

# 1. Einführung und Grundbegriffe

Vorlesung "Informationssysteme"  
Sommersemester 2017

# Überblick

- Was ist ein Informationssystem?
- Aufgaben eines Informationssystems (IS)
  - wichtige Anwendungsklassen
- Nutzung von Datenbanksystemen
  - Anforderungen und Aufgaben
- Wichtige Konzepte und Eigenschaften von DBS
- Architektur von DBS

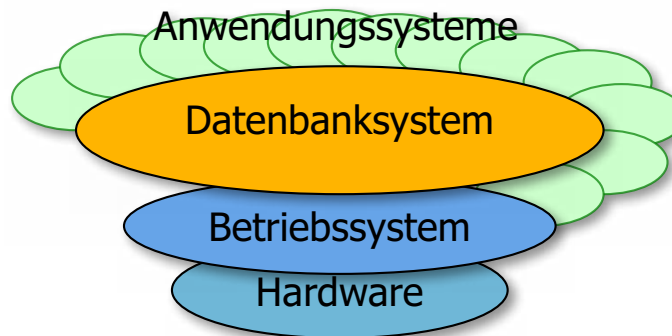
# Was sind Informationssysteme (IS)?

- Computergestützte Programmsysteme,
  - die Informationen erfassen, dauerhaft speichern, verarbeiten, verändern, analysieren, bereit stellen, anzeigen
- Betriebliche Informationssysteme
  - spiegeln Geschäftsmodell eines Unternehmens wider
  - organisieren und unterstützen Arbeitsabläufe
  - integrieren eine Vielzahl von Datenquellen
- Web-basierte Informationssysteme
  - stellen Informationsdienste über Web-Schnittstellen zur Verfügung
  - unterstützen die Abwicklung von Geschäftsvorgängen über das Internet (E-Business)

# Beispiele

- **Universitätsinformationssystem (KIS, QIS, ...)**
  - **Daten:** Fachbereiche, Studenten, Professoren, Vorlesungen, Prüfungen, ...
  - **Vorgänge:** typische Verwaltungsaufgaben
- **Produktionsbetrieb**
  - **Daten:** Abteilungen, Mitarbeiter, Produkte, Projekte, Lieferanten, ...
  - **Vorgänge:** Gehaltsabrechnung, Fertigung und Versand von Produkten, Lagerhaltung, ...
- **Bank**
  - **Daten:** Filialen, Kunden, Konten, ...
  - **Vorgänge:** Zahlungsverkehr, Kundenverwaltung, Geldanlage, Zinsberechnung, ...
- **Internet-Auktionshaus**
  - **Daten:** Produkte, Käufer, Verkäufer, Auktionen, ...
  - **Vorgänge:** Suche nach angebotenen Waren, Prüfen der Vertrauenswürdigkeit der Verkäufer, Durchführen der Auktionen, ...
- **Aber auch Suchmaschinen, soziale Netzwerke, Google-Anwendungen, ...**

# Aufbau eines rechnergestützten Informationssystems



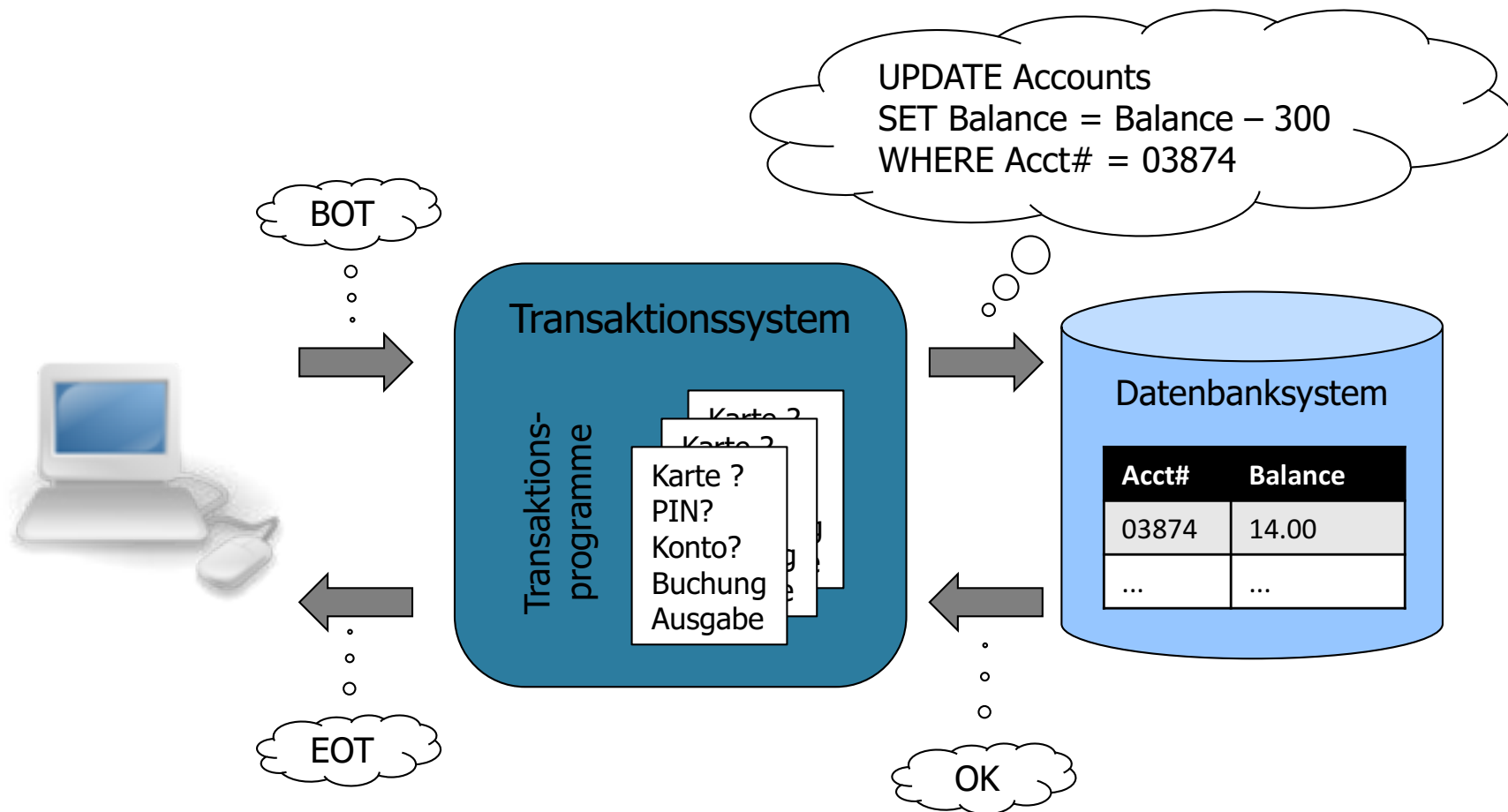
- Datenbanksystem (DBS) ist zentrale Komponente für KIS
  - Datenbank (DB): Sammlung gespeicherter operationaler Daten, die von den Anwendungssystemen eines bestimmten Unternehmens benötigt werden
  - Datenbankverwaltungssystem (DBVS, DBMS): standardisiertes Softwaresystem zur Definition, Verwaltung, Verarbeitung und Auswertung von DB-Daten.
- Grobdefinition: **DBS = DB + DBVS**

