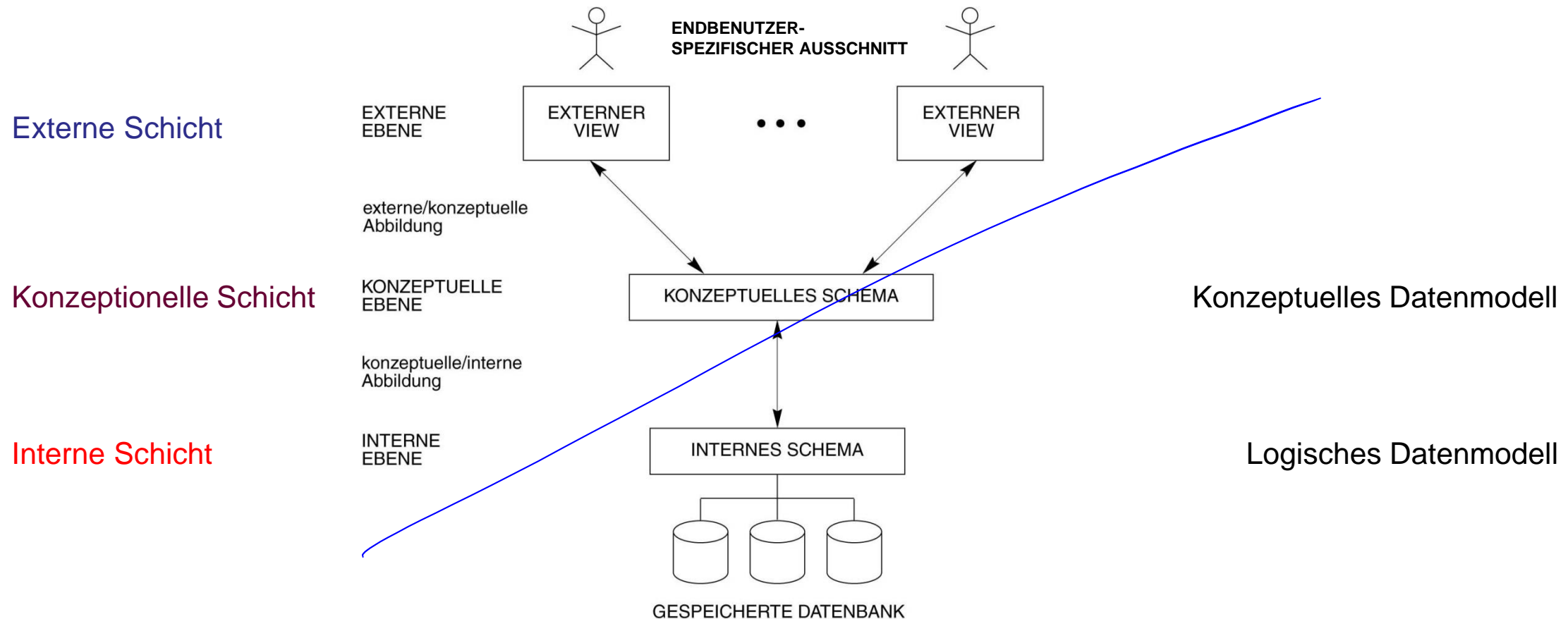


## Hinweis

Zuordnung der drei Schichten zu Kategorien von Datenmodellen nicht eindeutig:

