



Foliensatz Wirtschaftsinformatik

Lehrveranstaltungsinhalte

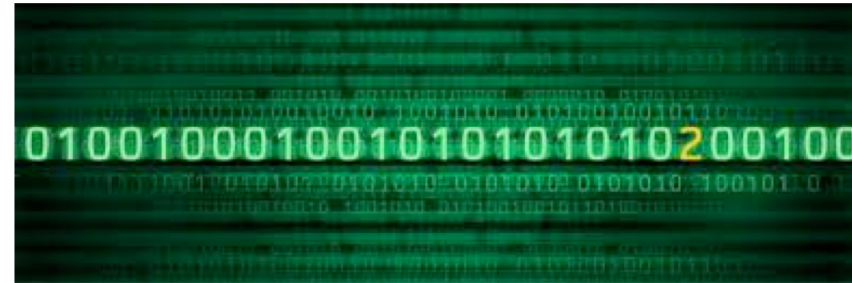
Termine	Inhalte	Kapitel	Textbuch-Seiten
1. LV-Termin	• Einführung • Informationssysteme in Wirtschaft und Gesellschaft • Geschäftsprozessmanagement	1, 2	1 – 56 57 – 96
2. LV-Termin	• Modellierung betrieblicher Informationssysteme • Unterstützung betrieblicher Leistungsprozesse durch ERP-Systeme	3, 4	97 – 134 135 – 188
3. LV-Termin	• Außenwirksame Informationssysteme und Electronic Commerce • Managementunterstützungs-systeme	5, 6	189 – 266 267 – 316
4. LV-Termin	• Planung, Entwicklung und Betrieb von Informationssystemen • Informationssicherheit und Datenschutz	7, 8	317 – 368 369 – 422
5. LV-Termin	• Datenspeicherung • Rechnersysteme	9, 10	423 – 494 495 – 539

Kapitel 9

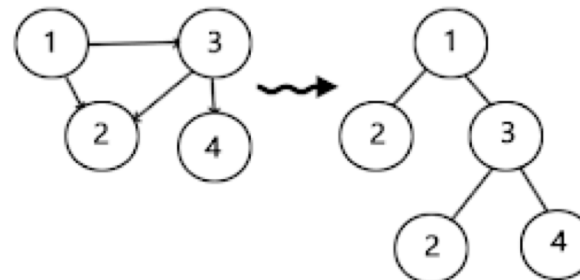
Datenspeicherung

Wiederholungsfragen Kapitel 9

- Welchen numerischen Wert hat die höchste vierstellige Dualzahl?



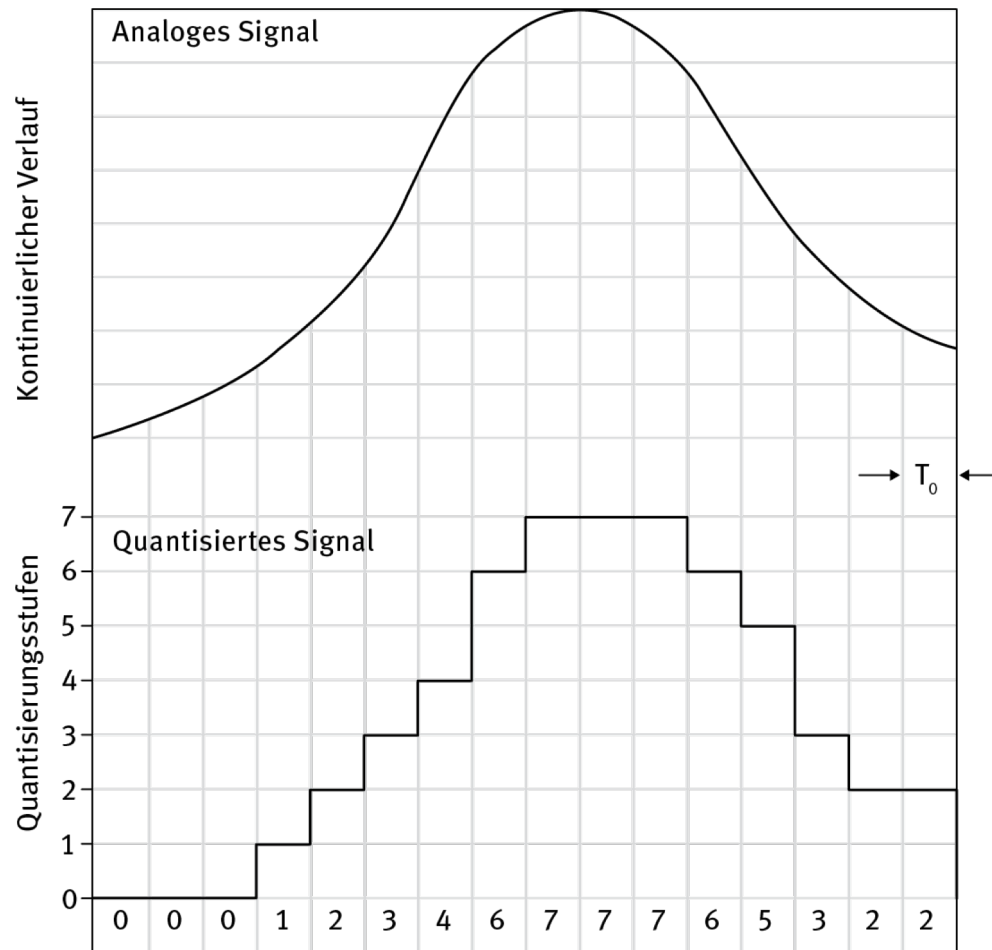
- Erklären Sie, inwieweit sich ein binärer Baum von einem gerichteten azyklischen Graphen unterscheidet. Ist jeder binäre Baum ein gerichteter azyklischer Graph und umgekehrt?



Analoge und digitale Daten

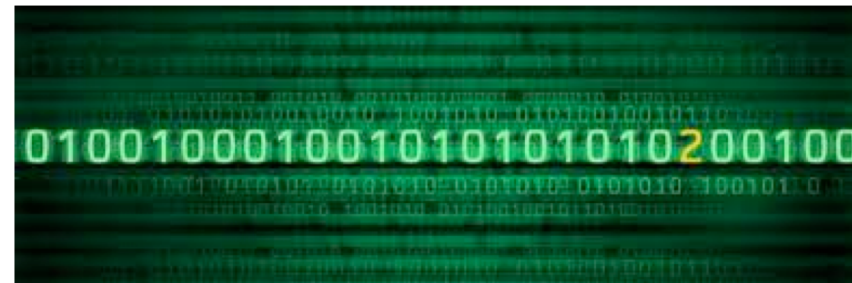
- **Digitale Daten** werden durch Zeichen repräsentiert. Ein *Zeichen* (Symbol) ist ein Element aus einer zur Darstellung von Information vereinbarten endlichen Menge von verschiedenen Elementen, dem sogenannten **Zeichenvorrat** (Alphabet).
- **Analoge Daten** werden durch kontinuierliche Funktionen repräsentiert. Die analoge Darstellung erfolgt durch eine physikalisch messbare Größe, die sich entsprechend *stufenlos* ändert.