# Zielsetzungen für betriebliche IS

- Anforderungen sind spezifisch für Unternehmensebene
- Operierende Ebene (Sachbearbeitung)
  - Verbesserung durch Auskunfts-, Berichts-, Buchungs-, Produktions-, Steuerungs-, Vertriebs- und Anwendungssysteme
  - **Kennzeichen:** Verarbeitung begrenzter Datenmengen und große Änderungshäufigkeit der Daten
- Planenden Ebene (mittleres Management)
  - Unterstützung und evtl. Teilautomatisierung durch benutzerorientierte Bereitstellung von Informationen, Suche und Auswertung von Daten im Dialog, Automatisierung von Routine-Entscheidungen, Einsatz von mathematisch-statistischen Methoden
  - **Kennzeichen:** teilweise nicht vorhersehbarer Informationsbedarf, verdichtete Daten, große Datenmengen, kein Änderungsdienst
- Unterstützung der strategischen Ebene
  - durch Bereitstellung von Daten für einen **überwiegend nicht vorhersehbaren Informationsbedarf**

# Wichtige Anwendungsklassen

- **OLTP (On-Line Transaction Processing)**  
Abwicklung von Transaktionen auf den **operationalen Daten**

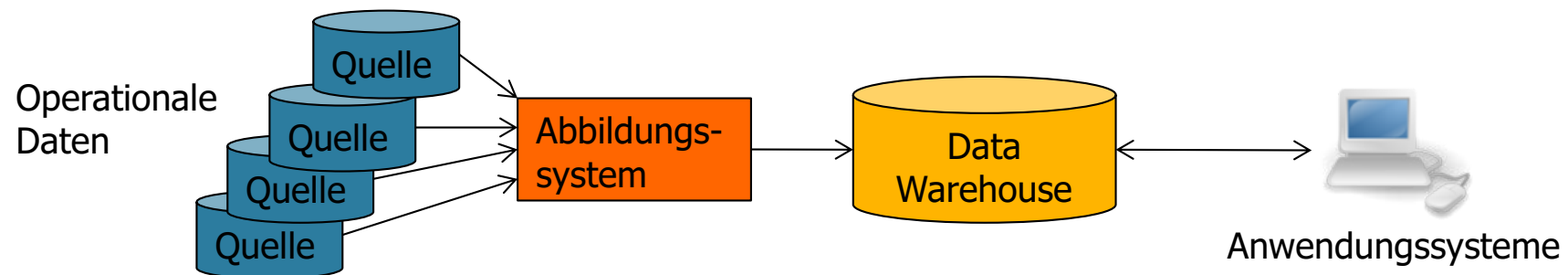


# Wichtige Anwendungsklassen (2)

## ■ OLAP (On-Line Analytical Processing)

Analyse betrieblicher Datenbestände, meist anhand eines **DW** (Data Warehouse)

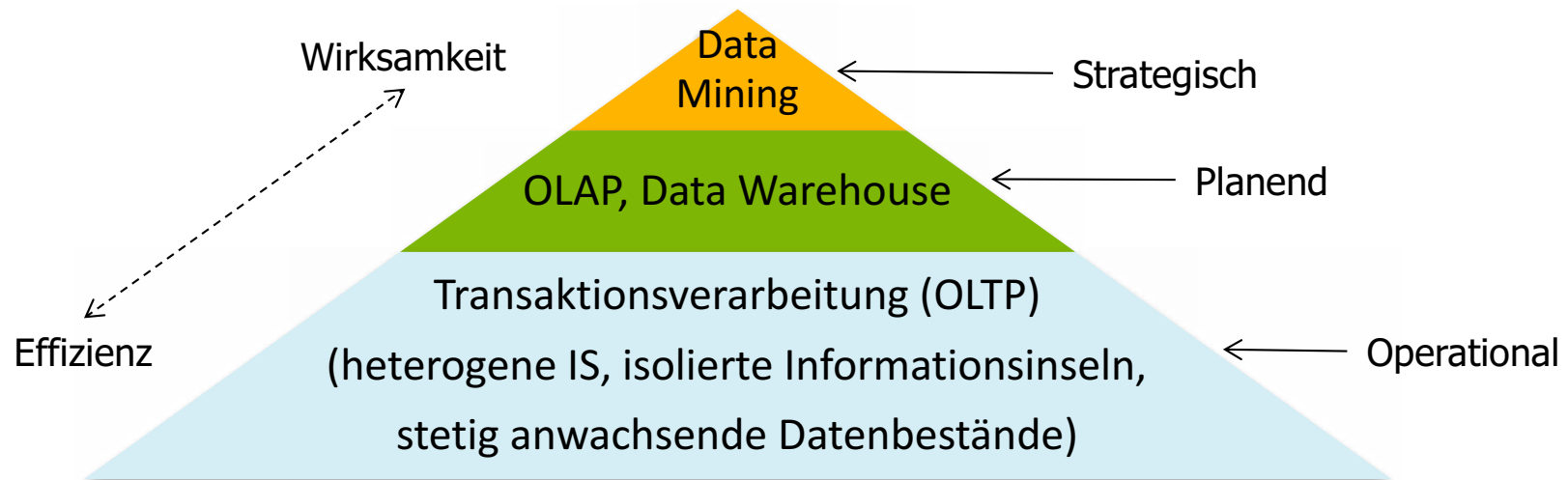
- als themenorientierte, **historische**, zeitlich veränderliche Datensammlung, getrennt von den operationalen Daten
- aus sehr **großen Datenbeständen** abgeleitet und **multidimensional** aufgegliedert
- **Voraggregation** der Daten aus Leistungsgründen
- **inkrementelle Aktualisierung** des DW aus den operationalen DBS (z. B. jede Nacht) erfolgen.





