Einführung in die Wirtschaftsinformatik I

WS 2024/25

Modellierung und Organisation von Daten II

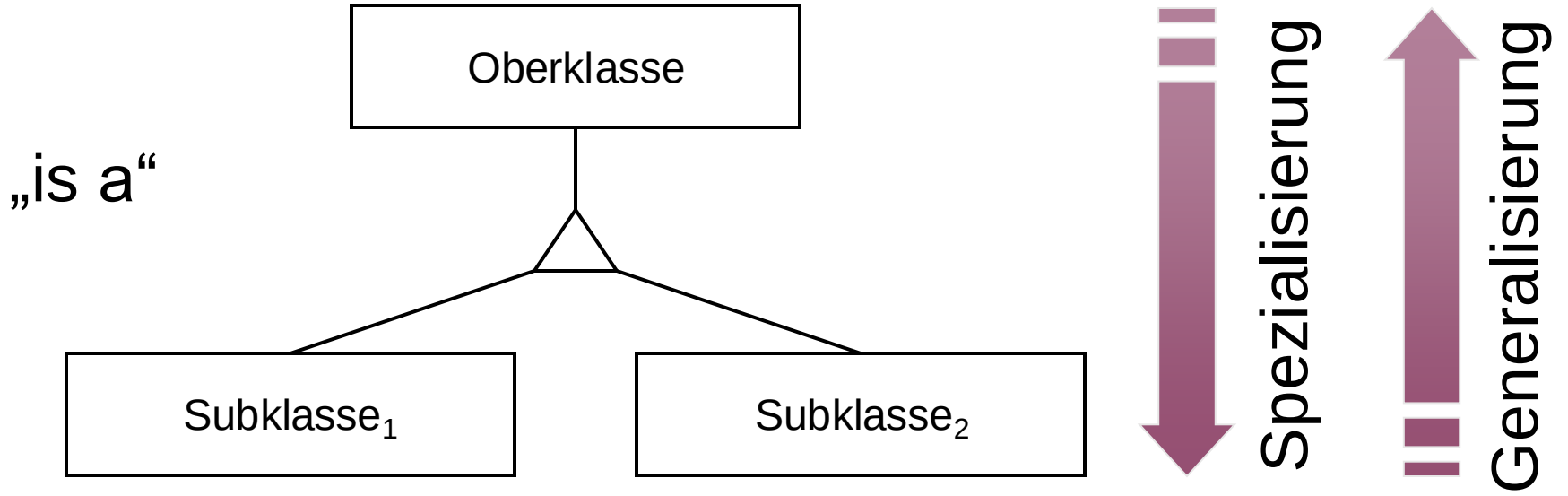
Dr. Rainer Wieching

Generalisierung / Spezialisierung

Haben verschiedene Entitätstypen gleiche Attribute, so können diese in einer Oberklasse zusammengefasst werden (**Generalisierung**).

Die ursprünglichen Entitätstypen werden **Subklassen** genannt, die durch Spezialisierung aus der Oberklasse abgeleitet werden

Von der **Oberklasse** erben die Subklassen die Attribute und die Beziehungen.



Generalisierung / Spezialisierung

Arten

- **Total**

  Alle Entitäten des Super-Typs befinden sich auch in einem der Sub-Typen. (Vollständigkeit)

- **Partiell**

  Eine Entität des Super-Typs muss keinem Sub-Typen angehören.

- **Disjunkt**

  Es gibt keine Überschneidungen unter den Sub-Typen

- **Nicht Disjunkt**

  Es gibt Überschneidungen unter den Sub-Typen.

Generalisierung / Spezialisierung

Total, disjunkt:
Oberklasse: Person
Subklassen: Mann, Frau

Total, nicht disjunkt:
Oberklasse: Hochschulangehöriger
Subklassen: Studierender, Angestellter

Partiell, disjunkt:
Oberklasse: Fahrzeug
Subklassen: Auto, Fahrrad

Partiell nichtdisjunkt:
Oberklasse: Person
Subklassen: Mann, Angestellter

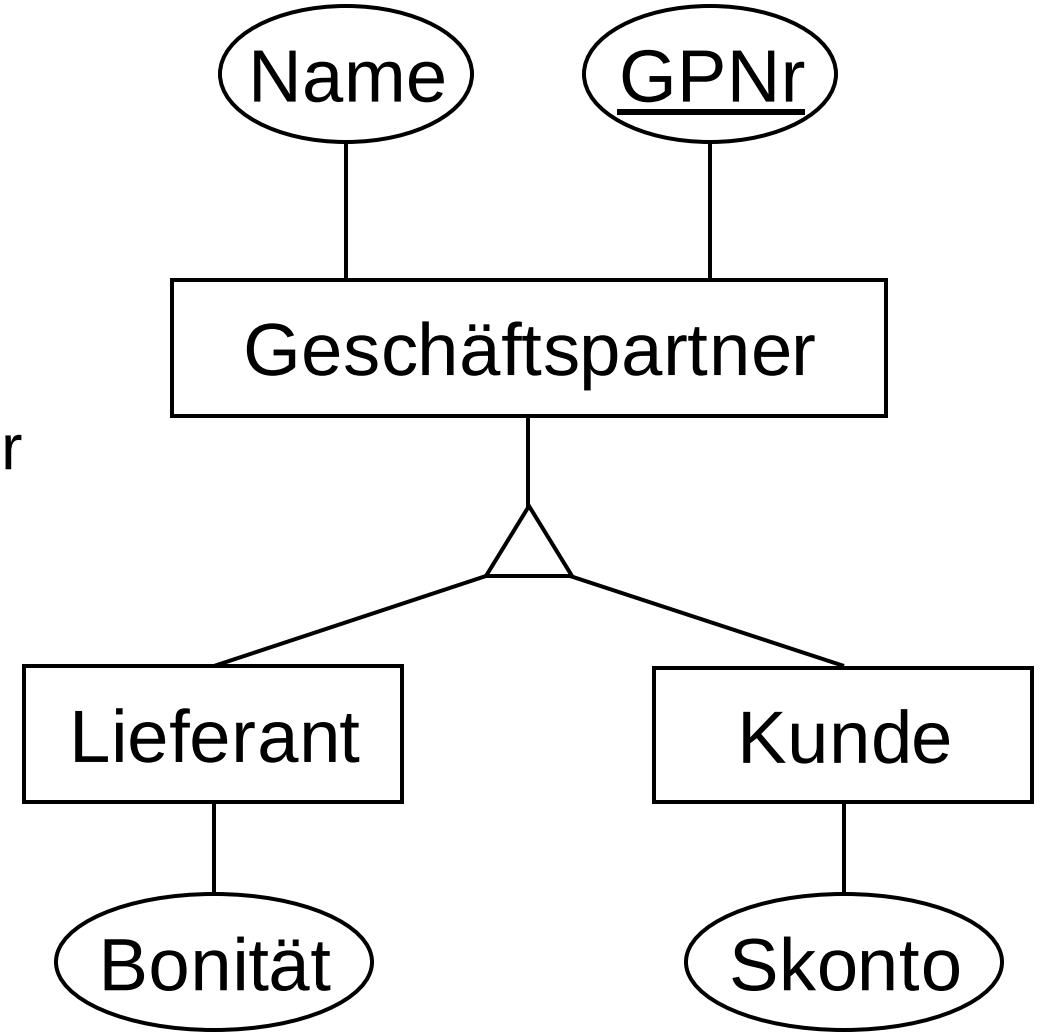
Generalisierung / Spezialisierung

Total, disjunkt:
Oberklasse: Person
Subklassen: Mann, Frau

Total, nicht disjunkt:
Oberklasse: Hochschulangehöriger
Subklassen: Studierender, Angestellter

Partiell, disjunkt:
Oberklasse: Fahrzeug
Subklassen: Auto, Fahrrad

Partiell nichtdisjunkt:
Oberklasse: Person
Subklassen: Mann, Angestellter



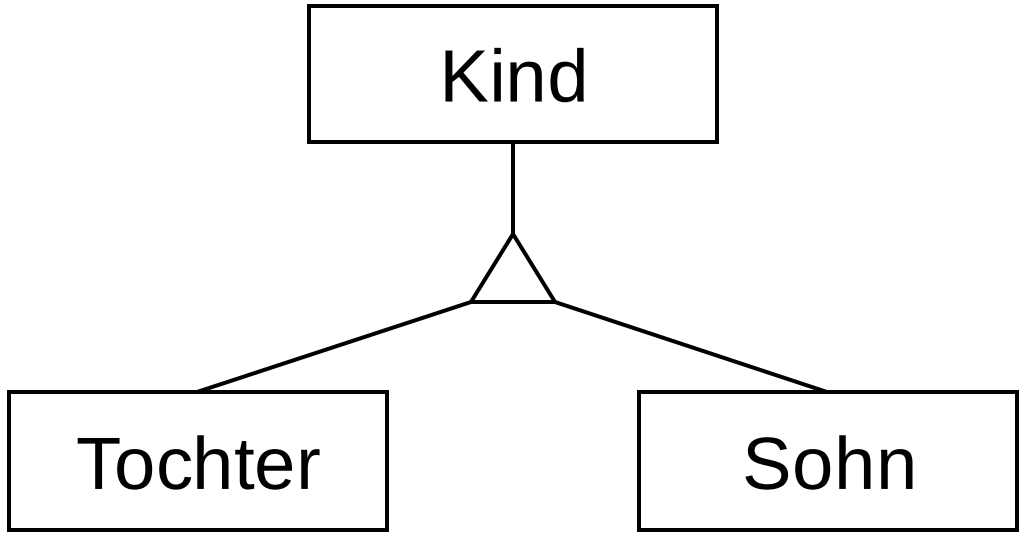
Generalisierung / Spezialisierung

Total, disjunkt:
Oberklasse: Person
Subklassen: Mann, Frau

Total, nicht disjunkt:
Oberklasse: Hochschulangehöriger
Subklassen: Studierender, Angestellter

Partiell, disjunkt:
Oberklasse: Fahrzeug
Subklassen: Auto, Fahrrad

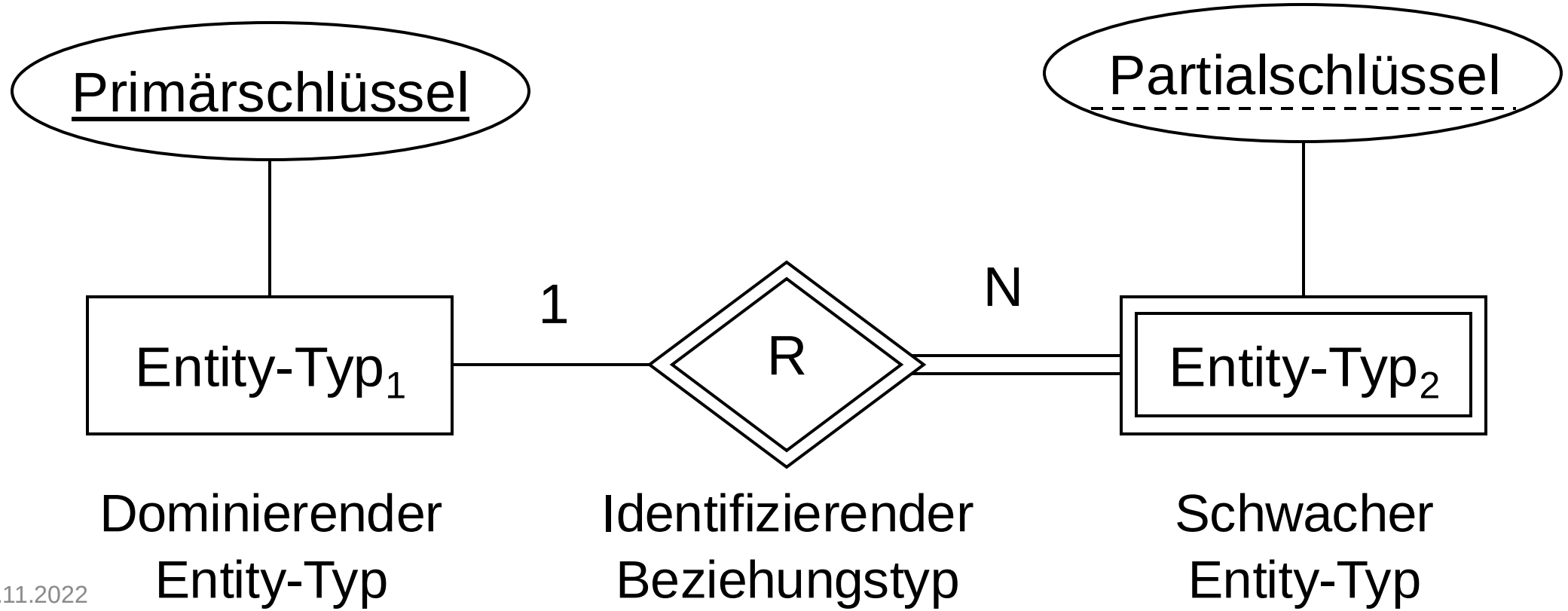
Partiell nichtdisjunkt:
Oberklasse: Person
Subklassen: Mann, Angestellter



Schwacher (existenzabhängiger) Entitätstyp

Die Existenz von Entities eines schwachen Entitytyps („weak entity“) hängt von der Existenz eines Entities des dominierenden Typen ab.

