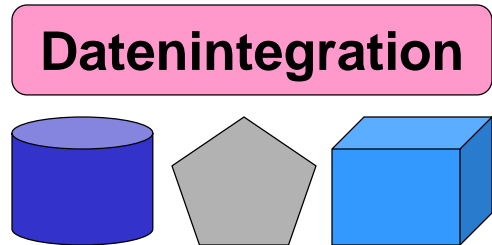


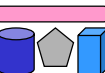
# Datenintegration



## Kapitel 2: Verteilung, Autonomie und Heterogenität

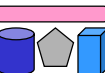
**Dr. Anika Groß**  
**Sommersemester 2016**

**Universität Leipzig**  
**Institut für Informatik**  
**<http://dbs.uni-leipzig.de>**



# Inhalt

- Verteilung
  - Physische Verteilung
  - Logische Verteilung
- Autonomie
  - Designautonomie
  - Kommunikationsautonomie
  - Ausführungsautonomie
- Heterogenität
  - Technische Heterogenität
  - Syntaktische Heterogenität
  - Datenmodellheterogenität
  - Strukturelle Heterogenität
  - Semantische Heterogenität



# Verteilung (Distribution)

- Ein verteiltes Informationssystem ist eine Sammlung mehrerer, logisch verknüpfter Informationssysteme, die über ein gemeinsames Netzwerk verteilt sind.

(Özsu, Valduriez: Principles of Distributed Database Systems. Prentice-Hall, 1991)

- Anwendungsentwicklung
  - Ohne Spezifikation der physischen Präsenz der Komponenten
  - Häufige Techniken: HTTP, CORBA, ...
  - Transparenz bzgl. Speicherort, Netzwerk, ...
- Arten von Verteilung
  - Physische Verteilung
  - Logische Verteilung

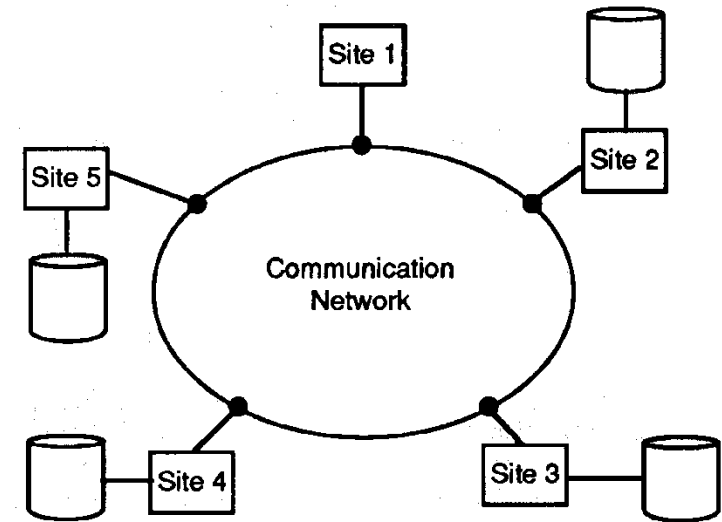
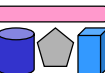
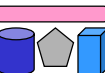


Figure 1.7 DDBS Environment



# Physische Verteilung

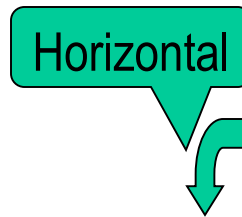
- Motivation: Hardware-Anforderungen
  - Höhere Sicherheit (disaster protection)
  - Lokale Nähe von Servern zu Clients
  - Historisch begründete Orte
  - Physikalische Einschränkungen (Hitze, Gewicht, Energie)
  - Monetäre Gründe (Grid, Cloud)
- Server stehen an unterschiedlichen Orten
  - Gleiches Gebäude, anderer Raum
  - Anderes Gebäude
  - Andere Stadt, anderes Land
- Shared Nothing
  - Server haben keine gemeinsamen, abhängigen Hardwarekapazitäten
    - Memory, Disk, CPU, ...
  - Mit Ausnahme des Netzwerks
  - Im Gegensatz zu shared-disk und shared-memory



