

Datenbanken

Grundlagen

- Dozent: Diana Troancă
- E-mail: [dianat \[at\] cs.ubbcluj.ro](mailto:dianat@cs.ubbcluj.ro)
- Website: www.cs.ubbcluj.ro/~dianat/
- Fragen und Feedback sind immer erwünscht: per e-mail oder per persönlichem Gespräch
- Anonymes Feedback möglich: auf meine Website
- Kursanforderungen - <http://www.cs.ubbcluj.ro/~dianat/bd.php>

Struktur und Klausur

- Vorlesung: jede Woche (2 Std)
 - Praktische Übungen: jede Woche (2 Std)
 - Seminar: jede 2te Woche (2 Std)
-
- Praktisches Test: in der letzten Übungsstunde
 - Schriftliche Prüfung: während der Prüfungszeit

Folien, Literatur

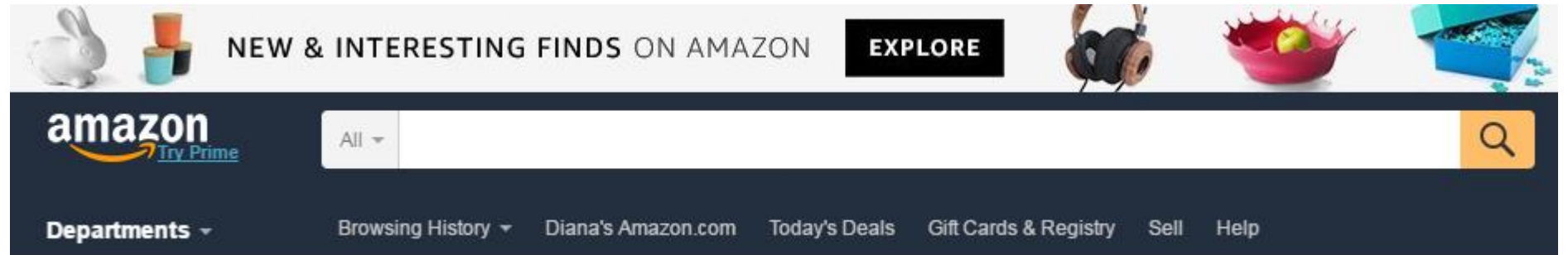
- Folien und andere Informationen zur Vorlesung, Seminar und Übungen werden unter www.cs.ubbcluj.ro/~dianat/ zur Verfügung gestellt
- Literatur:
 - **A. Kemper , A. Eickler. Datenbanksysteme – Eine Einführung. Oldenbourg Verlag, 2015. 10. Auflage.**
 - **A. Kemper, M. Wimmer. Übungsbuch Datenbanksysteme. Oldenbourg Verlag, 3. Auflage, 2012.**

Fragen?

1. Einführung

Datenbanken Grundlagen

Wo finden wir Datenbanken?



Was sind Datenbanken/ Datenbankensysteme(DBS)?

- “A collection of related data items” mit folgenden Eigenschaften:
 - Eine Datenbank repräsentiert einen bestimmten Ausschnitt der realen Welt durch einen Datenmodell
 - Eine Datenbank ist logisch konsistent und hat eine bestimmte Bedeutung
 - Eine Datenbank ist entworfen, aufgebaut und mit Daten gefüllt
 - Die Daten werden gespeichert für Aufzeichnungen (record-keeping) und Analyse