Analoge und digitale Werte



Beispiele:

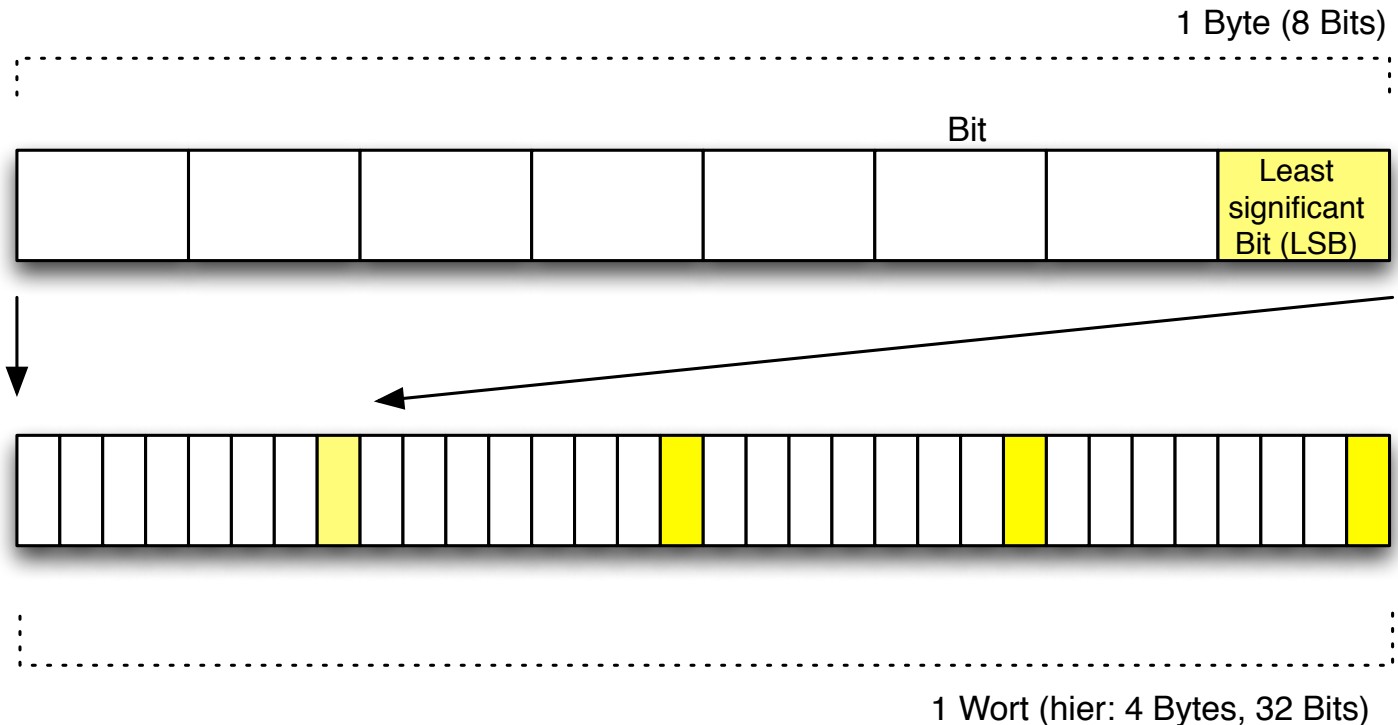
- Töne (Sprache, Musik)
- Quecksilberthermometer
- Optische Information



Bits und Bytes

Einheiten:

- Bit (0,1)
- Byte
- Wort



8 Bits == 1 Byte

4 oder 8 Bytes == 1 Wort (32 Bits oder 64 Bits)

Typische Größenangaben

- Speicher: KB (Kilo Byte), MB, GB, TB, PB
- Datenübertragung: Kbit (Kilo Bit), Mb, Gb,...

Daten

- Daten werden am Rechner letztendlich als **Bitfolgen** (Aneinanderreihung der Werte „0“ der „1“) gespeichert
- **Arten von Daten:**
 - Numerische Daten (Zahlen)
 - Ganze Zahlen (mit oder ohne Vorzeichen)
 - Gleitkommazahlen
 - Texte
 - Bilder, Grafiken, Töne, Filme,
- Daten (und auch Anweisungen) müssen **kodiert** werden
- Die gleiche Bitfolge kann als Zahl oder Text **interpretiert** werden

Stellenwertsysteme

- Unterscheidung **Ziffernwert** und **Stellenwert**

- Dezimalzahl „675“:

$$675 = 6 \times 10^2 + 7 \times 10^1 + 5 \times 10^0$$

- Ziffernwert: 0..9,
 - Stellenwert: Anzahl der Ziffernwerte hoch Position
 - Dezimalsystem: Stellenwertsystem zur Basis 10

- Allgemein:

$$W = \sum_{i=0}^{n-1} b_i \times B^i$$

- b_i Ziffernwert, B^i Stellenwert der i -ten Ziffer

- Beispiele für Stellenwertsysteme:

Dezimalsystem, Dualsystem, Oktalsystem, Hexadezimalsystem

Beispiele

$$W = \sum_{i=0}^{n-1} b_i \times B^i$$

- Zahlenwert zur Basis 10 (Dezimalzahl)

$$675 = 6 \times 10^2 + 7 \times 10^1 + 5 \times 10^0 = 600 + 70 + 5$$

- Zahlenwert zur Basis 2 (Dualzahl)

$$1100_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 = 8 + 4 + 0 + 0 = 12$$

- Maximaler Wert, der in einem Byte (8 Bits) dargestellt werden kann:

$$11111111_2 = 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255 = 2^8 - 1$$

- Dezimalzahl 65 als binärer Wert:

$$01000001_2 = 0 \times 128 + 1 \times 64 + 0 \times 32 + 0 \times 16 + 0 \times 8 + 0 \times 4 + 0 \times 2 + 1 \times 1 = 65$$

Repräsentationsgröße und Darstellungsbereiche

- Die **Repräsentationsgröße** bestimmt die vorgesehene Datenmenge in Bits oder Bytes, die für ein Datenelement vorgesehen ist. Sie bestimmt somit auch, wie viele unterschiedliche Werte in dem Datenelement dargestellt werden können.

Repräsentationsgröße	Minimaler Wert	Maximaler Wert
8 Bit (1 Byte)	0	$255 = 2^8 - 1$
16 Bit (2 Byte)	0	$65.535 = 2^{16} - 1$
32 Bit (4 Byte)	0	$4.294.967.295 = 2^{32} - 1$
64 Bit (8 Byte)	0	$18.446.744.073.709.551.615 = 2^{64} - 1$

Repräsentationsgröße und Darstellungsbereiche

- Die **Repräsentationsgröße** bestimmt die vorgesehene Datenmenge in Bits oder Bytes, die für ein Datenelement vorgesehen ist. Sie bestimmt somit auch, wie viele unterschiedliche Werte in dem Datenelement dargestellt werden können.

Repräsentationsgröße	Minimaler Wert	Maximaler Wert
8 Bit (1 Byte)	0	$255 = 2^8 - 1$
16 Bit (2 Byte)	0	$65.535 = 2^{16} - 1$
32 Bit (4 Byte)	0	$4.294.967.295 = 2^{32} - 1$
64 Bit (8 Byte)	0	$18.446.744.073.709.551.615 = 2^{64} - 1$

- Die Repräsentationsgröße beeinflusst u.A. die max. nutzbare Speichergröße, Dateigröße (32- vs. 64 Bit-Rechner) oder die bspw. die Zahl der möglichen Internetadressen (IPv4 Limitation), u.v.a.m.

