

# Datenbanksysteme

## 0 - Einführung und Organisatorisches

Prof. Dr. Katja Zeume

[katja.zeume@w-hs.de](mailto:katja.zeume@w-hs.de)

WHS Gelsenkirchen

# Kurze Vorstellung

## Prof. Dr. Katja Zeume

Raum: A4.1.08 oder A4.0.07 (Dekanat)

Email: [katja.zeume@w-hs.de](mailto:katja.zeume@w-hs.de)

Sprechstunde: nach Vereinbarung

## Themen

- Bachelor: Datenbanksysteme, Data on the Web, Knowledge Graphs
- Master: NOSQL Datenbanken, Datenbanktheorie
- Abschlussarbeiten (BA/MA), Seminarthemen, Wissenschaftliche Vertiefungen
- Ausdrucksstärke und Komplexität von Anfragesprachen
- Didaktische Konzepte für NoSQL-Datenbanken

## Modul Datenbanksysteme (DBA)

- 3. Semester Bachelor
- 3 SWS Vorlesung, 1 SWS Übung, 1 SWS Praktikum
- Prüfungsleistung Klausur (PO2016, PO2023)

## Lerninhalt

Kenntnisse über die wesentlichen Merkmale von Datenbanksystemen:

- Modellierung von **Daten** (mit Hilfe relationaler Datenbankschemata)
- Verarbeitung von Daten durch mehrere Nutzer und unter Beachtung von Integritätsbedingungen (**Anfragesprache SQL**)
- DB-Implementierung, Transaktionsmanagement, Integrität
- Ausblick zu anderen Datenbankmodellen

# Was sind Daten?

## Daten (Singular: Datum)

- **Daten** bestehen aus einzelnen Zeichen, Symbolen oder Zahlenwerte, die durch Beobachtungen oder Messungen gewonnen wurden.  
Beispiel: Eine Zahlenkette 19081923
- Als **Information** werden Daten bezeichnet, die mit Kontext angereichert wurden.  
Beispiel: Geburtstag 19.08.1923
- Um **Wissen** aus den Daten zu ziehen, ist es notwendig die vorliegende Daten auszuwerten, z. Bsp. zu kombinieren, um komplexere Sachverhalte zu verstehen. Das Wissen über den Geburtstag wird erst interessant wenn man weiß wer hier Geburtstag hat. In diesem Fall Edgar F. Codd ein bekannter Datenbankler.

Dies soll keine formale Definition sein! Die Begriffe werden umgangssprachlich oft anders verstanden und redundant verwendet.

# Beispiele für große Datensätze

## Knowledge Bases

- **DBpedia** (Universität Leipzig, Mannheim, HPI)  
<https://wiki.dbpedia.org/>
  - 3 Mrd. Einzeldaten
  - ~ 4,5 Mio. Einträge basierend auf Wikipedia Artikeln
  - davon ~ 4,2 Mio. Einträge klassifiziert in Ontologien:
    - ~ 1,4 Mio. Personen, ~ 700T. Orte,
    - ~ 100T. Musikalben, ~ 85T. Filme
- **Yago** (entwickelt am MPI in Saarbrücken)  
<https://yago-knowledge.org/>
  - 10 Mio. Entitäten
  - 120 Mio. Fakten über diese Entitäten
- **Google Knowledge Graph**
  - ~ 800 Mrd. Fakten (8 Mrd. Entitäten)