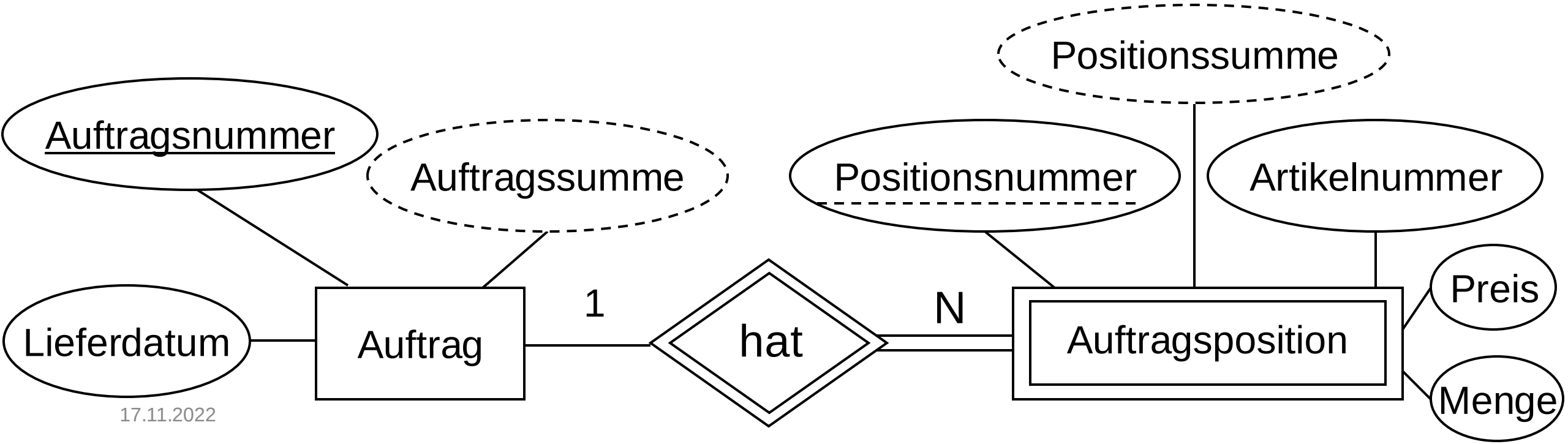
Abhängigkeit wird durch identifizierende Beziehung modelliert.



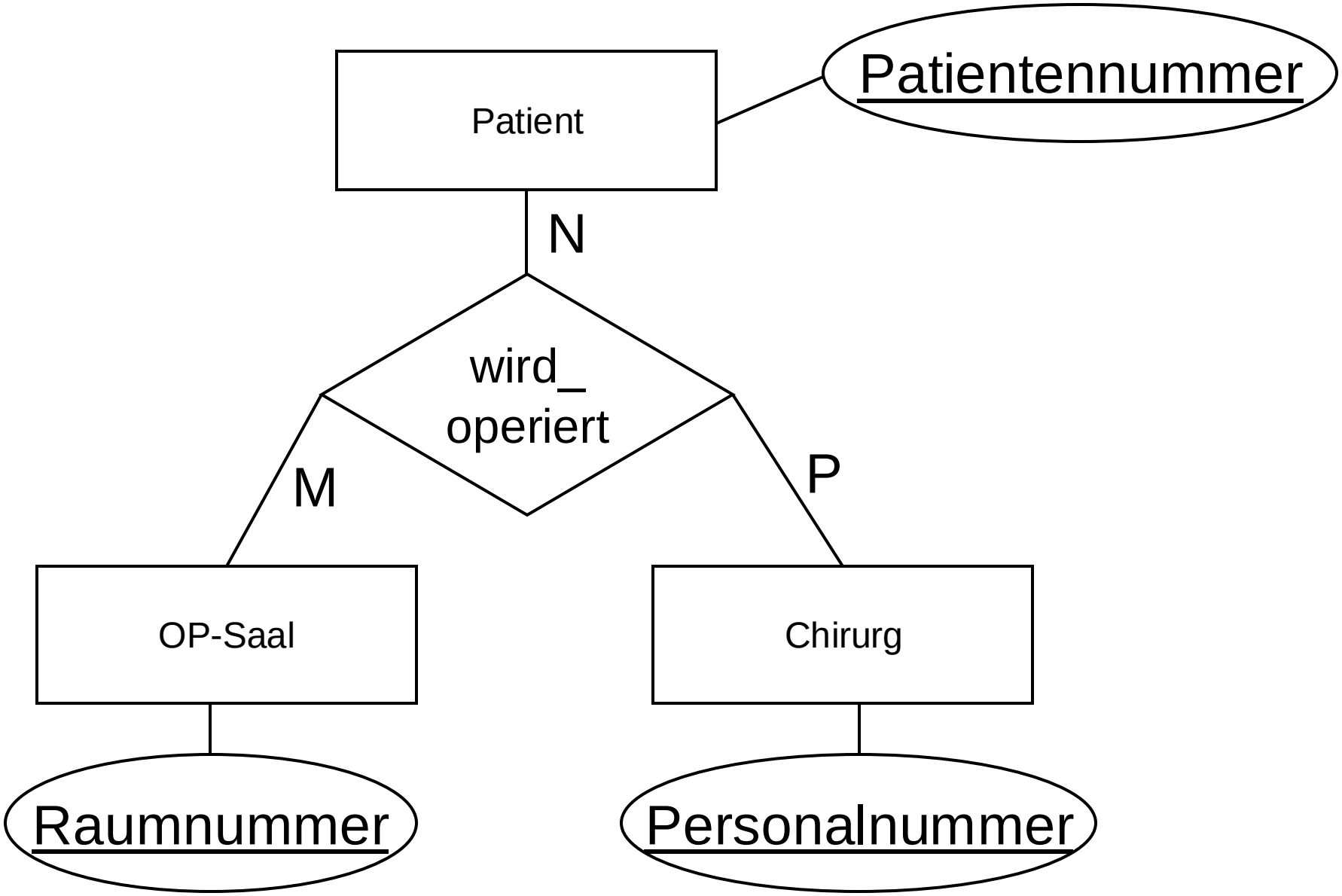
Beispiel: Schwacher (existenzabhängiger) Entitätstyp

Auftragsnummer: xyz **Lieferdatum:** TT.MM.JJJJ

Position	*Artikel*	*Bezeichnung*	*Preis*	*Menge*	*Positionssumme*
1	4711	Tapete	6,50 €	6	39,00 €
2	874	Kleister	4,00 €	2	8,00 €
3	...	...	...	...	...
			Auftragssumme:	**...**	



