

# 184.686 VU Datenbanksysteme

## Vorbesprechung und Einführung

Katja Hose

Institut für Logic and Computation

Sommersemester 2024



Informatics

# Wer sind wir

## Personen:

Katja Hose (katja.hose@tuwien.ac.at)

Felix Winter (felix.winter@tuwien.ac.at)

Timo Camillo Merkl (timo.merkl@tuwien.ac.at)

Aleksandar Pavlovic (aleksandar.pavlovic@tuwien.ac.at)

Sowie 21 TutorInnen

# Voraussetzungen

Anmeldung im TISS ([tiss.tuwien.ac.at](https://tiss.tuwien.ac.at))  
bis 11.03.2024, 23:59 Uhr

## STEOP oder nicht STEOP?

- abhängig von Ihrem Studium
- im Zweifel: Anmeldung in TISS möglich?

# Fragen und Kontakt

## Fragen

- Bitte stellen Sie welche! (Wenn Sie welche haben.)

## Kontakt

**TUWEL** Diskussionsforum zum Austausch untereinander  
(d.h. Sie dürfen auch sehr gerne antworten!)

**Email** Bitte ausschließlich an die Adresse  
`dbs-course@list.tuwien.ac.at`  
(Bitte von TU-Email-Adresse schreiben)

**Persönlich** Vor, während, und nach der Vorlesung

# Wichtige Informationsquellen

## **TISS** ([tiss.tuwien.ac.at](http://tiss.tuwien.ac.at))

- Anmeldung zur [LVA](#)
- Anmeldung zu den [Übungsgruppen](#)
- Anmeldung zu den [Prüfungen](#)
- [Aussendungen](#) der LVA Leitung

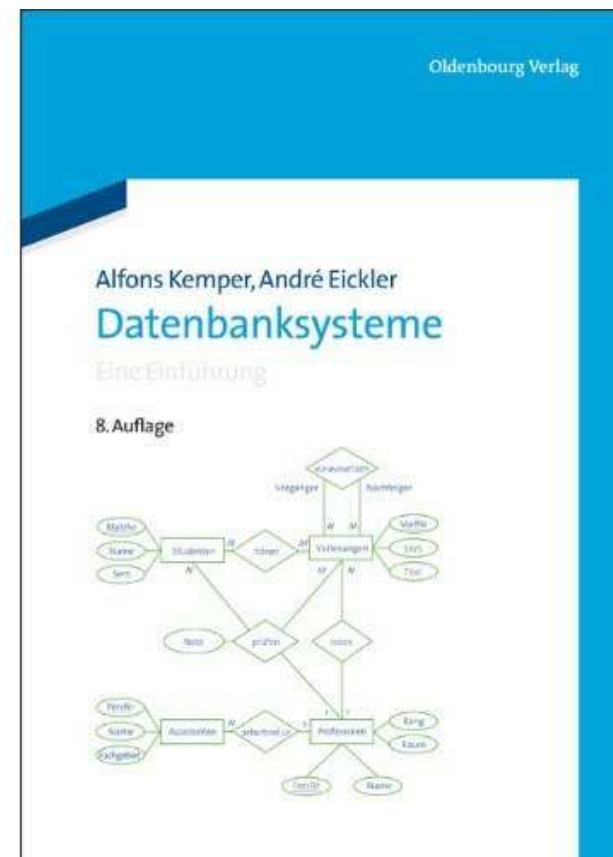
## **TUWEL** ([tuwel.tuwien.ac.at](http://tuwel.tuwien.ac.at))

- [eSQL Lernumgebung](#)
- Zentrale Informationsquelle
- Diskussionsforum
- Vorlesungsunterlagen und Informationen der LVA Leitung

# Lehrbuch

A.Kemper, A.Eickler

- 8. Auflage (9. oder 10.)
- Relationales Modell and Algebra
- SQL
- Datenbankentwurf
- ER Diagramme
- Anfragebearbeitung/-optimierung
- Transaktionen
- ...



# Vorlesung & Übungsmodus

- Vorlesung
  - Montag, 14:00 – 16:00
  - Dienstag, 10:00 – 12:00
- Übungsgruppen
  - Gruppen mit je ca. 40 Studierenden
  - 1,5h-Slots je Gruppe
  - **Verpflichtende** Anwesenheit – Beginn diese Woche
  - Gruppenanmeldung via TISS
  - Donnerstag, 13:00 – 19:00
  - Freitag, 09:00 – 19:00
- Quizzes
  - Online Quizzes im TUWEL Kurs
  - Die ersten zwei Quizzes sind verpflichtend und müssen innerhalb 1 Woche (genaue Daten im TUWEL) abgeschlossen sein!
- SQL Übung
  - Online SQL Übung im TUWEL Kurs

# Vorlesungsmodus

## Vorlesung

- Abstimmungen
  - Interaktive “Abstimmungen” während der Vorlesung
  - Fragen sind nicht immer perfekt, aber Sie werden etwas lernen
- Vorlesungsfolien
  - Die Folien haben Kategorien und Unterkategorien
  - “Mehr Folien sind besser als wenige überladene Folien ohne Beispiele”
  - Die Folien (und auch TUWEL) sind nach Themen organisiert und NICHT nach Vorlesungseinheiten
  - Die Folien **überstimmen** Inhalte aus dem Lehrbuch (im Besonderen bzgl. Notation)!
- Fragen dürfen gerne während der Vorlesung gestellt werden



# Übungsmodus

## Übungen

- Wöchentliche Übungsgruppen (Do und Fr)
- Vertiefen das Verstehen und Anwenden des Lernstoffs aus der Vorlesung
- Möglichkeit Fragen zu stellen und Training für die Prüfungen
- Die richtige Notation verwenden(!) – vermeiden von Punkteabzügen bei der Prüfung
- Übungen in Präsenz unter Aufsicht der TutorInnen

# Übungsmodus

## Quizzes

- Ein Quiz für jede Vorlesung auf TUWEL
- Wir empfehlen das Quiz ein paar Tage nach der Vorlesung zu absolvieren
- Erfolgreicher Abschluss von Quiz 1 & 2 ist Voraussetzung für die weitere Teilnahme
- Quizzes 3–12 liefern je 2 Bonuspunkte und sind freiwillig
- Unbegrenzte Anzahl an Lösungsversuchen
- mindestens 90% korrekte Antworten zum erfolgreichen Abschluss nötig

# Übungsmodus

## SQL Übung

- In eSQL Online Umgebung (Zugriff via TUWEL)
- **freiwillig**
- 10–15 SQL Anfragen sind zu formulieren
- automatisches Feedback durch das System
- mindestens 90% korrekte Anfragen für 5 Punkte

# Prüfungsmodus

## Prüfungen

- 2 schriftliche Prüfungen (in Präsenz)
- 1 Stunde
- Während der Prüfung ist keine Zeit, „die Lösung zu erträumen“
- Kein Buch, Notizen, ChatGPT, elektronische Hilfsmittel, etc.  
Nur Verstand, Stift und Papier
- Max. 100 Punkte pro Prüfung
- Mindestens 50 Punkte auf eine der beiden Prüfungen

## Aufwand – ECTS Breakdown

30h Vorlesungsbesuch

30h Vorlesungsvorbereitung und -nachbereitung

52h Übungsteilnahme, Quizzes, SQL

34h Prüfungsvorbereitung

4h Prüfung

---

150h (= 25h \* 6 ECTS)

# Benotung und Zeugnis

Max. 225 Punkte:

- max. 20 Punkte Quizze (Quizzes 3-12)
- max. 5 Punkte SQL Übung
- max. 100 Punkte Prüfung 1
- max. 100 Punkte Prüfung 2

Für eine positive Note wird benötigt:

- Erfolgreiches Absolvieren der Quizzes 1 & 2
- min. 50 Punkte in einer der beiden Prüfungen
- Insgesamt mindestens 113 Punkte
- Anwesenheit bei allen Übungsterminen

**! Zeugnis** ab Abschluss des ersten Quizzes **!**

# Notenschlüssel

Max. 225 Punkte:

[0 – 113[	Nicht Genügend
[113 – 138]	Genügend
]138 – 168]	Befriedigend
]168 – 196]	Gut
]196 – 225]	Sehr Gut

**! Zeugnis** ab Abschluss des ersten Quizzes **!**

# TUWEL-Kurs

<https://tuwel.tuwien.ac.at/course/view.php?idnumber=184686-2024S>

Bis Anmeldeschluss laufend Import Studierender aus TISS

- eSQL Lernumgebung für SQL-Übungsteil
- Quizzes und SQL Übung
- Diskussionsforum
- Unterlagen und Informationen



# Fragen und Information

**TUWEL** <http://tuwel.tuwien.ac.at/>

**TISS** dringende Aussendungen der LVA Leitung  
Aussendungen der LVA Leitung sind bindend  
⇒ bitte abonnieren

**E-Mail** [dbs-course@list.tuwien.ac.at](mailto:dbs-course@list.tuwien.ac.at)

# Acknowledgements

- Beitrag zu Beispielen, Folien usw.
  - Sebastian Skritek
  - Kristian Torp
  - Hua Lu
  - Sebastian Michel
  - Kai-Uwe Sattler
  - Rasmus Pagh
  - ...
- Lehrmaterial aus Büchern
  - A. Kemper: Datenbanksysteme: Eine Einführung
  - A. Silberschatz, H. Korth, and S. Sudarshan: Database System Concepts
  - ...