# Logische Verteilung

- Motiviert: Anwendungsanforderungen
  - Zuverlässigkeit (bei Ausfall eines Servers)
  - Verfügbarkeit (bei Ausfall eines Netzwerkteils)
  - Effizienz
- Redundanz
  - Replikation
  - Caching
- Partitionierung
  - Vertikal
  - Horizontal

Horizontal



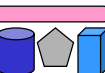
<u>Id</u>	Name	Wohnort
1	Müller	Leipzig
2	Meier	Berlin
3	Schulz	Dresden

Vertikal



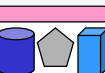
# Physische vs. logische Verteilung

- Definition von logischer Verteilung ist anwendungsabhängig, die von physischer Verteilung nicht
- Daten können *logisch verteilt* sein, obwohl sie *physisch unverteilt* sind
  - Schema verdoppeln und Daten verteilen
  - Zwei Filmquellen unverändert in ein Schema kopieren
- Daten können *physisch verteilt* sein, obwohl sie *logisch unverteilt* sind
  - Replikation und Caching:
    - Klare Master – Slave Beziehung
  - Performanzsteigerung durch Partitionierung
    - Auftrennung nach festen Kriterien
  - Verteilte Datenbanken
    - Strenge Kontrolle des „wo“ von Daten



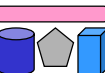
# Vor- und Nachteile der Verteilung

- Vorteile aus Sicht der Quellen und des IIS
  - Autonomie (gleich genauer)
  - Performance: Kapazität dort, wo sie gebraucht wird
  - Verfügbarkeit: Bei Ausfall eines Standorts
  - Erweiterbarkeit
  - Teilbarkeit (Verantwortung bei anderen Organisationseinheiten)
- Nachteile aus Sicht des IIS
  - Komplexität (Verwaltung, Optimierung)
  - Kosten
  - Sicherheit
  - Autonomie



# Autonomie

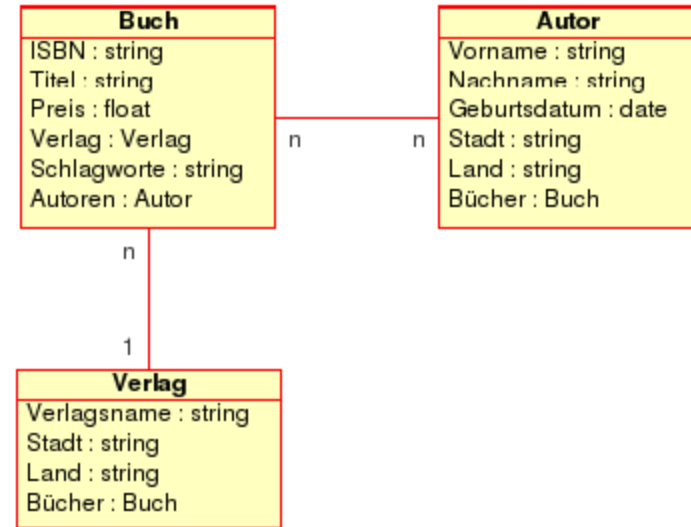
- Der Grad zu dem verschiedene IS unabhängig operieren können
- Bezieht sich auf Kontrolle, nicht auf die Daten selbst
  - (Recht auf) Weiterentwicklung, Administration
- **Klassen nach** [Özsu, Valduriez: Principles of Distributed Database Systems. Prentice-Hall, 1999]
  - Designautonomie
  - Kommunikationsautonomie
  - Ausführungsautonomie
- **Weitere Klassen**
  - Schnittstellenautonomie
    - Technischer Zugriff: Webservice+SOAP, Web+HTML, SQL+Tupel, ...
  - Zugriffsautonomie
    - Authentifizierung, Rechtevergabe, ...
  - Juristische Autonomie
    - Zugriff eingeschränkt für bestimmte Verwendungen (Copyright, Screen Scraping)





# Designautonomie (Entwurfsautonomie)

- Freiheit bezüglich ...
  - Datenmodell
    - Relational, hierarchisch, XML
    - (Siehe Beispiel rechts)
  - Schema
    - Abdeckung der Domäne (universe of discourse, miniworld)
    - Grad der Normalisierung
    - Benennung
  - Transaktionsmanagement
- Freiheit dies jederzeit zu ändern!
  - Besonders problematisch!