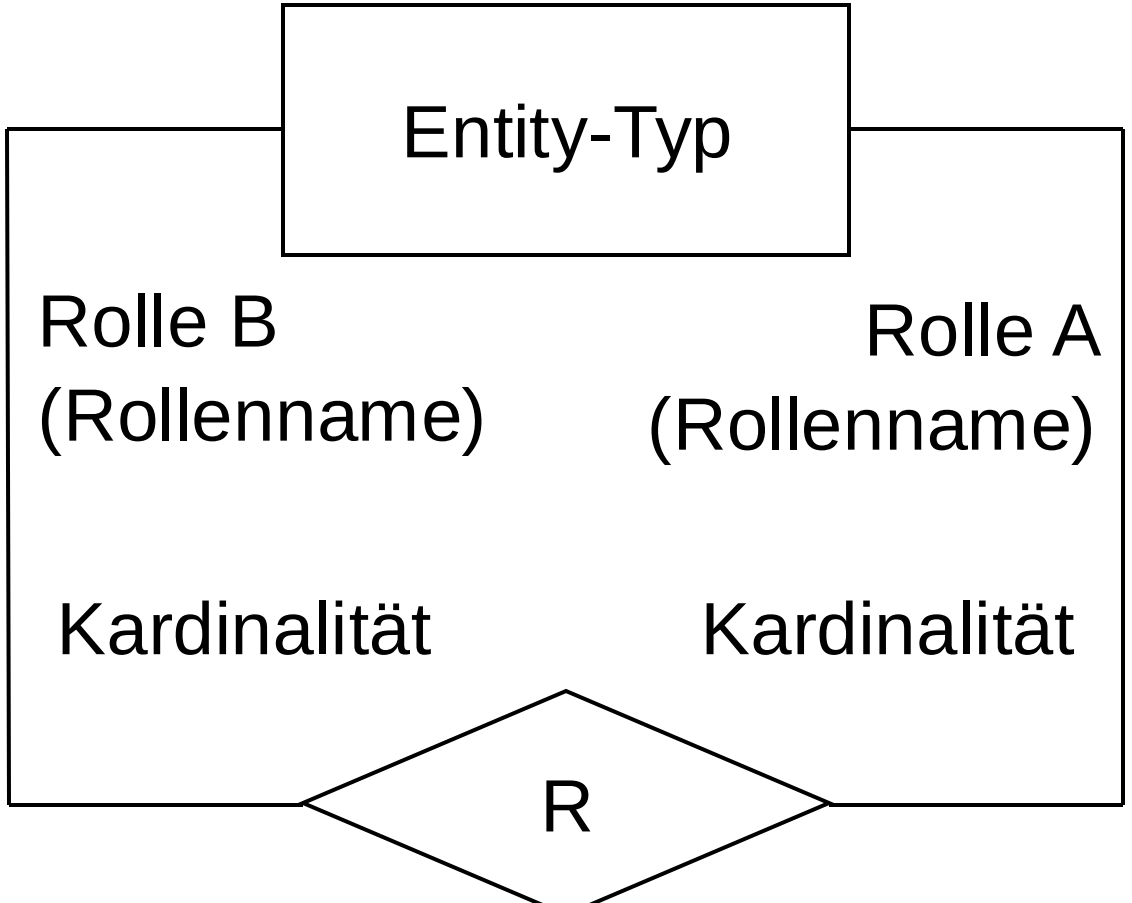
Beispiel für Beziehungstyp vom Grad 3



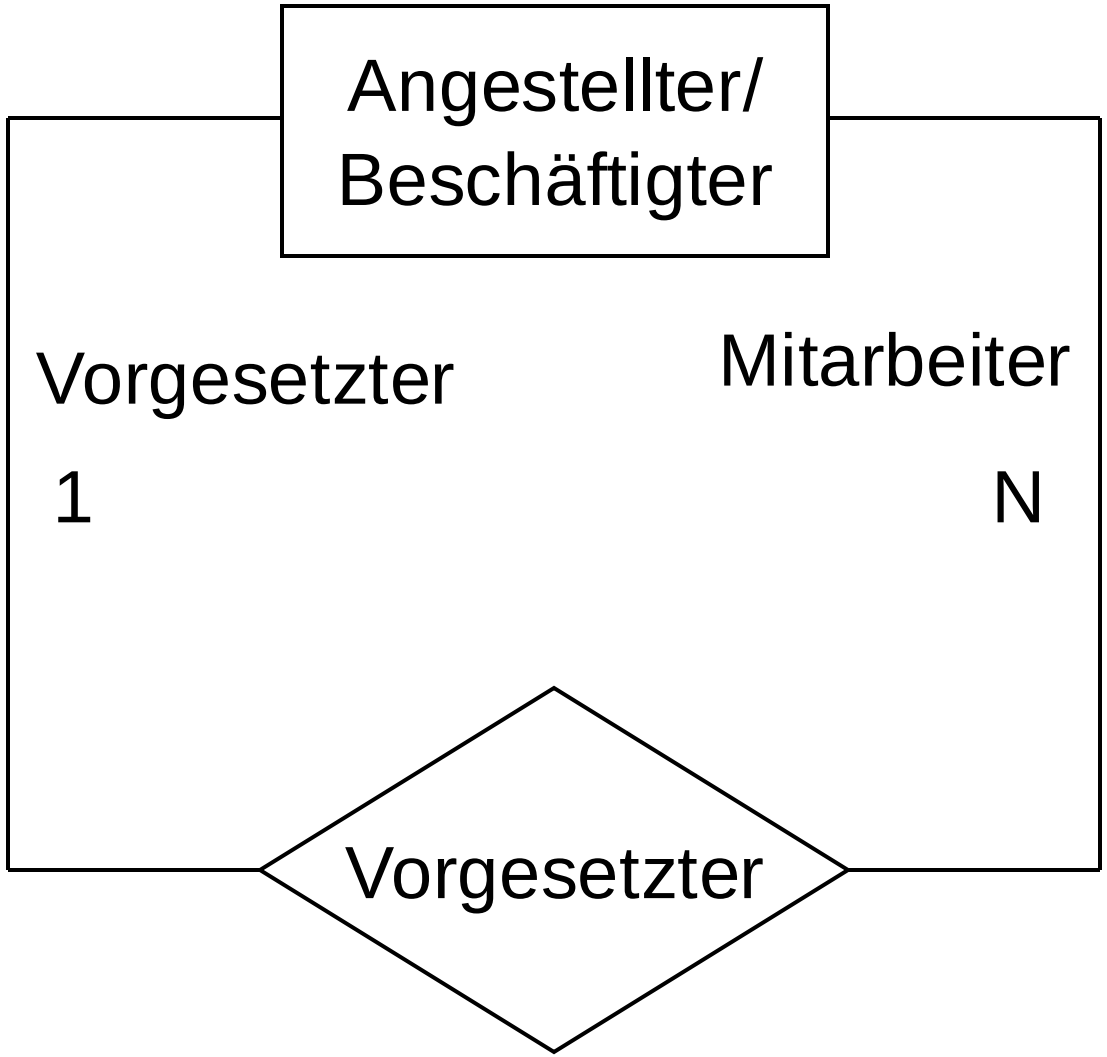
Rekursive Beziehung / Assoziation

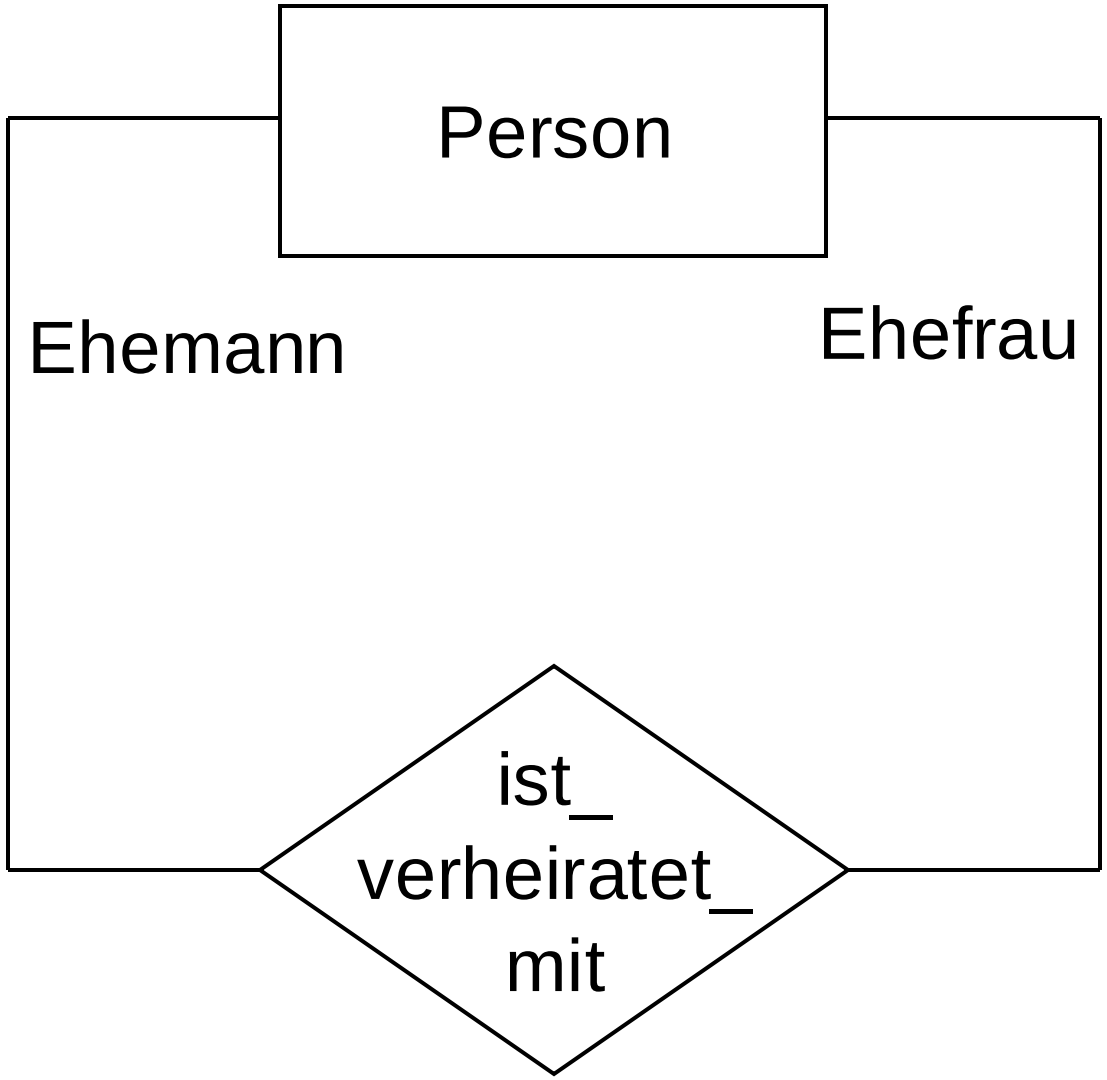
Derselbe Entity-Typ nimmt mehr als einmal an einem Beziehungstypen teil

Verdeutlichung der Semantik der Beziehung durch unterschiedliche Rollen



Beispiele für Rekursive Beziehung / Assoziation





Gliederung

1. Grundgedanken der ER-Modellierung

2. Datenorganisation

3. Relationales Datenbankschema

Definition Datenorganisation

Datenorganisation ist eine Zusammenfassung von Verfahren zur:

- **Strukturierung der Daten** (logische Datenorganisation)
- **Datenablage auf Speichermedien** und **Bereithaltung für den Zugriff** (physische Datenorganisation)



http://dzok-ulm.de/images/opener/archiv.jpg

Grundlegende Begriffe

Daten

Informationen, die weiterverarbeitet werden , z.B. Akustische bzw. elektromagnetische Signale, Zeichenketten bzw. Bitfolgen.

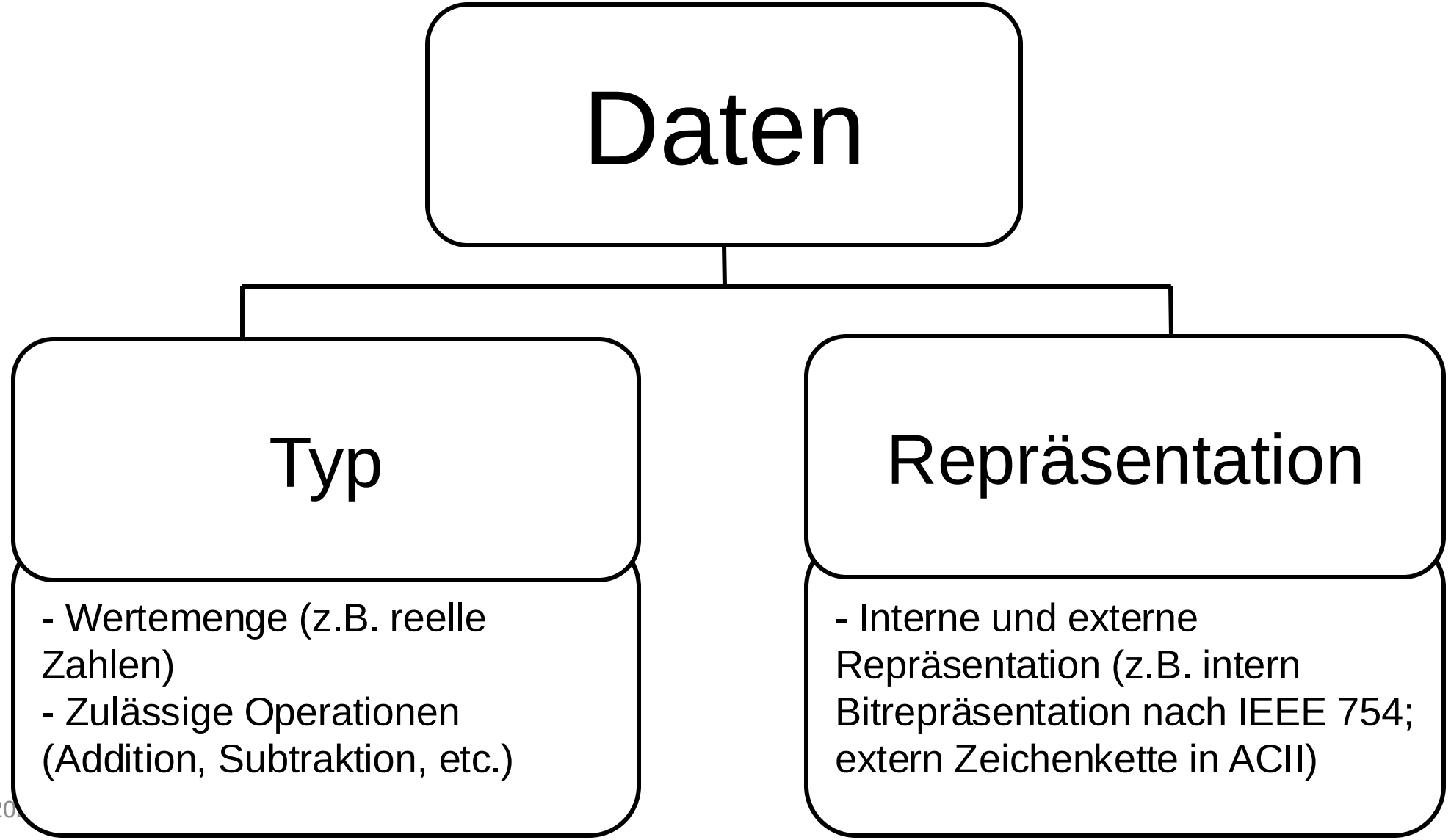
Information

Eine Mitteilung (Nachricht), die beim Empfänger eine Unbestimmtheit beseitigt, z.B. „Lagerbestand ist unter den Soll-Zustand“

Wissen

Im jeweiligen Kontext interpretierte Information,
z.B. „Kapazitäten der Produktion durch zusätzliche Schichten erhöhen.“

Daten und Datentypen



Klassifikation auf Basis ihrer Verwendung / Dynamik

Stammdaten: Daten, die nur selten verändert werden, z.B. Personalstamm, Kundenstamm, Materialstamm