# Lernziele

## Dies ist KEIN Kurs, ...

- in dem Sie zum Datenbankadministrator ausgebildet werden
- der Sie zu einem Experten in Oracle, MS SQL, DB2, Postgres oder einem anderen kommerziellen/Open-Source Systemen macht

## Dies IST ein Kurs, ...

- in dem Sie die Grundlagen lernen, die allen relationalen Datenbanksystemen zugrunde liegen
- der Ihnen hilft, relationale Datenbanksysteme (besser) nutzen zu können

# Lernziele

- Das relationale Modell verstehen und relationale Algebra anwenden
- Konzeptuelles Design einer Datenbank (ER-Modell, konzeptuelles Design)
- Erstellen und Bewerten eines relationalen Datenbankschemas und Normalisierung (logisches Design)
- Verwenden von SQL zum Erstellen, Modifizieren und Anfragen relationaler Datenbanken
- Verstehen, wie Daten physisch organisiert sind (physisches Design)
- Interpretieren von Anfrageausführungsplänen und Bewertung deren Effizienz
- Verständnis des Transaktionskonzepts und einiger Grundkonzepte (Concurrency Control und Recovery)

- 1 Organisation
  - Administrative Informationen
  - Voraussetzungen
  - Fragen und Kontakt
- 2 Ablauf
  - Wichtige Informationsquellen
  - Unterlagen
  - Vorlesung
  - Aufwand
  - Benotung und Zeugnis
  - TUWEL-Kurs
  - Lernziele
- 3 Motivation
  - Wo kommen Daten vor?
  - Probleme mit Daten
  - SQL

- Architektur
- Verfügbare Systeme

- 4 Vorlesungsinhalt
  - Vorlesungsüberblick
  - Die Vorlesung im Schnelldurchlauf
- 5 Erste Aufgaben: Quizzes und Übung

# Computerprogramme

```
int a = 42;  
int myArray[] = {0,7,8,12,43};
```

```
class Student {  
    int ID;  
    String firstname;  
    String lastname;  
    Date birthdate;  
    String dept_name;  
    double tot_cred;  
    ...  
}
```

```
Student kristianPedersen = new Student  
    (1234, "Kristian", "Pedersen", "7.7.1990",  
     "Computer Science", 20, ...);
```

# Computerprogramme

```
List<Student> allStudents = new LinkedList<Student>(100);

allStudents.add(
    new Student(1234, "kristian", "pedersen", "7.7.1990",
        "computer science", 20)
);

allStudents.add(
    new Student(1235, "tim", "pedersen", "32.4.2000",
        "computer science", 15)
);

allStudents.add(
    new Student(1235, "tom", "jensen", "11.3.2017",
        "Dept. of Computer Science", -20)
);
```



# Computerprogramme

```
File studFile = new File("/home/hose/lecture/allStudents.bin");

studFile.createNewFile();

FileOutputStream studFileOS =
    new FileOutputStream(studFile);

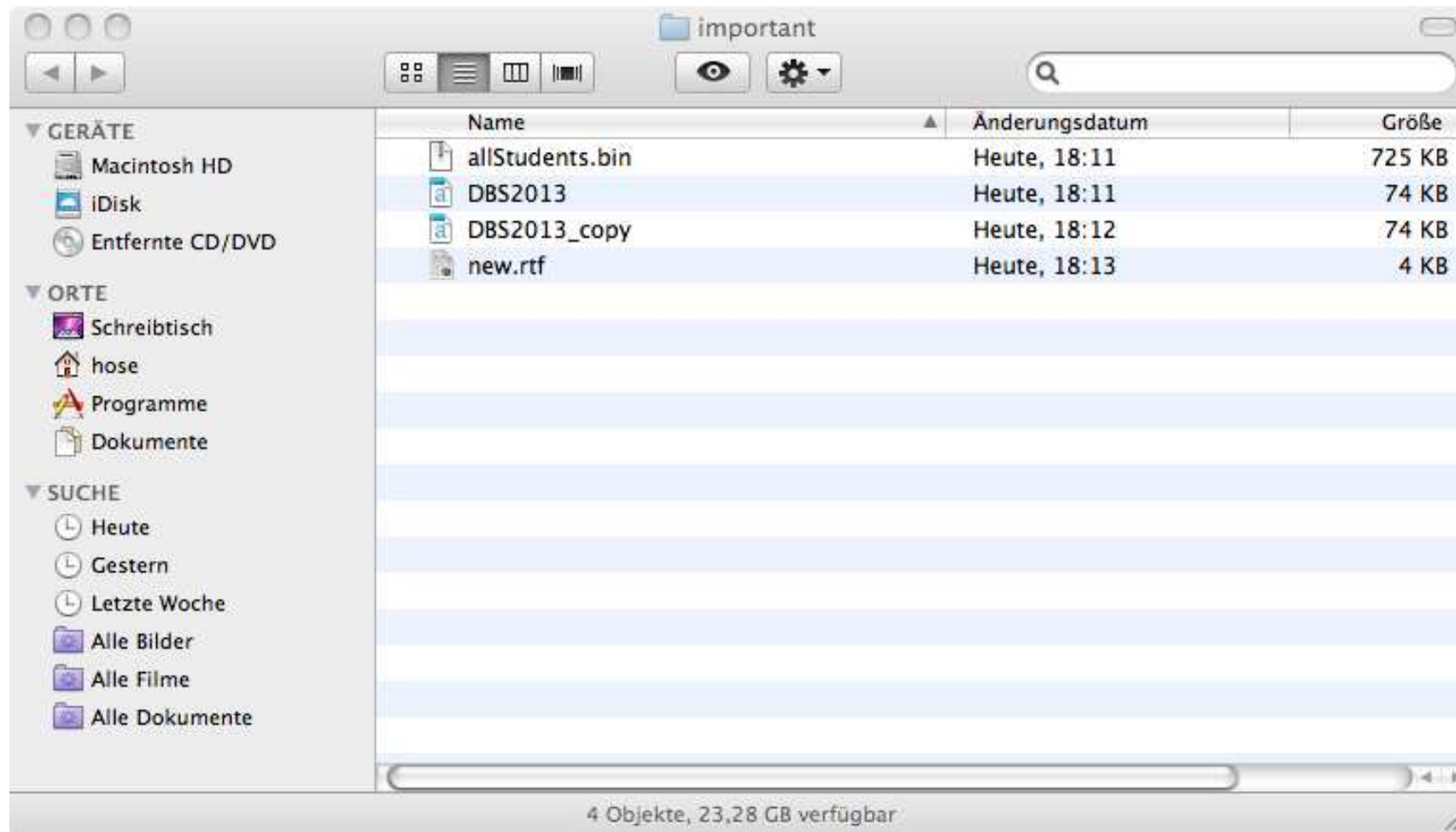
for (Student student : allStudents) {

    //write all student data into a file
    studFileOS.write(...);

}

studFile.close();
```

# Dateisysteme



... auf dem Handy...



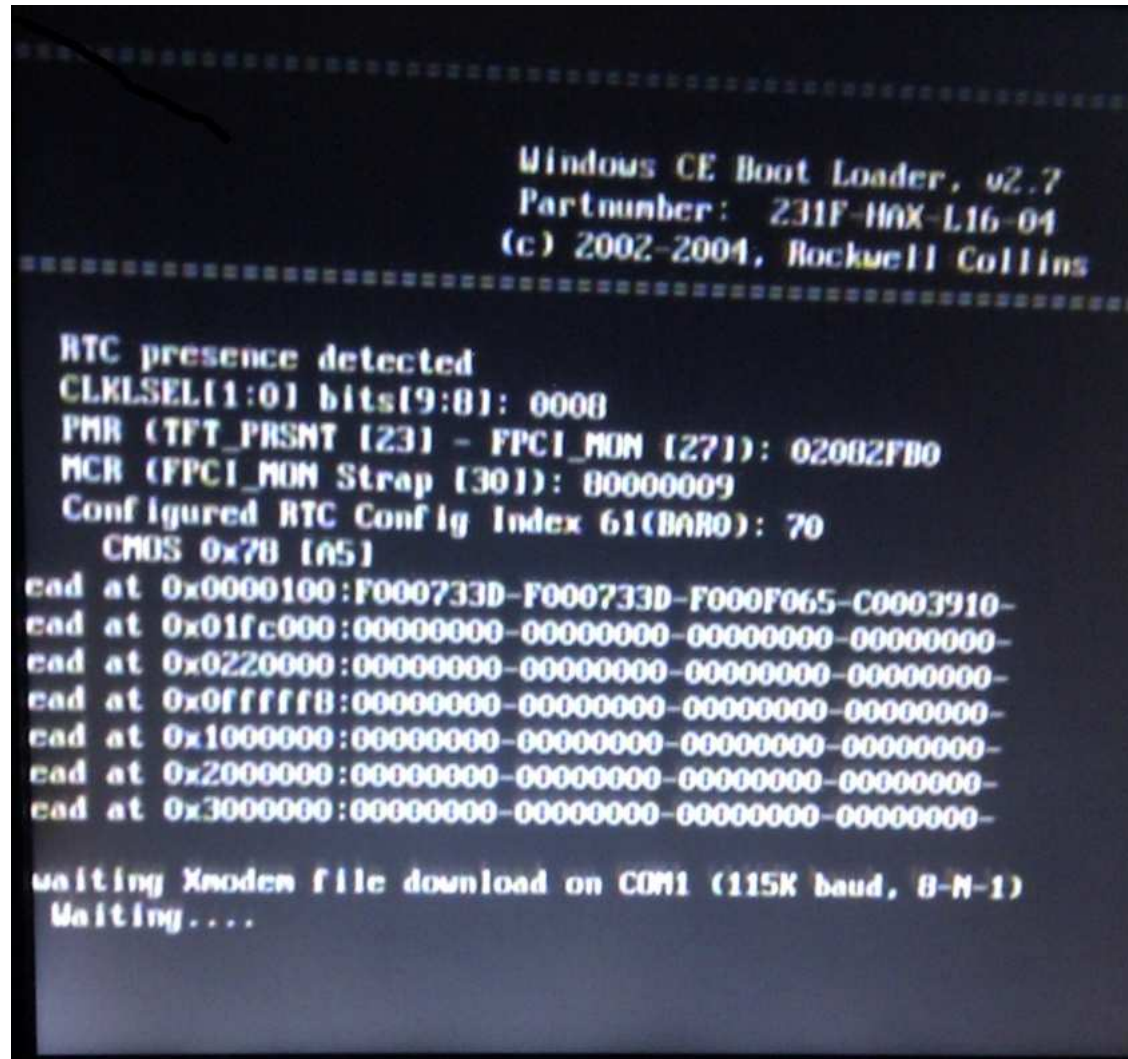
<http://www.apple.com/>

...in Flugzeugen...



<http://www.wikipedia.org/>

...in Flugzeugen...



```
Windows CE Boot Loader, v2.7
Partnumber: 231F-HAX-L16-04
(c) 2002-2004, Rockwell Collins

RTC presence detected
CLKSEL[1:0] bits[9:8]: 0000
PMR (TFT_PRST [23] - FPCI_MON [27]): 020B2F00
MCR (FPCI_MON Strap [30]): 00000009
Configured RTC Config Index 61(BARO): 70
CMOS 0x78 (A5)
eas at 0x0000100:F000733D-F000733D-F000F065-C0003910-
eas at 0x01fc000:00000000-00000000-00000000-00000000-
eas at 0x0220000:00000000-00000000-00000000-00000000-
eas at 0x0fffff8:00000000-00000000-00000000-00000000-
eas at 0x1000000:00000000-00000000-00000000-00000000-
eas at 0x2000000:00000000-00000000-00000000-00000000-
eas at 0x3000000:00000000-00000000-00000000-00000000-

waiting Xmodem file download on COM1 (115K baud, 8-N-1)
Waiting....
```

... in Autos...