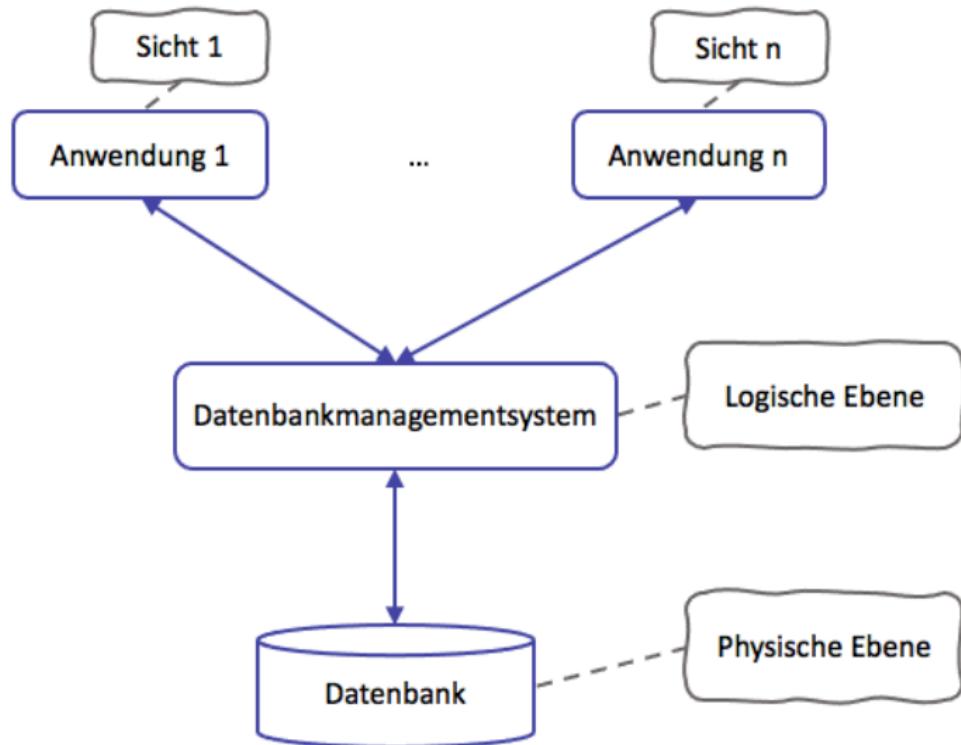
→ Modellieren, effizient und skalierbar Speichern, Verarbeiten

# Inhalt der Vorlesung - Überblick der Kapitel

1. Grundlegende Eigenschaften von Datenbanken
2. SQL Teil I
3. Datenbankentwurf
4. Relationale Algebra
5. Java-Schnittstelle JDBC
6. SQL Teil II - Erweiterte SQL Anfragen
7. Funktionale Abhängigkeiten und Normalformen
8. Transaktionsmanagement und Integritätssicherung
9. Implementierung von Datenbanken und Anfrageoptimierung
10. XML
11. Ausblick - Erweiterte DB Formate

Im Praktikum viele Übungen zu den Themen und SQL

# Kurzvorstellung Kapitel 1 - DB Eigenschaften



# Kurzvorstellung Kapitel 2 - SQL Teil I

**Tabelle: Student**

Film-ID	Titel	Erscheinungsdatum	Genre	Dauer
1	Wonka	2023-12-07	Family	117
2	Dune:Part Two	2024-03-02	Action	166
3	Barbie	2023-07-20	Comedy	114
4	Oppenheimer	2023-07-20	Thriller	180
5	John Wick 4	2023-03-23	Thriller	169

**SQL Anfrage:**

```
SELECT Titel, Genre  
      FROM Film  
     WHERE Dauer >= 150
```

**Ergebnis:**

Titel	Genre
Dune:Part Two	Action
Oppenheimer	Thriller
John Wick 4	Thriller

# Kurzvorstellung Kapitel 2 - SQL Teil I

The screenshot shows the PRAXIS IDE interface. At the top, there's a navigation bar with 'PRAXIS' and 'Version control'. Below it is a toolbar with icons for file operations like New, Open, Save, and a search icon. The main workspace has a 'Database' tab selected. On the left, a tree view shows a folder 'console\_1 [nordwind]' containing a 'SELECT' query:

```
1 ✓ SELECT "Artikelname"
2   FROM "Artikelv";
```

To the right of the query, the 'Database' panel shows the structure of the 'nordwind' database, specifically the 'Artikel' table:

- tables: 11
- columns: 10
  - Artikel-Nr: INTEGER (auto increment) = AUTOINCREMENT: sta
  - Artikelname: VARCHAR(40)
  - Lieferanten-Nr: INTEGER
  - Kategorie-Nr: INTEGER
  - Liefereinheit: VARCHAR(25)