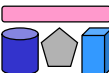


```
<xs:element name="Autor" minOccurs="0" maxOccurs="unbounded">
- <xs:complexType>
- <xs:sequence>
  <xs:element name="Vorname" type="xs:string" />
  <xs:element name="Nachname" type="xs:string" />
- <xs:element name="Buch" minOccurs="0" maxOccurs="unbounded">
- <xs:complexType>
- <xs:sequence>
  <xs:element name="ISBN" type="xs:string" />
  + <xs:element name="Verlag" minOccurs="1" maxOccurs="1">
  </xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
```

# Kommunikationsautonomie

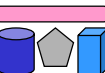
- Freiheit bezüglich der Wahl ...
  - mit Wem kommuniziert wird
    - z.B. Sperren von Clients bei zu vielen Zugriffen
  - Wann mit anderen Systemen kommuniziert wird
    - jederzeit Eintritt/Austritt aus integriertem System, Priorisierung von Zugriffen
  - Was (welcher Teil der Information) kommuniziert wird
    - z.B. beschränkte Anfrageergebnisse
  - Wie mit anderen Systemen kommuniziert wird (Anfragemöglichkeiten)
    - Anfragesprache und -prädikate, Sortierung, Write

<i>Beispiel</i>	SQL-Zugriff per JDBC	HTML Formular auf Website
Mit Wem?		
Wann?		
Was?		
Wie?		



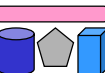
# Ausführungsautonomie

- Freiheit bezüglich der Wahl ...
  - wann Anfragen ausgeführt werden
  - wie Anfragen ausgeführt werden
  - der Scheduling-Strategien
  - Optimierungs-Strategien
  - ob globale Transaktionen unterstützt werden
  
- Beispiel:
  - Optimierung und Scheduling
    - Behandlung externer vs. lokaler Anfragen
    - Golden customers
    - Garantierte Antwortzeiten
  - Transaktionen
    - Dirty-read egal?



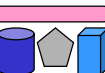
# Verteilung → Autonomie → Heterogenität

- Verteilung als „Ursache“ für Autonomie
- Autonomie als „Ursache“ für Heterogenität
  - Gestaltungsfreiheit → Unterschiedliche Entscheidungen → Heterogenität
- Besonders schlimm: Softwareentwickler
  - Das Recht, alles dauernd zu ändern
  - „Not invented here“ Syndrom
  - Wiederverwendung als ewiger Traum
- Standards grenzen Autonomie ein



# Heterogenität

- Heterogenität herrscht, wenn sich zwei miteinander verbundene Informationssysteme syntaktisch, strukturell oder inhaltlich unterscheiden.
- Heterogenitäten zu überbrücken ist die Kernaufgabe der Informationsintegration.
  - Erstellung oder “Erwecken des Anscheins” eines homogenen Systems
- Arten von Heterogenitäten
  - Technische Heterogenität
  - Syntaktische Heterogenität
  - Datenmodellheterogenität
  - Strukturelle Heterogenität
  - Semantische Heterogenität
- Weitere Klassifikationen möglich
  - Klare Trennung der Arten nicht immer möglich
  - Häufig Kombinationen von mehreren Arten in der Praxis



# Heterogenitäten: Übersicht

Gelöst, wenn ...