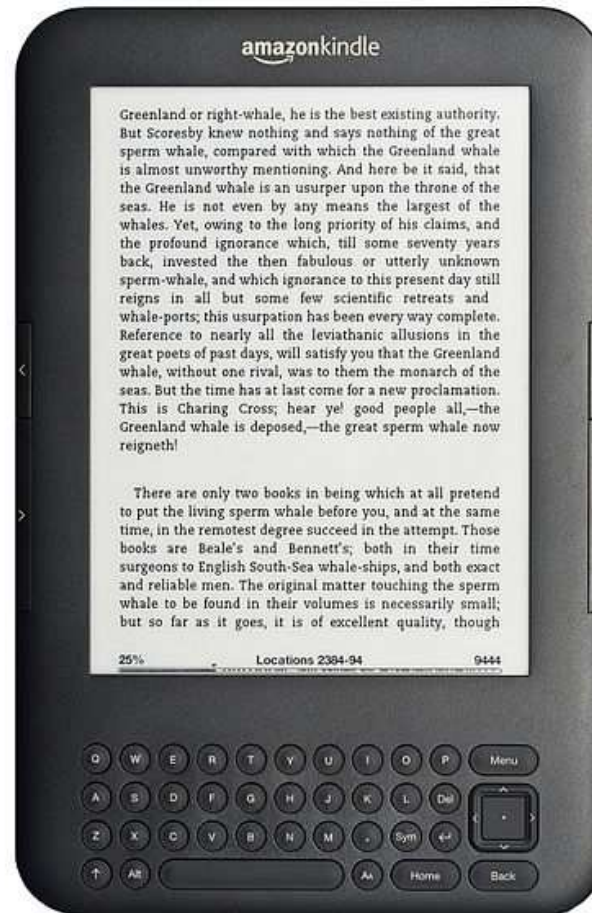
...in Kameras...



<http://www.wikipedia.org/>



# ... in Büchern...



<http://www.wikipedia.org/>



... in medizinischen Geräten...



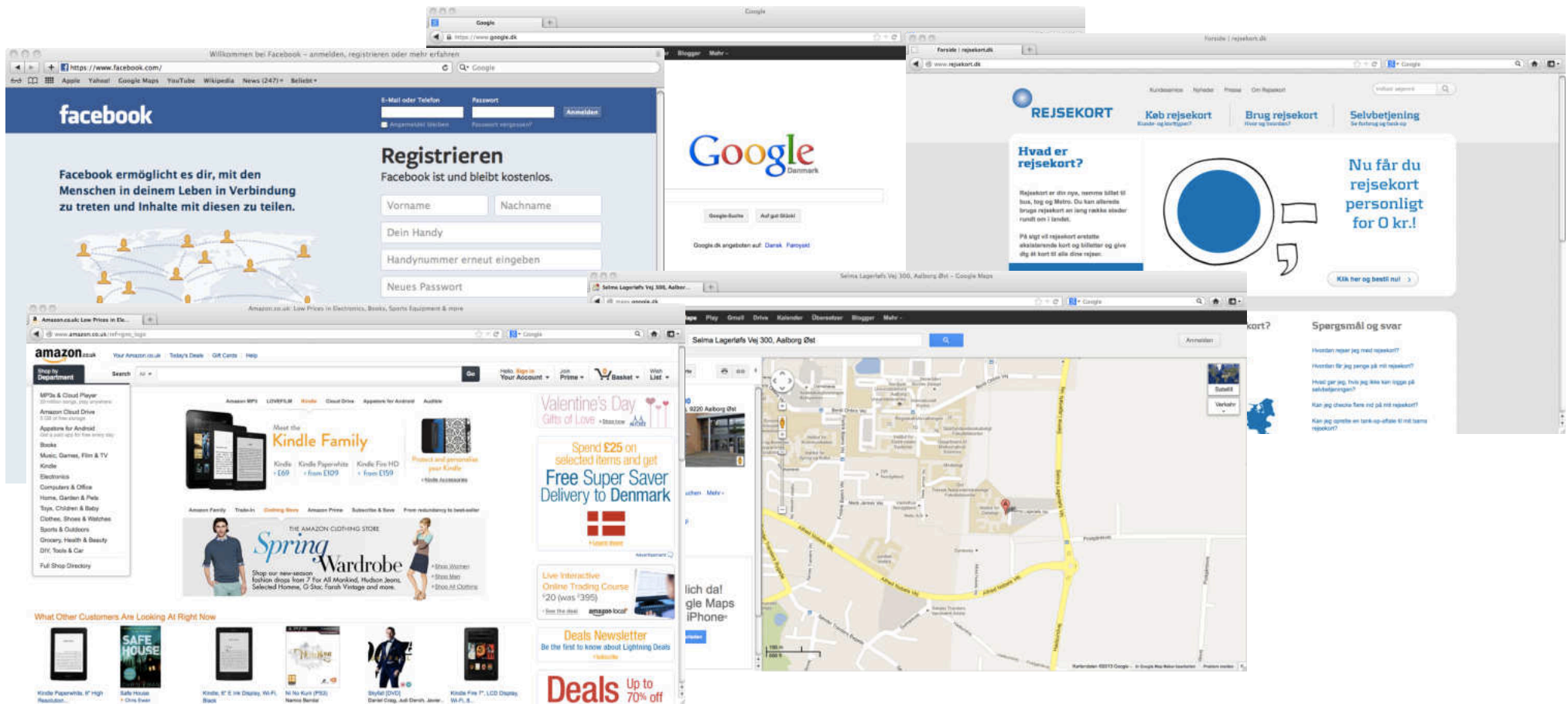
<http://www.wikipedia.org/>

...im/am Fahrrad...



Wo kommen Daten vor?

... in vielen anderen Anwendungen...



... in vielen anderen Anwendungen...

- Banking: Transaktionen
- Airlines: Reservierungen, Schedules
- Universitäten: Vorlesungen, Studierende, Lehrende
- Sales: KundInnen, Produkte, Käufe
- Online Retailers: personalisierte Kaufempfehlungen
- Manufacturing: Produktion, Lager, Bestellungen, Supply Chains
- Human Resources: Angestellte, Gehälter, Steuern



Datenbanken kommen überall vor...

<http://www.wikipedia.org/>

- 1 Organisation
  - Administrative Informationen
  - Voraussetzungen
  - Fragen und Kontakt
- 2 Ablauf
  - Wichtige Informationsquellen
  - Unterlagen
  - Vorlesung
  - Aufwand
  - Benotung und Zeugnis
  - TUWEL-Kurs
  - Lernziele
- 3 Motivation
  - Wo kommen Daten vor?
  - Probleme mit Daten
  - SQL

- Architektur
- Verfügbare Systeme

- 4 Vorlesungsinhalt
  - Vorlesungsüberblick
  - Die Vorlesung im Schnelldurchlauf
- 5 Erste Aufgaben: Quizzes und Übung

## Mögliche Probleme mit Daten

```
Student timPedersen = new Student  
(1235, "tim", "pedersen", "32.4.2000", "computer science", 15);
```

```
Student tomJensen = new Student  
(1235, "tom", "jensen", "11.3.2017", "Dept. of Computer Science", -20);
```

id, first name, last name, birthdate, department, ECTS

Gibt es hier potentiell Probleme?



# Mögliche Probleme mit Daten

Student timPedersen = new Student

(**1235**, "tim", "pedersen", "**32**.4.2000", "computer science", 15);

Student tomJensen = new Student

(**1235**, "tom", "jensen", "11.3.**2017**", "**Dept. of Computer Science**", **-20**);

id, first name, last name, birthdate, department, ECTS

## Mögliche Probleme

- Mehrere Studierende haben die gleiche ID 1235
- 32.4.2000 kein gültiges Datum
- 11.3.2017 sehr jung für einen Studierenden
- Sind "computer science" und "Dept. of Computer Science" dasselbe?
- Ist "-20" eine gültige Anzahl von ECTS?



# Computerprogramme und Dateien

```
Student timPedersen = new Student  
(1235, "tim", "pedersen", "32.4.2000", "computer science", 15);
```

```
Student tomJensen = new Student  
(1235, "tom", "jensen", "11.3.2017", "Dept. of Computer Science", -20);
```



<http://www.wikipedia.org/>

## Stromausfall

- Wurden beide Studierende auf Festplatte geschrieben?
- ...oder nur 1,5?
- Möglicherweise ist die Festplatte sogar beschädigt worden
- Wie komme ich an die aktuellen Daten?

# Computerprogramme und Dateien

```
Student timPedersen = new Student  
(1235, "tim", "pedersen", "32.4.2000", "computer science", 15);
```

```
Student tomJensen = new Student  
(1235, "tom", "jensen", "11.3.2017", "Dept. of Computer Science", -20);
```

... 343,323,111 weitere Studierende werden eingefügt

## Probleme

- Die Daten passen nicht mehr alle gleichzeitig in den Hauptspeicher.
- Wenn ich Daten ändere, müssen sie auch auf der Festplatte geändert werden.
- Effiziente Anfragebearbeitung (Suche, Sortieren,...)
- Robuste und ausgereifte Anfragesprache
- Zugriffsrechte
- Zusätzliche Informationen abspeichern
- ...

# Typische Motivation für DBMS

Am Anfang...