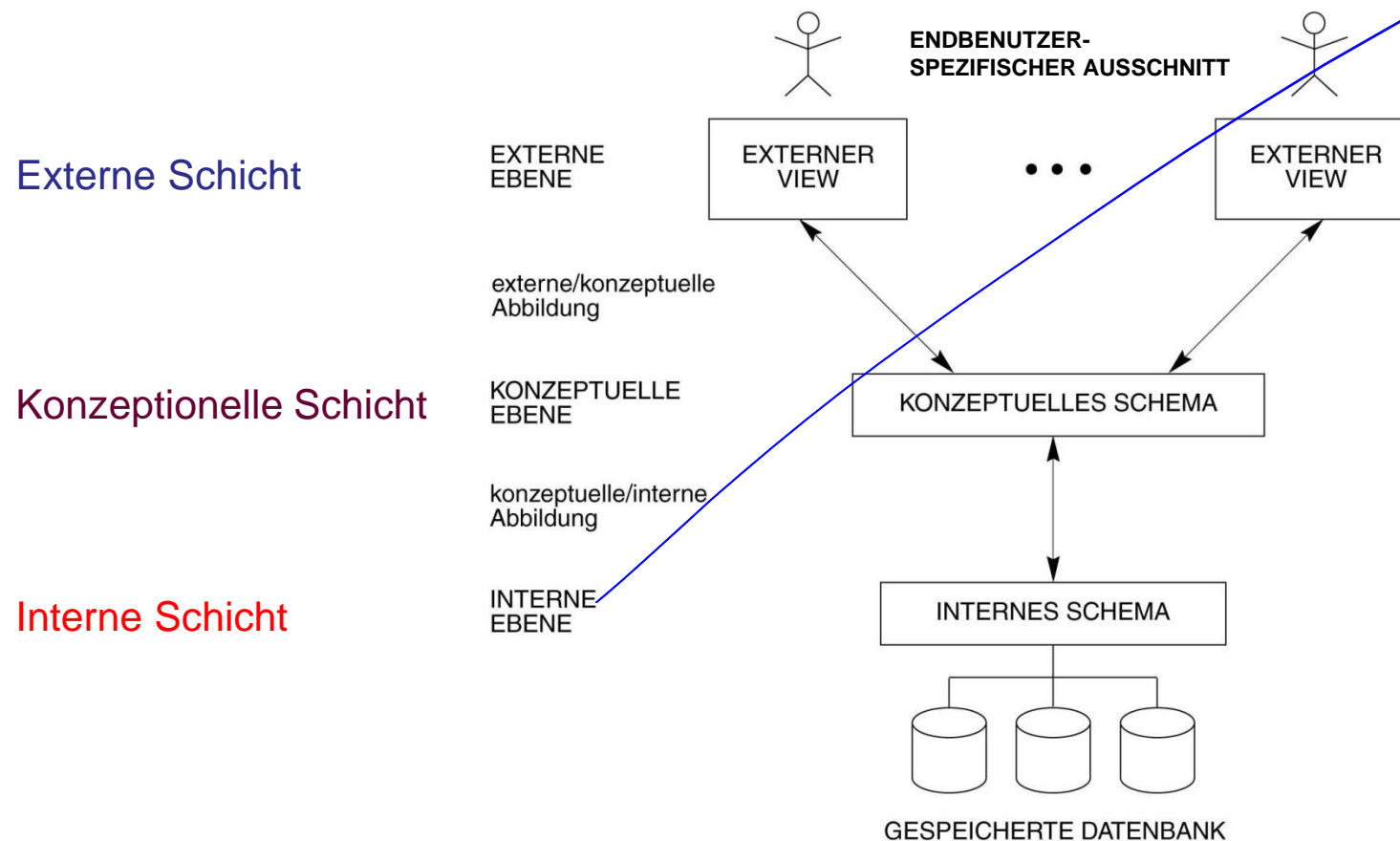
- Es liegt nahe, die konzeptuelle Ebene mit einem konzeptuellen Datenmodell zu beschreiben.
- Das interne Schema entspricht dann etwa einem logischen Datenmodell.
- Das physische Datenmodell ist in der DSA nicht explizit.

- Erste Standardisierungsbemühungen der DBMS-Hersteller gab es bereits in den 60/70er Jahren





- Erste Standardisierungsbemühungen der DBMS-Hersteller gab es bereits in den 60/70er Jahren