- Technische Heterogenität
  - Technische Realisierung des Datenzugriffs (Schnittstelle)
  - Technische Unterschiede in der Darstellung
- Syntaktische Unterschiede
  - Unterschiede in der Darstellung
  - Gleiche Dinge syntaktisch verschieden repräsentieren
- Datenmodellheterogenität
  - Unterschiede im verwendeten Datenmodell (z.B. hierarchisch, relational)
- Strukturelle Heterogenität
  - Strukturelle Unterschiede in der Darstellung
  - Gleiche Dinge verschieden modellieren (z.B. verschiedene DB-Schemas)
- Semantische Heterogenität
  - Bedeutungsunterschiede von Namen (Schema und Daten); Gleiches sagen, verschiedenes meinen (oder andersrum)

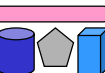
Das IntInfoSys kann eine Anfrage absetzen und kriegt „was“ zurück

In dem „was“ sind gleiche Dinge auch gleich dargestellt

Die Quelle liefert das „was“ im Datenmodell des IntInfoSys

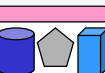
Die Quelle liefert das „was“ im Schema des IntInfoSys

Die Quelle meint mit Begriffen dasselbe wie das IntInfoSys



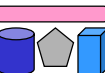
# Technische Heterogenität

- Anfragemöglichkeit / Schnittstelle
  - Anfragesprache
    - SQL, XQuery, ...
  - parametrisierte Funktionen
    - Webservice-Aufrufe
  - Formulare (“canned queries”)
    - HTML-Formulare
- Austauschformat
  - Binärdaten, XML, HTML, tabellarisch
- Kommunikationsprotokoll
  - HTTP, JDBC, SOAP, ...



# Heterogenität von Anfragesprachen

- Negation vs. keine Negation
  - Oft zu teuer
- Gleichheit / Ungleichheit
  - „=“ oder auch „>, <, ≥, ≤“
- Konjunktion (UND)
  - oder auch Disjunktion (ODER)
- Prädikate nur mit Konstanten (author = „Melville“)
  - Oder auch mit anderen Variablen (ResidenceCountry = Nationality)
- Gebundene und freie Variablen später
  - Feld muss mit Wert belegt sein (gebunden) oder kann unspezifiziert bleiben (frei)
- Andere Einschränkungen
  - Joins über maximal 3 Relationen
  - z.B. Prädikate nur über eine Auswahl von Werten





# Heterogenität von Anfragesprachen: Beispiele

## Erweiterte Suche Bücher

Je mehr Felder Sie ausfüllen, desto zielgerichteter können wir suchen. Es reicht jedoch aus, nur eines der Felder auszufüllen.

Gebundene / freie Variablen

ISBN: (10- oder 13-stellig, ohne Bindestriche)

Autor/in:

Titel:

Schlagwörter:

Verlag:

Verfeinern Sie Ihre Suche, indem Sie nur nach bestimmten Buchformaten suchen.

Nur gebraucht:

Format:

Ordnen nach:

Veröffentlichungsdatum:

Suche in:

Jetzt suchen

Feste Auswahl von Werten

Konjunktion / Diskjunktion

Nachrichten suchen

Nachrichten suchen in:

Unterordner durchsuchen

Alle Bedingungen erfüllen

Mindestens eine Bedingung erfüllen

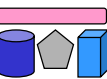
Betreff	enthält	Integration	+	-
Von	enthält	Rahm	+	-
Alter in Tagen	ist		+	-

Prädikate

ist  
ist größer als  
ist kleiner als

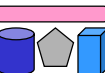
Vergleiche

Öffnen   Ablegen   Löschen   Ordner öffnen   Als virtuellen Ordner speichern...



# Heterogenität von Anfragesprachen: Probleme

- In einzelnen Systemen kein Problem
  - Aber: Probleme für integrierte Systeme
    - Überwindung der Heterogenität zwischen globaler Anfragesprache des integrierten Systems und lokaler Anfragesprache der Quelle
1. Globale Anfragesprache ist mächtiger als lokale Anfragesprache
    - Anfragen eventuell nicht ausführbar
    - Oder globales System muss kompensieren
  2. Lokale Anfragesprache ist mächtiger als globale Anfragesprache
    - Verpasste Chance, lokale (effiziente) Ausführung auszunutzen
  3. Gebundene und freie Variablen sind inkompatibel
    - Anfragen eventuell nicht ausführbar
  4. Übersetzung von Anfragesprachen notwendig
    - SQL – XQuery, SQL – HTTP, Web-Service – SQL, etc.
    - Oft nicht einfach möglich, da unterschiedliche Konzepte