Darstellung von positiven und negativen Zahlen

- Erstes Bit wird als Vorzeichen interpretiert, weitere Bits für Wertrepräsentation



- Wertbereiche für ganze Zahlen mit Vorzeichen:

Repräsentationsgröße	Minimaler Wert	Maximaler Wert
8 Bit (1 Byte)	-128	127
16 Bit (2 Byte)	-32.768	32.767
32 Bit (4 Byte)	-2.147.483.648	2.147.483.647
64 Bit (8 Byte)	-9.223.372.036.854.775.808	9.223.372.036.854.775.807

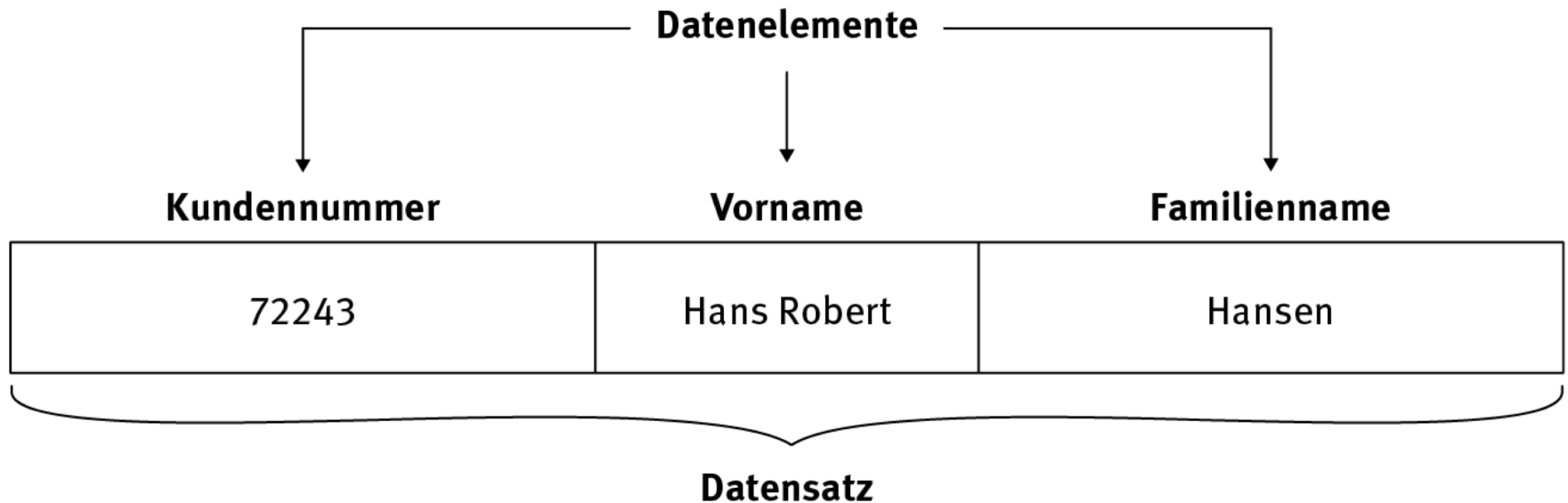
Codierung von textlicher Information

Zeichensatz:

- Legt fest, welche Bitfolge als welches Schriftzeichen codiert wird
- Schriftzeichen: Buchstaben, Ziffern, Leerzeichen, „Sonderzeichen“,...

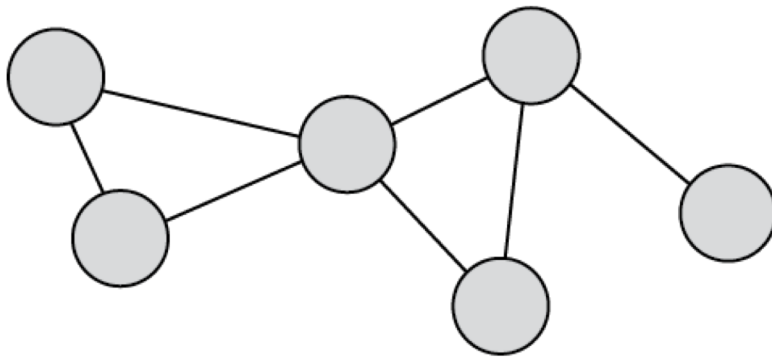
		Rechtes Halbbyte															
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Linkes Halbbyte	0000																
	0001																
	0010		!	“	#	\$	%	&	‘	(	)	*	+	,	–	.	/
	0011	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
	0100	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	0101	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
	0110	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
	0111	p	q	r	s	t	u	v	w	x	y	z	{		}	~	–

Datenstrukturen



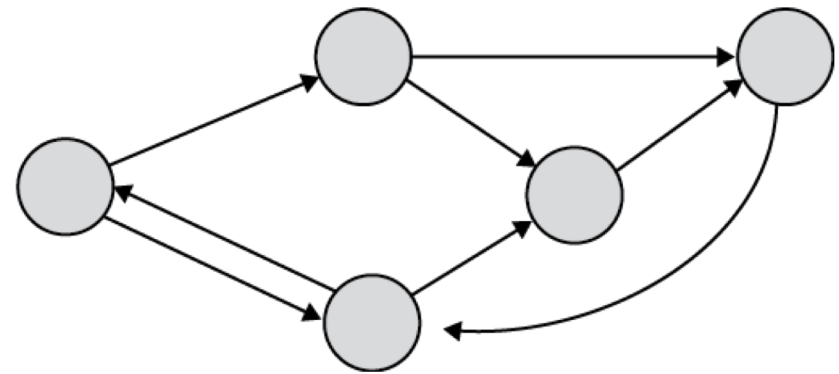
Ungerichtete und gerichtete Graphstrukturen

- Dienen zur Repräsentation von komplexeren Zusammenhängen
- Bestimmen Datenstrukturen und Algorithmen zur Verarbeitung