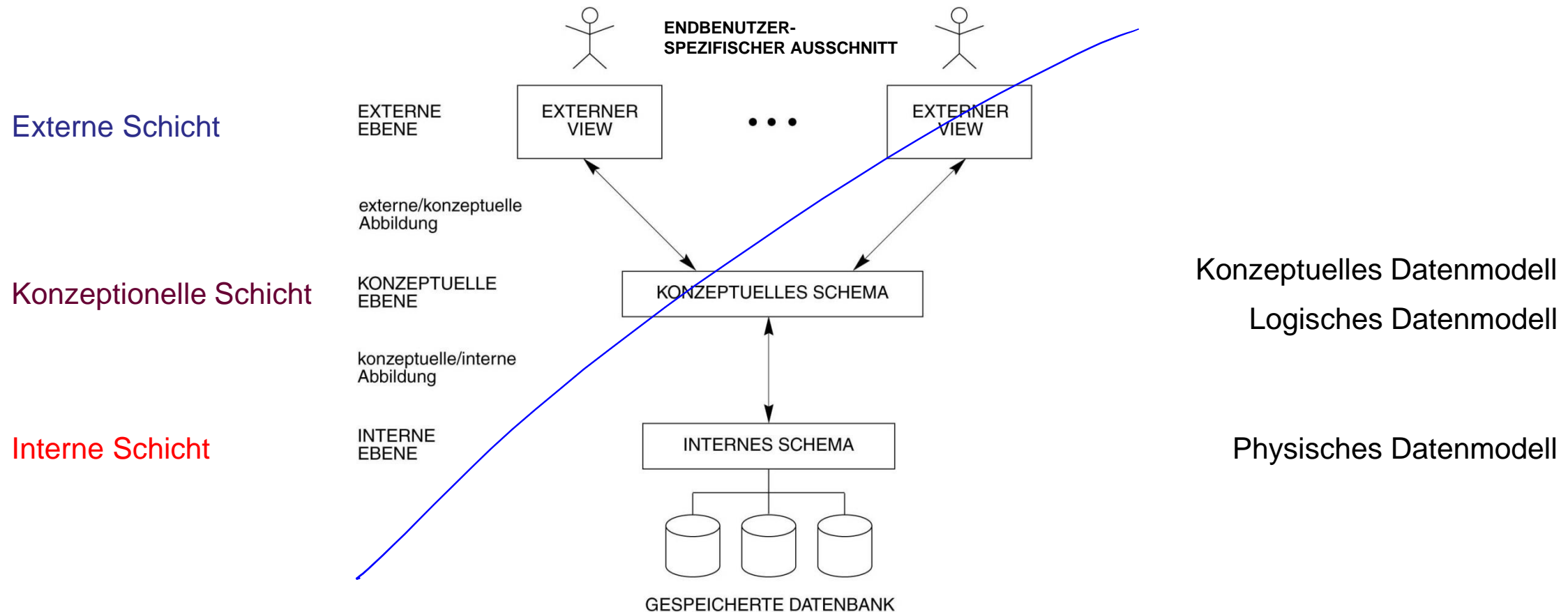


## Hinweis

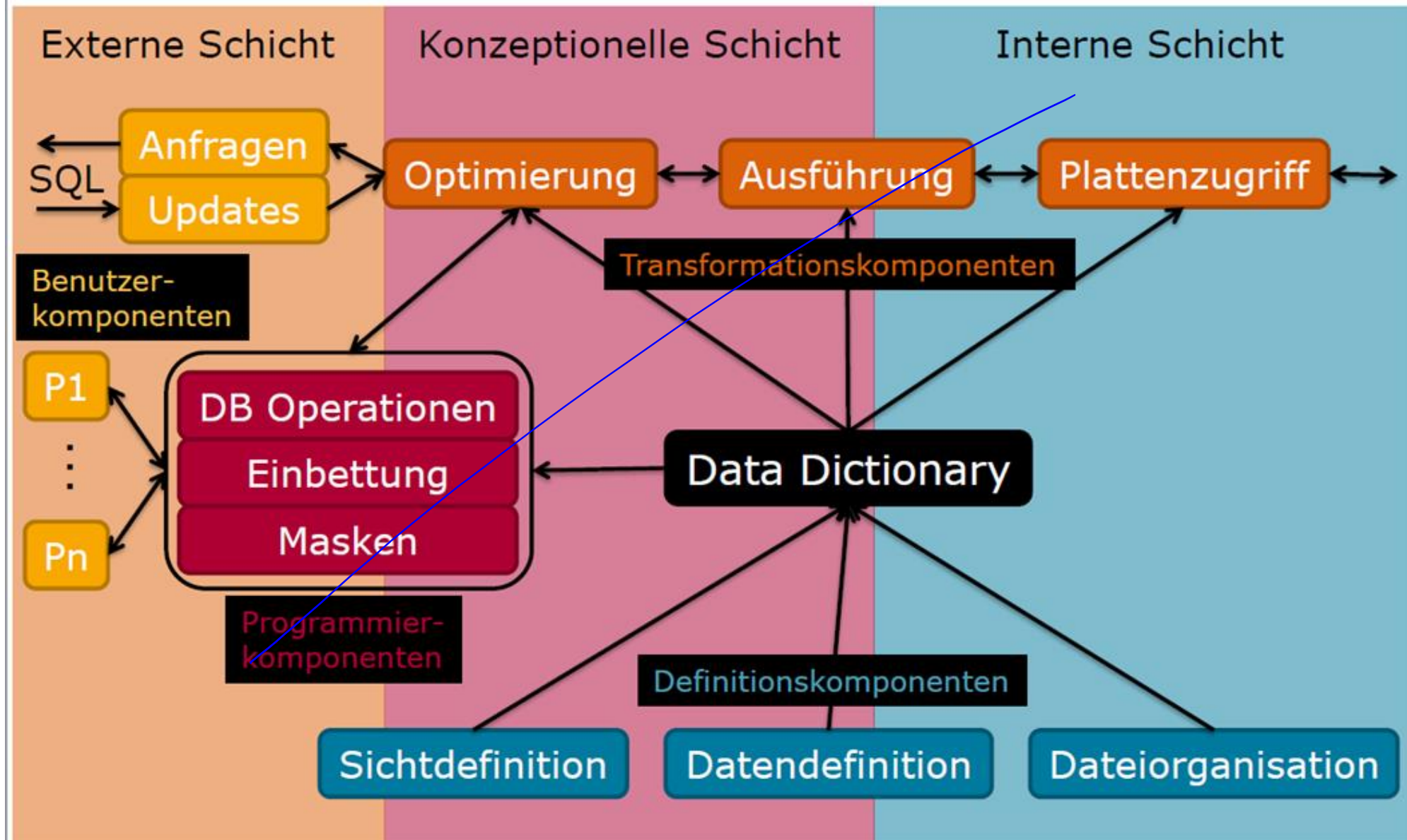
Alternatives Ebenenverständnis:

- Internes Schema beschreibt physische Speicherung durch ein physisches Datenmodell
- Konzeptuelle Ebene beschreibt Struktur der DB durch ein konzeptuelles Datenmodell