# Wichtige Anwendungsklassen (3)

- **DM (Data Mining):**
  - **selbständiges** Aufspüren von **inhärenten Daten- bzw. Informationsmustern** aus großen dynamischen Datenbeständen
    - Beziehungen zwischen Datensätzen oder zwischen Attributen eines Satzes
    - gewisse Regelmäßigkeiten oder Regelabweichungen in Attributwerten
  - Erkennen von unscharfen oder probabilistischen Regeln erforderlich



# Nutzung von Datenbanksystemen für IS

- Vielfältige, wiederkehrende Anforderungen und Aufgaben bei der Entwicklung von IS
    - Zuverlässige Verwaltung großer, persistenter Datenmengen
    - Effizienter Datenzugriff
    - Flexible Auswahl und Verknüpfung von Daten
    - Leistungsanforderungen
  - Adressierung dieser Anforderungen
    - durch Dateisysteme nicht bzw. nur sehr unzureichend gegeben
    - in der Anwendung zu komplex und teuer
- ⇒ Einsatz (historisch: Entwicklung) von Datenbanksystemen zur Übernahme von kritischen Datenverwaltungsaufgaben

# Vermeidung von Datenredundanz und Inkonsistenzen

## ■ Traditionelle Dateiverarbeitung

- jede Benutzergruppe pflegt eigene Dateien
  - Beispiel: Studierendensekretariat und Prüfungsamt verwalten Studierendendaten in getrennten Dateien
- viele Daten werden mehrfach (redundant) gespeichert
  - im Beispiel: Matrikelnummer, Name, Kontaktinfo, ...
  - redundanter Arbeitsaufwand, Verschwendung von Ressourcen (Speicher)
  - inkonsistente Änderungsstände!

## ■ Datenbanksystem

- integrierte Datenbank ohne Redundanz
  - vermeidet die o.g. Probleme
- Datenbankentwurf berücksichtigt die Sichten aller Benutzergruppen

# Flexible und mächtige Zugriffs- bzw. Anfragemöglichkeiten

- Wertbasierte Auswahl von Datensätzen und Verknüpfung von Sätzen

Beispiel: Gib mir die Kontaktinfos aller Studierenden, die bei der Prüfung "Informationssysteme" mit 2.0 oder besser bestanden haben

- Mit Dateisystem schwer bis unmöglich zu realisieren, erfordert komplexe Filter- und Verknüpfungslogik in der Anwendung

- Datenbanksystem

- einheitliche Modellierung und Speicherung der gesamten Informationen
- mächtige (deklarative) Anfragesprache zur Auswahl und Verknüpfung von Daten
- effiziente Zugriffsverfahren für große Datenmengen

# Unterstützung von Mehrbenutzerbetrieb

- Ein IS wird meist von vielen Benutzern gleichzeitig benutzt
  - Beispiele: Flugreservierung, Banktransaktionen, ...
- Unkontrollierter Zugriff auf die gleichen Daten kann problematisch sein (Mehrbenutzeranomalien, z.B. *lost update*)

## Abheben (Benutzer 1)

Lese Kontostand von 1234

$KtoStd := KtoStd - Betrag1$

Schreibe Kontostand von 1234

## Überweisen (Benutzer 2)

....

Lese Kontostand 1234

$KtoStd := KtoStd + Betrag2$

Schreibe Kontostand von 1234

- Dateisysteme bieten keine oder ungenügende Kontrolle
- Datenbanksysteme bieten Mehrbenutzerkontrolle mit Hilfe sog. Synchronisationsverfahren

# Zuverlässigkeit und Fehlertoleranz

- Auftreten von Fehlern kann zu Datenverlust führen
  - Programmfehler, Systemausfall, Hardwarefehler, ...
- Ziel: Wiederherstellen des letzten konsistenten (gemäß der realen Welt gültigen) Gesamtzustands der Daten
- Dateisysteme unzureichend, periodisches Sichern von Dateien nicht ausreichend
- Datenbanksysteme
  - Einsatz von Logging & Recoverymechanismen
  - dadurch Schutz vor Datenverlust

# Überwachung von Integritätsbedingungen

- Integritätsbedingungen sind logische Bedingungen, welche von korrekten Daten eingehalten werden müssen
  - können komplex sein und mehrere Informationseinheiten betreffen
    - Beispiel Universität: eine Prüfung darf nur für immatrikulierte und nicht beurlaubte Studierende durchgeführt werden und man hat maximal 3 Versuche
- Dateisysteme bieten keine Unterstützung, Prüfung der Bedingungen in der Anwendung ist unzureichend
- Datenbanksystem
  - unterstützt explizite Integritätsbedingungen
  - Prüfung der Bedingungen und Zurückweisen von Operationen, welche sie verletzen, zu wohldefinierten Zeitpunkten

# Zugriffskontrolle

- Der Zugriff (Lesen und insbesondere Ändern) auf Daten muß
  - auf bestimmte Benutzer oder Benutzergruppen einschränkbar und
  - für einzelne Bestandteile (Attribute) von Objekten und Teilmengen von Objekten kontrollierbar sein
- Dateisysteme bieten nicht die erforderliche Flexibilität
- Datenbanksysteme
  - Zugriffsprivilegien (Lesen, Ändern, ...) für Benutzer, Rollen
  - sehr flexibel und feingranular einsetzbar
    - auf Teile von Objekten verwendbar (z.B. spezifische Eigenschaften wie Name, Gehalt eines Angestellten)
    - auf Teilmengen von Objekten des gleichen Typs definierbar (z.B. Manager darf nur das Gehalt seiner eigenen Mitarbeiter sehen)



# Wichtige Konzepte und Eigenschaften von DBS

- Datenbankschema und Ausprägung
- Datenmodelle
- Datenabstraktion
- Datenunabhängigkeit
- Datenbanksprachen
- Transaktionskonzept