1 Excel-Sheet, 1 Benutzer

Dann...

mehr und mehr Excel-Sheets, mehr und mehr Benutzer

Tausende von Corona-Testergebnissen durch Excel-Nutzung verloren (UK)

BBC (Oktober 2020):

- Covid: Test error 'should never have happened'<sup>a</sup>
- Excel: Why using Microsoft's tool caused Covid-19 results to be lost<sup>b</sup>

---

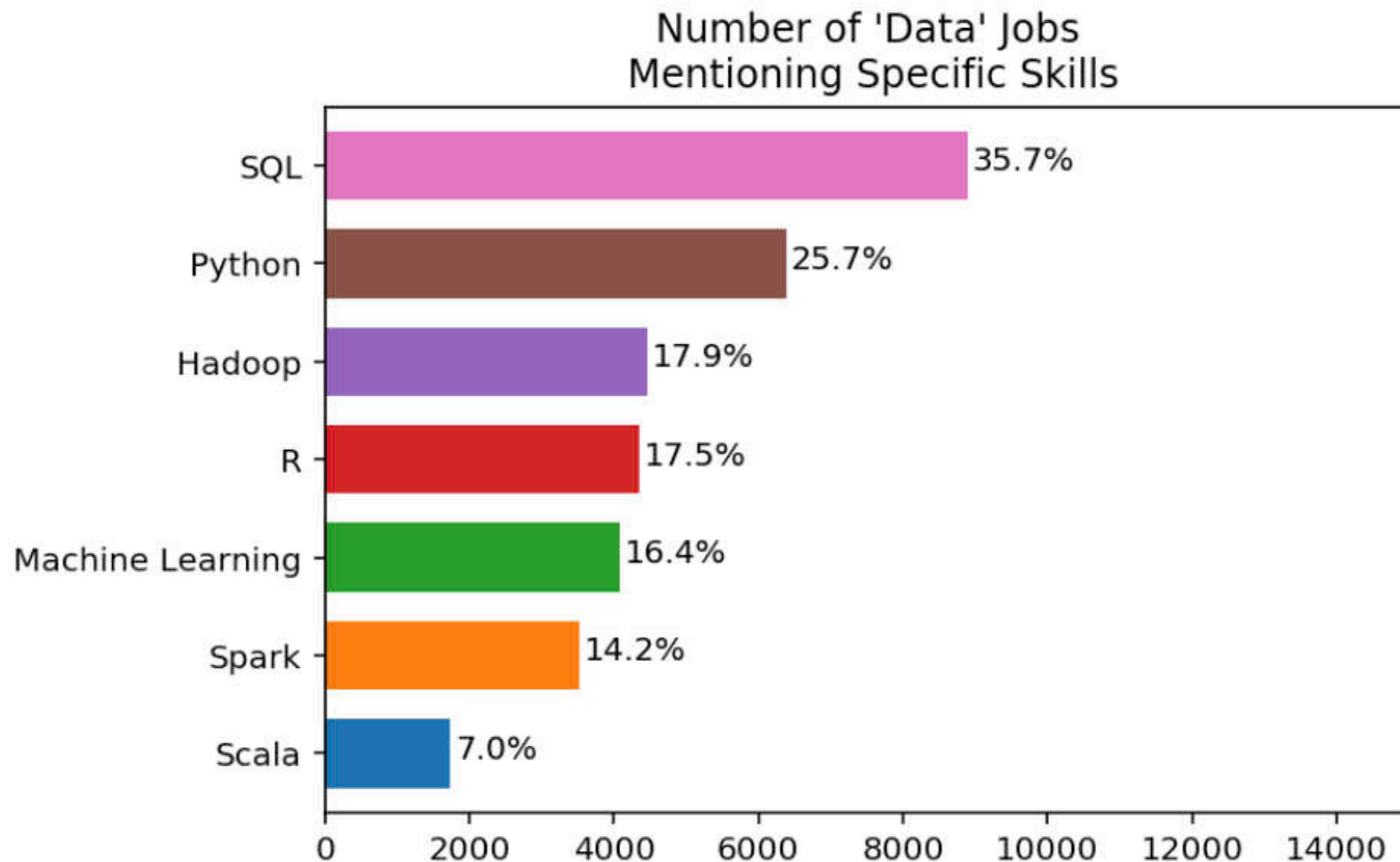
<sup>a</sup><https://www.bbc.com/news/uk-54422505>

<sup>b</sup><https://www.bbc.com/news/technology-54423988>

# Datenbanksysteme

Datenbanksysteme bieten Lösungen für all diese Probleme.

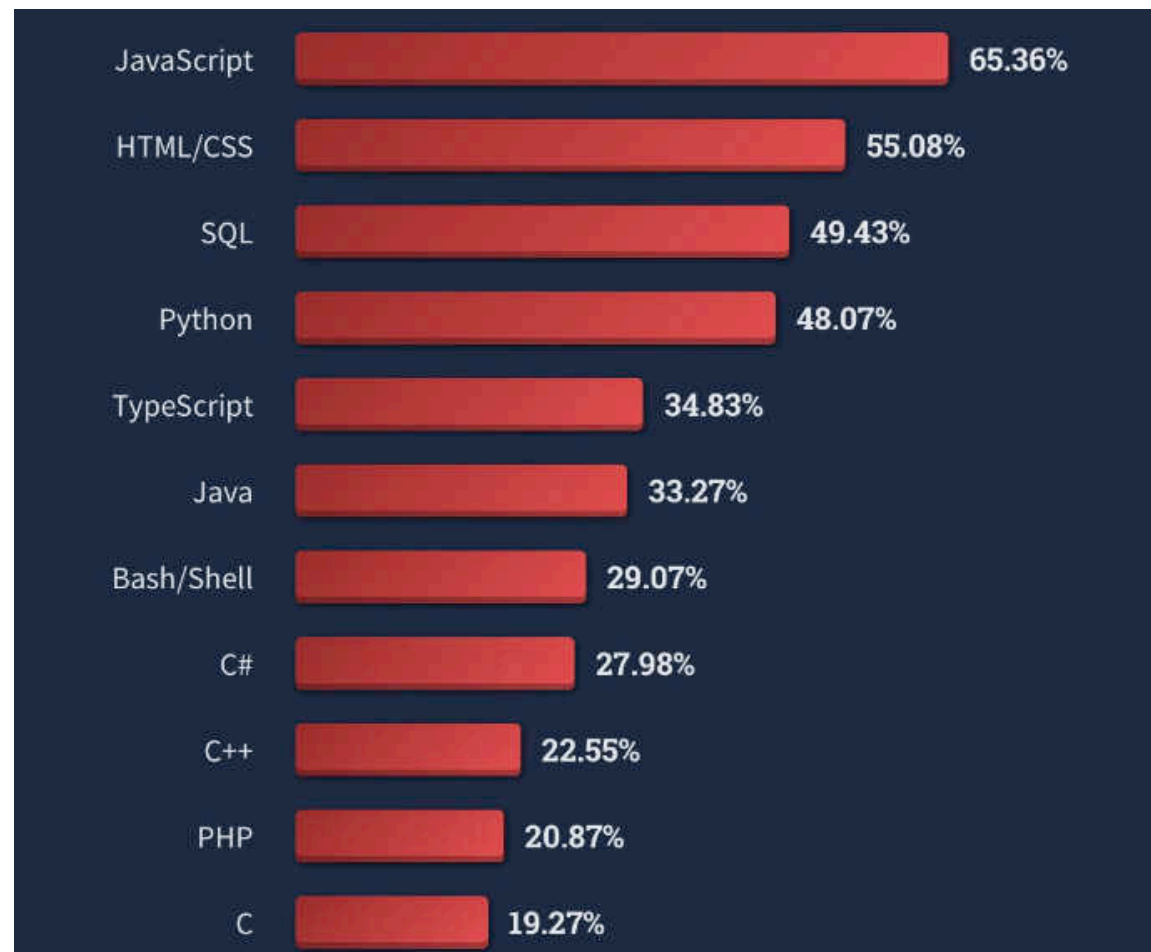
# SQL ist überall und in ständiger Nachfrage



Source: indeed.com

# SQL ist überall und in ständiger Nachfrage

Top 10 most popular languages



Stack Overflow Developer Survey 2022

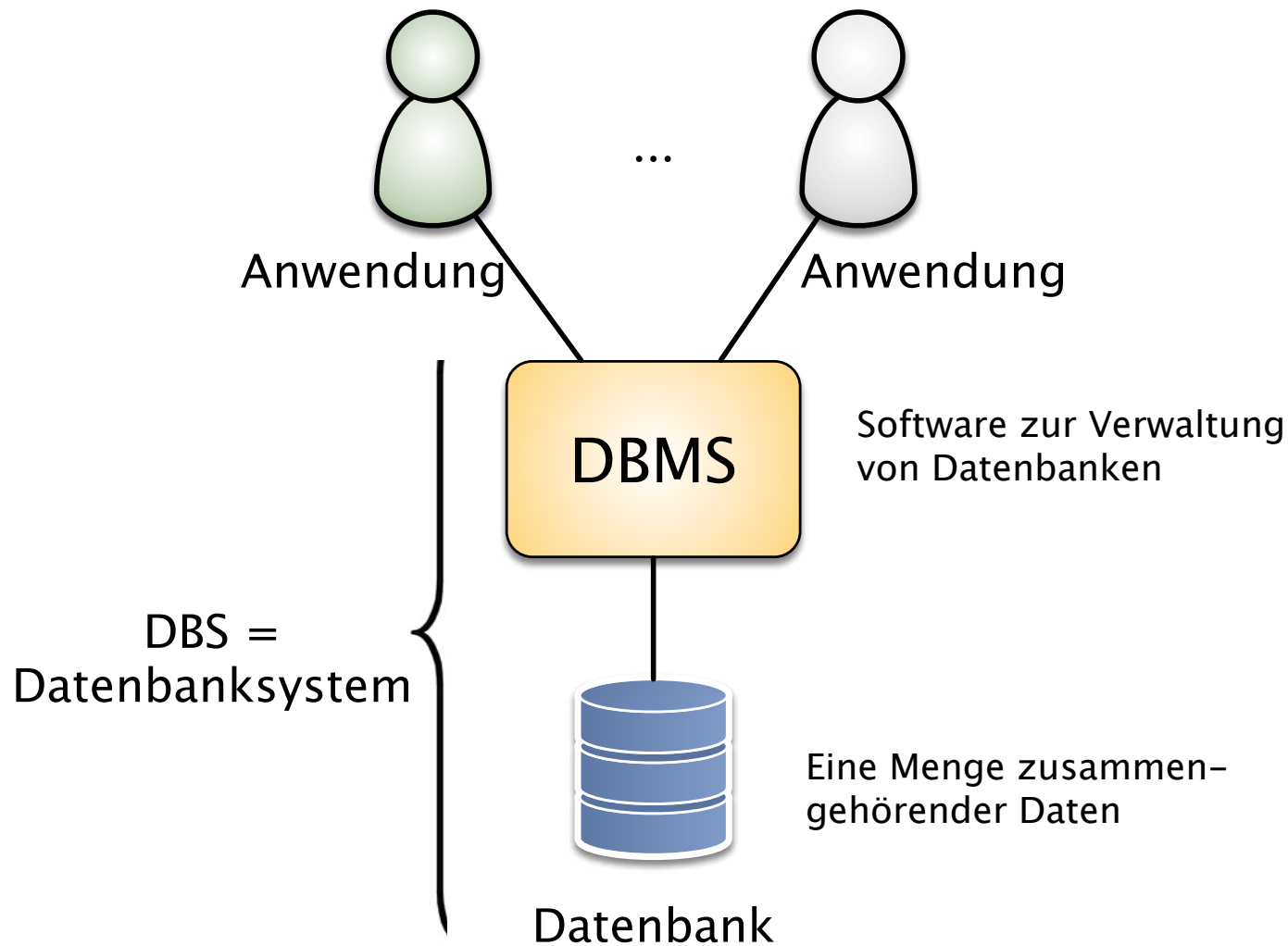
- 1 Organisation
  - Administrative Informationen
  - Voraussetzungen
  - Fragen und Kontakt
- 2 Ablauf
  - Wichtige Informationsquellen
  - Unterlagen
  - Vorlesung
  - Aufwand
  - Benotung und Zeugnis
  - TUWEL-Kurs
  - Lernziele
- 3 Motivation
  - Wo kommen Daten vor?
  - Probleme mit Daten
  - SQL

- Architektur
- Verfügbare Systeme

- 4 Vorlesungsinhalt
  - Vorlesungsüberblick
  - Die Vorlesung im Schnelldurchlauf
- 5 Erste Aufgaben: Quizzes und Übung



# Datenbanksysteme



# Anwendungsbeispiel: Music Store

The screenshot shows a music store interface for the album "Living With War" by Neil Young. Red annotations highlight the following database entities:

- Musiker**: Points to the artist name "Neil Young".
- Titel**: Points to the album title "Living With War".
- Jahr**: Points to the release date "Erschienen 02. Mai 2006".
- Preis**: Points to the album price "9,99 €".
- Rezension(en)**: Points to the album review section titled "ALBENREZENSION".
- Tracks**: Points to the track listing table at the bottom.