  - Kapselt Details der physischen Speicherung
- Externe Ebene bietet spezielle Sichten auf DB durch externe Schemata
  - Aspekte der DB können so verborgen werden

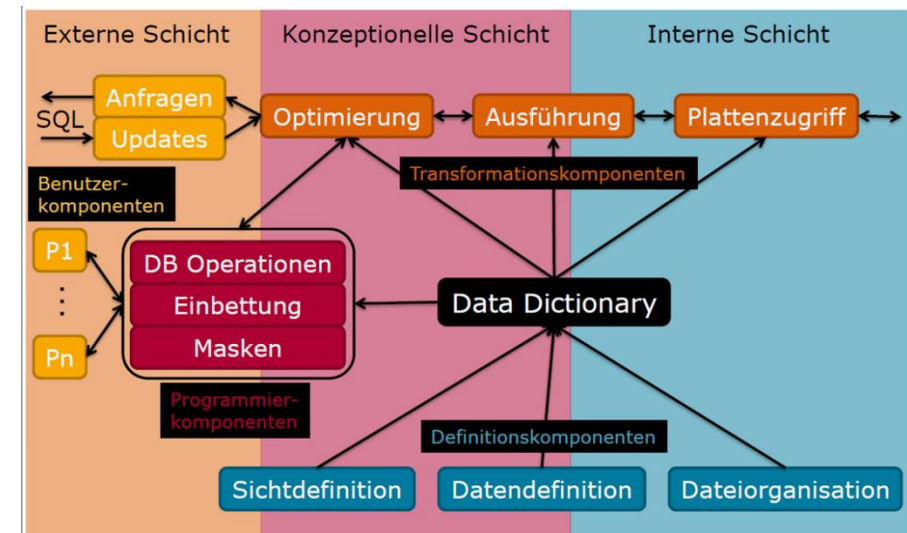
- Erste Standardisierungsbemühungen der DBMS-Hersteller gab es bereits in den 60/70er Jahren