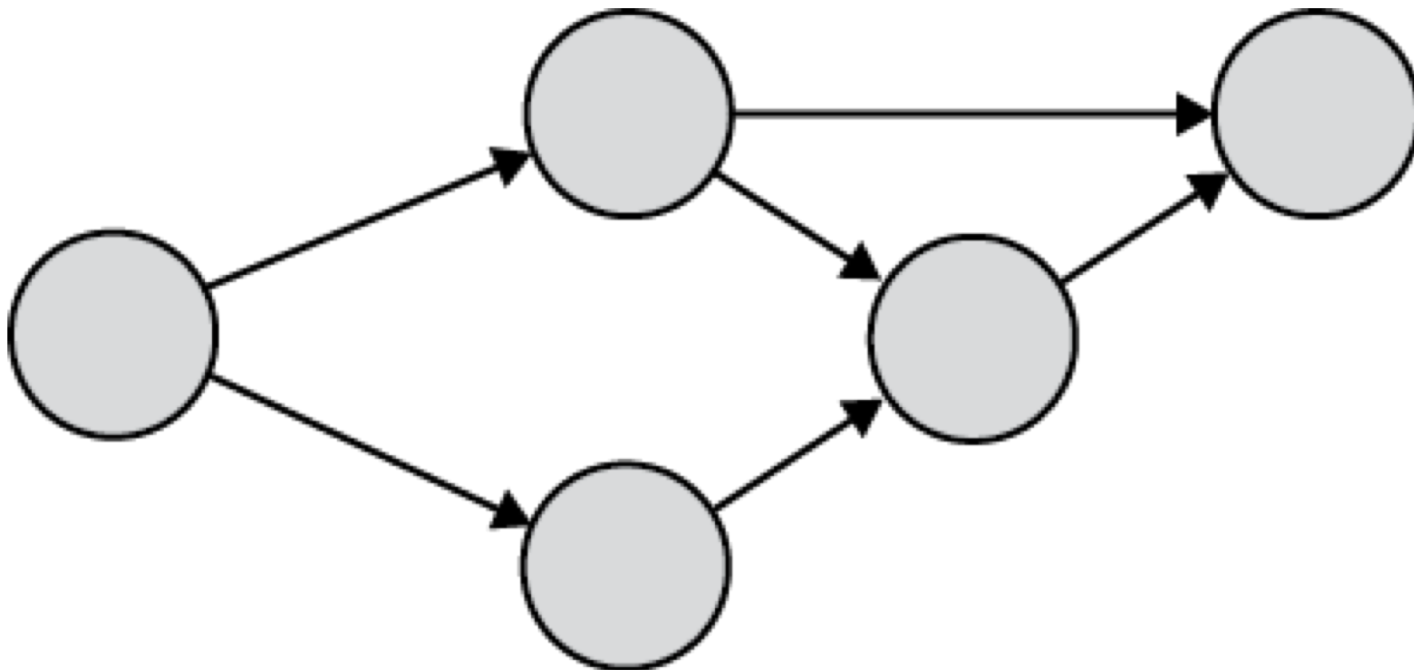
Beispiele:

- Arbeitsbeziehungen zwischen Mitarbeitern
- Straßennetz

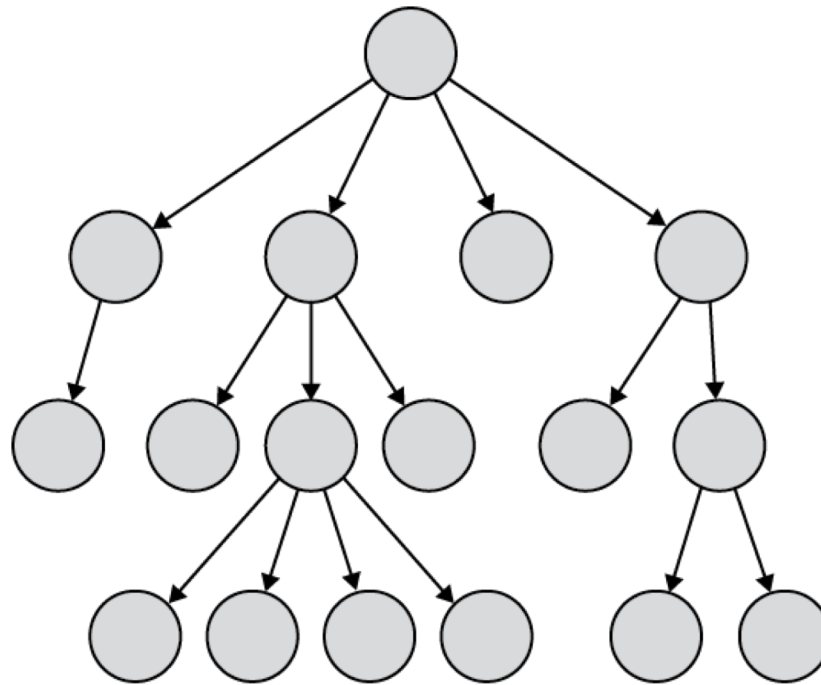


- Abhängigkeiten von Softwarekomponenten
- Ablauf in einem Geschäftsprozess

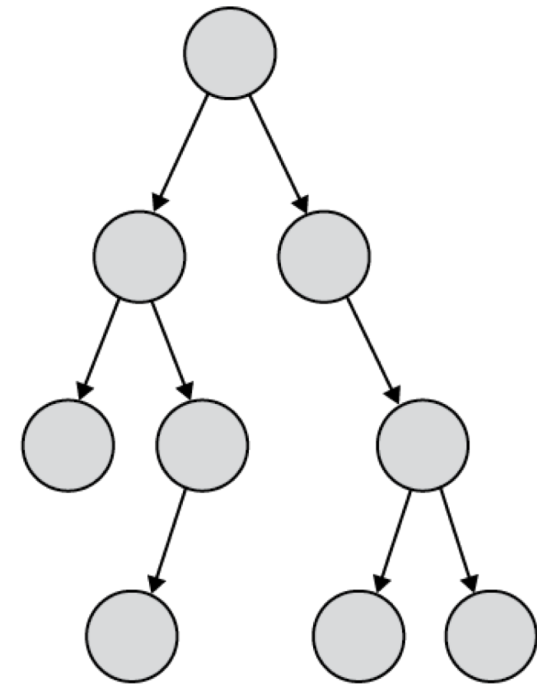
Gerichteter azyklischer Graph



Baumstrukturen



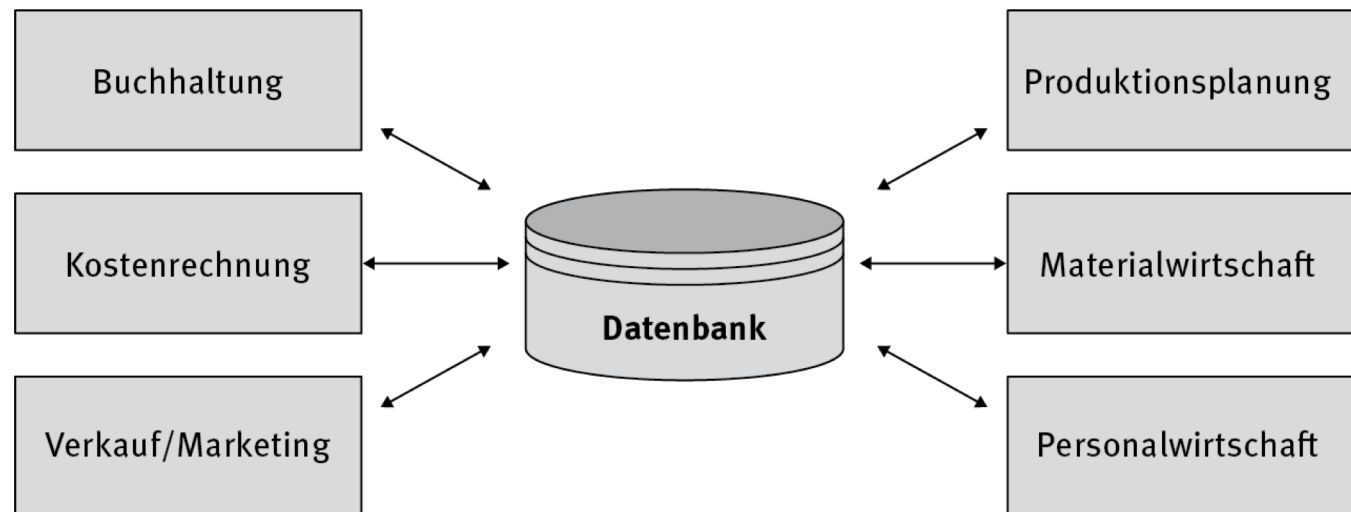
a) Baumstruktur der Ordnung 4



b) Binärbaum

Datenbank

- **Datenbank:** zentral verwalteter Datenbestand, der über anwendungsunabhängige Zugriffsverfahren nutzbar gemacht wird (Ziele: Entkopplung von Anwendungsprogrammen von Aspekten der Datenspeicherung, gemeinsam nutzbare Daten)
- **Datenbankverwaltungssystem** (DBMS) verwaltet Datenbank, regelt Zugriffsschutz, ermöglicht gleichzeitige Zugriffe von mehreren Anwendungsprogrammen, verwendet eigene Sprache für Definition und Abfrage von Daten.



ACID-Eigenschaften

- **Atomarität:** jede Operation in der Datenbank wird als untrennbares Ganzes ganz oder gar nicht durchgeführt
- **Konsistenz:** Alle Veränderungen in einer Datenbank sollen von einem gültigen Zustand in einen anderen gültigen Zustand des Datenbestands führen
- **Isolation:** Jede Operation soll von den Ergebnissen von anderen gleichzeitig aktiven Operationen unbeeinflusst ablaufen (keine Zwischenergebnisse von anderen gleichzeitigen Anfragen beeinflussen das Ergebnis)
- **Dauerhaftigkeit:** Sobald eine Datenbankoperation erfolgreich abgeschlossen wurde, muss das Ergebnis auch dauerhaft in der Datenbank abgespeichert sein (bestätigte Änderungen bleiben auch bei Rechnerabstürzen erhalten)

Relationales Datenmodell

- Daten werden durch eine Menge von **Relationen** definiert und in entsprechenden **Tabellen** gespeichert
- Eine Relation wird durch **Namen, Attribute** und **Einschränkungen** und **Abhängigkeiten** definiert

PERSON(Personennummer, Name, PLZ, Ort, Land)

- **Tabellen**
 - speichern Daten
 - Eine Ausprägung entspricht einer Zeile (ein Tupel) in der Tabelle
 - Merkmale (Attribute) der Ausprägung entsprechen den Spalten

Beispieltabelle: Person

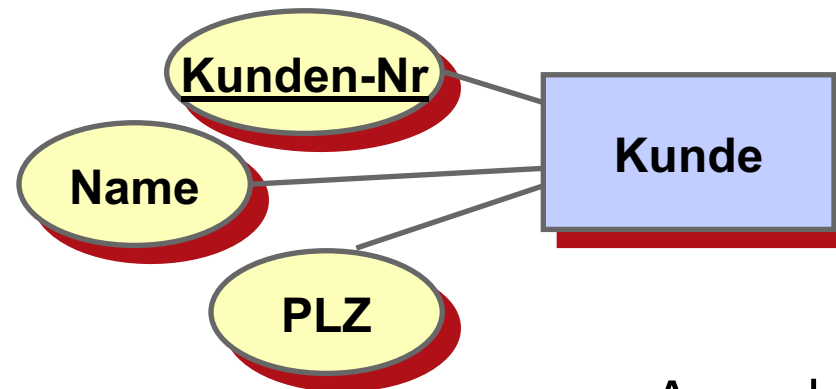
The diagram illustrates the components of a database table. Labels with arrows point to specific parts of the table structure:

- Relationenname**: Points to the table name 'Person' in the first column header.
- Primärschlüssel**: Points to the primary key attribute 'Personennummer' in the second column header.
- Attribute**: A bracketed label pointing to the headers of the five data columns: 'Name', 'PLZ', 'Ort', and 'Land'.
- Tupel**: Three arrows point from this label to the three data rows of the table, indicating each row is a tuple.

Person	<u>Personennummer</u>	Name	PLZ	Ort	Land
	P1	Kafka	110 00	Prag	Tschechien
	P2	Hansen	1010	Wien	Österreich
	P3	Torberg	1010	Wien	Österreich

Zusammenhang zwischen konzeptionellen Modell und Datenbankmodell

- Relation: KUNDE(KundenNr, Name, PLZ)



Ausschnitt aus dem
Fachkonzept

Relationsschema in Tabellenform

- Relation: KUNDE(KundenNr, Name, PLZ)

KUNDE	<u>Kunden-Nr.</u>	Name	PLZ
	1	Hansen	1010
	2	Torberg	8670
	3	Neumann	1010
	4	Schnitzler	8020
	.	.	.
	.	.	.

Werteinschränkungen und Abhängigkeiten

- **Werteinschränkungen** definieren erlaubte Werte von Attributen
 - Datentypen
 - Wertbereiche
- **Abhängigkeiten** zwischen Attributen definierten mögliche Werte im Zusammenhang mit anderen Ausprägungen
 - Funktionale Abhängigkeit
 - Inklusionsabhängigkeit

Funktionale Abhängigkeit

- Ausprägung einer Attributmenge bestimmt Ausprägung einer zweiten Attributmenge
- **Abhängigkeit** zwischen Attributen/Attributmengen **innerhalb einer Relation**
- Beispiel:

Kunden-Nr bestimmt Adresse

Kunden-Nr $\rightarrow$ Adresse

Interpretation: Für eine Kundennummer darf nur eine Adresse in einer Tabelle eingetragen werden.

- Diese Abhängigkeit wird bspw. verletzt, wenn mehrfach die gleiche Kundennummer vergeben wurde.

Inklusionsabhängigkeit

- Alle Ausprägungen einer Attributmenge müssen in den Ausprägungen einer zweiten Attributmenge enthalten sein
- Abhängigkeiten zwischen Attributen/Attributmengen unterschiedlicher Relationen
- Beispiel:

AUTOR(AutorenNr, Name)

VERFASST(AutorenNr, Buch-Nr)

- **Interpretation:**
 - AutorenNr_{VERFASST} inkludiert in AutorenNr_{AUTOR}
 - AutorenNr_{VERFASST} ist **Fremdschlüssel** (engl.: foreign key).
 - Die Abhängigkeit wird verletzt, wenn ein Tupel in Relation/Tabelle AUTOR gelöscht wird ohne das korrespondierende Tupel in Relation/Tabelle VERFASST ebenfalls zu entfernen
 - Möglich: Definition von automatisch kaskadierenden Löschoperationen

Definition von Relationenschemata

- **Normalisierung:**

Regeln zum Aufbau von Tabellen (Relationen) in relationalen Datenbanken

- **Bei Nichtbeachtung:**

Diverse Probleme (*Anomalien*) beim

- Abfragen,
- Einfügen,
- Ändern und
- Löschen

von Daten

Relationale Operationen

- **Selektion:** Auswahl einer Untermenge aller Tupel einer Tabelle
- **Projektion:** Auswahl einer Untermenge der Attribute einer Relation
- **Verbund:** Verknüpfung von Tabellen anhand selektierter Attribute

Sowohl Eingabewerte wie auch Ergebnisse von relationalen Operationen sind Tabellen.