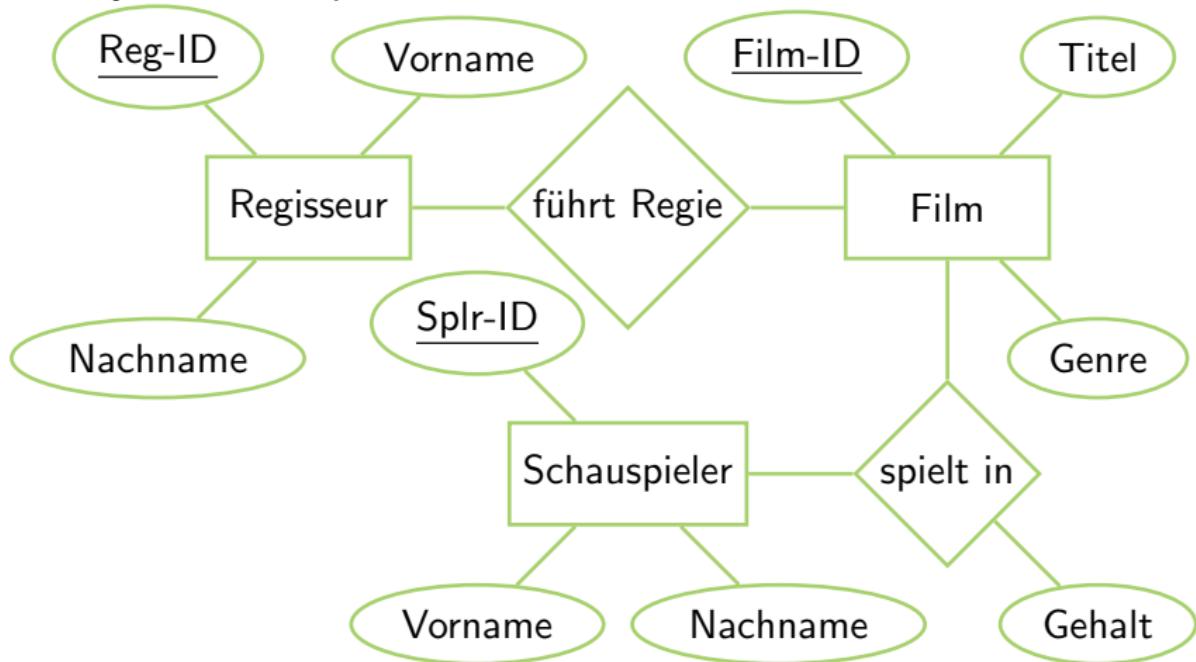
Below the database structure, the 'Output' tab displays the results of the executed query:

	"Artikelname"
1	Chai
2	Chang
3	Aniseed Syrup
4	Chef Anton's Cajun Seasoning
5	Chef Anton's Gumbo Mix
6	Grandma's Boysenberry Spread
7	Uncle Bob's Organic Dried Pears
8	Northwoods Cranberry Sauce
9	Mishi Kobe Niku
10	Ikura
11	Queso Cabrales

At the bottom, the status bar shows 'Database Consoles > nordwind > console\_1 [nordwind]'. The bottom right corner shows the time '2:16' and file encoding 'CRLF UTF-8 4 spaces'.

# Kurzvorstellung Kapitel 3 - Datenbankentwurf

Entity-Relationship-Modell:

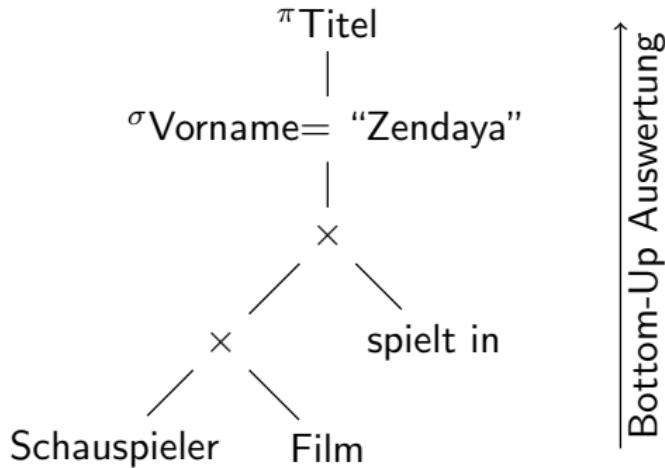


# Kurzvorstellung Kapitel 4 - Relationale Algebra

## Algebra-Ausdruck:

$$\pi_{Titel}(\sigma_{Vorname = "Zendaya"}((Schauspieler \bowtie Film) \bowtie spieltin))$$