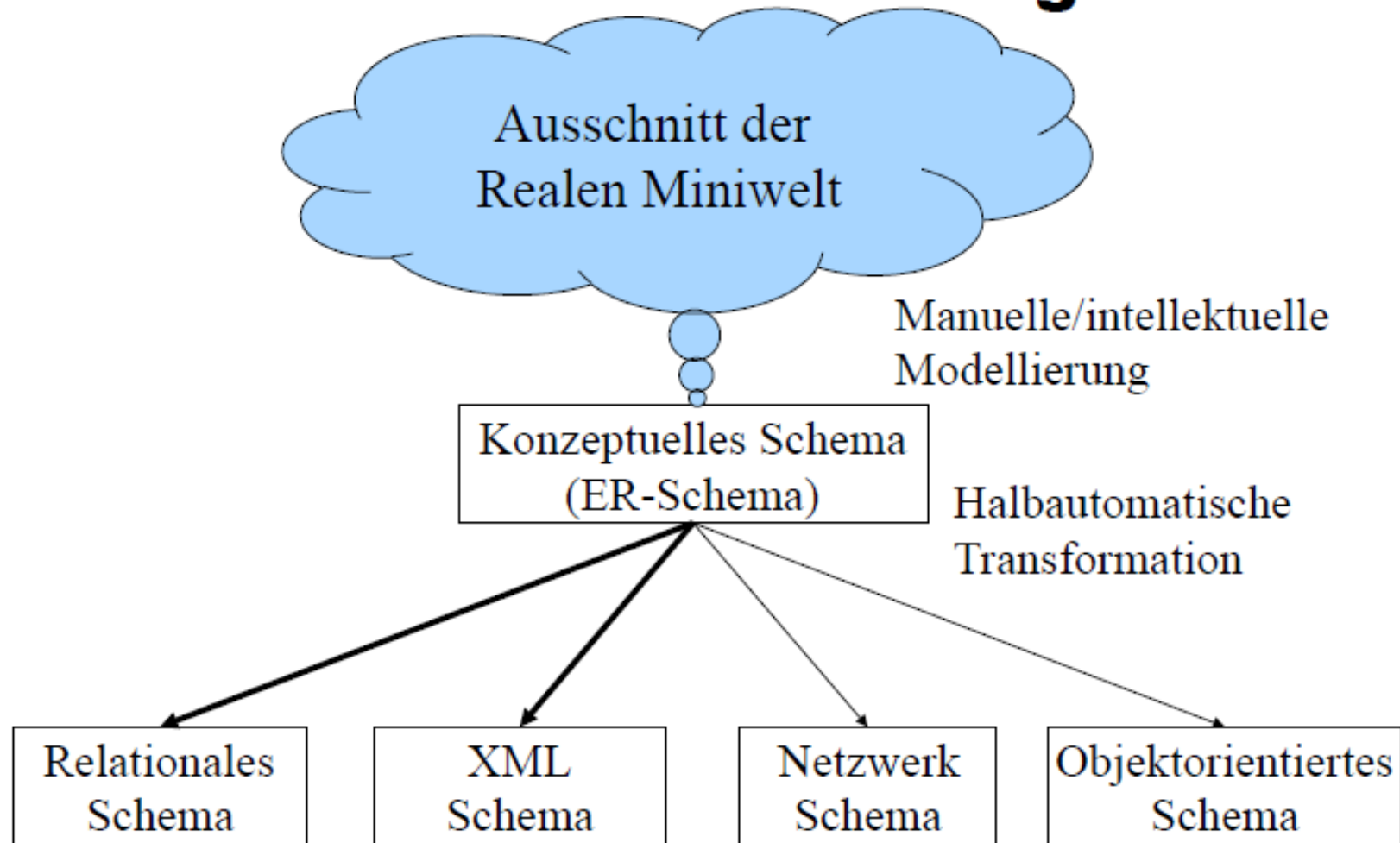
Ziel und Zweck der Datenbanken

- Datenbanken werden benutzt für effiziente Speicherung, Wiederfindung und Analyse von Daten (store and manage data)
- Einsatzgebiete für Datenbanksysteme:
 - Kontoführungsdaten bei Banken
 - Verwaltung der Kundendaten bei Versicherung
 - E-learning Plattformen
 - E-commerce Websites (Amazon, Emag, etc.)
 - Facebook
- Beispiele von non-computerized Datenbanken:
 - Telefonbuch
 - Wörterbuch