Bestandsdaten: Weisen Bestände aus, z.B. Lagerbestände, Kontostände

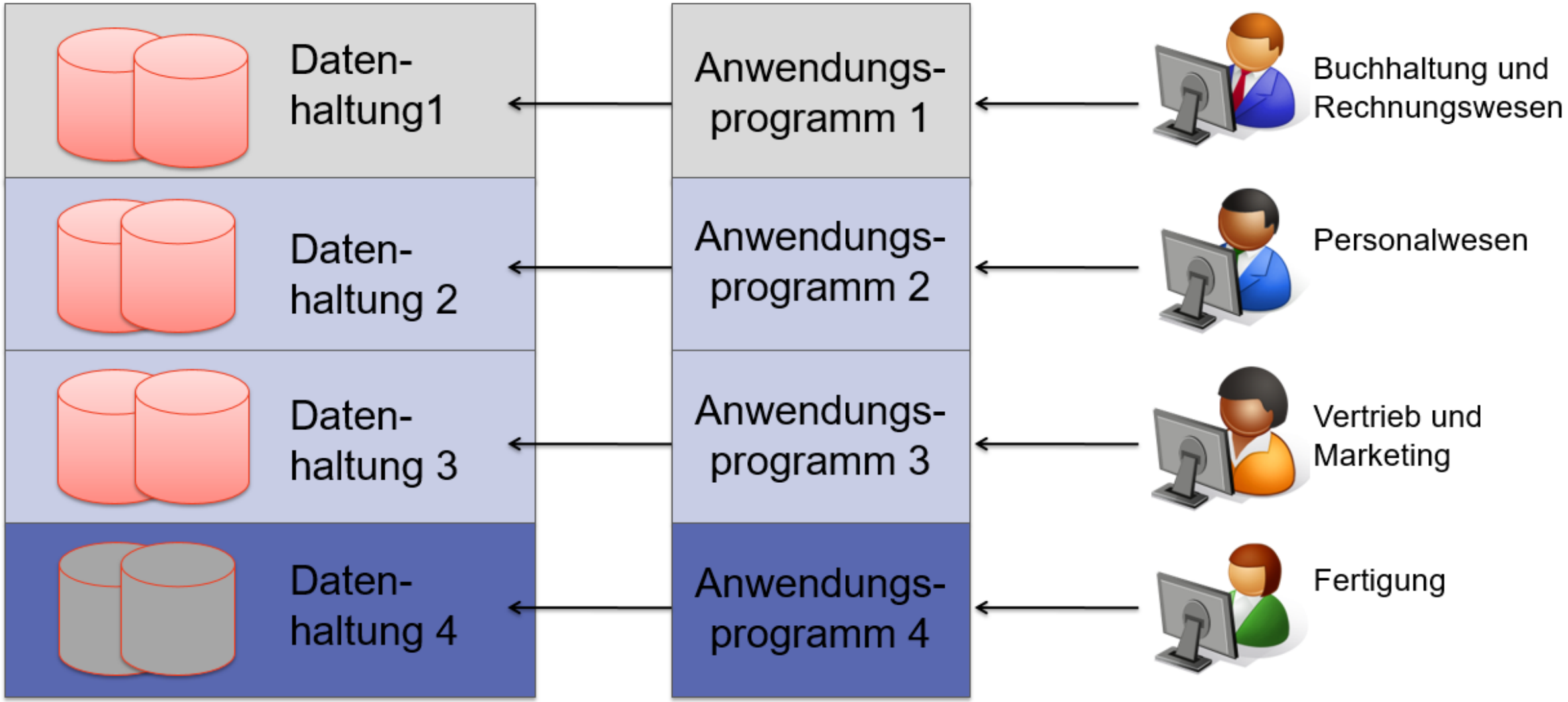
Bewegungsdaten: Daten mit mengen- oder wertmäßigen Zu- und Abgängen, z.B. Lagerentnahmen, Einzahlungen, Abbuchungen

Metadaten: Daten, die andere Daten beschreiben, z.B. Eigenschaften eines Datenfeldes, z.B. Fließtext oder feste Werte

Beispiel Krankenkasse

(087, Heinz, … (021, Grippeimpfung, …)	(087, 021, 997, offen)	(997, 087, 021, 80)	Daten
		 Bewegungs-daten	
- KK-Kunden (Kunden-Nr., Name, Adresse, etc.) - Behandlungsarten (Behandlungs-Nr., Adresse, etc.)	- Behandlungen (Kunden-Nr., Behandlungs-Nr., Rechnungs-Nr, Bezahlt?, etc.)	- Rechnungen (Rechnung-Nr, Kunden-Nr., Behandlungs-Nr., Kosten, etc)	Meta-daten

Traditionelle Datenverarbeitung – Separate Datenhaltung



Probleme separater Datenhaltung

Datenredundanz: Mehrfaches Vorkommen identischer Datenelemente in verschiedenen Tabellen/Verzeichnissen etc.; Redundanzen sind oftmals Ursache von inkonsistenten Daten, da mehrfach gespeicherte „gleiche“ Daten nicht überall gleich(zeitig) geändert werden

Dateninkonsistenz: Die selben Attribute haben verschiedene Werte. Dateninkonsistenz kann zu falschen Informationen (und damit evtl. falschen Entscheidungen) führen

Abhängigkeit zwischen Anwendung und Daten: Änderungen am Programmcode erfordern Änderungen an den zur Anwendung gehörenden Daten; Fehlender Datenaustausch und mangelnde Flexibilität; Mangelende Datensicherheit

Ziele der Datenorganisation (I)

Benutzerfreundlichkeit

- Leichte Änderbarkeit/Aktualisierbarkeit
- Schnelle Zugriffsmöglichkeiten (auf gespeicherte Daten)

Flexibilität

- Flexibilität der Auswertung und Verknüpfung der Daten

Datenschutz und Datensicherheit

- Schutz der Daten vor Verlust, Zerstörung, unbefugtem Zugriff
- Verhinderung der unberechtigten Verwendung der Daten

Ziele der Datenorganisation (II)

Datenintegrität

- Korrektheit/Vollständigkeit der Daten

Redundanzfreiheit

- Keine mehrmalige Speicherung gleicher Daten

Parallele Datenverarbeitung - Transaktionen

Eine Transaktion ist eine Folge von Operationen, die logisch eine Einheit bilden

Beispiel: Geld überweisen ist eine logische Einheit bestehend aus:

1. Geld von A abheben
2. Geld von A nach B transportieren
3. Geld bei B einzahlen



https://www.finanztip.de/blog/wp-content/uploads/2018/07/Blog_Edi-geld-ueberweisen-IconicBestiary_iStock-628883362.jpg

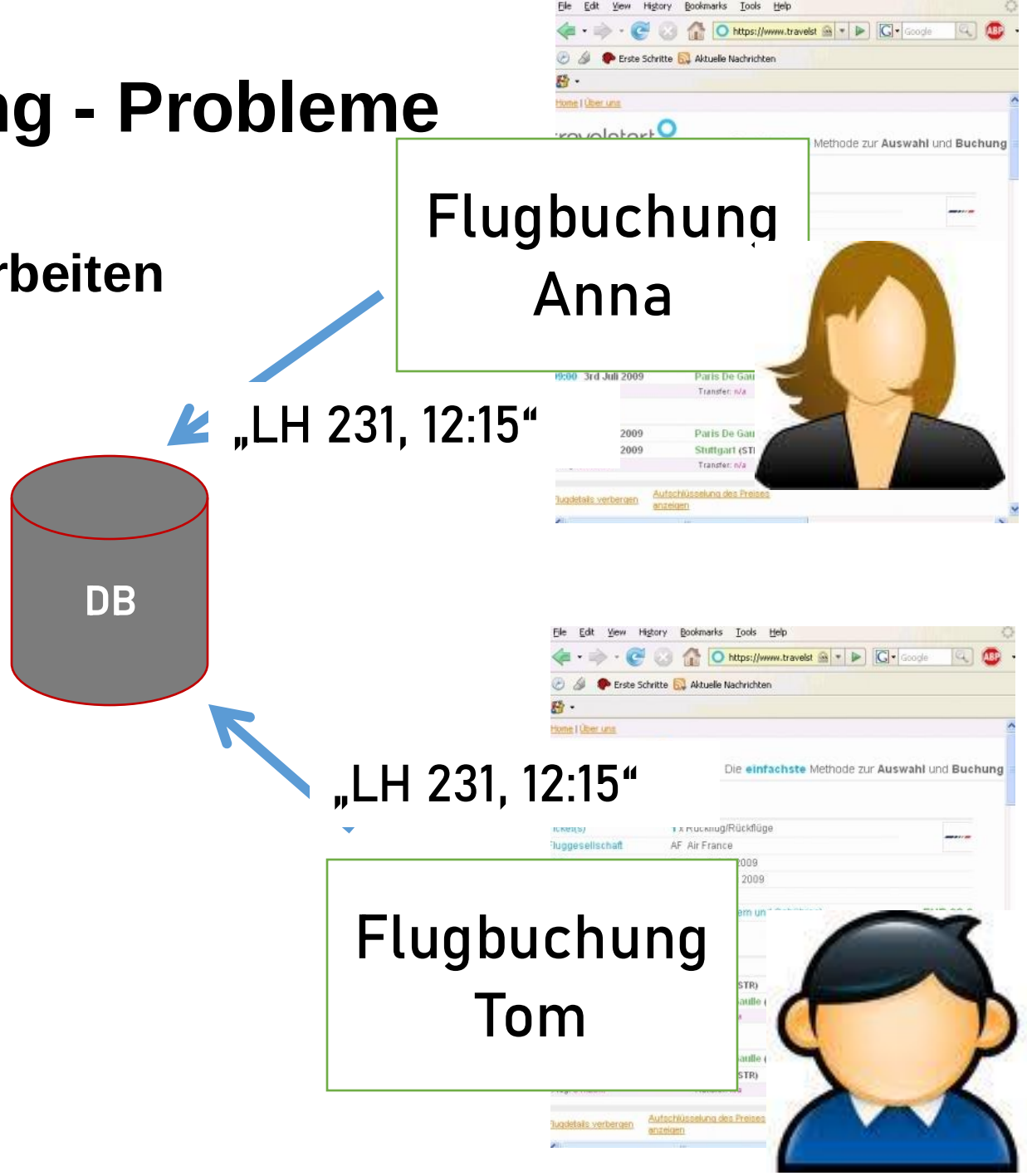
Parallele Datenverarbeitung - Probleme

Transaktion A und Transaktion B arbeiten gleichzeitig auf demselben Objekt

Beispiel: Flug buchen

Anna und Tom buchen gleichzeitig den letzten Platz des Fluges LH231

1. IstFrei
2. GebePlatzNr
3. ÜberweiseGeld
4. Drucke Ticket



Parallele Datenverarbeitung - Probleme

Transaktion A und Transaktion B arbeiten gleichzeitig auf demselben Objekt.

Beispiel: Flug buchen

Anna und Tom buchen gleichzeitig den letzten Platz des Fluges LH231

1. IstFrei
2. GebePlatzNr
3. ÜberweiseGeld
4. Drucke Ticket



Welche Fälle sollten nicht auftreten?

Parallele Datenverarbeitung - Probleme

Klassifikation der verschiedenen Fälle die nicht auftreten sollten:

Lost Updates: Zwei Transaktionen modifizieren parallel denselben Datensatz und nach Ablauf dieser beiden Transaktionen wird nur die Änderung von einer der beiden Transaktionen übernommen.

Dirty Read: Daten einer noch nicht abgeschlossenen, anderen Transaktion werden gelesen.

Non-Repeatable Read: Wiederholte Lesevorgänge liefern unterschiedliche Ergebnisse.

Phantom Read: Suchkriterien treffen während einer Transaktion auf unterschiedliche Datensätze zu, weil eine (während des Ablaufs dieser Transaktion laufende) andere Transaktion Datensätze hinzugefügt, entfernt oder verändert hat.

Ziele der Datenverwaltung: Transaktionsisolation

ACID-Prinzip (Atomicity, Consistency, Isolation und Durability)

Atomizität: Eine Transaktion ist eine atomare Verarbeitungseinheit; sie wird entweder vollständig oder überhaupt nicht ausgeführt.

Konsistenz: Eine Transaktion ist konsistenzbewahrend, wenn ihre vollständige Ausführung die Datenbank von einem konsistenten Zustand in einen anderen überführt.