# Datenbankschema und Ausprägung

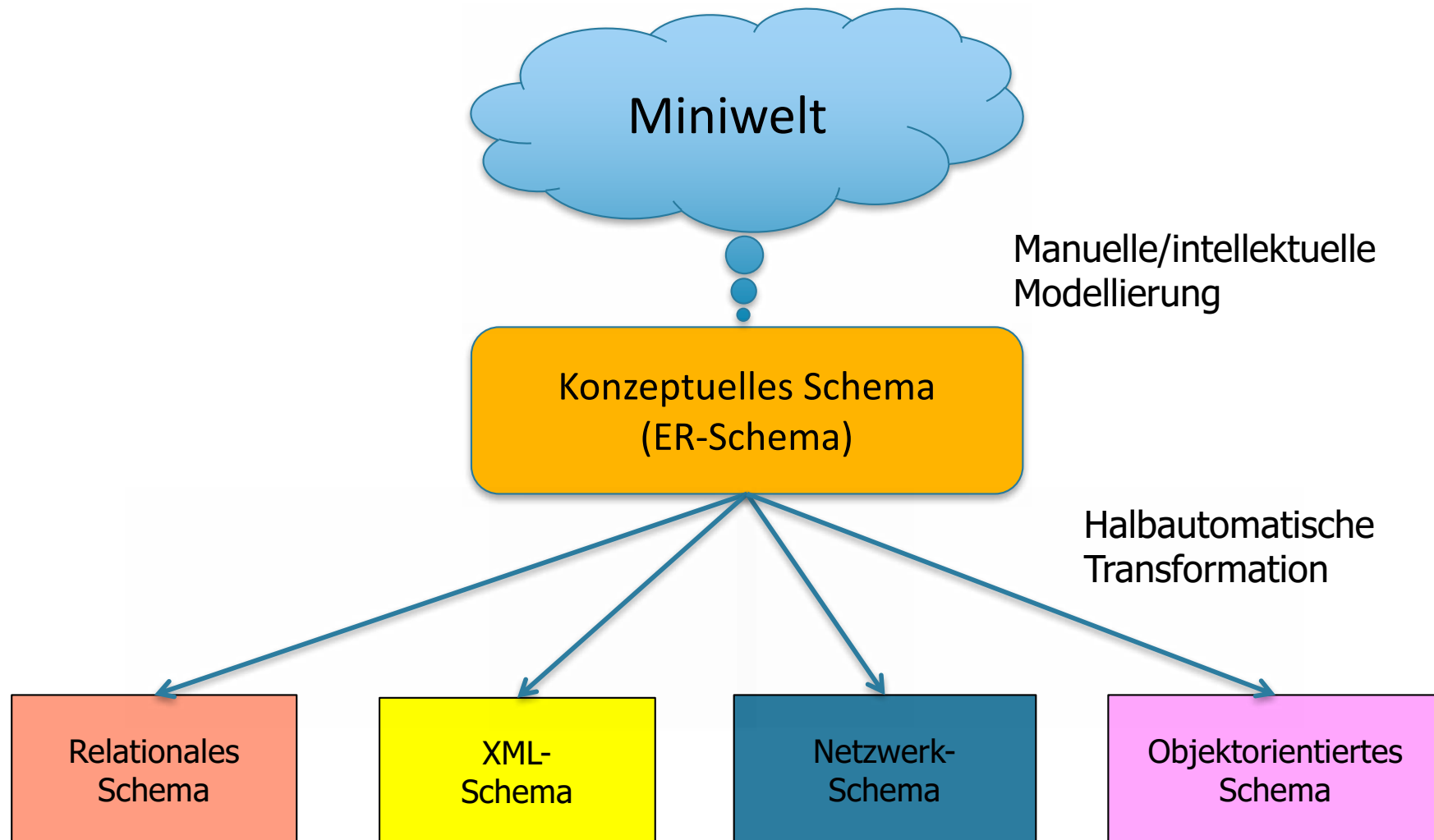
- Datenbankschema (auch: Intension)
  - legt die Struktur der gespeicherten DB-Objekte fest
  - Metadaten (Daten über Daten), weitgehend zeitinvariant
  - wird im Datenbankkatalog aufbewahrt
- Ausprägung oder Instanz (auch: Extension)
  - die eigentlichen Daten(-sätze)
  - veränderlich

<b>ANGESTELLTER</b>				
<b>PNR</b>	<b>NAME</b>	<b>TAETIGKEIT</b>	<b>GEHALT</b>	<b>ALTER</b>
496	Peinl	Pförtner	2100	63
497	Kinzinger	Kopist	2800	25
498	Meyweg	Kalligraph	4500	56
...				