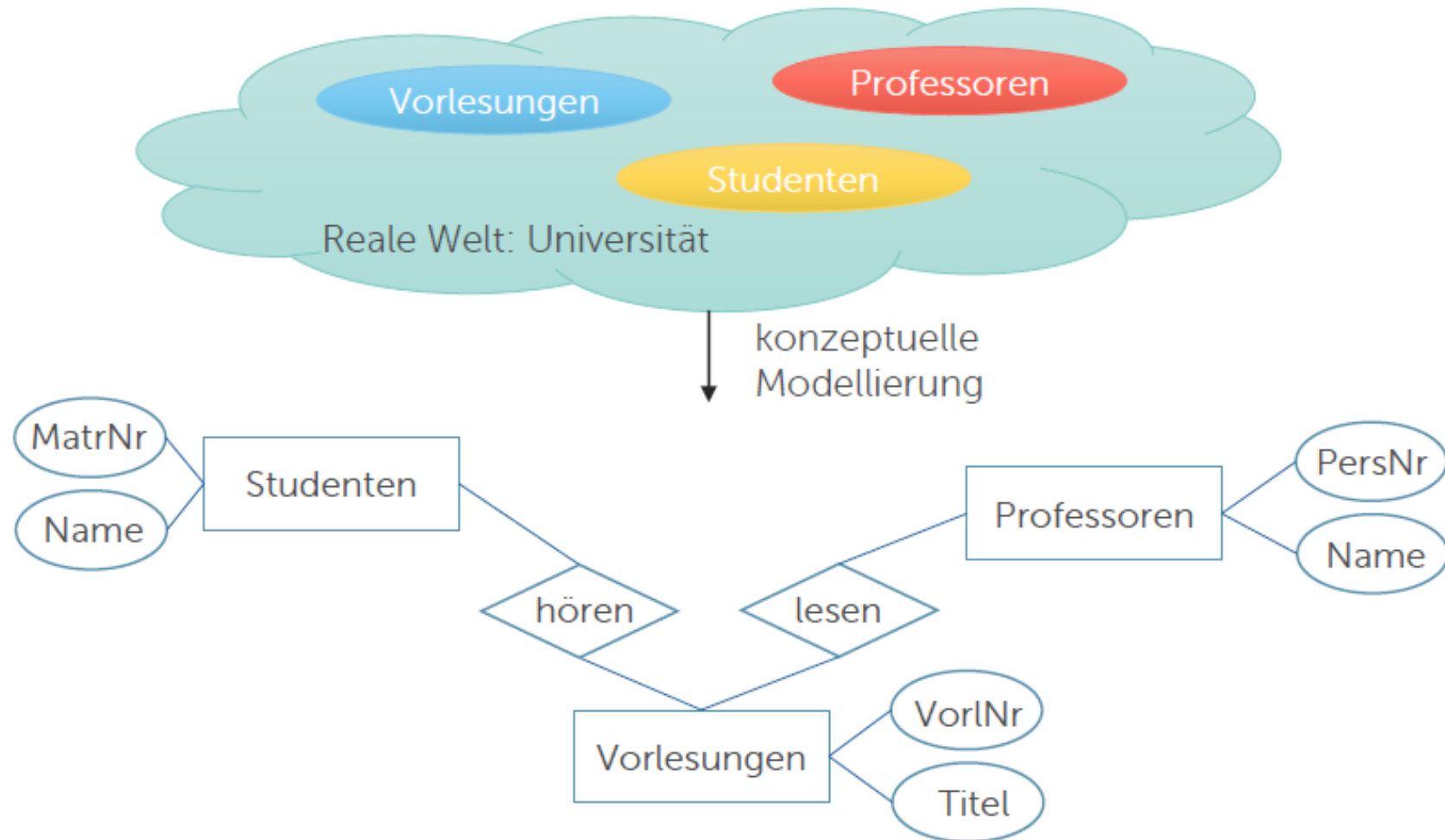
Datenmodell

- Datenmodell
 - legt fest, welche Konstrukte zur Beschreibung der Daten existieren
- Schema
 - Eine konkrete Beschreibung einer bestimmten Datensammlung, unter Verwendung eines Datenmodells