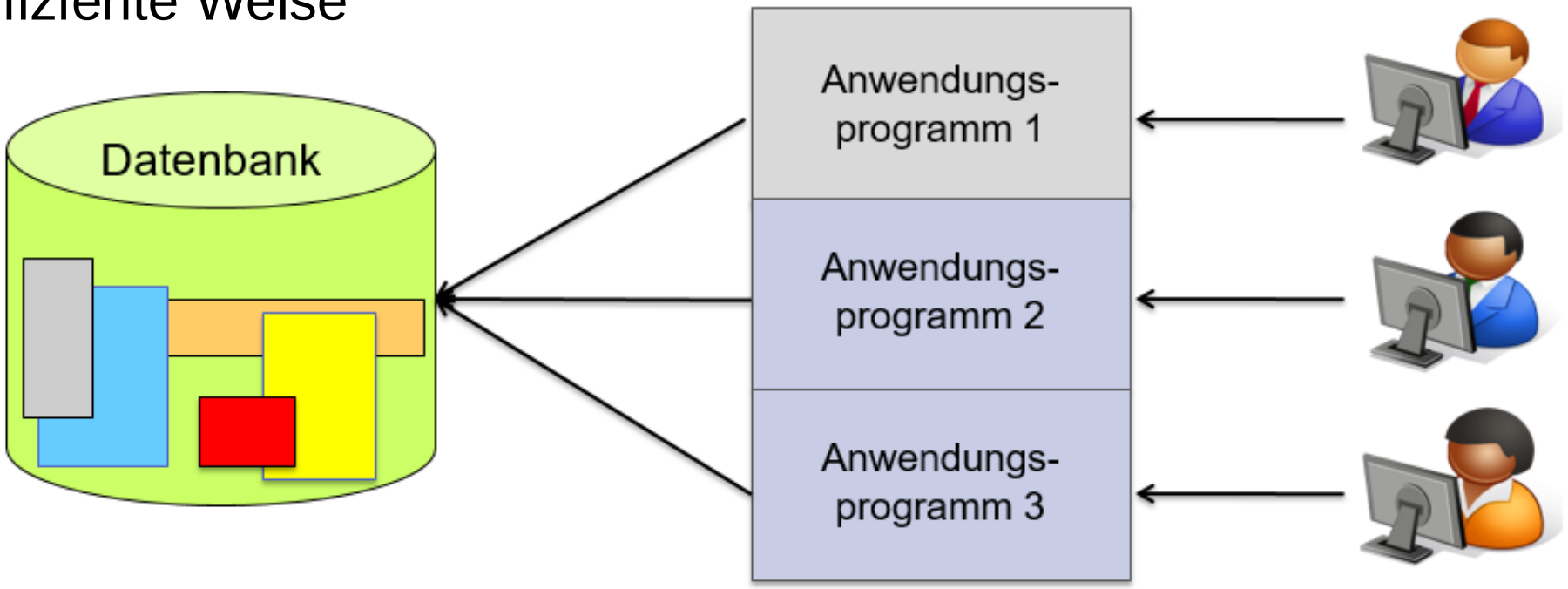
Isolation: Eine Transaktion soll so ausgeführt werden, als sei sie isoliert von anderen Transaktionen.

Dauerhaftigkeit: Die von einer bestätigten Transaktion in die Datenbank geschriebenen Änderungen müssen in der Datenbank fortbestehen.

Datenbanken

Möglichst redundanzfreie Sammlung von Daten, die den Zugriff von mehreren Benutzer oder Anwendungen ermöglicht

- gleichzeitiger Zugriff (ACID Prinzip)
- auf effiziente Weise



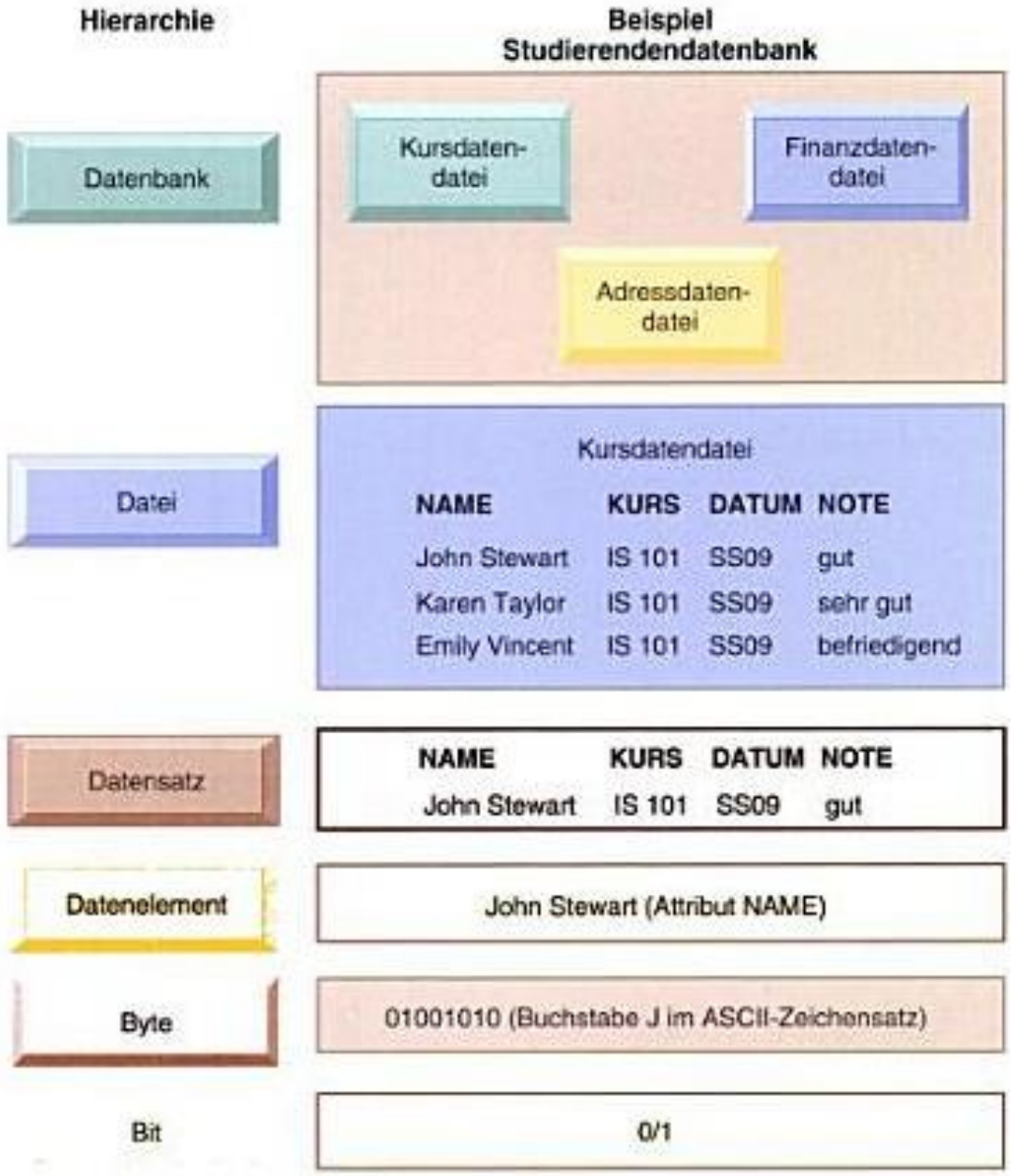
Hierarchischer Aufbau

Datenbank: Sammlung logisch zusammengehörender Dateien (und deren Verknüpfungen)

Datei: Zusammenfassung logisch zusammengehörender, gleichartiger Datensätze

Datensatz: Inhaltlich zusammenhängende Datenelemente werden zu einem Datensatz zusammengefasst

Datenelement: Die kleinste gespeicherte logische Dateneinheit



Datenbankmanagementsysteme (DBMS)

Sammlung von Programmen zum Erstellen und Verwalten einer Datenbank, die es mehreren Anwendungen gleichzeitig ermöglicht auf die Daten in effizienter Weise zuzugreifen.

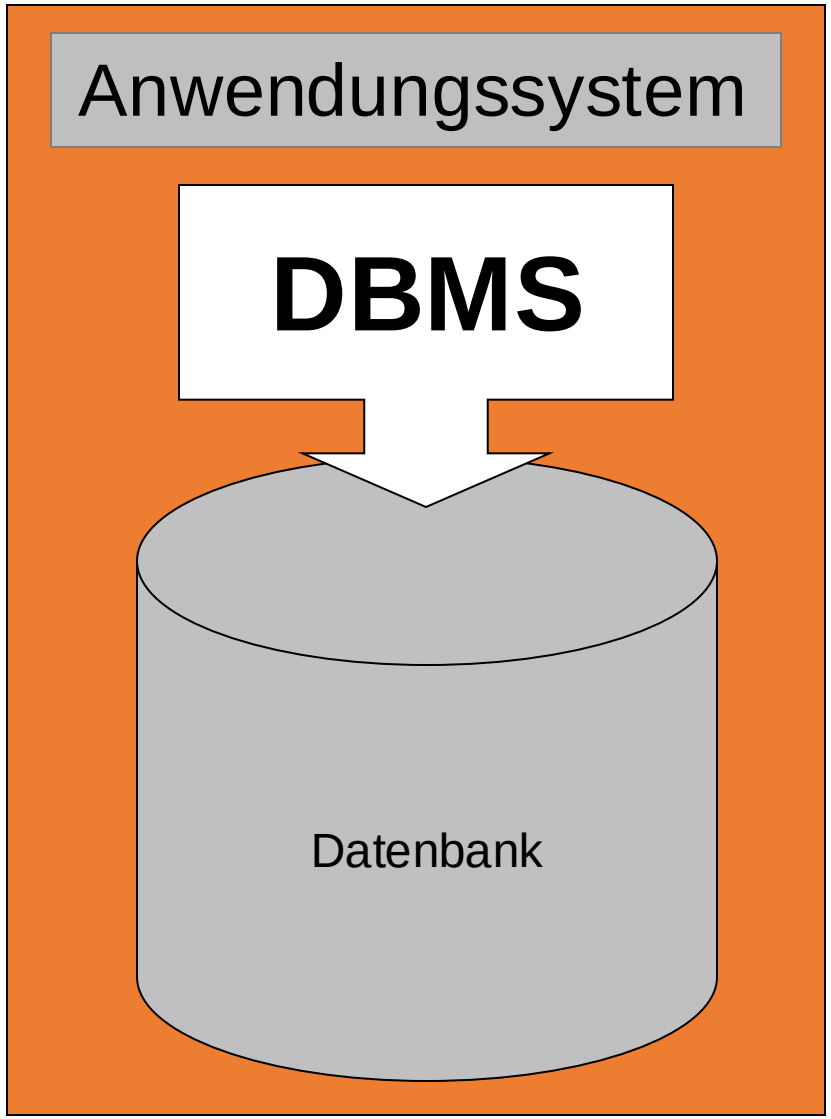
- Ermöglicht die Speicherung, Extrahierung und Manipulierung von Daten ohne eigene Dateien erstellen zu müssen
- Löst einige Probleme traditioneller Datenverarbeitung, z.B. Einhaltung des ACID- Prinzips, Redundanzfreiheit, etc.
- Reduziert Komplexität der Programmierung/Modellierung der Datenverwaltung
  - Trennt die physische und logische Datenstruktur
  - Trennt Programmcode und Daten