# Globale Anfragesprache mächtiger als lokale

- Beispiel: Integr.System erlaubt SQL, Datenquelle "nur" HTML-Formular

```
SELECT *  
FROM Books  
WHERE Author = "Defoe"  
AND Year = 1979
```

Defoe, Robinson Crusoe, 1979

3 ↑ Year = 1979

Int.System kompensiert  
Anfragemächtigkeit

Defoe, Robinson Crusoe, 1986  
Defoe, Robinson Crusoe, 1979  
Defoe, Moll Flanders, 1933

1

Autor/in: Defoe

Titel:

Schlagwörter:

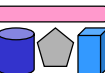
ISBN:

Verlag:

2

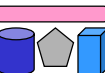
Ignoriere Jahr-Bedingung, da  
kein entsprechendes Prädikat

Was passiert bei anderen Anfragen?



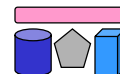
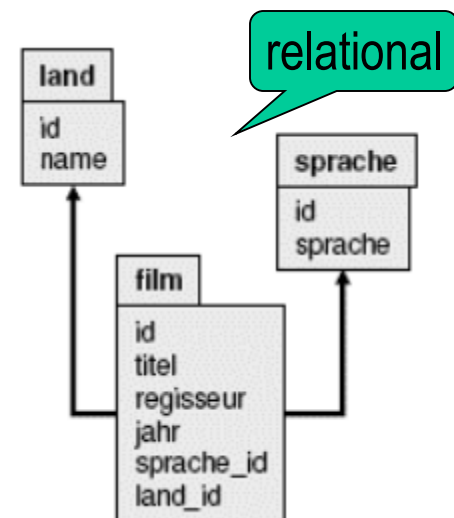
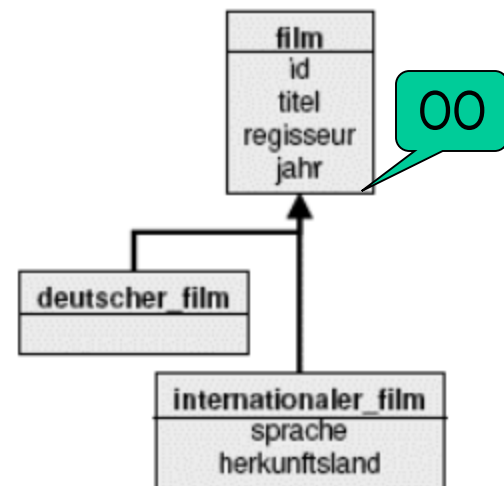
# Syntaktische Heterogenität

- Unterschiedliche Darstellung desselben Sachverhalts
  - Dezimalpunkt oder –komma
  - Euro oder €
  - Comma-separated oder tab-separated
  - HTML oder ASCII oder Unicode
  - Notenskala 1-6 oder „sehr gut“, „gut“, ...
  - Binärcodierung oder Zeichen
  - Datumsformate (18. April 2008, 18.4.2008, 4/18/2008, ...)
- Überwindung in vielen (einfachen) Fällen nicht problematisch
  - Umrechnung, Übersetzungstabellen, ...
- Probleme u.a. bei “nicht-standardisierten” Darstellungen
  - Beispiel: Konferenznamen
    - “VLDB 2008”, “Very Large Databases, 2008”, “34. VLDB”, ...



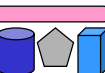
# Datenmodellheterogenität

- Typische Datenmodelle
  - Relational, XML, Objektorientiert
    - Siehe Beispiel rechts
  - Domänenspezifisch (OPEN-GIS, ...)
  - Proprietär (UniProt, ...)
- Unterschied: Zum Austausch oder zur Speicherung
  - XML als Speicherformat?
  - Black-Box-Sicht – was zählt, ist was die Quelle liefert
- Erfordert Konvertierung
  - Spezielle Semantik geht unter Umständen verloren
  - XML-Schachtelung im relationalen Modell? Part-of? Is-a?