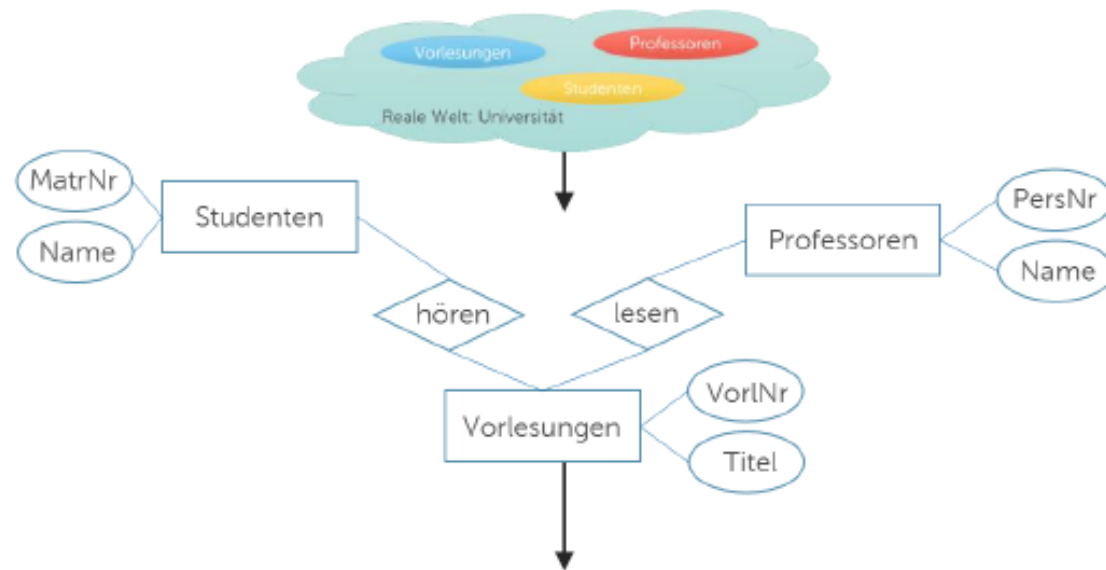
Datenmodellierung



Modellierungsbeispiel



Modellierungsbeispiel



Studenten	
MatrNr	Name
293948	Schlegel
292305	Strufe
...	...

hören	
MatrNr	VorlNr
292305	24
224833	24
...	...

Vorlesung	
VorlNr	Titel
24	DB Grundlagen
41	Betriebssysteme
...	...