Aufgaben des DBMS

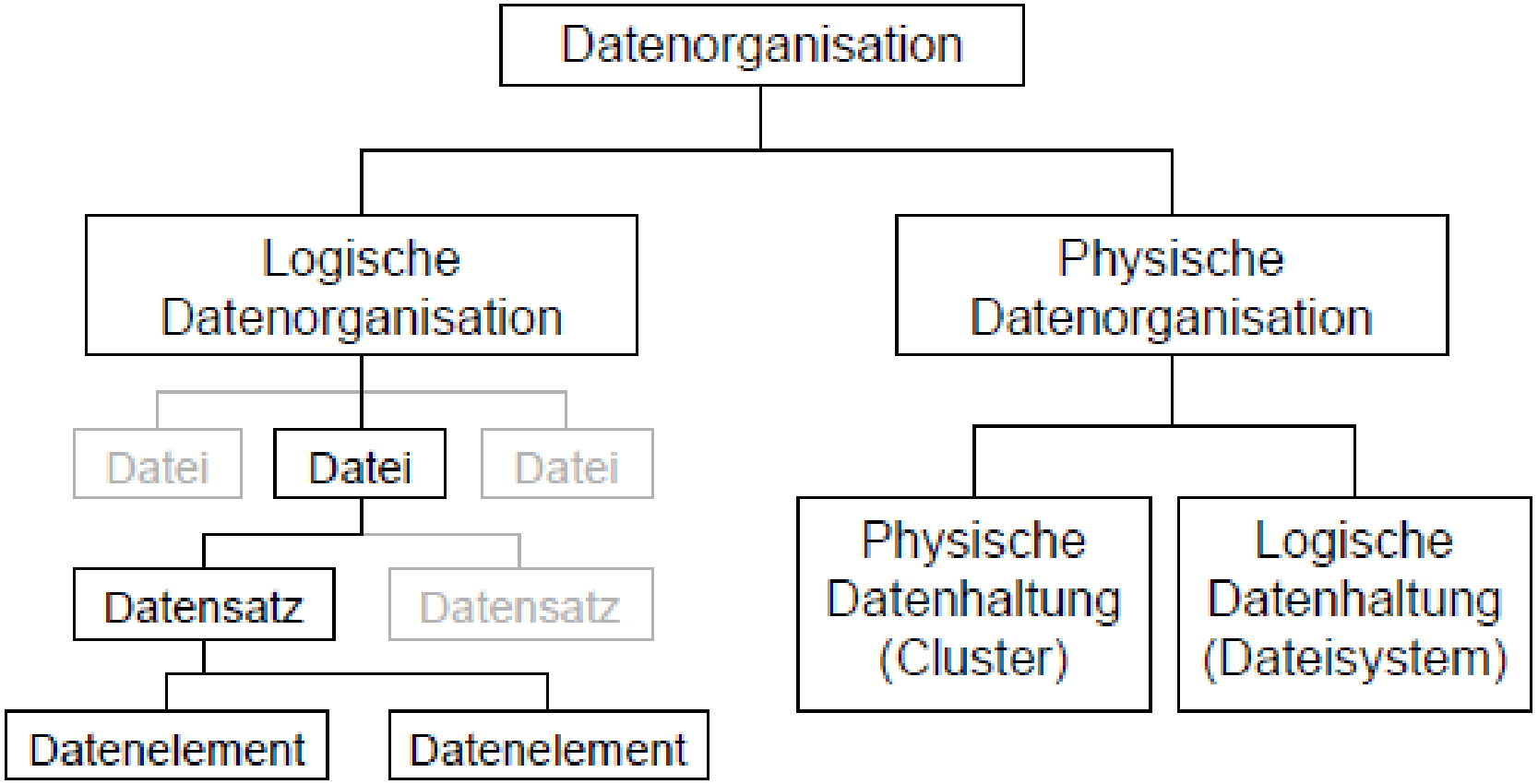
Das DBMS ist die **Schnittstelle zwischen dem Anwendungssystem und der Datenbank** und bietet den alleinigen physikalischen Zugriff auf den Datenbereich.

Das DBMS verantwortet dabei die **Benutzerverwaltung** sowie die **Transaktionsverwaltung** und setzt das **Datenmodell** um.

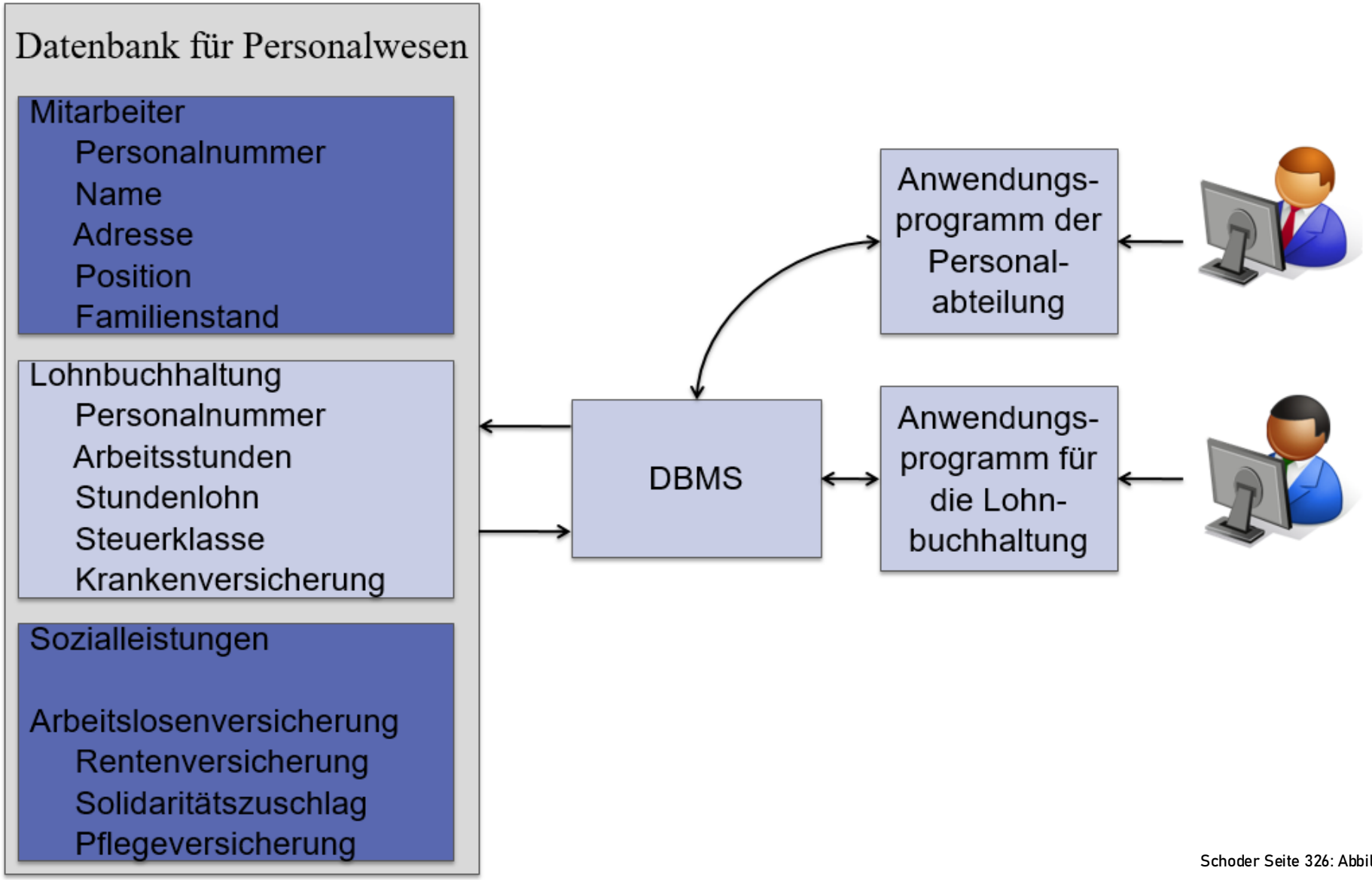
Ein DBMS bietet **Abfragesprachen**, um Daten zu lesen/schreiben und **Spezifikationssprachen**, um Datenmodelle zu erstellen.



Aufbau der Datenorganisation



Beispiel



Schoder Seite 326: Abbildung 7.4

Gliederung

1. Grundgedanken der ER-Modellierung

2. Datenorganisation

3. Relationales Datenbankschema

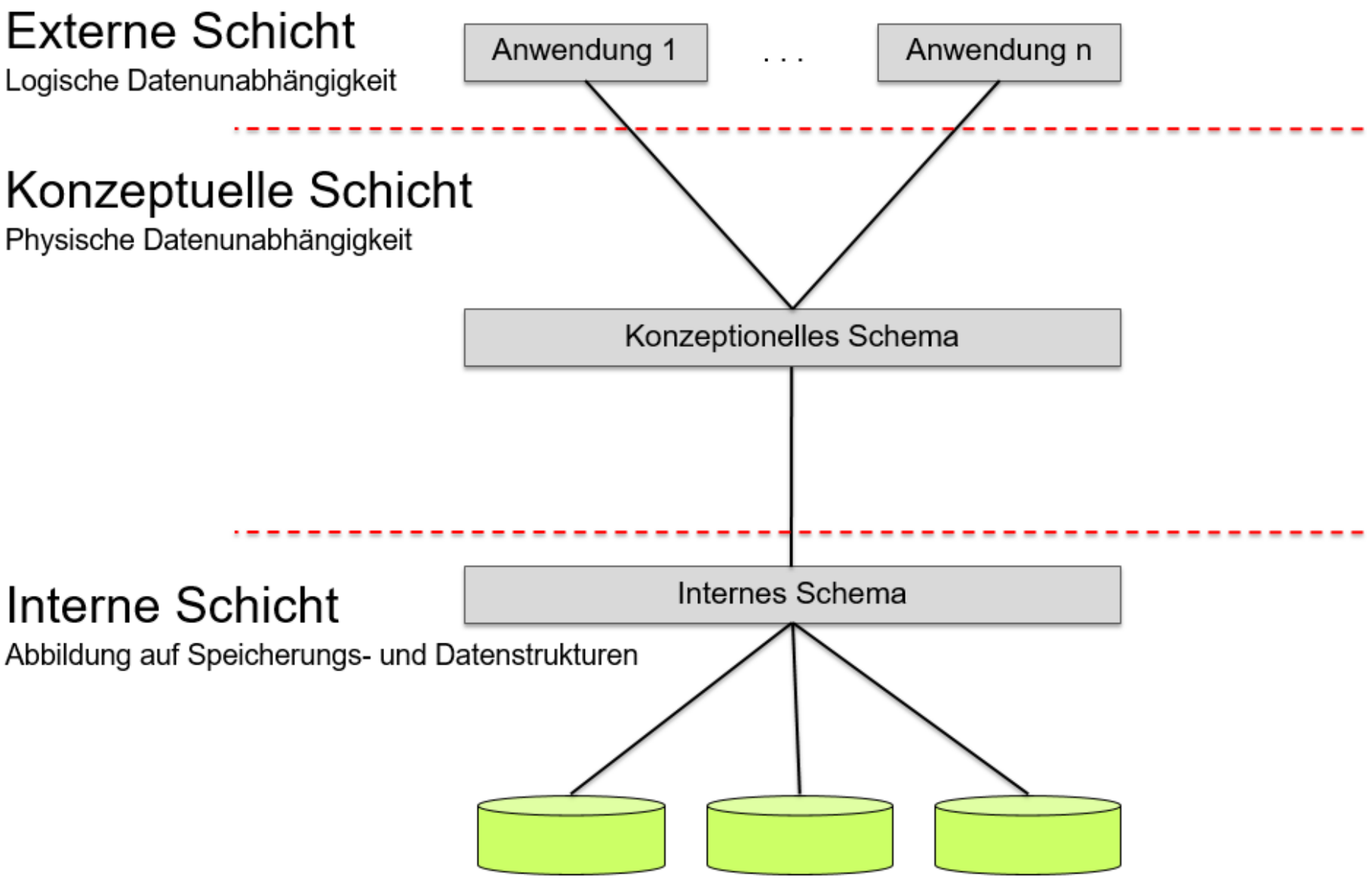
SPARC/ANSI 3-Ebenen Modell

Das 3-Ebenen Modell wurde 1975 vom **Standard Plaing Requirement Committee (SPARC)** entwickelt.

Die Ziele waren eine **Standardisierung von Datenbankarchitekturen**, ein hohes Maß an **Datenunabhängigkeit** und die Möglichkeit **Änderungen auf einer Ebene durchzuführen** ohne andere Ebenen zu beeinträchtigen.