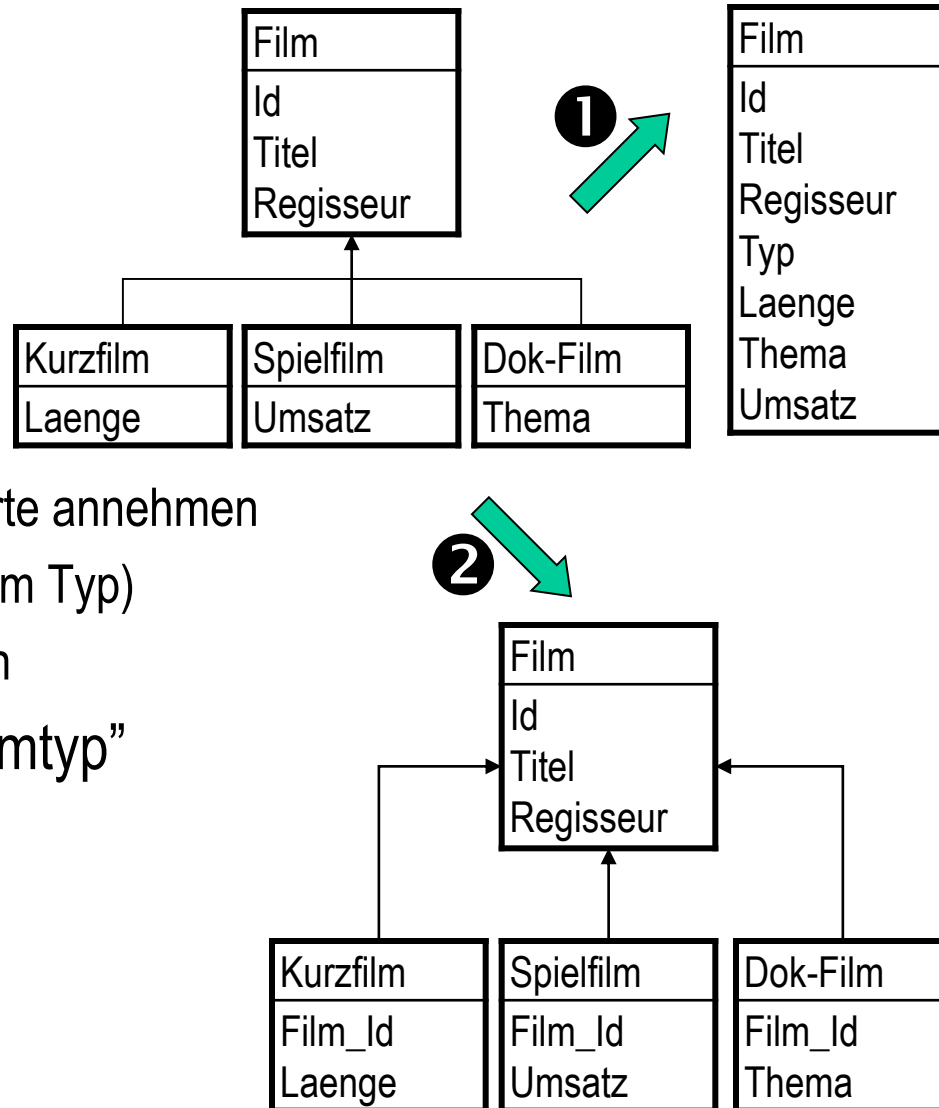
# Strukturelle Heterogenität

- Allgemein
  - Gleiche Dinge in unterschiedlichen Schemata ausdrücken
    - Andere Aufteilung von Attributen auf Tabellen
    - Setzt intensionale Überlappung voraus („gleiche Dinge“)
  - Kann meist durch Anfragesprachen überwunden werden
    - Sichten / Views
  - Meistens mit semantischer Heterogenität verbunden
    - Ausnahme: 1:1 Beziehungen
- Spezialfall: Schematische Heterogenität
  - Verwendung anderer Elemente eines Datenmodells
  - Kann meist nicht durch Anfragesprachen überwunden werden