Selektion

Ausgangstabelle

<u>Personennummer</u>	Name	PLZ	Ort	Land	
P1	Kafka	110 00	Prag	Tschechien	
P2	Hansen	1010	Wien	Österreich	
P3	<u>Personennummer</u>	Name	PLZ	Ort	Land
P4	P2	Hansen	1010	Wien	Österreich
P5		Adams	CB30EQ	Cambridge	England
P6		Clemens	06876	Redding	USA
P7		Torberg	1010	Wien	Österreich
P8		Musil	9020	Klagenfurt	Österreich
P9		Neumann	1010	Wien	Österreich
P10	P7	Torberg	1010	Wien	Österreich
P11		Schnitzler	1010	Wien	Österreich
	P9	Neumann	1010	Wien	Österreich
	P11	Schnitzler	1010	Wien	Österreich

Ergebnistabelle

Projektion

Ausgangstabelle

<u>Personennummer</u>	Name	PLZ	Ort	Land
P1	Kafka	110 00	Prag	Tschechien
P2	Hansen	1010	Wien	Österreich
P3	Schiller	99423	Weimar	Deutschland
P4	Rosegger	8670	Krieglach	Österreich

	Name	PLZ	Ort
P5	Adams		
P6	Clemens		
P7	Torberg		
P8	Musil		
P9	Neumann		
P10	Canetti		
P11	Schnitzler		
	Kafka	110 00	Prag
	Hansen	1010	Wien
	Schiller	99423	Weimar
	Rosegger	8670	Krieglach
	Adams	CB3 0EQ	Cambridge
	Clemens	06876	Redding
	Torberg	1010	Wien
	Musil	9020	Klagenfurt
	Neumann	1010	Wien
	Canetti	8001	Zürich
	Schnitzler	1010	Wien

Erzeugung einer Tabelle, die aus einer Unter-
menge der Attribute einer
Ausgangstabelle besteht.

Hier: Attribute **Name, PLZ, Ort**

Ergebnistabelle

Verbund

Produkt

<u>Produkt-Nr.</u>	Bezeichnung	Preisgruppe
Prod1	Notizblock A4 kariert	G3
Prod2	Notizblock A5 liniert	G2
Prod3	Notizblock A4 liniert	G3
Prod4	Notizblock A6 glatt	G1
Prod5	Kopierpapier 500 Blatt	G7
Prod6	Notizblock A4 glatt	G3



Preis

<u>Preisgruppe</u>	Betrag
G1	0,50
G2	1,50
G3	3,00
G4	3,25
G5	5,00
G6	5,50

Verknüpfung von Tabellen anhand selektierter Attribute.
Hier: Verknüpfung anhand von „Preisgruppe“

Ergebnis: Tabelle mit Attributen:

Produkt-Nr, Bezeichnung, Preisgruppe, Betrag

SQL: Definition einer Tabelle

```
CREATE TABLE Produkt (  
    ProduktNr INTEGER NOT NULL UNIQUE,  
    Bezeichnung CHAR (250) NOT NULL,  
    Preisgruppe CHAR (10) NOT NULL,  
    PRIMARY KEY(ProduktNr),  
    FOREIGN KEY(Preisgruppe) REFERENCES Preis(Preisgruppe)  
)
```

SQL: Abfrage von Daten

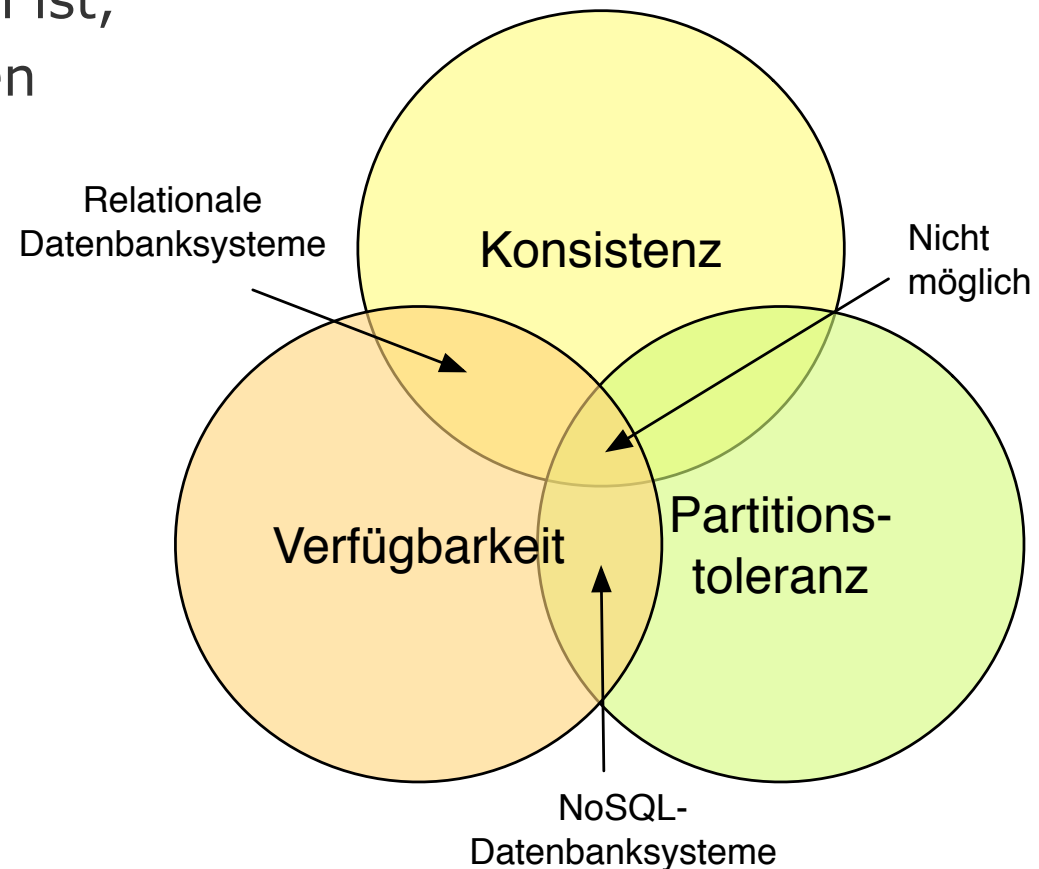
```
SELECT ProduktNr, Bezeichnung, Preis  
FROM Produkt, Preis  
WHERE Produkt.Preisgruppe = Preis.Preisgruppe AND Betrag <= 3.0
```

Skalierbare Datenspeicherung und Big Data

- **Big Data:** Datenkollektionen, deren Größe die Fähigkeiten einzelner Rechnersysteme überschreiten, um diese zu speichern, zu durchsuchen, zu analysieren und zu verwalten (offene Definition)
- **Verteiltes Datenbanksystem:** ein Datenbanksystem, das für die Speicherung, Abfrage und Verwaltung eines Datenbestands mehrere getrennte Rechner verwendet, die über ein Kommunikationsnetzwerk (beispielsweise das Internet) miteinander verbunden sind.
- **Problem:** Was passiert, wenn Daten über mehrere Rechner konsistent gehalten werden müssen (ACID-Prinzip), aber einzelne der Rechner nicht erreichbar (Neustart, Netzwerkproblem, usw.)