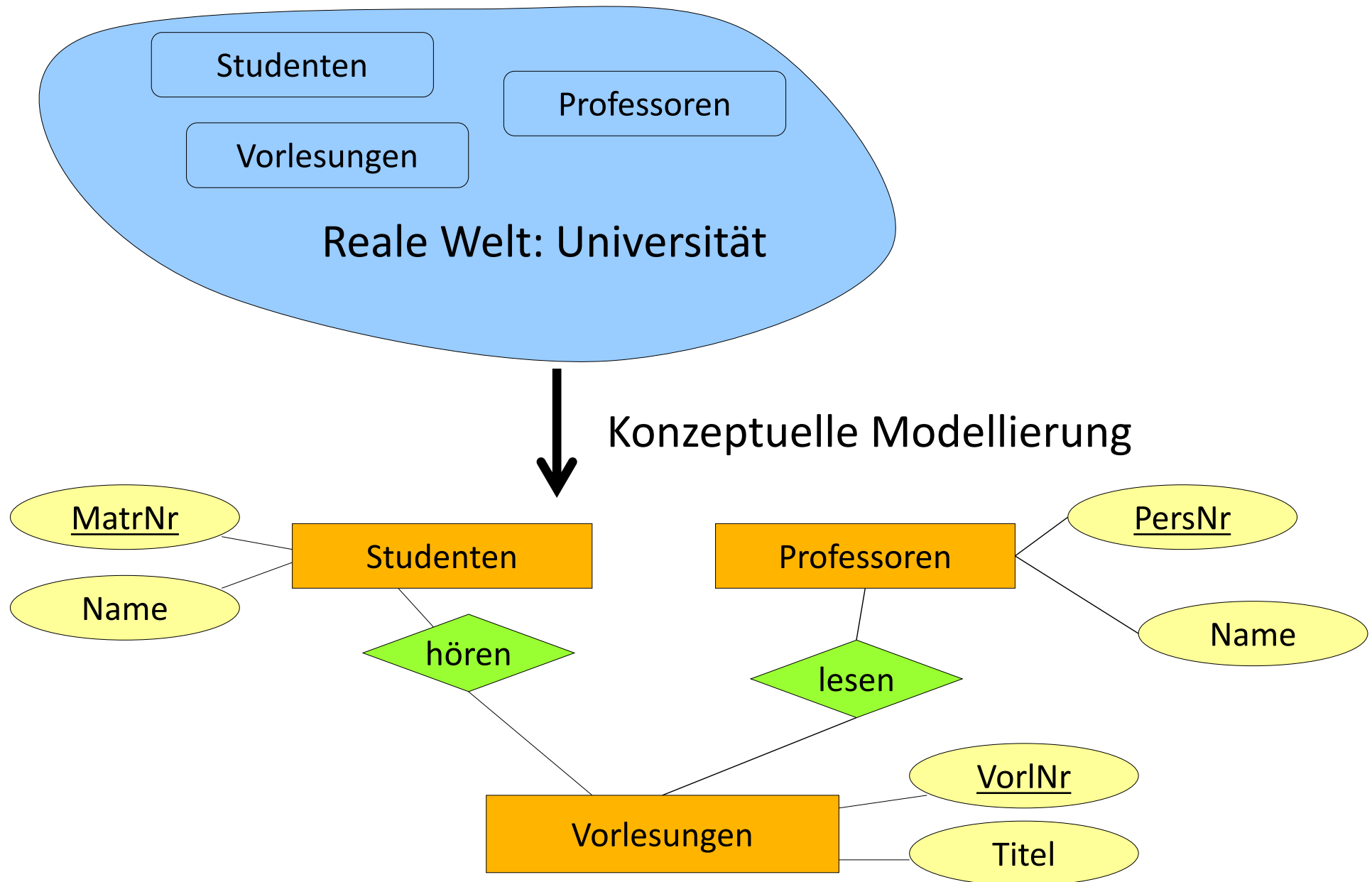
# Datenmodelle

- Datenmodell
  - enthält Modellierungskonstrukte zur Abbildung eines Ausschnitts der realen Welt (Miniwelt, *universe of discourse*)
  - "Sprache" zur Beschreibung der Struktur einer DB, also Datentypen/-strukturen, Beziehungen, Integritätsbedingungen
- Kategorien von Datenmodellen
  - Konzeptuelle Datenmodelle bzw. Informationsmodelle
    - unabhängig vom eingesetzten DBS
    - Beispiele: Entity-Relationship-Modell (ERM), UML
  - Logische (Implementations-) Datenmodelle
    - werden direkt von DBVSen unterstützt
    - definieren außerdem (generische) Operationen zur Datenmanipulation
    - dominant: relational (und objekt-relational)
    - weitere: hierarchisch, netzwerkartig, objektorientiert, semistrukturiert/XML
  - Physische Datenmodelle
    - zur Beschreibung der Datenspeicherung und –organisation

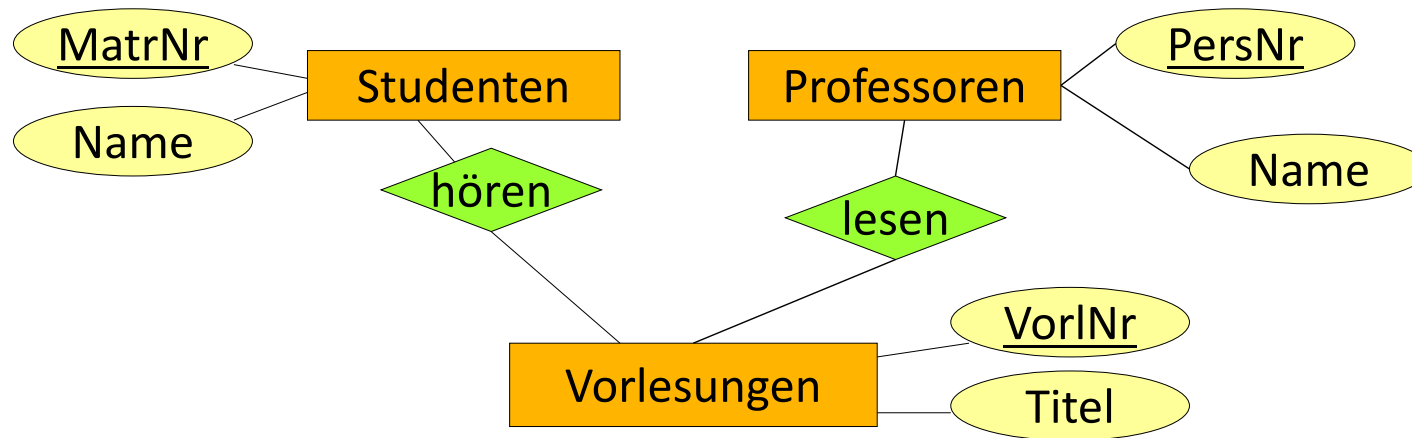
# Datenmodellierung



# Modellierung einer kleinen Beispielanwendung



# Transformation in Relationales Schema



Transformation

**Studenten**

<u>MatrNr</u>	Name
...	

**Professoren**

<u>PersNr</u>	Name
...	

**Hören**

<u>MatrNr</u>	<u>VorlNr</u>
...	

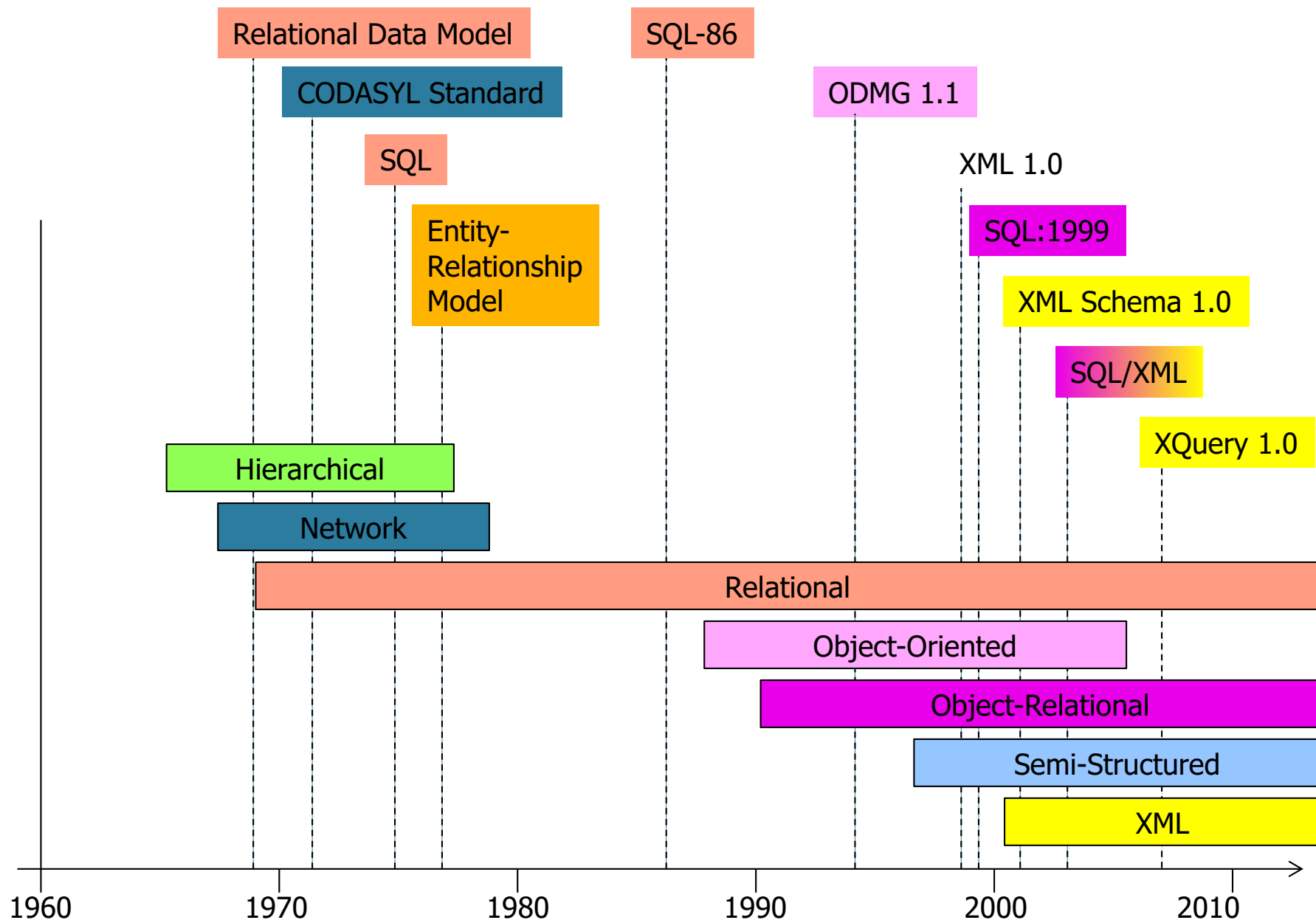
**Lesen**

<u>PersNr</u>	<u>VorlNr</u>
...	