- DBMS-Architekturvorschlag von 1975
- Verfeinert die Drei-Schichten-Architektur
  - Interne Ebene / Betriebssystem verfeinert
  - Mehrere interaktive und Programmier-Komponenten
  - Schnittstellen sind bezeichnet und normiert



- Definitionskomponenten
  - DDL, Sichten, Dateiorganisation, Indizes
- Programmierkomponenten
  - Entwicklungsumgebung und Programmiersprache
  - Integration von DB-Operationen
- Benutzerkomponenten
  - Anfrageinterface für Experten
  - DB-Anwendungen für Laien
- Transformationskomponenten
  - Anfrageausführung und Darstellung der Ergebnisse
- Data Dictionary
  - Metadaten (in relationalen Tabellen)



- **Logische Datenunabhängigkeit**  
(Anwendungsunabhängigkeit)
- Ziel: Änderungen am konzeptuellen und externen Schemata haben keine Auswirkungen auf andere externe Schemata und Anwendungsprogramme
- Beispiele:
  - **DB-Erweiterung** durch neue Datensatztypen/Datenfelder
    - i.d.R. kein Problem
  - **DB-Reduktion:** löschen bestehender Datensatztypen/Datenfelder
    - Wirkt sich auf externe Schemata aus, die diese Schemakonstrukte nutzen
  - **Erweiterung** („Verschärfung“) / **Reduktion** („Entschärfung“) **von Einschränkungen** der Schemata
    - Wirkt sich u.U. nicht auf externe Schemata und Applikationen aus

- **Physische Datenunabhängigkeit** (Implementierungsunabhängigkeit)
- Ziel: Änderungen der Dateiorganisationen und Zugriffspfade haben keinen Einfluss auf das konzeptuelle Schema
- Beispiel für Änderungen **interner Schemata**:
  - Umorganisation physischer Dateien um (zusätzliche) Strukturen für einen **effizienteren Zugriff auf die Daten einer DB** zu realisieren
  - Bleiben die **Daten gleich**, ist keine Änderung konzeptuellen / logischen DB-Schemas erforderlich

Client-Server-Architekturen

# Schema

JDBC

ORDBMS ER-Modell Aktive Datenbanken

# Relational

ER Daten SQL DML  
NoSQL Recovery Big Data  
OODBMS DDL EER-Modell  
Transaktion Datenbank Normalisierung  
Architektur

# Modell

Relationale Algebra

## DB-Sprachen



- Nach Abschluss des DB-Entwurfs mit konzeptionellen DMe folgt die Implementierung
  - **Definition der Schemata**
- Sprachen zur Implementierung
  - View Definition Language (VDL)
    - Externes Schema
  - Data Definition Language (DDL)
    - Logisches Schema
  - Storage Definition Language (SDL)
    - Internes Schema
- Sprachen zum DB-Zugriff („Manipulation“)
  - Data Manipulation Languages (DML)
  - **Updates**
    - Einfügen, Ändern, Löschen von Daten
  - **Queries**
    - (Reines) Anfragen von Daten

- Heutige DBMSe bieten VDL, DDL, SDL und DML **in einer Sprache integriert**
- Die Structured Query Language (SQL) integriert z.B. Aspekte von
  - View- und Data-Definition
  - Data-Manipulation
- Explizite SDL-Aspekte zur physischen Speicherung sind heute aus SQL entfernt
- SQL-DML
  - **Mengenorientierte DML**
    - Kann viele Datensätze mit nur einer Anweisung verarbeiten
    - Arbeitet auf Mengen von Datensätzen
  - **Deklarative Sprache**
    - Nutzer/Programme beschreiben **welche** Daten geliefert werden sollen
    - Es wird nicht beschrieben, **wie** auf die Daten zugegriffen werden soll
    - „was interessiert“ statt „wie findet man das“

## Code-Beispiel

```
CREATE TABLE Student (  
    Name          VARCHAR2(100) NOT NULL,  
    StudentNumber NUMBER(10)     PRIMARY KEY,  
    Class         NUMBER(2)      DEFAULT 1 NOT NULL,  
    Major         VARCHAR2(10)  
);  
  
INSERT INTO Student (Name, StudentNumber, Class, Major)  
VALUES ('Smith', 17, 1, 'CS');  
  
INSERT INTO Student (Name, StudentNumber, Class, Major)  
VALUES ('Brown', 8, 2, 'CS');  
  
SELECT Name  
FROM Student  
WHERE Major = 'CS';
```

