

# Datenbanken

## Kapitel 1: Grundlagen



1

## Vorlesung

### **Vorlesung (ab 2024-10-01)**

- Di, 13:45 Uhr (K001 + YouTube)
- Mi, 10:00 Uhr (K001 + YouTube)



2

# Übung

## Übungblätter (ab 2024-10-11)

- Jede Woche ein Übungsblatt; insgesamt 12 Stück
- Aufgabe 1 (gibt i.d.R. 10 Punkte): Abgabe im ELO
- Weitere Aufgaben zur Besprechung in Übung

## Übung (ab 2024-10-16)

- Mi, 13:45 Uhr (K140)
- Mi, 15:30 Uhr (K220)
- Fr, 10:00 Uhr (K220)

Achtung: Es ist nicht möglich, Punkte für die Übung zu bekommen, wenn im ELO die Deadline abgelaufen ist oder die Abgabe nicht vollständig eingereicht wurde.

3

# Klausur

- Zulassungsvorraussetzung: keine
- Schriftliche Klausur
- 90 Minuten Zeit
- Hilfsmittel: Handschriftlich  
doppelseitig beschriebenes DIN-A4-Blatt
- Bonus: 1 Notenstufe besser (z. B. 1,0 statt 1,3), wenn mind. 100 Punkte in der Übung erreicht wurden.



Nicht auf Nachklausur und zukünftige Semester übertragbar, aber Sie können jedes Semester nochmal an den Übungen teilnehmen und den Bonus sammeln.

Als Hilfsmittel für die Klausur ist ein handschriftlich doppelseitig beschriebenes Blatt zugelassen. Nicht zwei einseitige Blätter, nicht kopiert, nicht ausgedruckt, keine Tablet-Zeichnung. Sondern: Ein (1) handschriftlich (mit Stiften) doppelseitig (Vorder- und Rückseite) beschriebenes DIN-A4-Blatt.

4

# Literatur

- *J. Schildgen*  
Sprachkurs SQL - Das Datenbanken-Hörbuch
- *A. Kemper, A. Eickler*  
Datenbanksysteme - Eine Einführung
- *A. Heuer, K.-U. Sattler, G. Saake*  
Datenbanken: Konzepte und Sprachen
- *C. J. Date*  
An Introduction to Database Systems

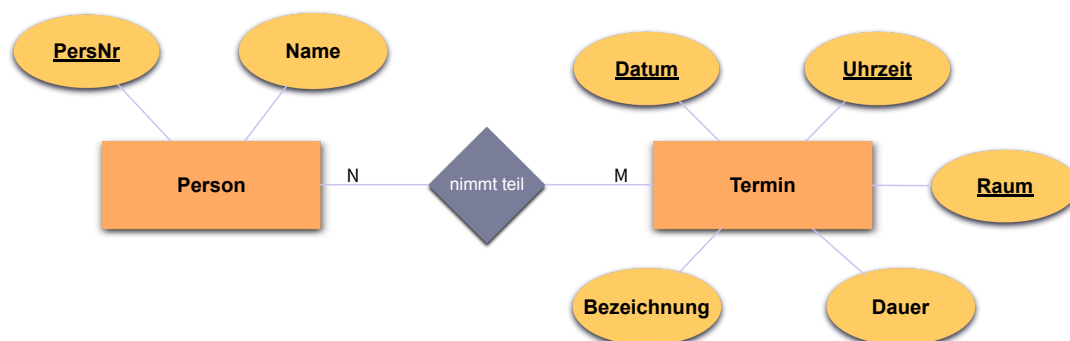


Der Aufbau dieser Vorlesung sowie die verwendeten Beispiele orientieren sich größtenteils am Sprachkurs SQL, dem Datenbanken-Hörbuch. Ein Kopfhörer-Symbol rechts oben auf Folien verweist auf die jeweiligen Kapitel (Tracks) im Hörbuch.