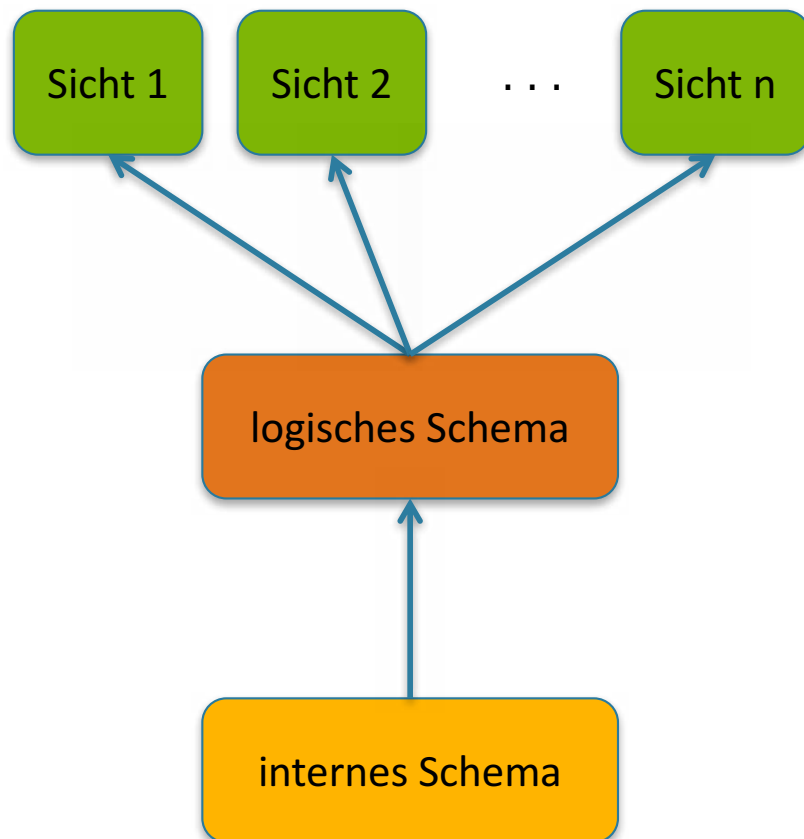
**Vorlesungen**

<u>VorlNr</u>	Titel
...	

# Datenmodelle - Evolution und Historie



# Datenabstraktion



## **externe Ebene (Sichten)**

jedes externe Schema (mit log. DM spezifiziert) beschreibt die Sicht einer Benutzergruppe als Teilmenge des log. Schemas

## **logische Ebene**

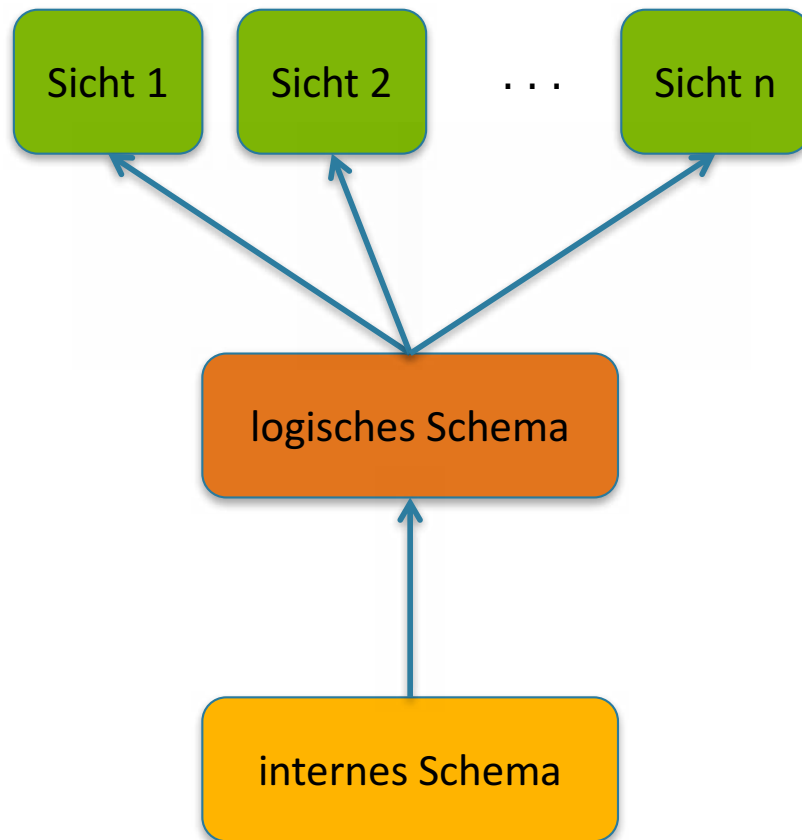
DB-Schema (mit log. DM spezifiziert) beschreibt die Struktur der gesamten DB für alle Nutzer

## **physische Ebene**

internes Schema (mit phys. DM beschrieben) legt fest, wie die Daten gespeichert werden



# Datenunabhängigkeit



## **logische Datenunabhängigkeit**

einfache Änderungen des log. Schemas (Umbenennungen, neue Satztypen und Attribute) können in den Sichten verborgen werden; Anwendungen nutzen nur die Sichten und bleiben unberührt

## **physische Datenunabhängigkeit**

log. Schema verbirgt mögliche Änderungen im internen Schema (physische Ebene); Anwendungen und Sichtendefinitionen auf der externen Ebene bleiben unberührt

# Datenbanksprachen

- **Datendefinitionssprache (*Data Definition Language, DDL*)**
  - zur Definition des logischen Schemas im DBS
  - beinhaltet oft Sprache für internes und externes Schema:
    - Speicherdefinitionssprache (*Storage Definition Language, SDL*)
    - View-Definitionssprache (*View Definition Language, VDL*)
- **Datenmanipulationssprache (*Data Manipulation Language, DML*)**
  - Anfragesprache (Query Language, QL)
  - Änderungsoperationen zum Einfügen, Ändern und Löschen
- **Art der DML**
  - prozedural: navigierend, satzorientiert, in Programm genutzt
  - deklarativ: mengenorientiert – "welche" Daten sind gewünscht
- **Nutzung der DML: interaktiv oder in einem Programm**

# Beispiel: SQL

- DDL

```
create table Studenten(  
  MatrNr char(6) primary key,  
  Name    varchar(30));
```

```
create table Vorlesungen(  
  VorlNr char(10) primary key,  
  Titel   varchar(30));
```

```
create table Hören(  
  MatrNr char(6)  
    references Studenten,  
  VorlNr char(10)  
    references Vorlesungen,  
  primary key (MatrNr, VorlNr));
```

...