The track listing table contains the following data:

Titel	Dauer	Interpret	Album	Preis
1 After the Garden	3:21	Neil Young	Living With War	0,99 €
2 Living With War	5:03	Neil Young	Living With War	0,99 €
3 The Restless Consumer	5:46	Neil Young	Living With War	0,99 €
4 Shock and Awe	4:51	Neil Young	Living With War	0,99 €
5 Families	2:24	Neil Young	Living With War	0,99 €
6 Flags of Freedom	3:40	Neil Young	Living With War	0,99 €
7 Let's Impeach the President	5:07	Neil Young	Living With War	0,99 €
8 Lookin' for a Leader	4:02	Neil Young	Living With War	0,99 €
9 Roger and Out	4:23	Neil Young	Living With War	0,99 €
10 America the Beautiful	2:56	Neil Young	Living With War	0,99 €

# 3-Ebenen-Architektur

Konzeptuelle Sicht:

*Darstellung in Tabellen (Relationen)*

MusikerIn	<u>MNr</u>	Name	Land
	103	Apocalyptica	Finland
	104	Subway To Sally	Germany
	105	Teräsbetoni	Finland

Album	<u>ANr</u>	Title	Jahr	Genre	<u>MNr</u> → MusikerIn
	1014	Amplified	2006	rock	103
	1015	Nord Nord Ost	2005	rock	104
	1016	Vaadimme Metallia	2008	metal	105
	1021	Engelskrieger	2003	rock	104
	1025	Reflections	2006	rock	103

# 3-Ebenen-Architektur

Externe Sicht:

*Daten in einer flachen Relation*

<u>ANr</u>	Titel	Jahr	Genre	MusikerIn
1014	Amplified	2006	Rock	Apocalyptica
1015	Nord Nord Ost	2005	Rock	Subway To Sally
1016	Vaadimme Metallia	2008	Metal	Teräsbetoni
1021	Engelskrieger	2003	Rock	Subway To Sally
1025	Reflections	2006	Rock	Apocalyptica

# 3-Ebenen-Architektur

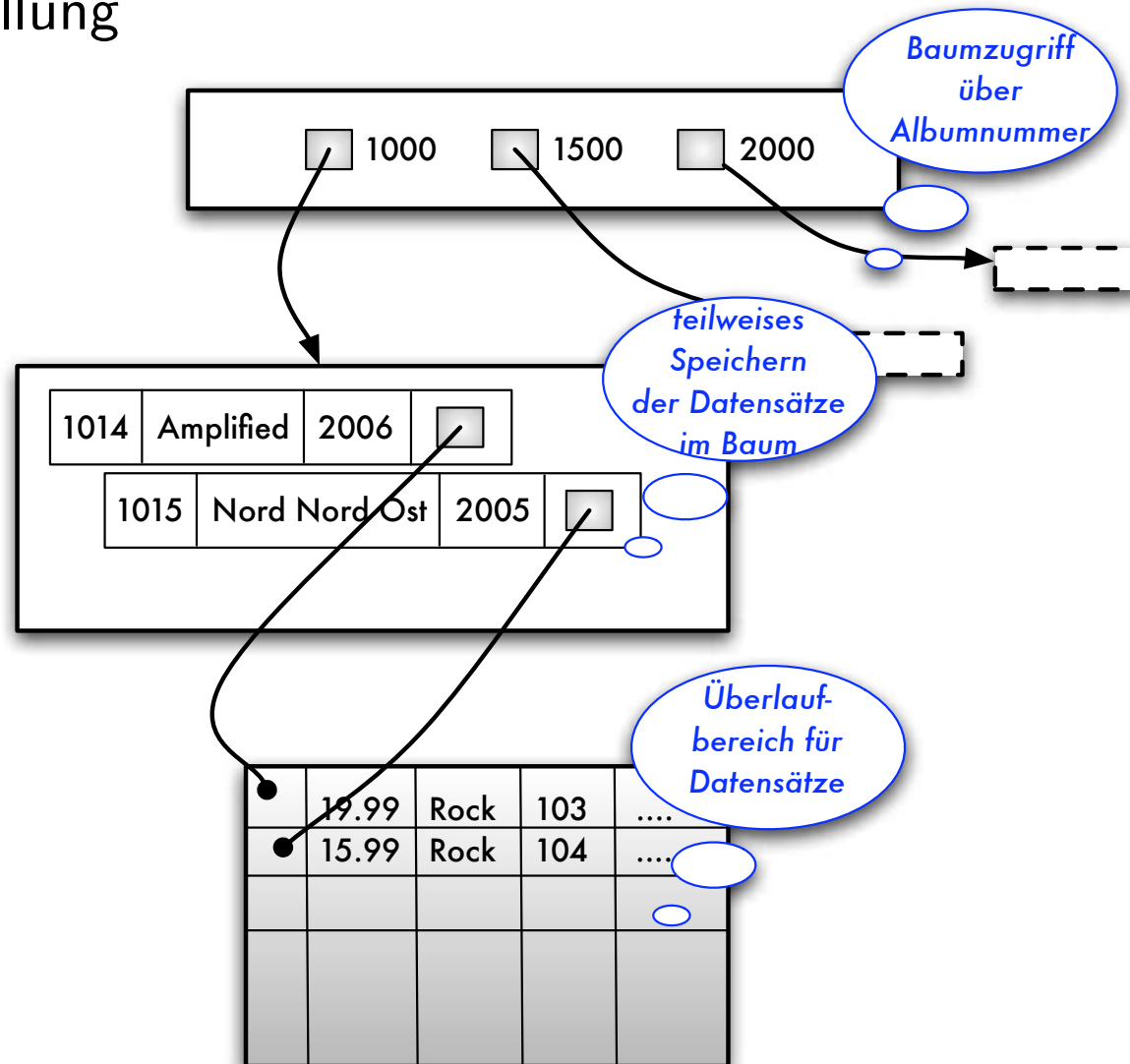
External schema:

*Daten in einer hierarchisch aufgebauten Relation*

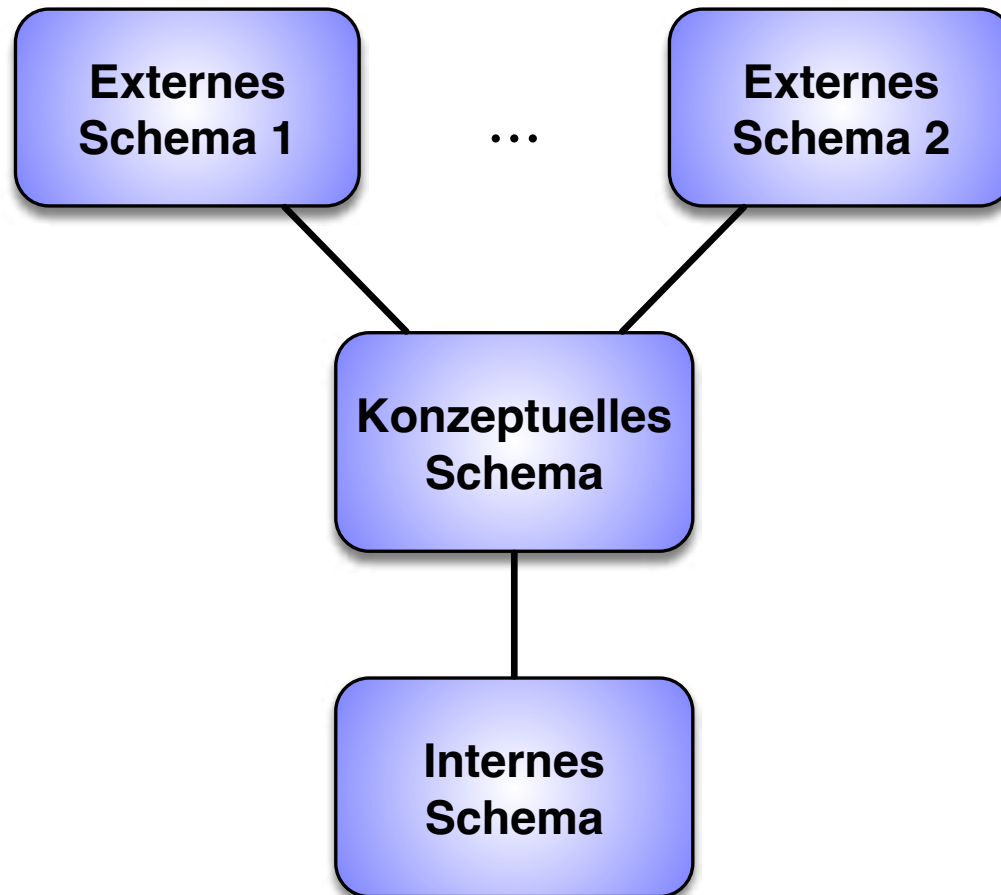
MusikerIn	Album		
	Titel	Jahr	Genre
Apocalyptica	Amplified	2006	Rock
	Reflections	2003	Rock
Subway To Sally	Nord Nord Ost	2005	Rock
	Engelskrieger	2003	Rock
Teräsbetoni	Vaadimme Metallia	2008	Metal

# 3-Ebenen-Architektur

## Interne Darstellung



## 3-Ebenen-Architektur



**Benutzersichten**

**Logische Ebene**

**Physische Ebene**

# Physische Datenunabhängigkeit

Änderungen der Dateiorganisationen und Zugriffspfade (physische Ebene) haben keinen Einfluss auf das konzeptuelle Schema (logische Ebene).