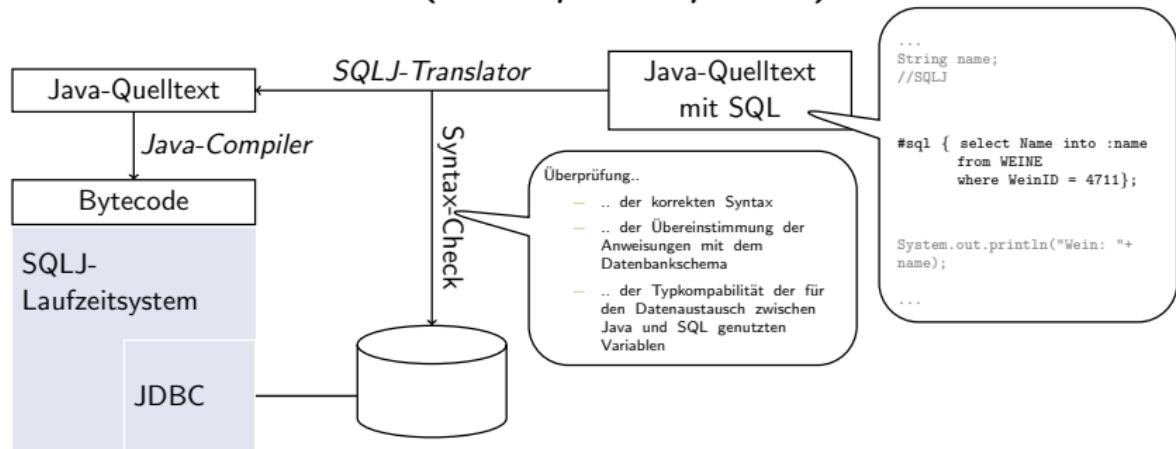
### Anfragebaum:



- Mathematische Formalisierung von SQL Anfragen
- Grundlage für Verifikation und Anfrageoptimierung

# Kurzvorstellung Kapitel 5 - JDBC

## Embedded SQL für Java (SQLJ / SQL / OLB)

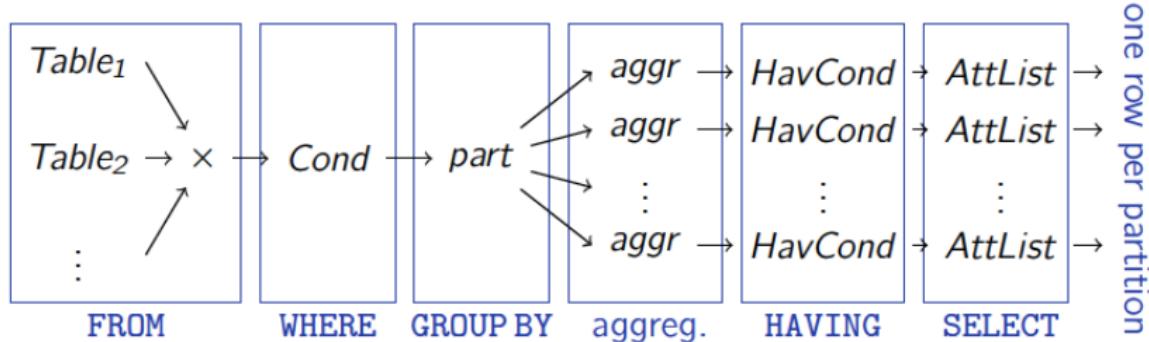


## SQLJ & JDBC

- SQLJ unterstützt von Hause aus nur fertig ausformulierte SQL-Ausdrücke - **statisches SQL**
- Über den Umweg von JDBC ist aber auch **dynamisches SQL** möglich

# Kurzvorstellung Kapitel 6 - SQL Teil II

## Auswertung der Anfrage:



[ Quelle Teuber ]

- Der **HAVING**-Teil wird **nach** **GROUP BY** und Aggregierungen ausgewertet. Er wird **vor** dem **WHERE**-Teil ausgewertet.
- Bedingungen, die **nur** die Attribute aus **GROUP BY** benutzen, können auch im **WHERE** Teil stehen.

## Auswertung einer SELECT-Anfrage (Übersicht):

1. FROM
2. WHERE
3. GROUP BY
4. HAVING
5. SELECT
6. ORDER BY

→ HAVING kommt nur in Kombination mit GROUP BY vor.



# Kurzvorstellung Kapitel 7 - Funktionale Abhangigkeiten und Normalformen

## Funktionale Abhangigkeit (FA) oder *Functional Dependency (FD)*

Gegeben eine Relation  $R$  und Attributmengen  $X_1, \dots, X_n$  und  $Y_1, \dots, Y_m$ .

Wir schreiben

$$X_1, \dots, X_n \rightarrow Y_1, \dots, Y_m,$$

wenn  $X_1, \dots, X_n$  die Attribute  $Y_1, \dots, Y_m$  eindeutig bestimmt.