**Studenten**

<u>MatrNr</u>	Name
...	

**Hören**

<u>MatrNr</u>	<u>VorlNr</u>
...	

**Vorlesungen**

<u>VorlNr</u>	Titel
...	

- DML: Anfragen

```
select Name  
from Studenten  
where MatrNr = '123456';
```

```
select s.Name, s.MatrNr, h.VorlNr  
from Studenten s, Hören h  
where s.MatrNr = h.MatrNr;
```

- DML: Änderungen

```
insert into Studenten  
values ('456789', 'Heinz Müller');
```

```
delete from Hören  
where MatrNr = '123456';
```

```
update Vorlesungen  
set Titel = 'Datenbanksysteme'  
where  
  Titel = 'Datenbankanwendung';
```

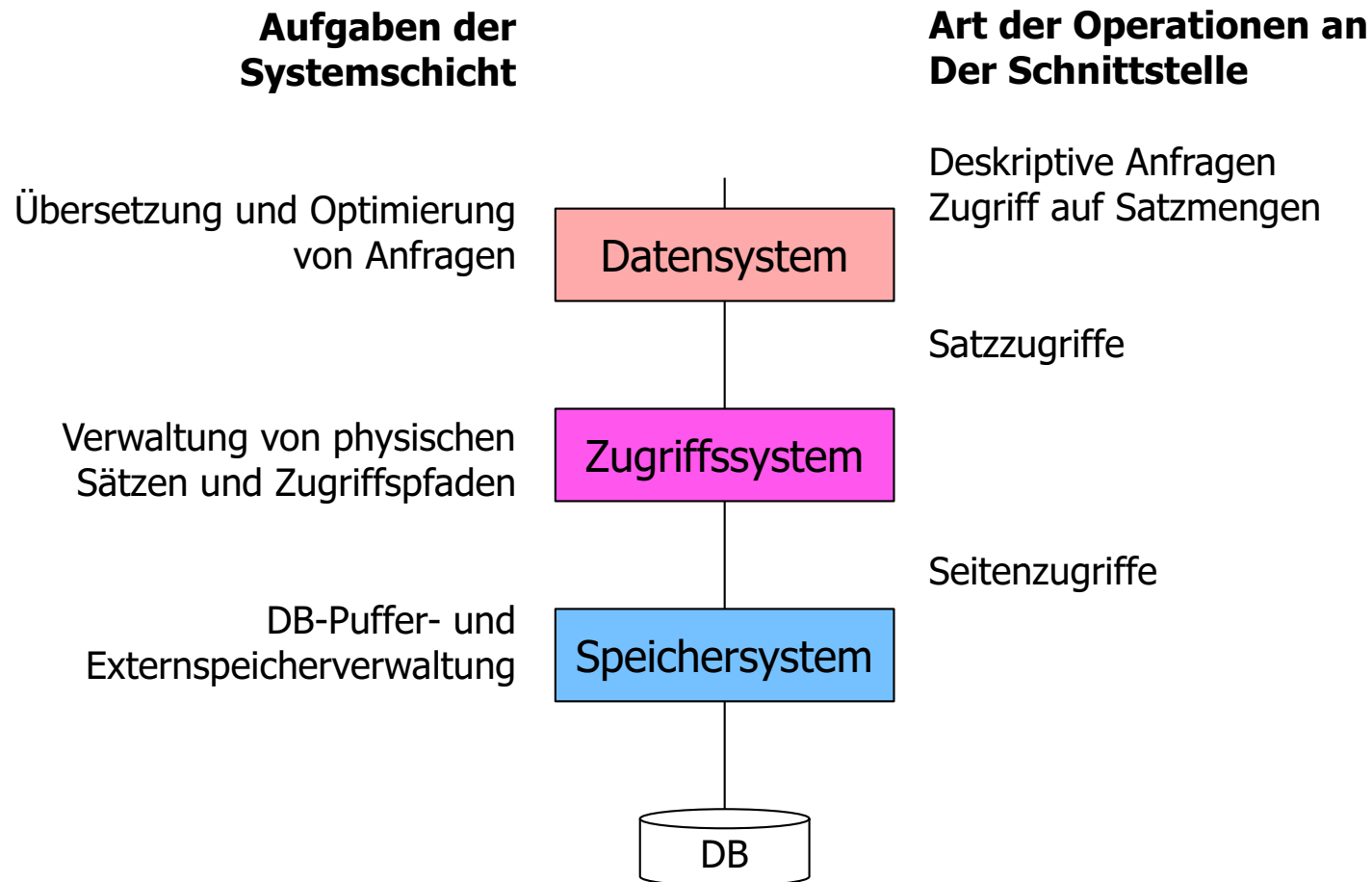
# Transaktionskonzept

- DBS unterstützt Mehrbenutzerbetrieb, Zuverlässigkeit, Fehlertoleranz und Integrität der Daten
- Transaktionskonzept als neues Verarbeitungsparadigma
  - ist Voraussetzung für die Abwicklung betrieblicher Anwendungen (mission-critical applications)
  - erlaubt "Vertragsrecht" in rechnergestützten IS zu implementieren
- Transaktion erlaubt "Bündelung" mehrere DB-Zugriffsoperationen zu einer Einheit
  - Beispieltransaktion "Überweisung" (Operationen 1 – 6)
    - (1) Lese Kontostand von A in a;
    - (2)  $a := a - 100$ ;
    - (3) Schreibe a in DB;
    - (4) Lese Kontostand von B in b;
    - (5)  $b := b + 100$ ;
    - (6) Schreibe b in DB;