Beispiele von Änderungen auf physischer Ebene

- Neue Festplatte wird hinzugefügt
- Neuer Prozessor wird hinzugefügt
- Dateien werden in mehrere Dateien aufgeteilt



# Logische Datenunabhängigkeit

Änderungen auf der logischen Ebene haben keine Auswirkungen auf externe Schemata und Anwendungsprogramme.

Beispiel für Änderungen auf logischer Ebene

- Ein weiteres Attribut im konzeptuelles Schema hinzufügen  
→ Benutzer sehen immer noch dieselben Attribute wie zuvor
- Ändern des Namens eines Attributs auf der konzeptuellen Ebene  
→ In der Sicht des Benutzers hat das Attribut weiter denselben Namen

# Definitionen

- Mini-Welt
  - Ein Teil der realen Welt, über den Informationen gespeichert werden
- Daten/Information
  - Bekannte Fakten über die Mini-Welt, die aufgezeichnet werden können und eine implizite Bedeutung haben
- Datenbank (DB)
  - Eine Menge zusammengehörender Daten
- Datenbankmanagementsystem (DBMS)
  - Software zur Verwaltung von Datenbanken
- Datenbanksystem (DBS)
  - Eine Datenbank und ein DBMS
- Datenbankinstanz
  - Der Inhalt einer DB zu einem konkreten Zeitpunkt

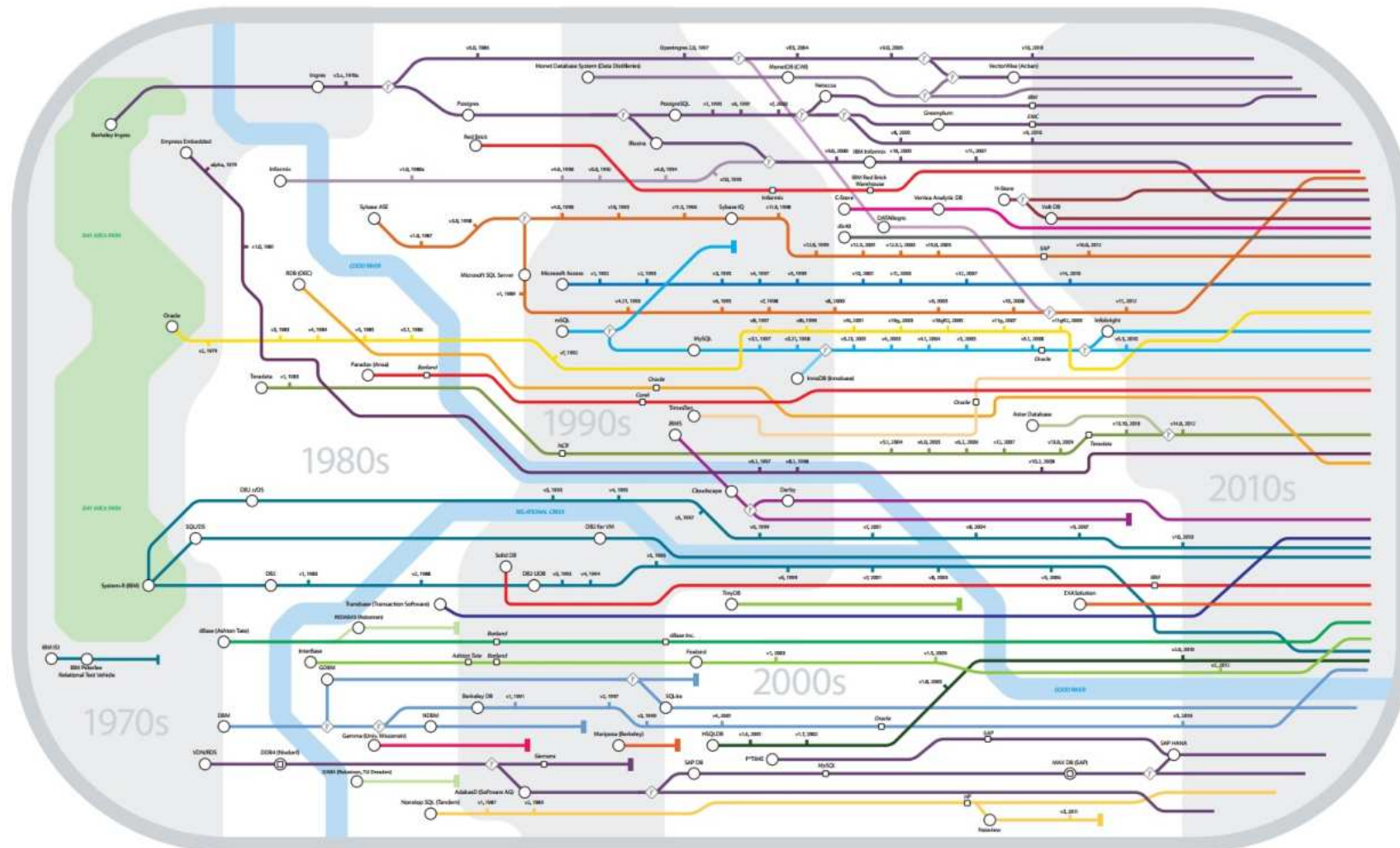
- 1 Organisation
  - Administrative Informationen
  - Voraussetzungen
  - Fragen und Kontakt
- 2 Ablauf
  - Wichtige Informationsquellen
  - Unterlagen
  - Vorlesung
  - Aufwand
  - Benotung und Zeugnis
  - TUWEL-Kurs
  - Lernziele
- 3 Motivation
  - Wo kommen Daten vor?
  - Probleme mit Daten
  - SQL

- Architektur
- Verfügbare Systeme

- 4 Vorlesungsinhalt
  - Vorlesungsüberblick
  - Die Vorlesung im Schnelldurchlauf
- 5 Erste Aufgaben: Quizzes und Übung

# Ahnentafel – Relationale Datenbankmanagementsysteme

Genealogy of Relational Database Management Systems



# Populäre Datenbanksysteme

## Die Großen

- Oracle
- Microsoft SQL Server
- IBM DB2

## Die Kleinen

- PostgreSQL
- MySQL
- ...

Kurzes Video "History of databases"

<https://www.youtube.com/watch?v=KG-mqHoX0XY>

# Zum selbst Ausprobieren...

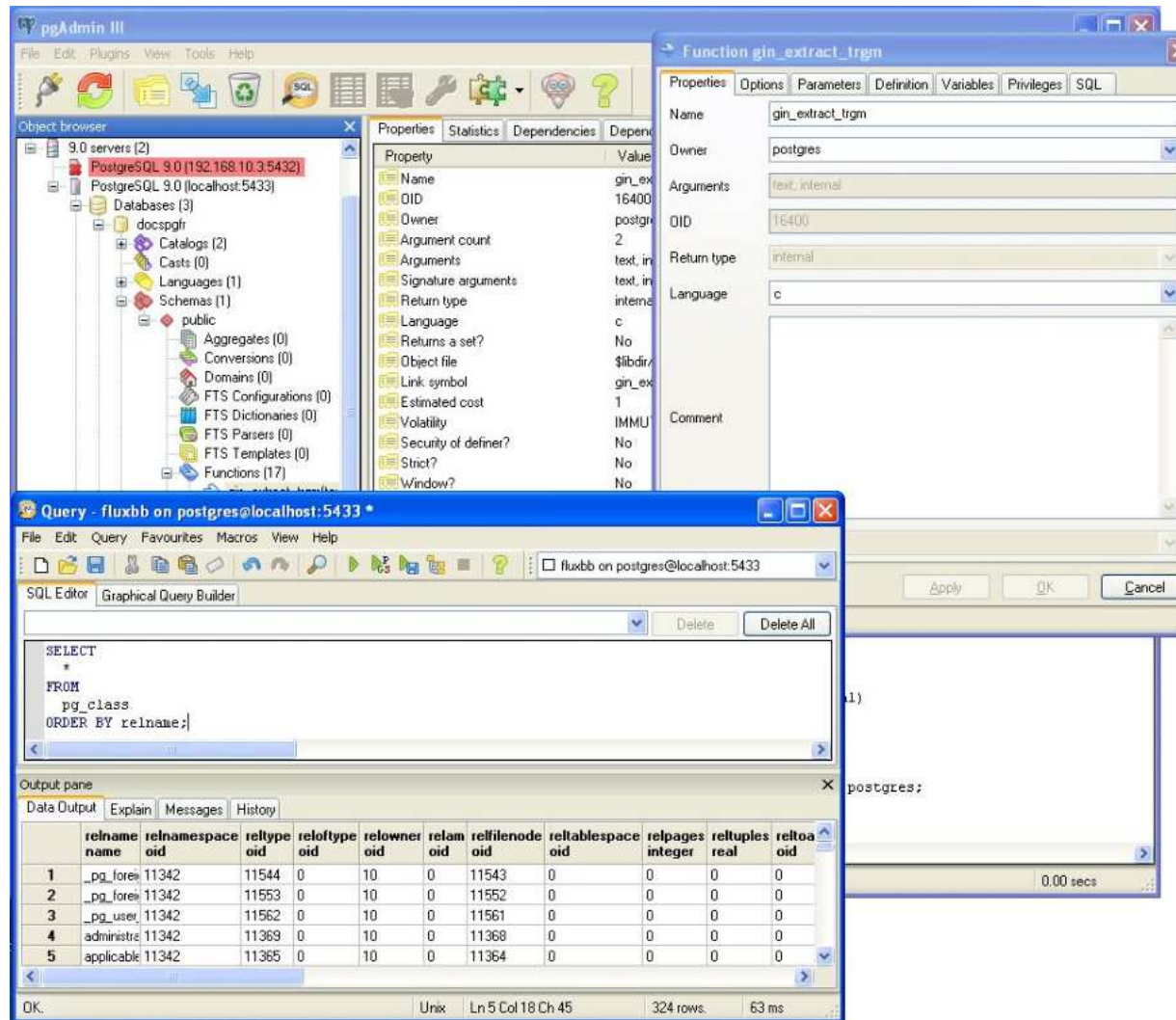
## PostgreSQL

- “The world’s most advanced open source database”
- Einfach zu Installieren
- Gute SQL Unterstützung
- Transaktionen
- Gute Dokumentation (es gibt auch Bücher dazu)



<http://www.postgresql.org/>

# PGAdmin3 user interface



<http://www.pgadmin.org/>



- 1 Organisation
  - Administrative Informationen
  - Voraussetzungen
  - Fragen und Kontakt
- 2 Ablauf
  - Wichtige Informationsquellen
  - Unterlagen
  - Vorlesung
  - Aufwand
  - Benotung und Zeugnis
  - TUWEL-Kurs
  - Lernziele
- 3 Motivation
  - Wo kommen Daten vor?
  - Probleme mit Daten
  - SQL

- Architektur
- Verfügbare Systeme

- 4 Vorlesungsinhalt
  - Vorlesungsüberblick
  - Die Vorlesung im Schnelldurchlauf

- 5 Erste Aufgaben: Quizzes und Übung

# Kernfragen

## Kernfragen

- ❶ Wie organisiert (modelliert und verwendet) man Daten?
- ❷ Wie speichert man Daten zuverlässig und dauerhaft?
- ❸ Wie verarbeitet man riesige Mengen an Daten effizient?
- ❹ Wie stellt man parallelen Zugriff für viele Benutzer zur Verfügung?