5

## Aufbau der Vorlesung

1. Datenbanken-Grundlagen
2. Logischer DB-Entwurf: ER-Modellierung



# Aufbau der Vorlesung

1. Datenbanken-Grundlagen
2. Logischer DB-Entwurf: ER-Modellierung
3. Das Relationenmodell

## termine

<u>Datum</u>	<u>Uhrzeit</u>	<u>Raum</u>	<u>Dauer</u>	<u>Bezeichnung</u>
2020-10-14	14:15	17-123	90	Dings-Meeting
2020-10-15	10:15	17-222	60	Treffen mit Jürgen
2020-10-21	14:15	17-222	60	Treffen mit Frau Becker
2020-10-21	16:00	Cafeteria	45	Kaffee trinken

7

# Aufbau der Vorlesung

1. Datenbanken-Grundlagen
2. Logischer DB-Entwurf: ER-Modellierung
3. Das Relationenmodell
4. Relationale Algebra

$\pi_{Uhrzeit, Bezeichnung} \sigma_{Datum='2020-10-14'}(Termine)$

8

# Aufbau der Vorlesung

1. Datenbanken-Grundlagen
2. Logischer DB-Entwurf: ER-Modellierung
3. Das Relationenmodell
4. Relationale Algebra
5. SQL

```
SELECT uhrzeit, bezeichnung FROM termine  
WHERE datum = '2020-10-14'
```

9

# Aufbau der Vorlesung

1. Datenbanken-Grundlagen
2. Logischer DB-Entwurf: ER-Modellierung
3. Das Relationenmodell
4. Relationale Algebra
5. SQL
6. Entwicklung von DB-Anwendungen
7. Indexstrukturen
8. Transaktionen

10

# Informationssysteme

**Ein computergestütztes Programmsystem,  
welches Informationen...**

- erfasst,
- dauerhaft speichert,
- verarbeitet,
- verändert,
- analysiert,
- bereitstellt,
- anzeigt.



11

## Anwendungsklasse: OLTP

**OLTP (OnLine Transaction Processing)**

Abwicklung von Transaktionen auf den operationalen  
Daten

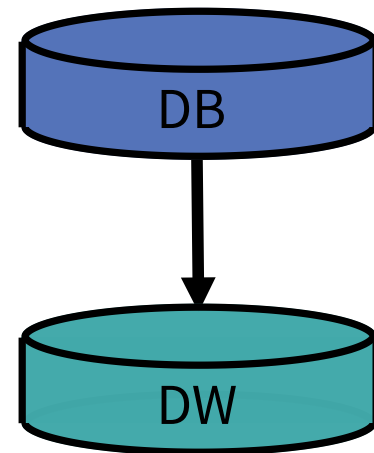
```
Gib mir alle Details zu Buch 2817;  
Berechne die Durchschnittsbewertung von Buch 2817;  
Kunde 123456 bestellt Buch 2817;
```

# Anwendungsklasse: OLAP

## OLAP (OnLine Analytical Processing)

Analyse von Datenbeständen;  
meist in einem Data-Warehouse

Finde diejenigen Bücher, die in den letzten 3 Monaten die höchsten Umsätze generierten;



Das Data-Warehouse ist getrennt von den operativen Daten. In regelmäßigen Zeitabständen (z. B. einmal nachts) werden die Änderungen im operationalen DBS in das Data-Warehouse geladen. Dort können komplexe Datenanalysen durchgeführt werden.

13

## Data Mining

Anwendung statistischer Methoden, um in großen Datenmengen

- Querverweise / Beziehungen
- Trends
- Muster
- (Un-)Regelmäßigkeiten

zu erkennen.



Ziel des Data-Minings ist es, aus vorhandenen Datenbeständen Wissen zu gewinnen. Eine Spezialisierung ist beispielsweise Text Mining, welches aus Verfahren zur Analyse meist unstrukturierter Texte besteht.

14

# Transaktionskonzept

- DBMS unterstützt Mehrbenutzerbetrieb
- TA = Folge mehrerer DB-Operationen als eine Einheit
- TAs laufen parallel
- Fehlerhafte TAs werden vollständig zurückgesetzt

```
konto1.kontostand -= 100;  
konto2.kontostand += 100;  
commit;
```

Transaktion

Die hier dargestellte Transaktion (TA) besteht aus zwei Schreiboperationen auf zwei Konten. Die Commit-Operation schließt die Transaktion ab. Tritt beispielsweise bei der Ausführung des zweiten Befehls ein Fehler auf (Konto existiert nicht, Limit erreicht, Hardwarefehler,...), wird auch der Rest der TA, also der erste Befehl, nicht ausgeführt. Mittels eines Rollbacks statt dem Commit kann die Transaktion manuell abgebrochen und zurückgesetzt werden.