CAP-Theorem

Das CAP-Theorem (Abkürzung von engl.: consistency, availability, partition tolerance) besagt, dass es in einem verteilten System unmöglich ist, gleichzeitig die Eigenschaften der **Konsistenz**, der **Verfügbarkeit** und der **Partitionstoleranz** zu garantieren.



NoSQL-Datenbanken

- Eigenschaften:
 - ein meist **nicht relationales** Datenbanksystem
 - für **hochskalierbare** Anwendungen
 - mit meist **schwacher Konsistenzgarantie** („eventually consistent“)
- Nicht relational:
 - Oft einfachere Datenmodelle; Ansätze:
 - Attribut-Wertpaare
 - Spaltenorientierte Speicherung
 - Strukturierte Dokumente
 - hierarchische (Baum-) Strukturen
 - Datenformate: XML, JSON, ...

XML

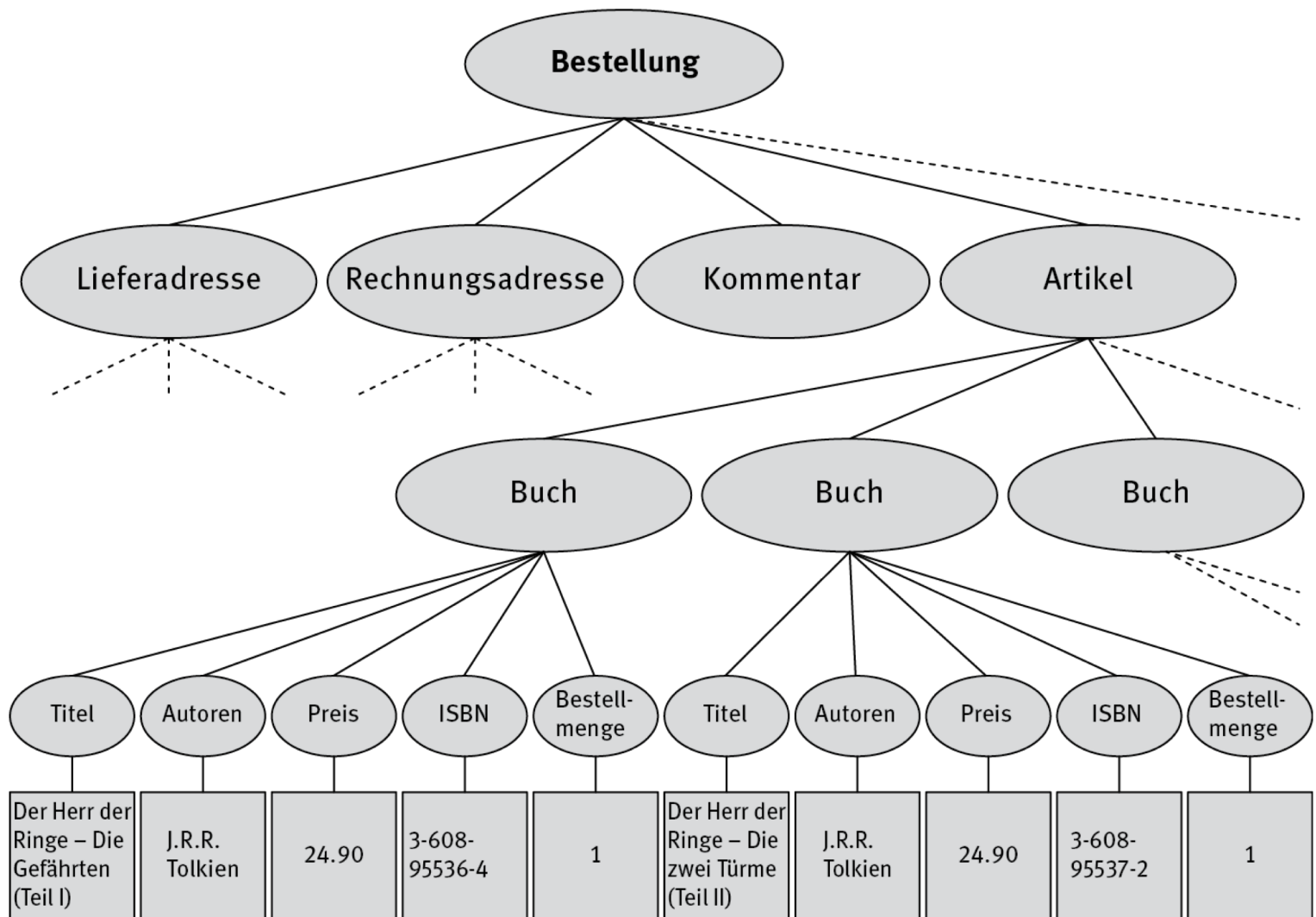
- Hierarchisches Modell
- Dokumentzentriert
- Ursprung:
Datenaustausch
- Basis vieler weiterer
Standards
- Durch Schemadefinition:
Dokumenttypen
- Abfragesprachen:
 - Xpath
 - XQuery

```

<?xml version="1.0" ?>
<Bestellung Datum="30.06.2015">
  <Lieferadresse>
    <Name>Peter Schmidt</Name>
    <Straße>Schleicher-Allee 11</Straße>
    <Stadt>Wien</Stadt>
    <Postleitzahl>1090</Postleitzahl>
    <Land>Österreich</Land>
  </Lieferadresse>
  <Rechnungsadresse>
    <Name>Andrea Müller</Name>
    <Straße>Getreidestraße 77</Straße>
    <Stadt>Wien</Stadt>
    <Postleitzahl>1200</Postleitzahl>
    <Land>Österreich</Land>
  </Rechnungsadresse>
  <Artikel>
    <Buch Kategorie="Belletristik" Verlag="Klett-Cotta">
      <Titel>Der Herr der Ringe - Die Gefährten (Teil 1)</Titel>
      <Autoren>J.R.R. Tolkien</Autoren>
      <Preis Währung="Euro">24.90</Preis>
      <ISBN-Nummer>3-608-95536-4</ISBN-Nummer>
      <Bestellmenge>1</Bestellmenge>
    </Buch>
    <Buch Kategorie="Belletristik" Verlag="Klett-Cotta">
      <Titel>Der Herr der Ringe - Die zwei Türme (Teil 2)</Titel>
      <Autoren>J.R.R. Tolkien</Autoren>
      <Preis Währung="Euro">24.90</Preis>
      <ISBN-Nummer>3-608-95537-2</ISBN-Nummer>
      <Bestellmenge>1</Bestellmenge>
    </Buch>
  </Artikel>
</Bestellung>

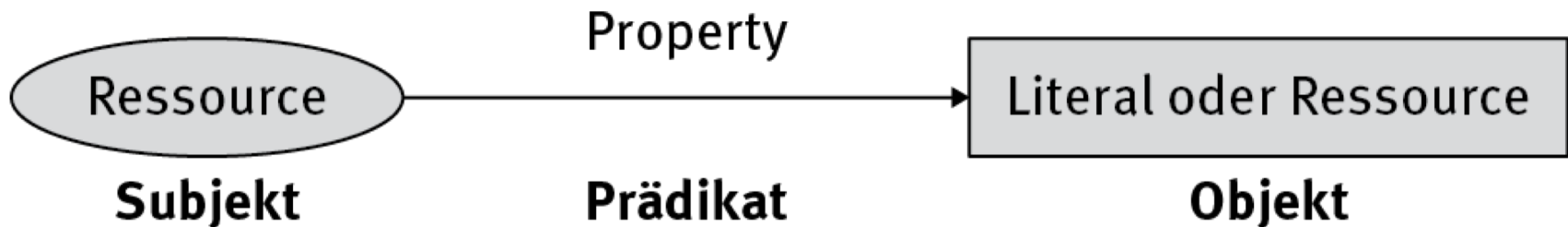
```