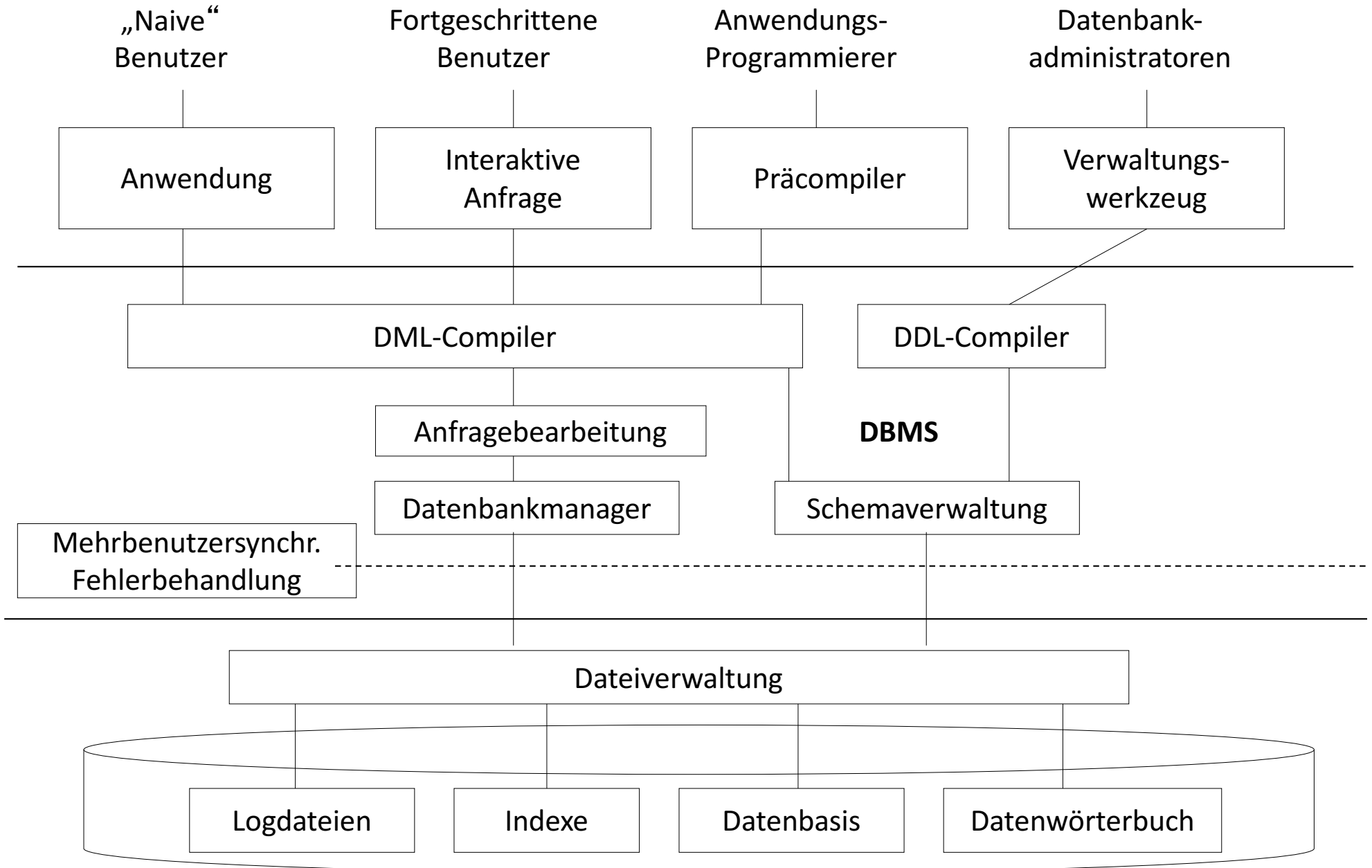
# Eigenschaften von Transaktionen (ACID-Paradigma)

- **Atomicity (Atomarität)**
  - TA ist kleinste, nicht mehr weiter zerlegbare Einheit („alles-oder-nichts“)
  - Entweder werden alle Änderungen der TA festgeschrieben oder gar keine
- **Consistency (Konsistenz)**
  - Endzustand einer TA muss die definierten Integritätsbedingungen erfüllen
  - andernfalls wird sie komplett (siehe Atomarität) zurückgesetzt
  - Zwischenzustände während der TA-Bearbeitung dürfen inkonsistent sein
- **Isolation**
  - Nebenläufig (parallel, gleichzeitig) ausgeführte TA dürfen sich nicht gegenseitig beeinflussen
  - Alle anderen parallel ausgeführten TA bzw. deren Effekte dürfen nicht sichtbar sein
- **Durability (Dauerhaftigkeit)**
  - Wirkung einer erfolgreich abgeschlossenen TA bleibt dauerhaft in der DB
  - auch nach einem Systemfehler (HW- oder System-SW) gewährleistet

# Architektur: Vereinfachtes DBMS-Schichtenmodell



# Architekturübersicht eines DBS (vereinfacht)



# DBS im Einsatz

- Wie sieht der Herstellermarkt aus?
  - Kommerzielle Systeme
    - Oracle, IBM, Microsoft mit den größten Anteilen
    - daneben Sybase, NCR Teradata
  - OpenSource-System
    - MySQL, Postgres, Derby, ...
- Für **strukturierte** Daten
  - Starke Dominanz von relationalen DBMS
  - Hierarchische DBMS und (flache) Dateien spielen weiterhin eine Rolle
- Trend für DBS
  - 85% der weltweit verfügbaren Daten sind semi- oder unstrukturiert (Internet, wiss. Aufzeichnungen und Experimente usw.)
  - Unterstützung für XML, Dokumente, Information Retrieval, Multimedia, ...



# Zusammenfassung

- Information und Informationssysteme
  - wichtige Anwendungsklassen für
    - operierende Ebene: OLTP
    - planende/kontrollierende Ebene: DW, OLAP
    - strategische Ebene: OLAP, DM
- Einsatz von DBS zur Bewältigung von wiederkehrenden, kritischen Datenverwaltungsaufgaben
  - Vermeidung von Redundanzen, Inkonsistenzen
  - Flexible und mächtige Zugriffs- und Verknüpfungsmöglichkeiten
  - Unterstützung von Mehrbenutzerbetrieb
  - Zuverlässigkeit und Fehlertoleranz
  - Datenintegrität und Zugriffskontrolle

# Zusammenfassung (2)

- Wichtige Konzepte und Eigenschaften von DBS
  - Trennung von Schema und Ausprägung
    - Schema weitgehend zeitinvariant, Ausprägung ändert sich laufend
  - Datenmodelle zur Modellierung einer Miniwelt
    - konzeptuelle, logische und physische Modellierung
  - Datenabstraktion und Datenunabhängigkeit
    - gezielte Unterstützung von Benutzergruppen
    - Anwendung bleibt isoliert von Änderungen der internen und physischen Strukturmodelle
  - Datenbanksprachen
    - Datendefinition und Datenmanipulation
    - deklarative, mengenorientierte Anfragen
  - Transaktionskonzept (ACID) als wichtiges Verarbeitungsmodell
- Vereinfachte (Schichten-) Architektur für DBS