15

## ACID-Transaktionen

- Atomarität
- Konsistenz
- Isolation
- Dauerhaftigkeit

```
konto1.kontostand -= 100;  
konto2.kontostand += 100;  
commit;
```



Transaktion

Das hier dargestellte ACID-Paradigma beschreibt die Eigenschaften von Transaktionen in Datenbanken. Transaktionen sind eine Sammlung von Aktionen, die stets als Gesamtheit - atomar - ausgeführt werden. Des Weiteren überführen Transaktionen die Datenbank von einem Konsistenten Zustand in einen konsistenten Zustand. Parallel laufende Transaktionen beeinflussen sich nicht. Und Änderungen an der Datenbank, die mit einem Commit bestätigt werden, bleiben dauerhaft in ihr gespeichert.

16



# Geht's auch ohne Datenbank?

Speicherung von Daten in Dateien

- (-) Komplexe Anwendungsentwicklung
- (-) Ineffizient (langsam)
- (-) Keine Anfragesprache
- (-) Kein ACID
- (-) Redundanz



Verwenden Anwendungen statt einer DB simple Dateien, ist die Entwicklung von Einfüge-, Änderungs- und Suchfunktionen sehr aufwändig. Da sich nicht um ACID gekümmert wird, können Inkonsistenzen entstehen, Änderungen verlorengehen und vieles mehr. In Datenbanken gilt das Prinzip der Redundanzfreiheit. Jede Info wird nur einmal gespeichert. In Dateien ist oft der Fall, dass ein und der selbe Fakt an vielen Stellen zu finden ist.

17

## DB, DBMS, DBS

### Datenbank (DB)

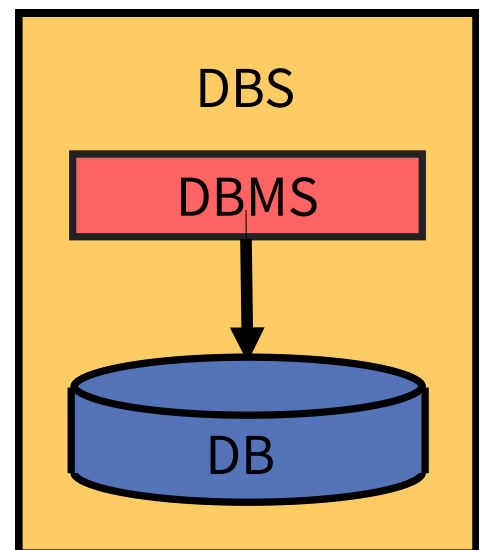
Sammlung von Daten

### Datenbankmanagementsystem

Standardsoftware zur Definition, Verwaltung, Verarbeitung und Auswertung von DB-Daten

### Datenbanksystem

DBS = DB + DBMS



# Anfragesprachen

Benutzer und Anwendungen sprechen mit dem DBMS in einer Anfragesprache (Query Language).

## **Data Definition Language (DDL)**

zur Definition des DB-Schemas

## **Data Manipulation Language (DML)**

für den lesenden und schreibenden Zugriff auf die Daten

19



# Anfrageausführung

```
SELECT uhrzeit, bezeichnung FROM termine  
WHERE datum = '2020-10-14'
```

SQL-Query

1. Benutzer schickt die Anfrage an das DBMS
2. DBMS generiert Anfrageplan
3. DBMS optimiert den Plan
4. DBMS führt den Anfrageplan aus
  - Daten aus DB lesen
  - Operationen durchführen (Join, Index-Scan, ...)
5. Ergebnis wird zurück an den Benutzer geschickt



20

# Metadaten und Daten

termine				
<u>Datum</u>	<u>Uhrzeit</u>	<u>Raum</u>	Dauer	Bezeichnung
2020-10-14	14:15	17-123	90	Dings-Meeting
2020-10-15	10:15	17-222	60	Treffen mit Jürgen

Metadaten

Daten

Metadaten (Datenbankschema):

- Festlegung der Struktur der zu speicherenden Daten
- weitgehend zeitinvariant; gespeichert im DB-Katalog

Daten (Ausprägungen):