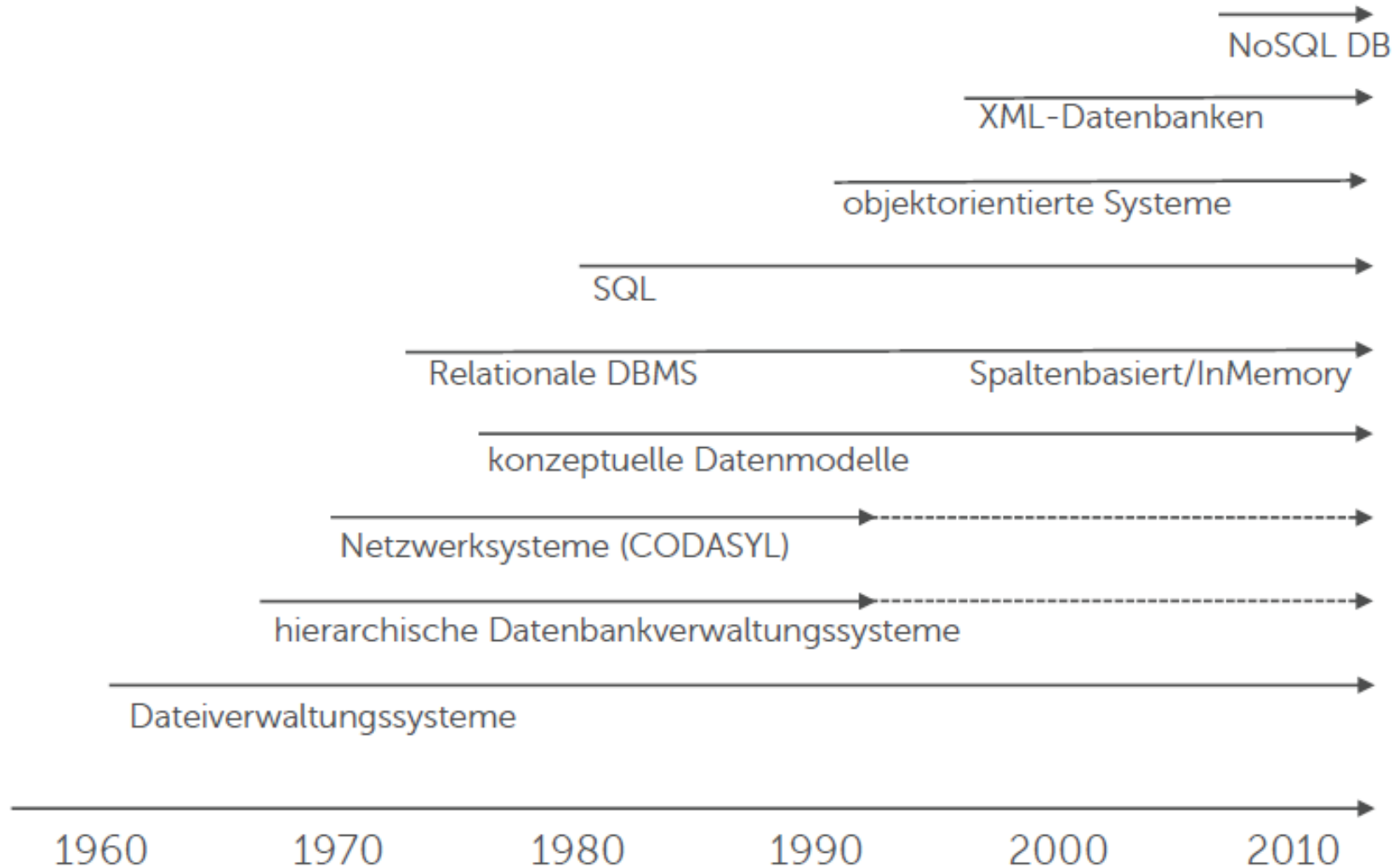
Konzeptuelle Modelle

- Entity-Relationship-Modell (ER-Modell)
- Unified Modeling Language (UML)

Logische Modelle

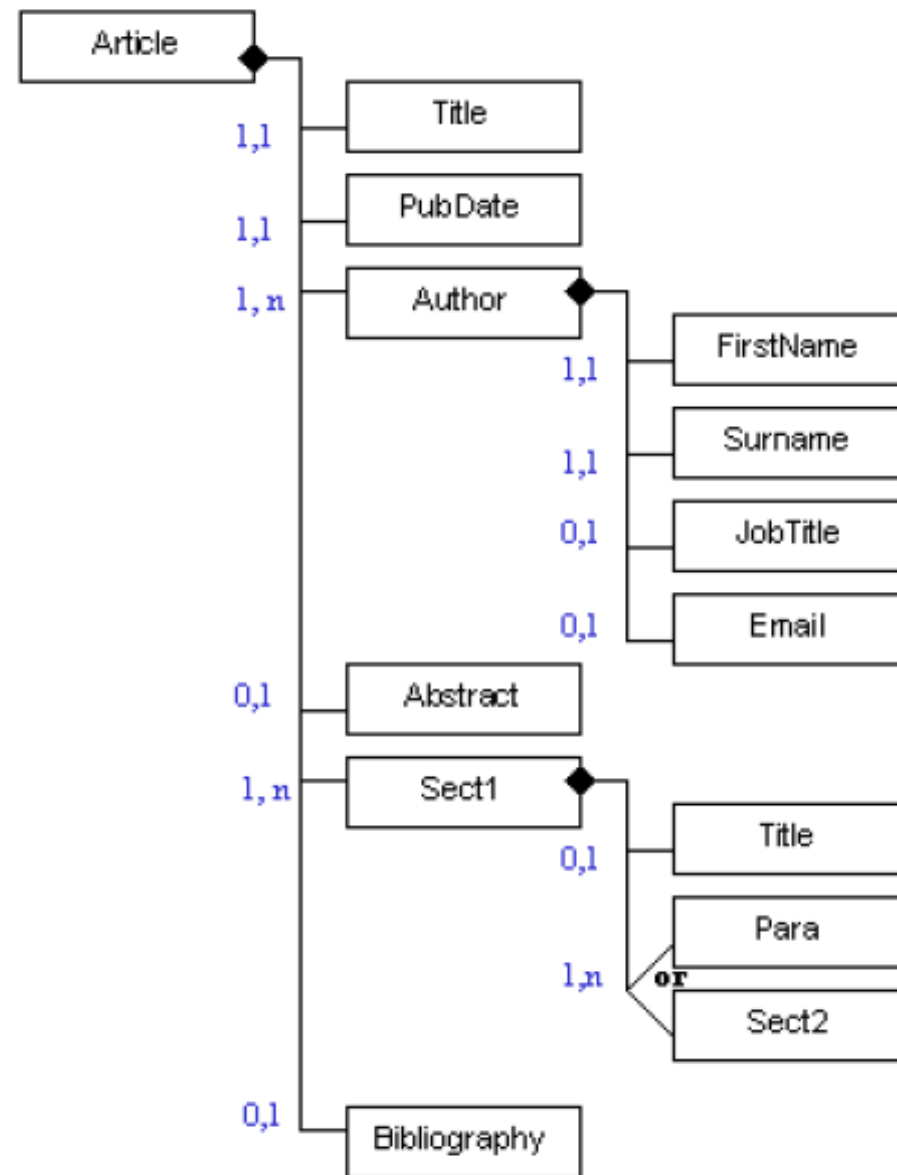
- Hierarchisches Datenmodell
- Netzwerkmodell
- Relationales Datenmodell
- Deduktives Datenmodell
- Objektorientiertes Datenmodell
- XML Schema