# Die Vorlesung im Schnelldurchlauf

- Relationales Modell
- Relationale Algebra
- Entity Relationship Modell (ER Modell)
- Datenbankentwurf
- SQL
- Transaktionen
- Recovery
- Anfragebearbeitung und -optimierung
- ...

# Am Anfang. . .

Abbilden eines Teilaspekts der realen Welt (Mini-Welt)

- Was wollen wir Abbilden?
- Welcher Grad an Details?
- Welche Entitäten spielen eine Rolle?
- Und welche Rolle spielen sie?
- Wie sind Entitäten miteinander verbunden?

# Anforderungsanalyse

Reale Welt: Universität → Anforderungsanalyse

Pflichtenheft

- “Studierende hören Vorlesungen”
- “ProfessorInnen halten Vorlesungen”
- “Studierende könnten anhand ihrer Matrikelnummer eindeutig identifiziert werden”
- ...

# Anforderungsanalyse

- Identifikation von Organisationseinheiten
- Identifikation von Beziehungen
- Identifikation von Prozessen
- Formalisierung
- Ziel: Anforderungskatalog/Pflichtenheft

# Entity-Relationship-Modellierung



How the customer  
explained it



How the Project  
Leader understood it

<http://www.fuki.ch/>



# Entity-Relationship-Modellierung



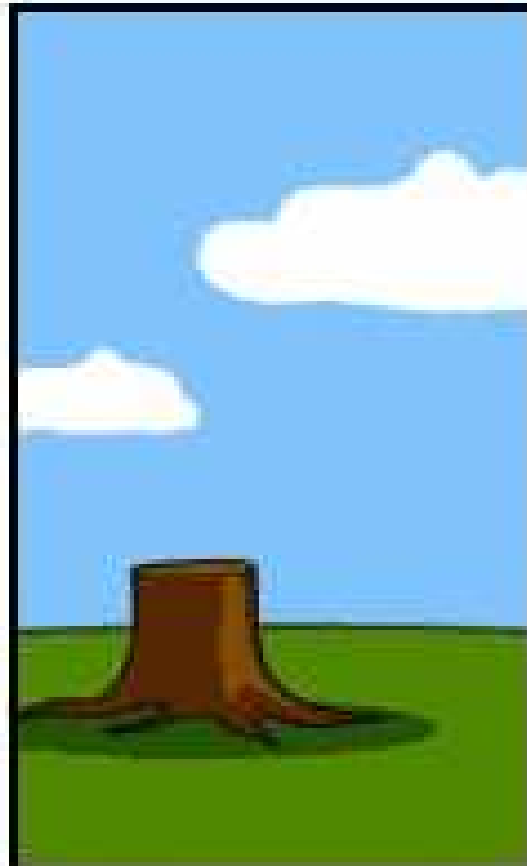
How the Analyst  
designed it



How the Programmer  
wrote it

<http://www.fuki.ch/>

# Entity-Relationship-Modellierung



How it was supported

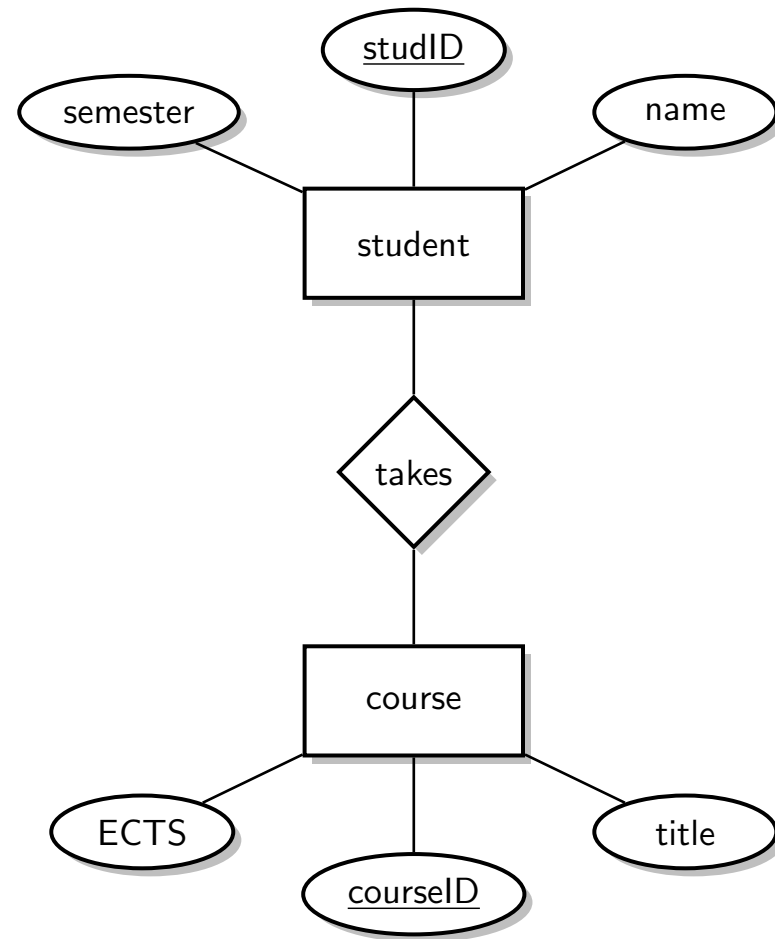


What the customer  
really needed

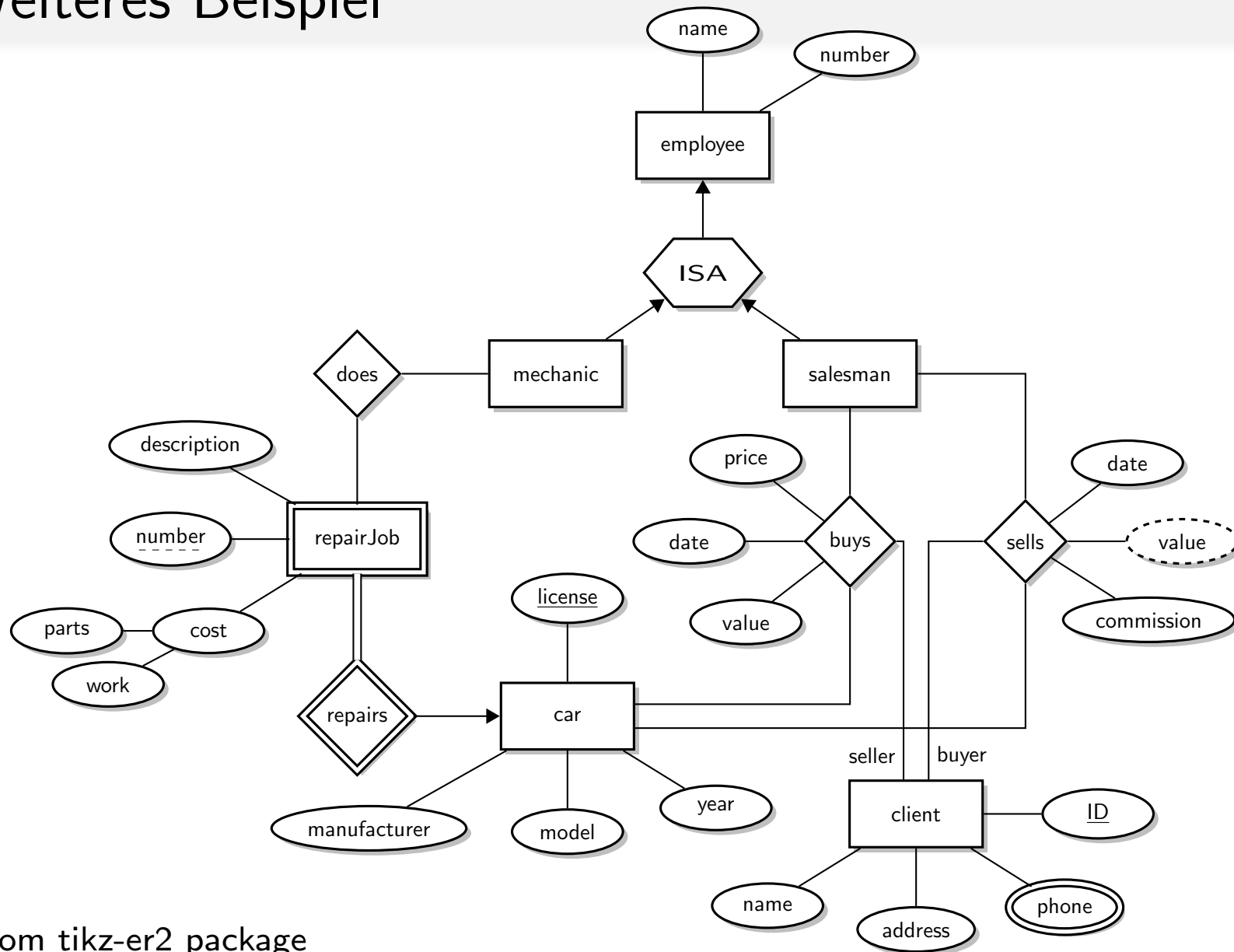
<http://www.fuki.ch/>

# Entity-Relationship-Modellierung

1. Entität → Entitätstyp
2. Beziehung → Beziehungstyp
3. Attribut (Eigenschaft)
4. Schlüssel (Identifikation)
5. Rolle



# Weiteres Beispiel



example from tikz-er2 package

# Relationales Modell

Abbilden des ER-Modells in Relationen

Zum Beispiel:

- **client**(ID, phone, address, name)
- **purchase**(value, data, price, car, client)
- ...

oder...