Anwendung des **Seperation of Concerns – Prinzips.**

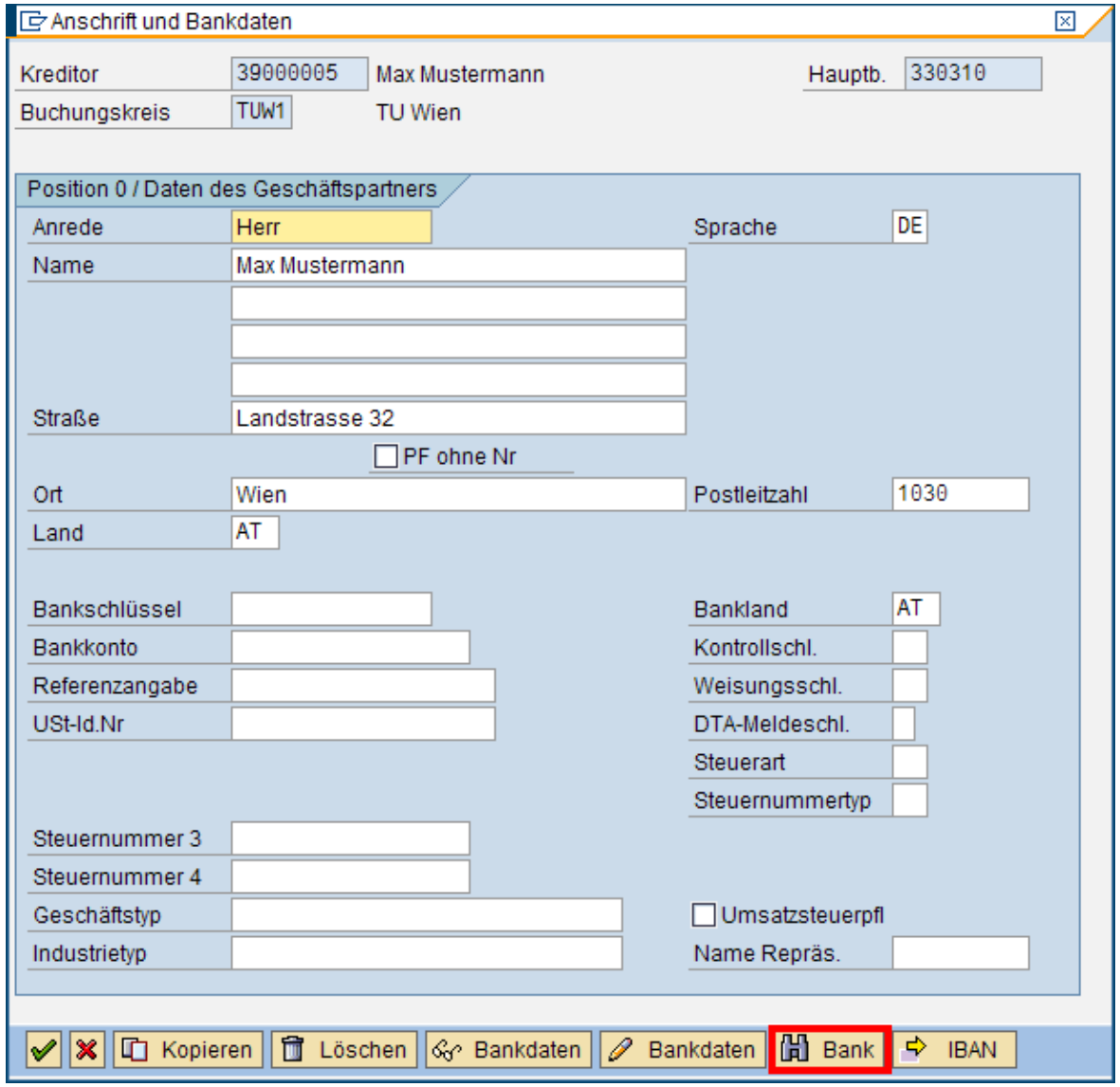
SPARC/ANSI 3-Ebenen Modell



Externe Schicht

Die externe Sicht eines oder mehrerer Benutzer auf die Daten.

Für den Benutzer nicht relevante Daten werden vor ihm verborgen.



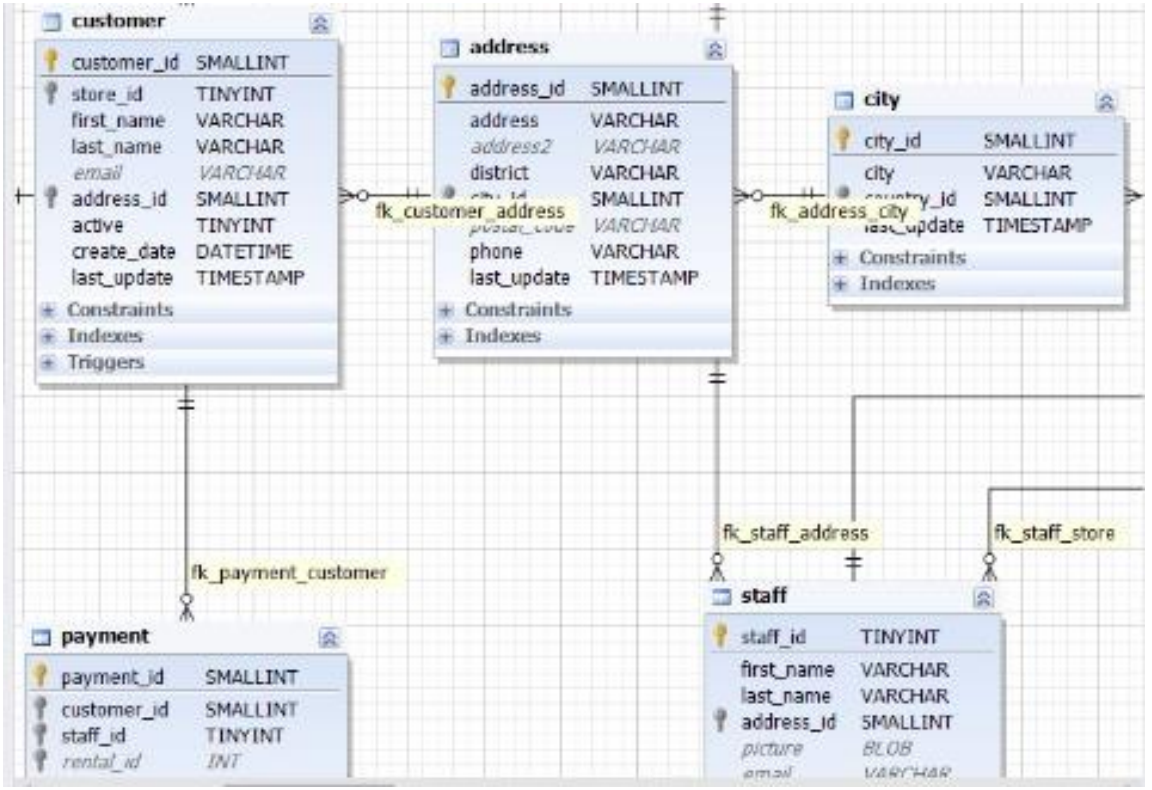
https://docplayer.org/docs-images/26/2086830/images/48-0.png

Logische / Konzeptuelle Schicht

Das konzeptuelle Schema legt die Strukturen der konzeptuellen Sicht der gesamten Datenbank fest.

Berücksichtigt werden **Entitäten, Datentypen, Beziehungen und Integritätsbedingungen.**

Physikalische Speicherstrukturen werden verborgen.



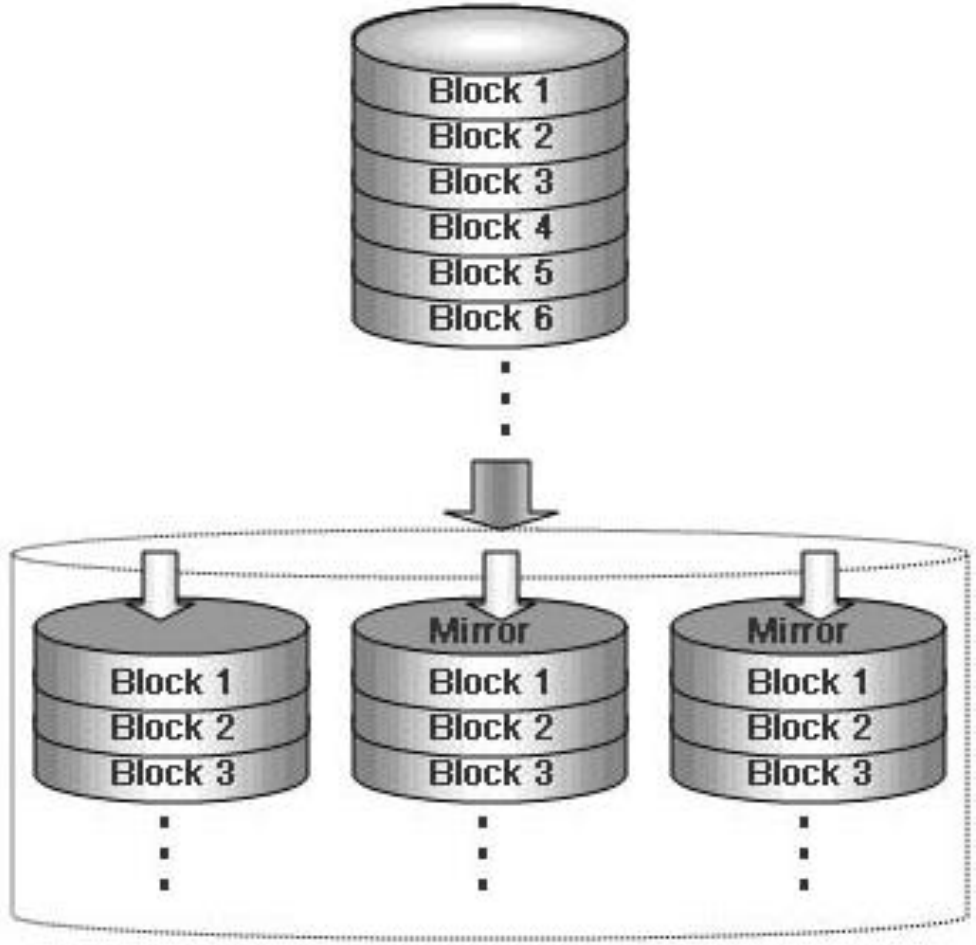
http://www.databaseanswers.org/images/dbforge_screenshot.jpg

Physische / Interne Schicht

Beschreibt die **physikalischen Speicherstrukturen** der Datenbank.

Fokus auf effiziente Abspeicherung der Datenbank-Objekte auf dem Speichermedium

- Physische Datenhaltung (Cluster)
- Logische Datenhaltung (Dateisystem)



DBMS Sprachen

Data Query Language (DQL)

- Sprache zur Formulierung von Abfragen

Data Manipulation Language (DML)

- Sprache zur Bearbeitung des Datenbestandes (Ändern, Löschen, …)

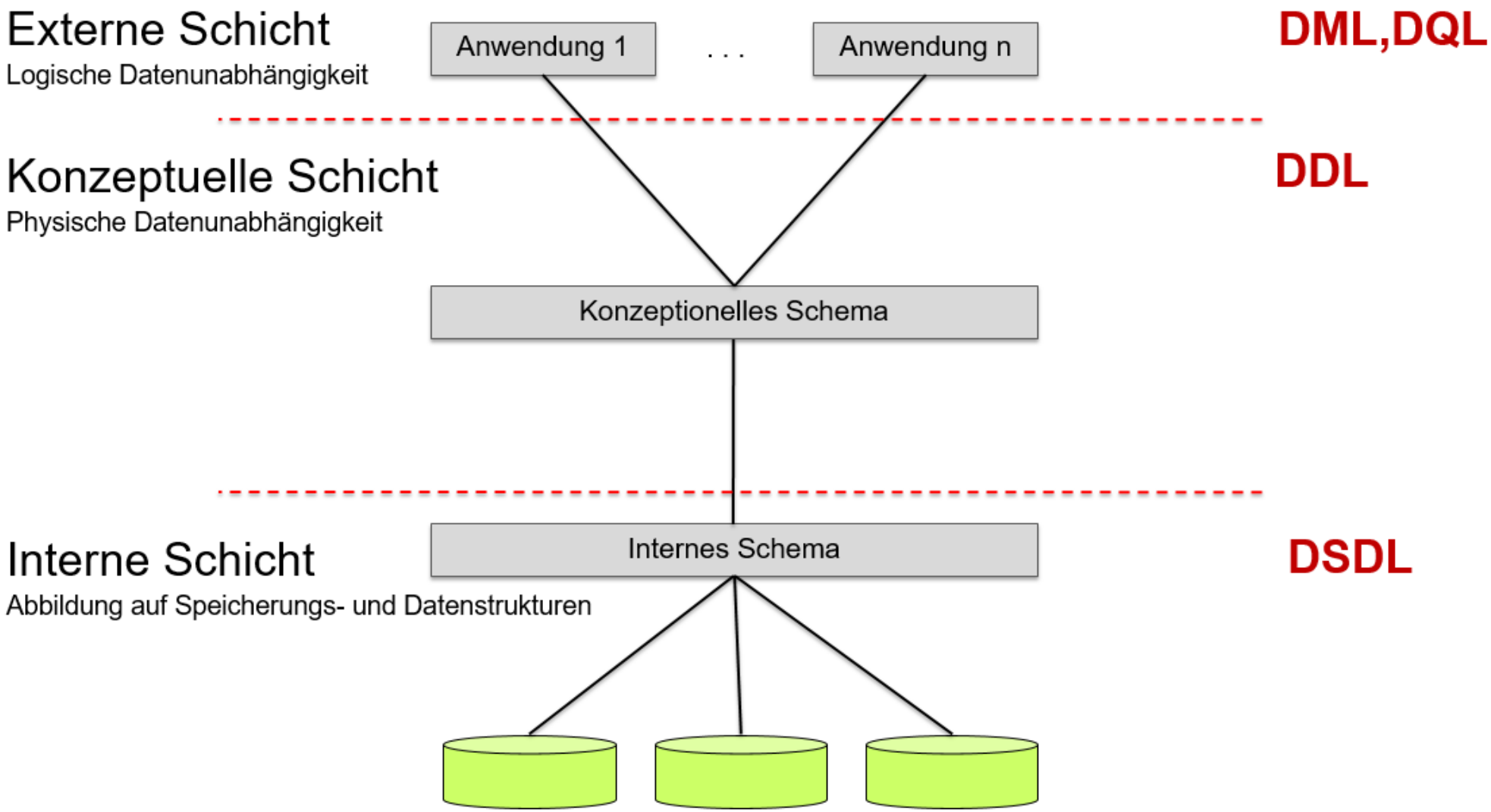
Data Definition/Description Language (DDL)