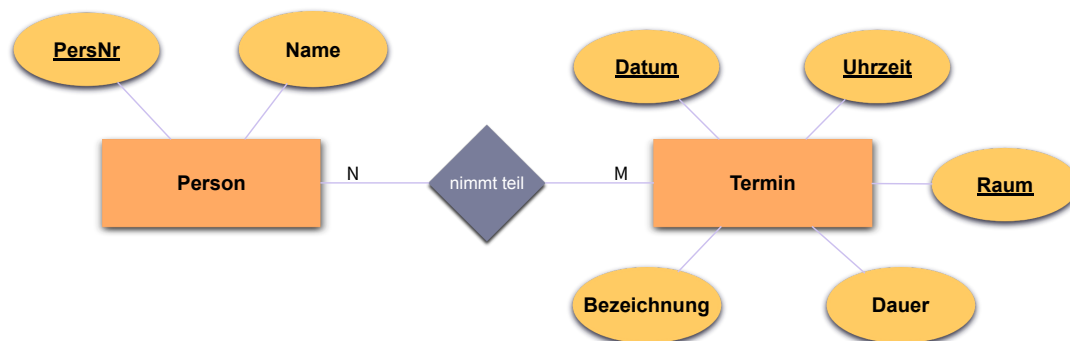
- veränderlich

21

# Datenmodellierung

## Konzeptionelles Datenmodell

- Unabhängig vom eingesetzten DBMS
- ER-Diagramme, UML, ...



22

# Datenmodellierung

## Logisches Datenmodell

- Prominentestes Datenmodell: Das Relationenmodell
- Andere: objekt-relational, hierarchisch, netzwerkartig, objektorientiert (→ NoSQL)

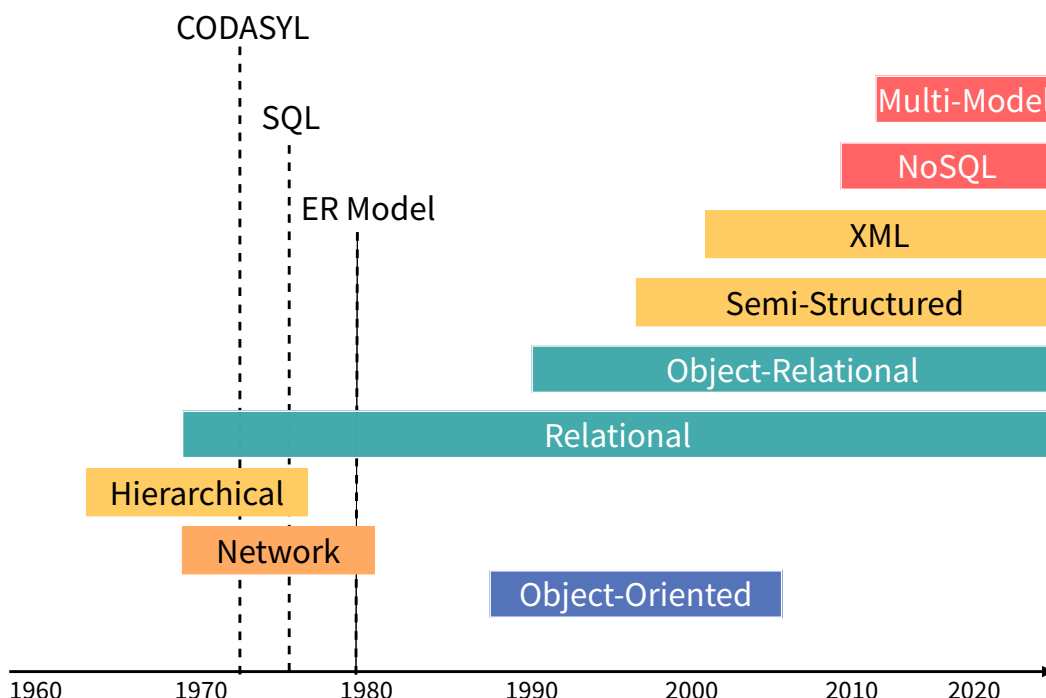
```
CREATE TABLE termine (  
  datum DATE, uhrzeit CHAR(5), raum VARCHAR(15), dauer INT,  
  bezeichnung VARCHAR(200), PRIMARY KEY(datum, uhrzeit, raum));
```

## Physisches Datenmodell

- Interne Speicherung und Organisation der Daten

23

## Evolution der Datenmodelle



24

# Datenabstraktion

## Externe Ebene (Sichten / Views)

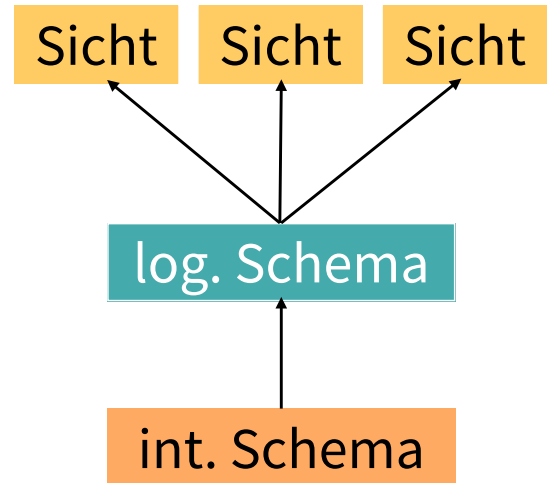
Sicht auf eine Teilmenge des logischen Schemas für eine bestimmte Benutzergruppe