- **student**(studentID, name)
- **attend**(studentID, courseID)
- **lecture**(courseID, title)

# Anfragen im relationalen Modell

## Relationale Algebra

- Selektion  $\sigma$
- Projektion  $\pi$
- Kreuzprodukt  $\times$
- Join (Verbund)  $\bowtie$
- Umbenennung  $\rho$
- Differenz  $-$
- ...

## Beispiele

$\sigma_{client.zip=9000}(client)$

$\pi_{client.name, client.phone}(\sigma_{value > 50000}(purchase \bowtie_{purchase.client=client.ID} client))$

# Anfragen im relationalen Modell

## Relationale Algebra

- Selektion  $\sigma$
- Projektion  $\pi$
- Kreuzprodukt  $\times$
- Join (Verbund)  $\bowtie$
- Umbenennung  $\rho$
- Differenz  $-$
- ...

## Beispiele

$\sigma_{client.zip=9000}(client)$

$\{c \mid c \in client \wedge c.zip = 9000\}$

# Tabellen und SQL

student	
<u>studID</u>	name
26120	K. Pedersen
25403	T. Jensen
...	...

attend	
<u>studID</u>	<u>courseID</u>
25403	5022
26120	5001
...	...

course	
<u>courseID</u>	title
5001	DBS
5022	Robotics
...	...

```
SELECT name
FROM student, attend, course
WHERE student.studID = attend.studID AND
      attend.courseID = course.courseID AND
      course.title = 'DBS'
```



# Tabellen und SQL

student	
<u>studID</u>	name
26120	K. Pedersen
25403	T. Jensen
...	...

attend	
<u>studID</u>	<u>courseID</u>
25403	5022
26120	5001
...	...

course	
<u>courseID</u>	title
5001	DBS
5022	Robotics
...	...

**UPDATE** course  
    **SET** title = 'Database Systems'  
**WHERE** courseID = 5001

# Integritätsbedingungen

- ... sind ein zusätzliches Sicherheitssystem
- Ziel: Dateninkonsistenzen vermeiden
- Strategie: versuche zu verhindern, dass inkosistente Daten in die DB eingefügt werden.
- Bedingungen an die möglichen Ausprägungen der Datenbank

```
CREATE TABLE student  
  (studID INTEGER PRIMARY KEY,  
   name VARCHAR(30) NOT NULL,  
   semester INTEGER CHECK semester BETWEEN 1 AND 13)
```

# Tabellen

Tabelle mit Rechnungen:

<b>invoiceNo</b>	<b>clientName</b>	<b>clientStreet</b>	<b>clientPhone</b>	<b>clientZIP</b>
3422341	Pedersen	Niels Bohrs Vej	9635 1234	9220
3459235	Pedersen	Niels Bohrs Vej	9635 1234	9220
3512344	Pedersen	Niels Bohrs Vej	9635 1234	9220
4197515	Pedersen	Niels Bohrs Vej	9635 1234	9220
4205235	Pedersen	Niels Bohrs Vej	9635 1234	9220

Was passiert, wenn Pedersen eine neue Telefonnummer bekommt?

# Normalformen

Kurz: Vermeiden von Redundanzen

- Regeln für einen guten relationalen Entwurf. Verschiedene Stufen
- z.B. erste Normalform: keine mengenwertige Attribute (z.B., rot, gelb, blau, grün)

student			
studID	street	zip	city
26120	Niels Bohrs Vej	9220	Aalborg
27550	Frederik Bajers Vej	9220	Aalborg
27550	Oddesundvej	9220	Aalborg
28106	Rendsburggade	9000	Aalborg
28106	Prinsensgade	9000	Aalborg
...	...	...	...

Einfügeanomalien, Updateanomalien, Löschanomalien

# Transaktionen

## Typisches Beispiel

- ① Lese den Kontostand von  $A$  in die Variable  $a$ :  $read(A, a);$
- ② Reduziere den Kontostand um 50 Euro:  $a := a - 50;$
- ③ Schreibe den neuen Kontostand in die Datenbasis:  $write(A, a);$
- ④ Lese den Kontostand von  $B$  in die Variable  $b$ :  $read(B, b);$
- ⑤ Erhöhe den Kontostand um 50 Euro:  $b := b + 50;$
- ⑥ Schreibe den neuen Kontostand in die Datenbasis:  $write(B, b);$

ACID-Paradigma: Atomarität, Konsistenz (Consistency), Isolation, Dauerhaftigkeit

# Transaktionen

Neulich am Geldautomaten...?

	Transaktion 1	Transaktion 2
1.	$x := \text{balance}$	
2.	$x := x + 2000$ <i>// salary</i>	$y := \text{balance}$
3.	$\text{balance} := x$	$y = y - 100$ <i>// withdrawal</i>
4.		$\text{balance} := y$

Dies ist ein Beispiel eines "Lost Updates"...

Es gibt noch viele weitere Anomalien.

# Transaktionen

## Mehrbenutzersynchronisation

- Verhindern von Problemen, wenn mehrere Benutzer gleichzeitig Daten lesen/schreiben.
- Das Datenbanksystem erlaubt gleichzeitigen Zugriff und verhindert Konflikte.

## Recovery

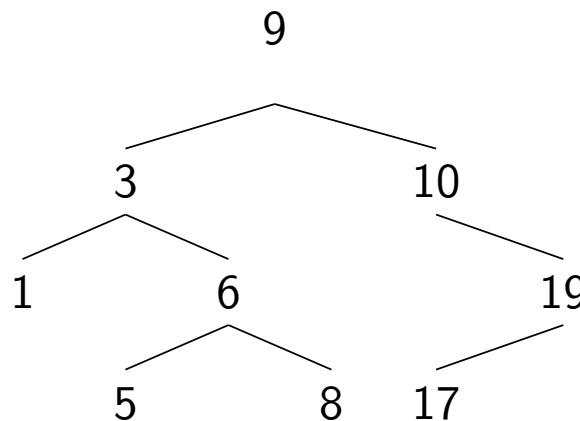
### Fehlersituation zum Zeitpunkt $t$

- Wirkungen der Transaktionen, die zuvor erfolgreich beendet wurden (commit), müssen vorhanden sein.
- Wirkungen der Transaktionen, die abgebrochen wurden, müssen vollständig eliminiert werden.

# Indexierungstechniken

Was ist mit Performance?

- **Problem:** Effiziente Suche in großen Datenmengen
- Abbildung: Schlüssel (z.B., Matrikelnummer)  $\rightarrow$  Menge von Einträgen
- Daten passen nicht in den Hauptspeicher
- Festplattenzugriffe sind teuer (insbesondere die wahlfreien (=zufälligen) Zugriffe)



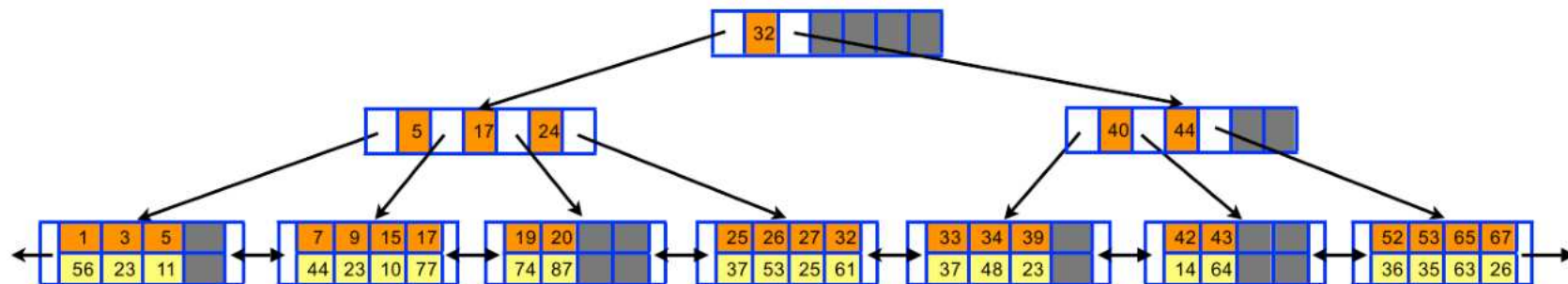
ca. 10ms für jeden Zugriff auf standard Festplatten (nicht SSD)



# B+ Baum

Lesen geschieht nicht Bit für Bit sondern in ganzen Blöcken.  
Also viel besser als binärer Baum:

- $n$ -ary Baum ( $n \gg 2$ )
- Eine Seite auf der Festplatte = ein Knoten/Blatt
- Daten werden nur in Blättern gespeichert
- Sehr starker "fan-out" (=Ausfächerung)  
→ sehr flach → wenige Zugriffe auf Festplatte



# Anfrageoptimierung

Wie werden Anfragen überhaupt ausgeführt?

- Überführen der Anfrage in einen ausführbaren Plan
- Wo liegen die Daten, wie sind sie organisiert, wie kann die Anfrage bearbeitet werden? Wie teuer ist die Ausführung der Anfrage?  
→ Kostenmodell
- Idee: finde den günstigsten Plan

Wie wird optimiert?

- “Nach unten schieben” von Selektionen
- Reihenfolge der Joins

... ( $\sigma_{value > 50000}(purchase \bowtie_{purchase.client=clients.id} clients)$ )

**besser:** ... ( $\sigma_{value > 50000}(purchase) \bowtie_{purchase.client=clients.id} clients$ )

- 1 Organisation
  - Administrative Informationen
  - Voraussetzungen
  - Fragen und Kontakt
- 2 Ablauf
  - Wichtige Informationsquellen
  - Unterlagen
  - Vorlesung
  - Aufwand
  - Benotung und Zeugnis
  - TUWEL-Kurs
  - Lernziele
- 3 Motivation
  - Wo kommen Daten vor?
  - Probleme mit Daten
  - SQL

- Architektur
- Verfügbare Systeme

- 4 Vorlesungsinhalt
  - Vorlesungsüberblick
  - Die Vorlesung im Schnelldurchlauf
- 5 Erste Aufgaben: Quizzes und Übung

# Quiz 1 & 2 und Übung

## Quiz 1 & 2

- Das erste Quiz muss bis zum 18. März 14:00 erfolgreich ( $> 0,9$  Punkte) abgeschlossen sein!
- Das zweite Quiz muss bis zum 19. März 10:00 erfolgreich ( $> 0,9$  Punkte) abgeschlossen sein!
- Lesen Sie die Anleitung zu den Quizzes auf Moodle

## Übung

- Bitte beachten Sie, dass die Übungen diese Woche beginnen und die Teilnahme verpflichtend ist!