(Zur Lesbarkeit werden die Mengenklammern oft weggelassen.)

## Normalformen

- 1. bis 3. Normalform
- Boyce-Codd-Normalform (BCNF)

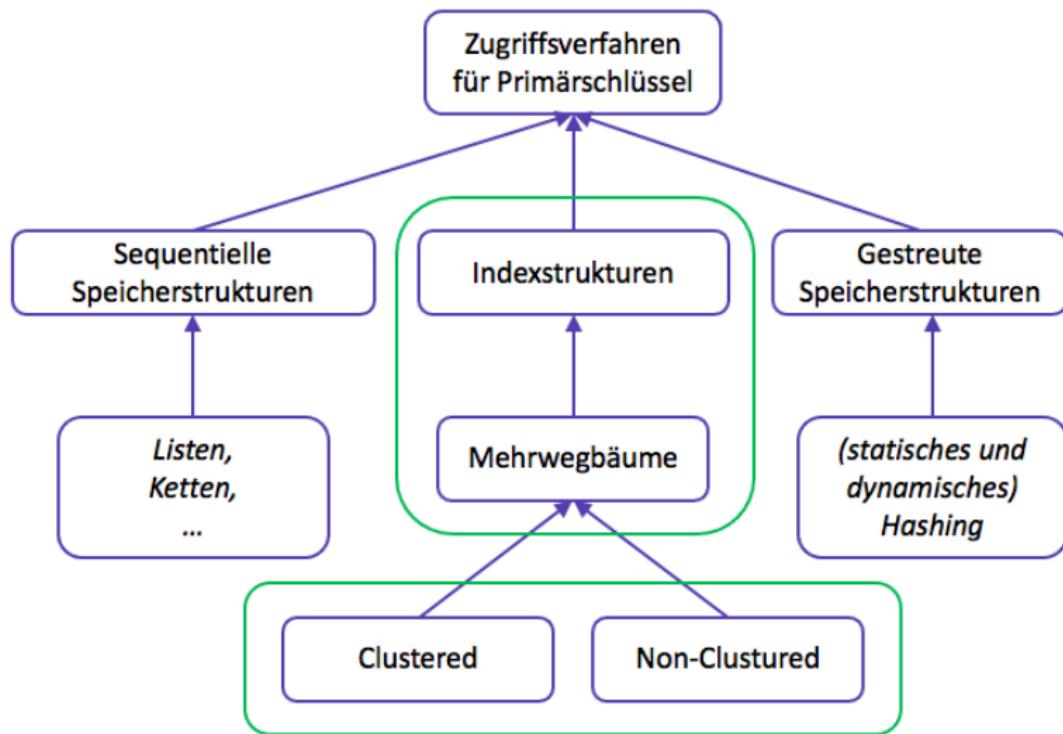
# Kurzvorstellung Kapitel 8 - Transaktionsmanagement und Integritätssicherung

## ACID-Prinzip

- **Atomaricity** Transaktionen werden “ganz oder gar nicht” ausgeführt.
- **Consistency** Transaktionen führen nur zu zulässigen Zuständen.
- **Isolation** Transaktionen werden isoliert für einen Nutzer ausgeführt.
- **Durability** Das Ergebniss der Transaktion ist persistent.

Isolationsstufe	Dirty R.	Inkons. L.	Lost Up.	Phantom P.
read uncommitted	ja	ja	ja	ja
read committed	nein	ja	ja	ja
repeatable read	nein	nein	nein	ja
serializable	nein	nein	nein	nein