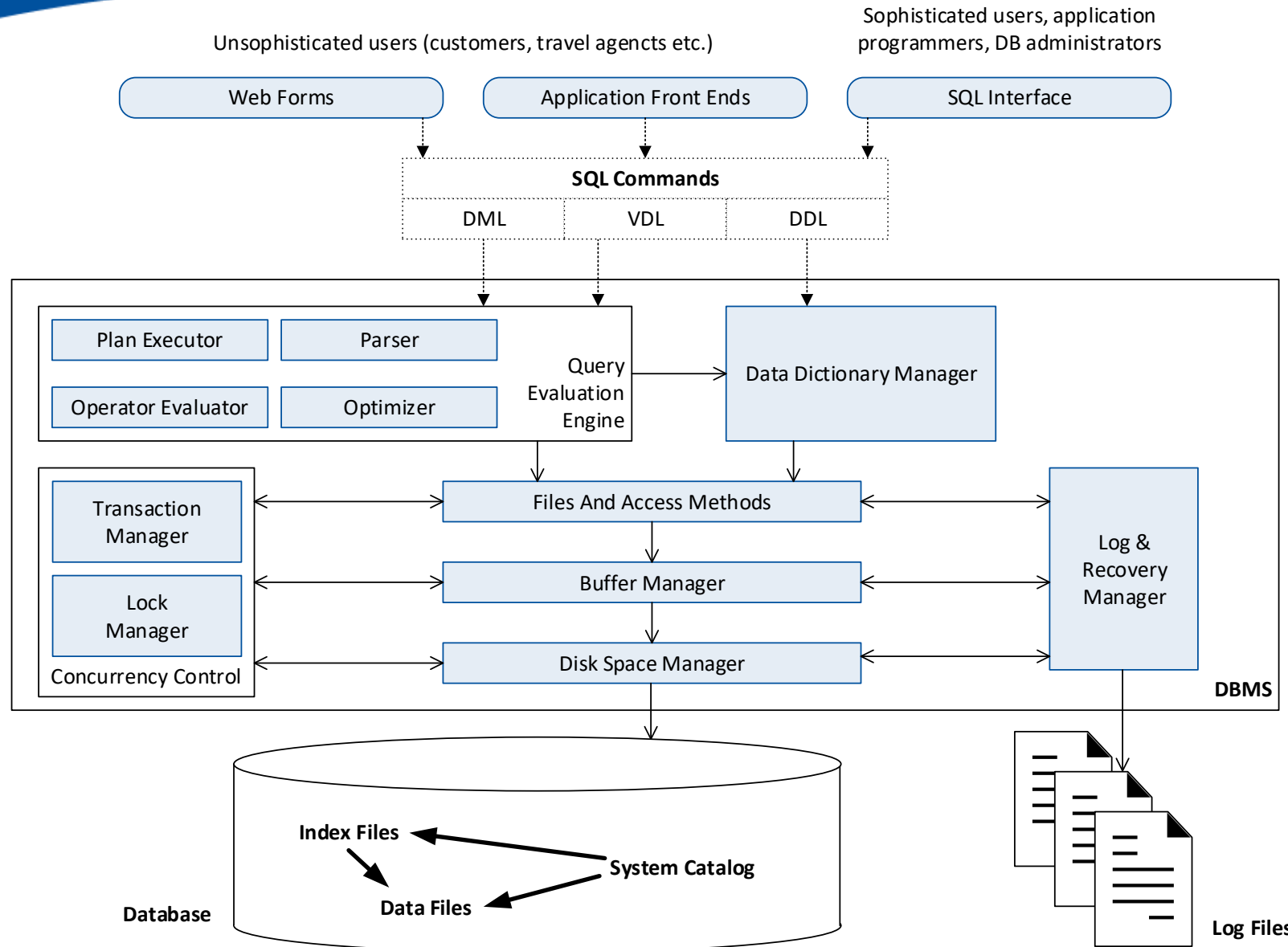
## Hinweis

Wird im Verlauf des Moduls vertieft

**Erzeugen eines  
Schemakonstruktes**

**Ablegen von  
Datensätzen in DB**

**Abfragen von  
Datensätzen aus DB**



**Quelle**  Angelehnt an: Ramakrishnan, Gehrke: Database Management Systems. Second Edition. Graw McHill, 2000.

**shows command flow**

**shows interaction**

**shows references**

- 67.000 Tabellen
- 700.000 Spalten
- 10.000 Sichten
- 13.000 Indizes
- 100.000.000 Zeilen
- 57 GB Daten (nach Initialisierung)
- 270.000.000 Zeilen Code