Historische Entwicklung von DBMS



Hierarchisches Datenmodell

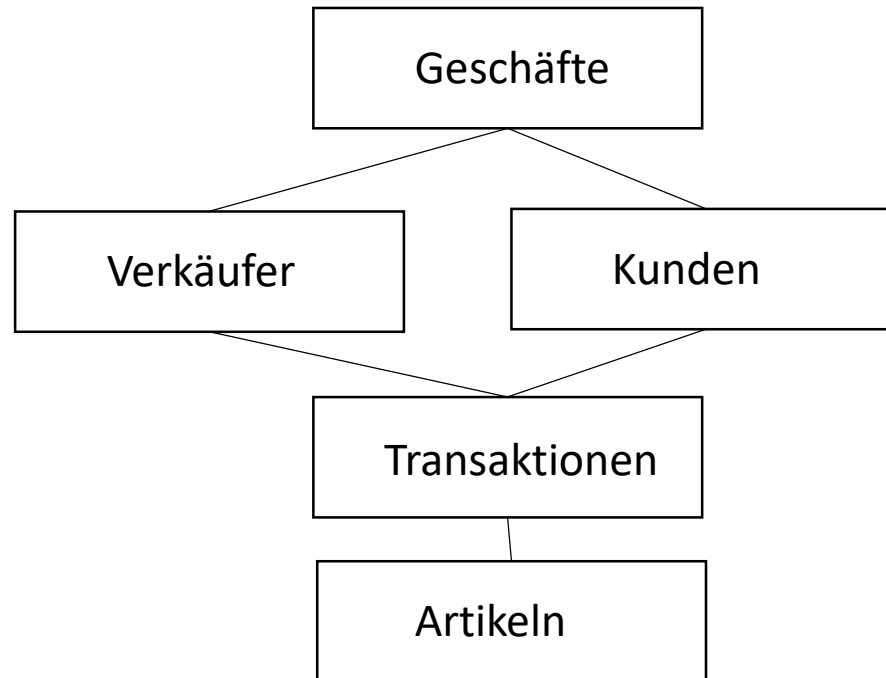
- Wurde in den 60er definiert
- Stellt die Daten in einer hierarchischen Baumstruktur dar



Entität *Article* – Hierarchisches Datenmodell

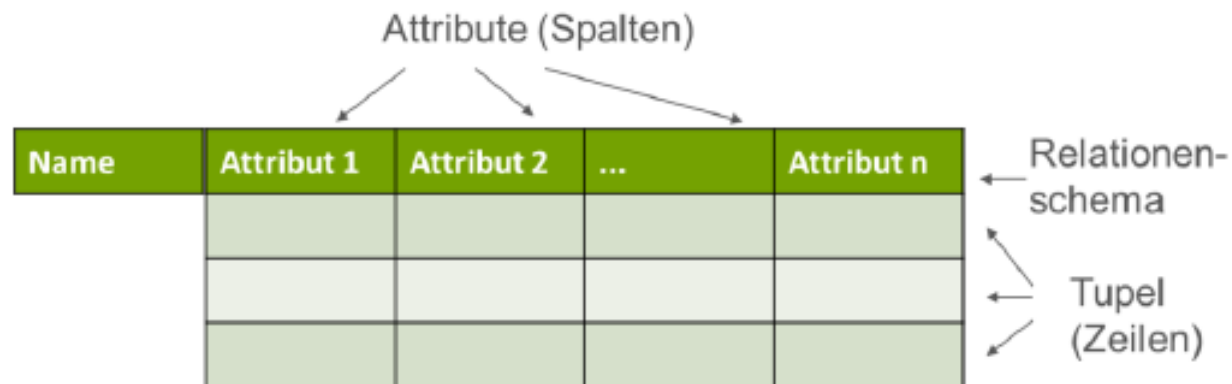
Netzwerkmodell

- Eine Erweiterung von dem Hierarchisches Datenmodell
- Stellt die Daten in Form eines Graphs dar