# Strukturelle Heterogenität: Beispiel

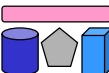
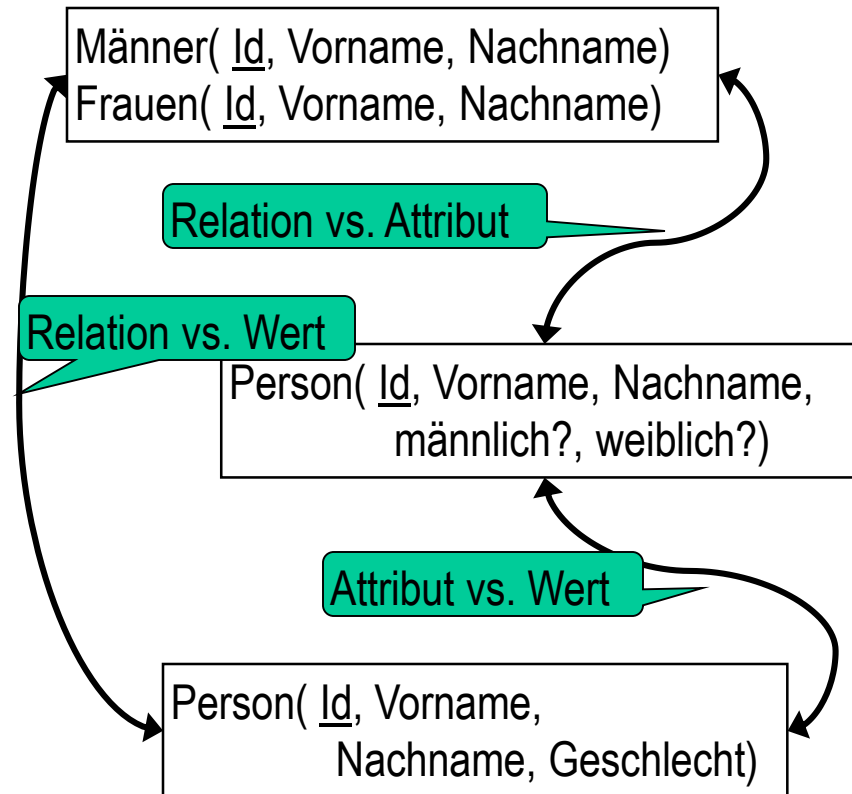
- Heterogenität auf Grund unterschiedlichen Abbildungen eines OO-Modells in ein relationales Modell
- Gleichwertig? Ja, aber nur durch zusätzliche Integritätsbedingungen
  - 1: Attribut Typ darf nur bestimmte Werte annehmen
  - 1: Umsatz nicht immer belegt (abh. vom Typ)
  - 2: Keine gleichen Film\_Id's in Tabellen
- Anfragen: “Anzahl der Filme pro Filmtyp”



# Schematische Heterogenität: Problemfälle



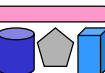
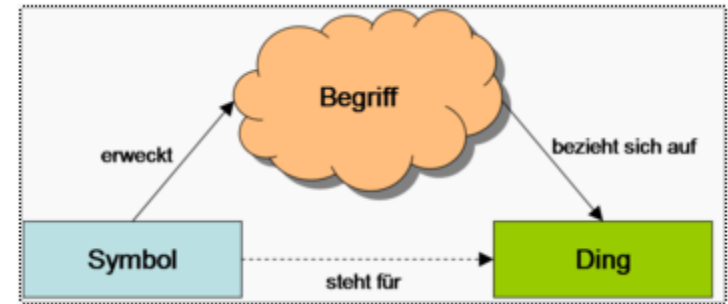
- Modellierung
  - Relation vs. Attribut
  - Relation vs. Wert
  - Attribut vs. Wert
  - Siehe Beispiel rechts
- Normalisiert vs. Denormalisiert
  - Attribut im Tupel vs. Auslagern in Tabelle
- Geschachtelt vs. Fremdschlüssel
  - 1:N-Assoziationen
- Weitere Konflikte
  - Fehlende (evtl. ableitbare) Attribute
  - Integritätsbedingungen, z.B. unterschiedliche Datentypen