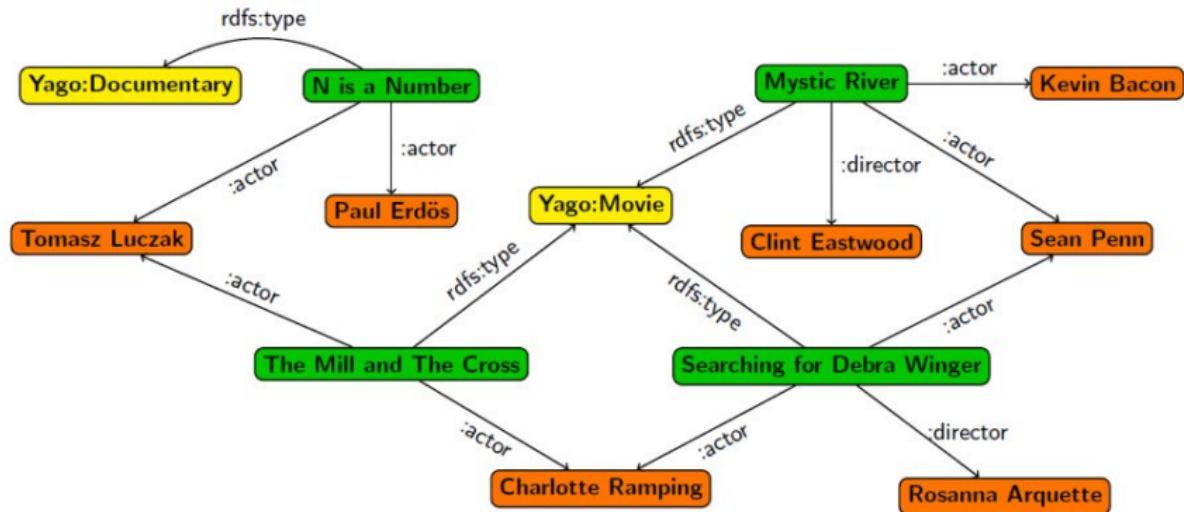
# Kurzvorstellung Kapitel 9 - DB-Implementierung und Anfrageoptimierung



# Kurzvorstellung Kapitel 10 - XML

```
<kino>
    <film>
        <film-id> 1 </film-id>
        <titel> Wonka </film-id>
        <genre> Family </genre>
    </film>
    <film>
        <film-id> 2 </film-id>
        <titel> Dune: Part Two </film-id>
        <genre> Action </genre>
    </film>
    <spielt in>
        <film-id> 1 </film-id>
        <schauspieler-id> 4 </schauspieler-id>
    </spielt in>
    <spielt in>
        <film-id> 2 </film-id>
        <schauspieler-id> 4 </schauspieler-id>
    </spielt in>
    <schauspieler>
        <schauspieler-id> 4 </schauspieler-id>
        <vorname> Timothée </vorname>
        <nachname> Chalamet </nachname>
    </schauspieler>
</kino>
```

# Kurzvorstellung Kapitel 11 - Ausblick Erweiterte Datenbankformate



# Organisation des Kurses

## Vorlesung

- In Präsenz
- Dienstags 11:30 - 13:50 (2x5min Pause)
- Prof. Dr. Katja Zeume

## Praktika (unterstützt durch Frank Laarmann)

- ab dem 14.10.
- Di 14:35 - 16:10 / 16:15 - 17:50 (A 4.2.11)
- Mi 9:50 - 11:25 (A4.2.09)
- Do 13:45 - 15:20 / 15:25 - 17:00 (A 4.2.09)
- Nehmen Sie unbedingt an der Umfrage (offen bis 9.10.25) zur Praktikumsverteilung teil. **Jetzt!**

## Freiwilliges Bonusprojekt - Datenbanken

- Ca. nach Kapitel 7 Veröffentlichung eines Bonusprojekts
- **Inhalt:** Datenbankentwurf und Analyse, Umsetzung mit JavaDB, Auswertung in SQL.
- Abgabe freiwillig, alleine oder in einer kleinen Gruppe.
- Das Projekt (die Abgaben) werden am Vorlesungsende stichprobenartig vorgestellt und besprochen.

# Materialien zur Vorlesung

## Kursmaterialien

- Alle Kursmaterialien finden Sie im Moodle-Kurs.
- **Schreiben Sie sich bitte in den Kurs DBA(Zeume) ein.**

## Klausurvorbereitung

- 3 Termine pro Jahr
- Aufgaben ähnlich der Übungen
- Üben, Üben, Üben!
- Wiederholung zum Ende der Vorlesung
- Fragen stellen! Jederzeit!

# Literatur

- Heuer, Sattler, Saake. Datenbanken: Konzepte und Sprachen.  
mitp-Verlag
- Kemper, Eickler. Datenbanksysteme
- Elmasri, Navathe. Grundlagen von Datenbanksystemen
- Saake, Heuer: Datenbanken - Implementierungstechniken,  
mitp-Verlag, 2005
- Abiteboul, Hull, Vianu. Foundations of Databases. 1995
- Garcia-Molina, Ullman, Widom. Database Systems - The Complete Book. Education International, 2002
- Ramakrishnan, Gehrke. Database Management Systems
- Liu, Özsü. Encyclopedia of Database Systems

Jetzt geht es los!