XML: Hierarchische Struktur



RDF: Resource Description Framework

- Ursprung: Definition von Meta-Daten (Information über Information)
- Ermöglicht Aussagen über im Web bezeichnbare Dinge (Ressourcen) mittels Vokabularen
- Sprache des „Semantic Web“, Grundlage für „Linked Open Data“
- Abfragesprache: SPARQL



Beispiel: JSON-Dokument

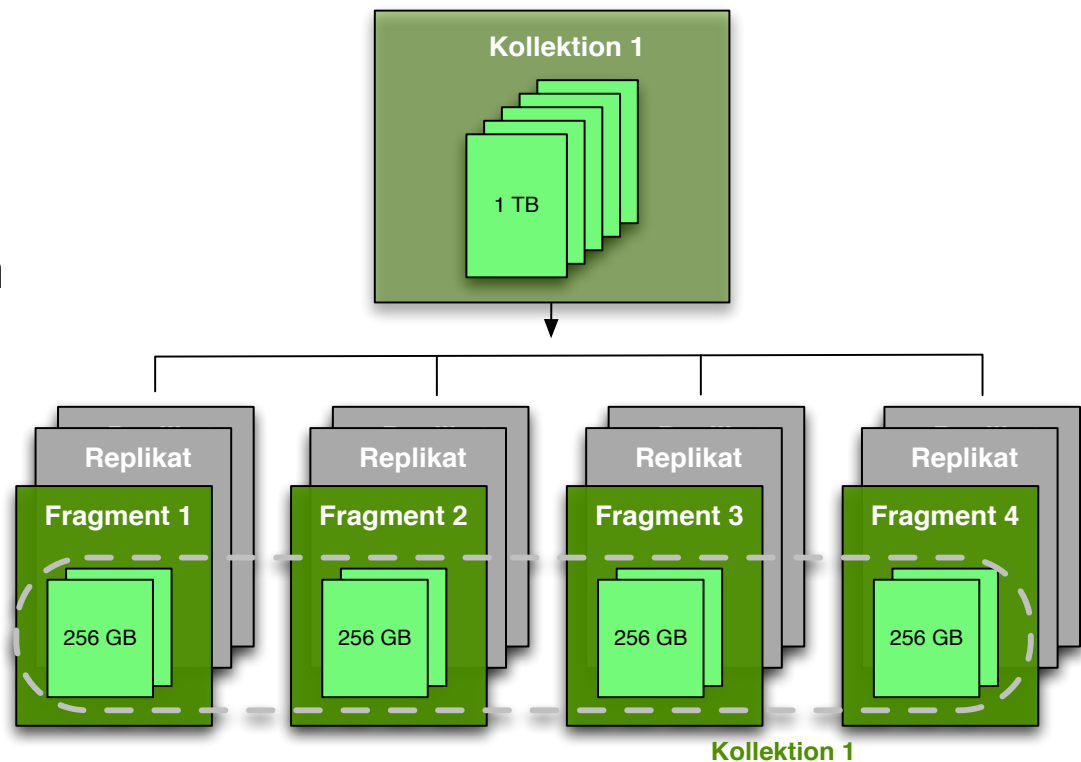
- JSON:
 - JavaScript Object Notation
 - offener Standard für die Beschreibung baumstrukturierter Daten
 - Ursprung: Web-Programmierung

```
{  
  "Titel": "Wirtschaftsinformatik",  
  "Autor": [  
    {"Vorname": "Hans Robert", "Familienname": "Hansen"},  
    {"Vorname": "Jan", "Familienname": "Mendling"},  
    {"Vorname": "Gustaf", "Familienname": "Neumann"} ],  
  "Verlag": "De Gruyter",  
  "Schlagworte": ["Wirtschaftsinformatik", "Einführung"]  
}
```

Vertreter einer NoSQL Datenbank: MongoDB

- Dokumentzentriert
- JSON-ähnliche Dokumente (BSON: binary JSON)
- Gute Unterstützung für verteilte, skalierbare Daten

- **Datenbank-replikation**
(mehrere Kopien des selben Datenbestands werden automatisiert auf unterschiedlichen Rechnern synchronisiert)
- **Datenbank-fragmentierung**
(Teile der Daten auf unterschiedlichen Rechnern)



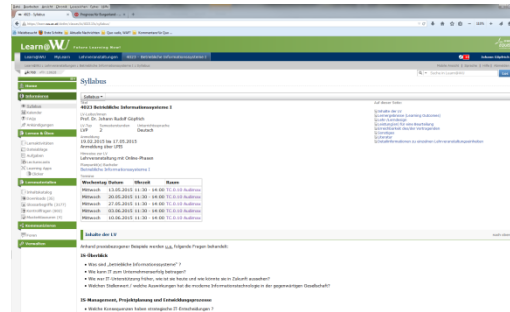
Diskussionsfragen Kapitel 9

- Warum gibt es so viele verschiedene Standards für die Codierung von Texten? Warum müssen Sie sich als Wirtschaftsinformatiker mit diesen Standards beschäftigen?
- Sollte nicht jedes Programm, das Daten verarbeitet, diese in einem Datenbanksystem ablegen? Was sind die Vor- und Nachteile von Datenbanksystemen?
- Warum benötigt man für die Verarbeitung von Big Data andere Ansätze als für traditionelle Datenbank Anwendungen? In welchen Fällen besitzen relationale Datenbanksysteme weiterhin Vorteile?



Fallbezogene Aufgabe zu Kapitel 9

Learn@WU:



- Welche Zeichensätze und Zeichencodierungen eignen sich für das E-Learning-System?
- Wie sollte die von dem Informationssystem verarbeitete Information gespeichert werden (Dateien, Datenbankverwaltungssystem, Mischformen)? Geben Sie Beispiele an!
- Welches konkrete Datenmodell sollte gewählt werden? Begründen sie Ihre Empfehlung.
- Geben sie drei Beispiele für mögliche Verletzungen der referenziellen Integrität an, die in dem System unbedingt vermieden werden sollten. Was wären die konkreten Folgen?
- Benötigt das System Datenaustauschformate? Geben Sie Beispiele an, in denen sich XML-Dokumente für den Datenaustausch eignen würden.
- Der Nettodatenbestand umfasst über 25 Millionen Datenobjekte und belegt 2 Terabyte. Handelt es sich hierbei um „Big Data“? Begründen Sie Ihre Antwort.