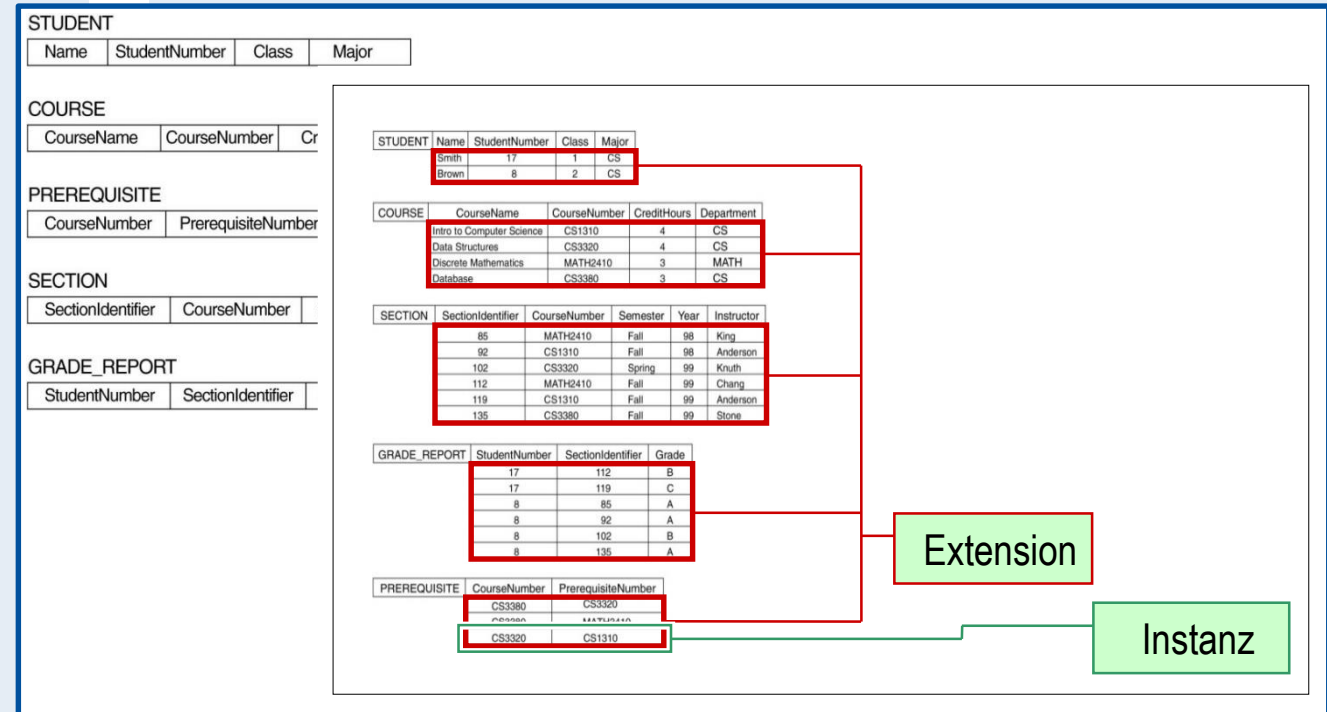
**Quelle**

Rudolf Munz, Keynote BTW 2007

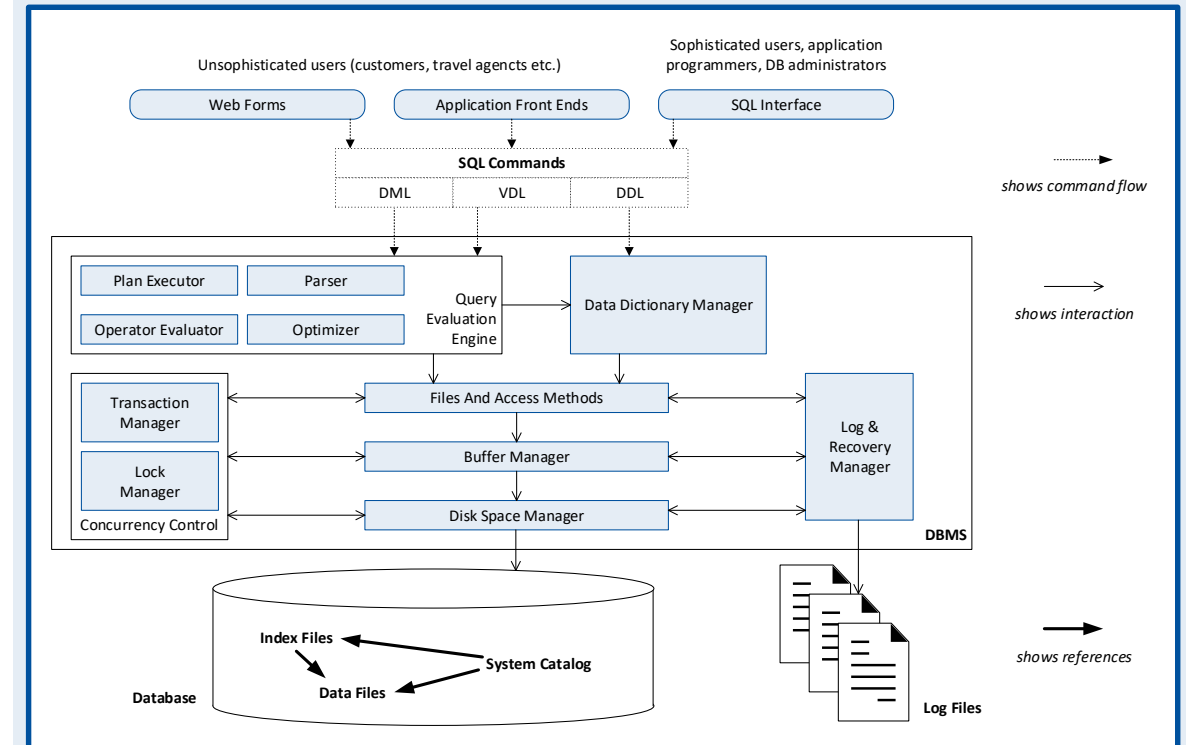
- Was ist eine Datenbank?
  - Darstellung einer „Miniwelt“
  - Sammlung von Daten
    - Logisch zusammenhängend
    - Inhärente Bedeutung
  - i.d.R. für bestimmten Zweck entworfen, entwickelt und mit Daten gefüllt
- Eigenschaften von Datenbanken
  - Persistente Speicherung großer Datenmengen
  - Metadaten in DB-Katalog
  - Integritäts- und Konsistenzbedingungen
  - Abstraktion (Programm- und Datenunabhängigkeit)
  - Mehrbenutzerfähigkeit
  - Transaktionen (ACID-Eigenschaften)
  - Datenschutz (Rechteverwaltung)
  - Views (individualisierte Sichten auf Daten)
  - Recovery

- Datenabstraktion
  - Datenmodelle:
    - konzeptuell (ER-Modell)
    - logisch
      - hierarchisches Modell
      - Netzwerkmodell
      - (flaches) relationales Modell
      - ~~hierarchisches relationales Modell~~
      - ~~objektorientierte Modelle~~
      - ~~objektrelationale Modelle~~
      - ~~XML-basierte Modelle~~
  - Physisch

- Schema (Intension), Instanz und DB-Zustand (Extension)



- Datenunabhängigkeit
  - Drei-Schichten-Architektur
  - Schemaänderungen auf einer Ebene haben nicht zwangsläufig Auswirkung auf höhere Ebene
  - Genauere Betrachtung:
    - logische und physische Datenunabhängigkeit
- DB-Sprachen
  - VDL, DDL, SDL und DML
- DB-Systemumgebung





- Datenbanken-Entwurfsprozess
- Modellierung (allgemein)
- Komponenten von ER-Modellen
- Beispiele für ER-Modelle