# Semantische Heterogenität

- Fremdwörterduden zu “Semantik”
  - „Teilgebiet der Linguistik, das sich mit den Bedeutungen sprachlicher Zeichen und Zeichenfolgen befasst“
  - „Bedeutung, Inhalt eines Wortes, Satzes oder Textes“
- „Semantische Heterogenität ist ein überladener Begriff ohne klare Definition. Er bezeichnet die Unterschiede in Bedeutung, Interpretation und Art der Nutzung.“ [Özsu, Valduriez: Principles of Distributed Database Systems. Prentice-Hall, 1991]
- Semantik vs. Syntax
  - Programmiersprachen
    - Syntax: EBNF, Grammatiken, ...
    - Semantik: Wirkung der Ausführung (operationale Semantik, Fixpunktsemantik, ...)
  - Natürliche Sprachen
    - Syntax: formale Struktur (Worte, Phrasen, Sätze) → “Ich esse Butterbrot ein.”
    - Semantik: inhaltliche Struktur (Referenz, Bedeutung) → “Ich esse einen Schrank.”



# Semantik vs. Struktur

- Strukturelle Heterogenität
  - Betrifft Schemas, Bedeutung der Labels im Schema egal
  - Annahme bisher: Gleiche Label -> Gleiche Semantik

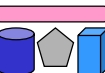
```
Männer( Id, Vorname, Nachname)  
Frauen( Id, Vorname, Nachname)
```

```
A( Id, X, Y)  
B( Id, X, Y)
```

```
Person( Id, Vorname, Nachname,  
        Männlich, weiblich)
```

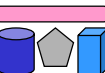
```
P( Id, X, Y, a, b)
```

- Semantische Heterogenität
  - Betrifft Daten, d.h deren „Bedeutung“



# Semantische Heterogenität: Problemfälle

- Definition von Konzepten/Begriffen
  - Uneinigkeit, was ein Konzept ist (Gen, Transaktion, Bestellung, Mitarbeiter, ...)
  - Semantisch überlappende Weltausschnitte mit einander entsprechenden Klassen
- “-nym” Worte
  - Synonym: verschiedene Worte, gleiche Semantik (Bsp: senkrecht, vertikal)
  - Homonym: gleiche Worte, verschiedene Semantik (Bsp: Bank, Schloss)
  - Auto-Antonym: gleiche Worte, gegenteilige Semantik (Bsp: Untiefe, übersehen)
  - ...
- Einheiten
  - cm vs. inch, \$ vs. €



# Konzepte und Kontext (Beispiel)

- Wieviele Mitarbeiter hat IBM?

- Konzept: Mitarbeiter
- Konzept: IBM
- Weitere Konzepte, z.B. Zeitpunkt

Auch temporäre MA, Diplomanden, Berater? Stellen oder Köpfe?

Welche Region? Welcher Geschäftsbereich?

- Wann machen Studenten der Uni Leipzig ihren Abschluss?

- Konzept: Studiendauer
- Konzept: Student

Mit Urlaubssemester? Zählen Semester an anderen Unis? Fach- oder Studiensemester?

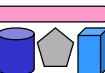
Diplom? Bachelor? Master? Nebenfach? Promotion? Lehramt?

- Semantik eines Namens hängt vom Kontext ab

- Bsp: “England = Großbritannien?” → Ja für uns, Nein für Schotten

- Wie definiert man die Bedeutung eines Namens?

- Formale Wissensrepräsentation (Ontologien, OWL)
- Dokumentieren (mit lauter Namen)
- Standards vereinbaren (schwierig)



# Zusammenfassung

- Verteilung
  - Physische und logische Verteilung
- Autonomie
  - Designautonomie, Kommunikationsautonomie, Ausführungsautonomie
  - “Eigenständigkeit” der Informationssysteme
- Heterogenität = Folge von Verteilung und Autonomie
  - Überbrückung = wichtiger Schritt bei Integration
  - Unterschiedliche Arten von Heterogenität, die “aufeinander aufbauen”
  - Technisch → Syntaktisch → Datenmodell → Strukturell → Semantisch