Relationales Datenmodell

- Wurde Anfang 70er von Ted Codd von IBM erfunden (1981 Turing Award)
- Am meisten benutztes Datenmodell (wird in den nächsten Vorlesungen ausführlich beschrieben)
- Relation als eigene Datenstruktur





Kunde

KNr	Name	Vorname	Strasse	PLZ
1	Pop	Marcel	Abator	534123
2	Martin	Alex	Teiului	324123

Relationales Datenmodell

SQL

- Ende 70er wurde die Brauchbarkeit des relationalen Modells bewiesen
- SQL (Structured Query Language) entwickelt

Objektorientiertes Datenmodell

- Konzepte: Klasse, Attribute, Methoden
- Relationen zwischen den Klassen: Assoziation, Aggregation, Vererbung
- Wird als Modell für Programmiersprachen benutzt
- In Datenbanken, aus Effizienz Gründen, nicht so viel benutzt

Schema vs. Data

- Datenbank Schema - Intension
 - beschreibt die Struktur der Datenbank (MetaDaten)
 - Zeitunabhängig (wird selten geändert)
- Ausprägung/Datenbankinstanz - Extension
 - Der Datenbankzustand zu einem bestimmten Zeitpunkt (snapshot), gegeben durch die aktuell existierenden Inhalte und Beziehungen und deren Attribute, wird **Datenbankinstanz** genannt
 - Die eigentlichen Daten einer Datenbank verändern sich im Laufe der Zeit häufig.
 - DBMS versichert, dass die Datenbank immer in einem validen Zustand ist

Schema vs. Data

- Traditionales Data Management und Analyse
 - We never deduce from the extensions to the intension
 - But, by applying new intensional knowledge (via SQL) we are able to define intensions not covered by the original model (ex. average)
- Given Big Data (billions of extensions) it's getting possible to deduce the intension, at least in a probabilistic sense

Datenbankmanagementsystem (DBMS)/ Datenbankverwaltungssystem (DBVS)

- Eine Datenbank wird von einem laufenden DBMS verwaltet und für Anwendungssysteme und Benutzer unsichtbar auf nichtflüchtigen Speichermedien (damit die Daten nicht verloren gehen) abgelegt.
- DBMS ist eine Software, die für das Datenbanksystem installiert und konfiguriert wird
- Das DBMS legt das Datenbankmodell fest
- Bietet Tools für die bequeme, mühelose Verwaltung von Daten (ohne low-level Details)

Beispiele von DBMS

- Record-based (Tuple-basierte) Datenmodelle:
 - Relationales Datenmodell (MySQL, MS SQL Server, Oracle, DB2, Informix, MS Access, FoxBase, Paradox)
 - Hierarchisches Datenmodell (IBM's DBMS)
 - Netzwerkmodell (wird in IDMS benutzt)
- Objekt-basierte Datenmodelle
 - Objektorientiertes Datenmodell (Objectstore, Versant)
 - Objektrelationales Datenmodell (Illustra, O2, UniSQL)

Schwerpunkt der Vorlesung

- Relationale Datenbanken und DBMS:
 - Etablierter Stand der Technik und bestens erforscht
 - Flexibel und universell einsetzbar
 - In allen Größen und zu allen Preisen verfügbar
 - Von vielen Tools unterstützt

Gründe für DBS-Einsatz

- Strukturierte Daten
- Effizienz und Skalierbarkeit (große Datenmengen)
- Integrität, Fehlerbehandlung und Fehlertoleranz
- Persistenz der Daten (nicht unkontrolliert verändern)
- Mehrbenutzersynchronisation
- Datenintegrität
- Deklarative Anfragesprachen: Benutzer sagt DBS was für Daten geholt werden sollen und nicht wie
- Datenunabhängigkeit: abstrakte Schichtenarchitektur