## Logische Ebene

DB-Gesamt-Schema

## Physische Ebene

Internes Schema / Speicherung der Daten



25

# Datenabstraktion (Bsp.)

## Externe Ebene (View für Webshop-Anwendung)

<u>Produktnr</u>	Bezeichnung	Preis	Bewertung
17	Schokoriegel	0.89	4.5
29	Spülmaschinentabs	3.99	2.0

## Logische Ebene

<u>Produktnr</u>	Bezeichnung	Preis	<u>Kundennr</u>	<u>Produktnr</u>	Bew
17	Schokoriegel	0.89	5	17	4
29	Spülmaschinentabs	3.99	8	17	5
			5	29	2

**Physische Ebene:** 0010110000000100101111010101...

26

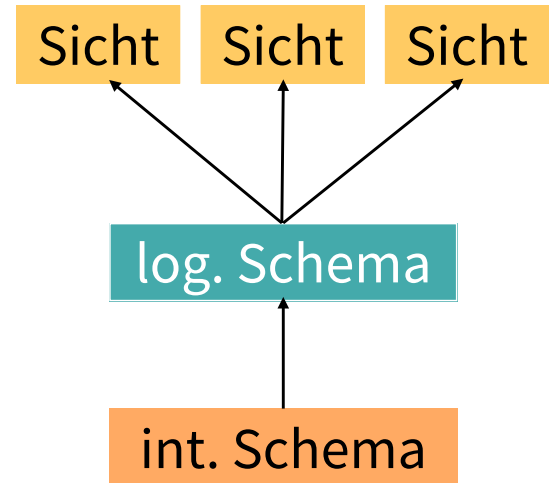
# Datenunabhängigkeit

## Logische Datenunabhängigkeit

Änderungen am logischen Schema (z. B. Spaltenumbenennung) erfordern lediglich eine Anpassung der externen Sichten. Anwendungen bleiben unberührt.

## Physische Datenunabhängigkeit

Änderungen an internen Speicher- oder Zugriffsstrukturen haben keine Auswirkungen auf das logische oder externe Schema.



Wird in dem Beispiel auf der vorherigen Folie auf logischer Ebene die Spalte Bezeichnung in Produktbezeichnung umbenannt, kann die entsprechende Sicht auf externer Ebene einfach angepasst werden und alle bisherigen Anwendungen, die über die Sicht auf die Daten zugreifen, funktionieren auch weiterhin noch.

27

# Architektur von DBMS

Vereinfachtes Modell: 3-Schichten-Modell

## Datensystem

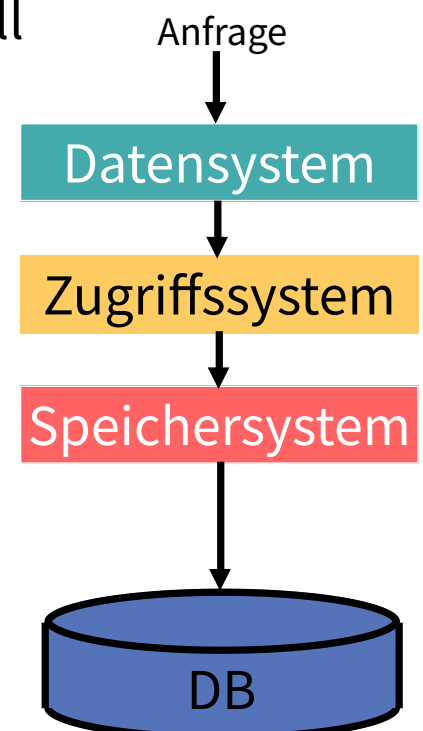
Übersetzung und Optimierung von Anfragen

## Zugriffssystem

Verwaltung von Datensätzen und Indexten

## Speichersystem

Blockverwaltung und Caching



28

# Kapitelzusammenfassung

- OLTP / OLAP
- ACID-Transaktionen
- DB + DBMS = DBS
- DDL / DML
- Metadaten / Daten
- Konzept. / Log. / Phys. Datenmodell
- Ext. / Log. / Phy. Datenebene
- Datensystem / Speichersystem / Zugriffssystem