- Sprache zur Beschreibung der logischen Datenstrukturen einer Datenbank

Data Storage Description Language (DSDL)

- Sprache zur Beschreibung der physischen Organisation der Daten innerhalb des Datenbanksystems

DBMS Sprachen

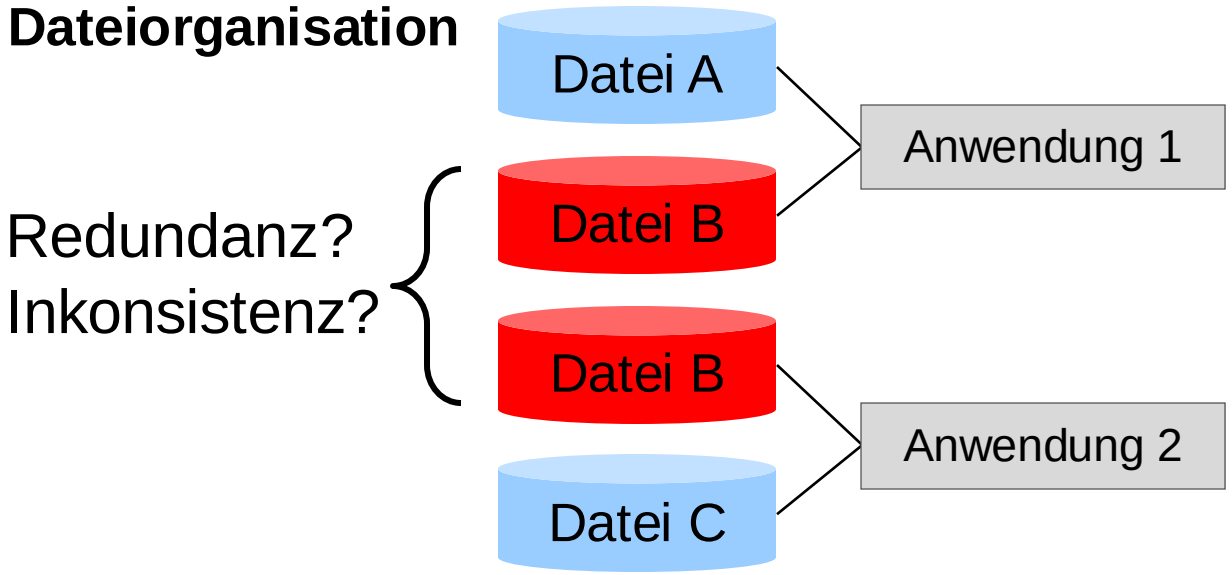


3-Ebenen Modell - Zusammenfassung

Benutzerebene	• Erzeugt Sichten (Views), um Benutzern die Daten darzustellen • Wie kann ich mit welchen Daten den Benutzer bei seiner Arbeit unterstützen?
konzeptionelle Ebene	• Welche Daten werden gespeichert? • In welcher Beziehung stehen sie zueinander? • Wer darf auf welche Daten zugreifen? • Wie wir die Datenintegrität hergestellt und geschützt? • Wie werden Redundanzen verhindert? • Abbildung der Wirklichkeit in Tabellen
physikalische Ebene	• Zugriffsverfahren • Speicherstruktur

Datei- vs. Datenbankorganisation : Zusammenfassung

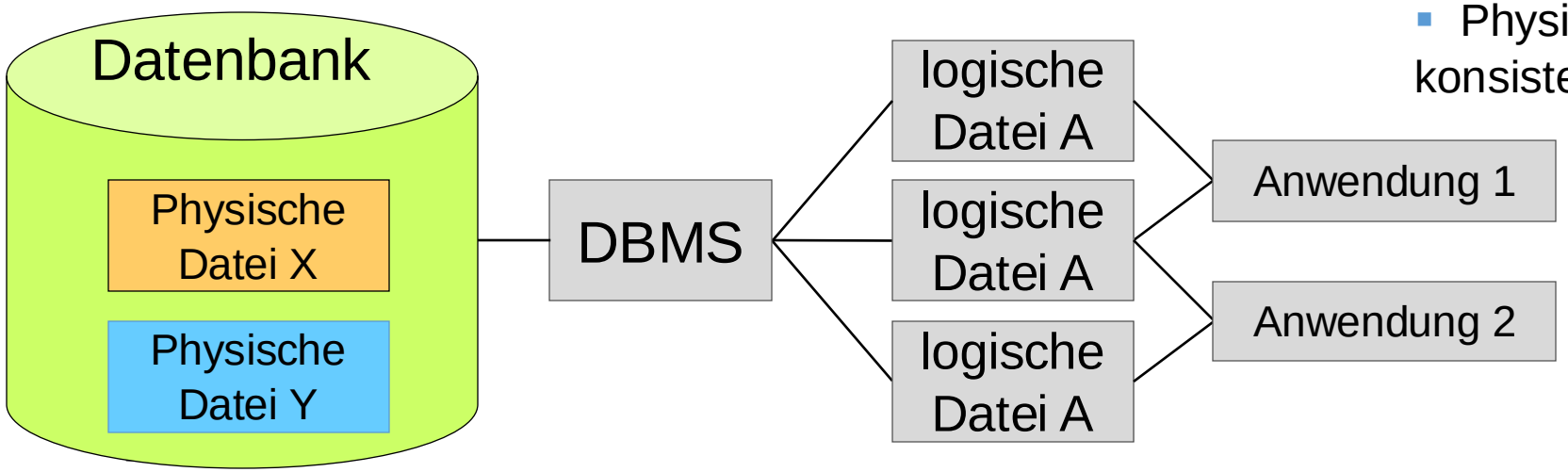
Dateiorganisation



Dateiorganisation:

- Wiederholte Speicherung gleicher Daten. Datei B ist redundant
- erhöhter Speicherplatzbedarf
- Konsistenzprobleme

Datenbankorganisation



Datenbankorganisation :

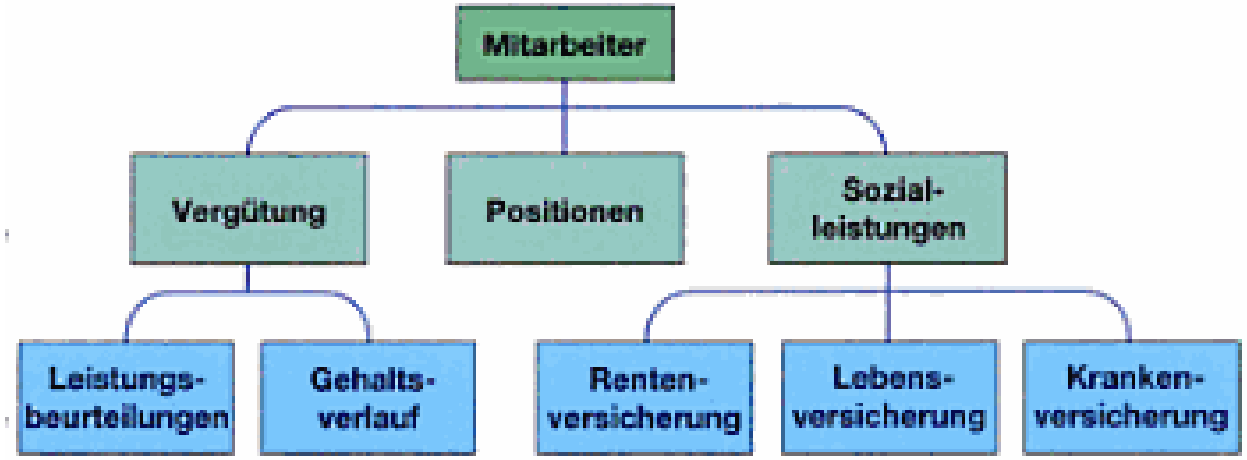
- Physische Daten sind redundanzfrei und konsistent
- Die Daten sind unabhängig von der einzelnen Programmen, die auf sie zugreifen

Hierarchisches Datenmodell

Das hierarchische Datenmodell ist das älteste logische Datenmodell.

Die Datensätze werden **durch Eltern-Kind-Beziehungen** verknüpft (liefert Baumstruktur).

Hierarchische Datenspeicherung erlebt Renaissance mit XML

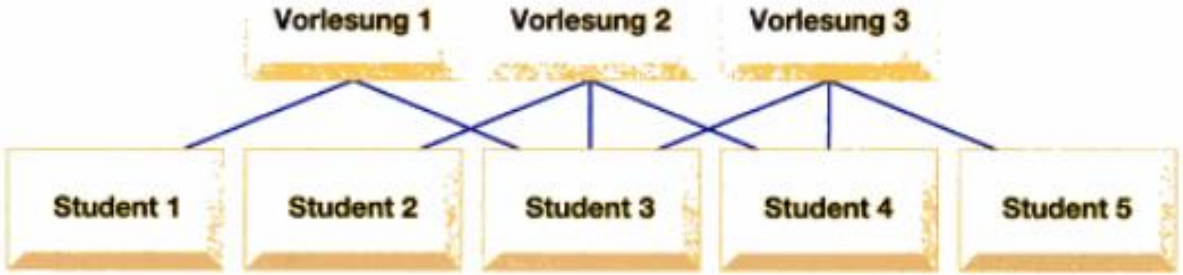


Laudon et al. 2010, 299

Netzwerk Datenmodell

Das Netzwerk-Datenmodell ist ein älteres Datenmodell und sehr nah an der Implementierung.

Das Modell fordert keine strenge Hierarchie sondern kann auch m:n-Beziehungen abbilden, D.h. ein Datensatz kann mehrere Vorgänger haben. Auch können mehrere Datensätze an höchster Stelle stehen.



Laudon et al. 2010, 299

Relationales Datenbankmodell (I)

Relationale Datenbankmodelle sind in **Tabellen (Relations)** organisiert, die voneinander weitgehend unabhängig sind.

Eine **Tabelle** ist ein Konstrukt aus Zeilen (Reihen, rows, auch Tupel genannt) und Spalten (columns).

Eine **Zeile** ist eine Sammlung von Attributen eines Datensatzes (Objekt, auch Entität), die Spalten stellen Attribute gleicher Art (z.B. Name, Vorname) dar.

Tabellen können dadurch verknüpft werden, dass sie (mindestens) ein gleiches Attribut beinhalten.

Es können **beliebig viele Sichtweisen** (views) erzeugt werden, die eine Zusammenstellung verschiedener **Attribute** und verschiedener Tabellen beinhaltet.

Der **Bereich der Werte**, die ein Attribut (d.h. ein Eintrag in einer bestimmten Spalte) annehmen kann, nennt man **Wertebereich** (domain) des Attributes.

Relationales Datenbankmodell (II)

Tabelle = Relation

Spalten=Colums

KNr	Name	Vorname	PLZ	Ort	Strasse
1	Hugenbubel	Herbert	12345	Steinhausen	Tomatenweg 10
2	Spitzweg	Carl	81526	Himmelhausen	Sternenweg 18
3	Cron	Maria	36645	Blaupausen	Kopierstr. 18

Rows(Tupel)

Sammlung aller Attribute eines Datensatzes

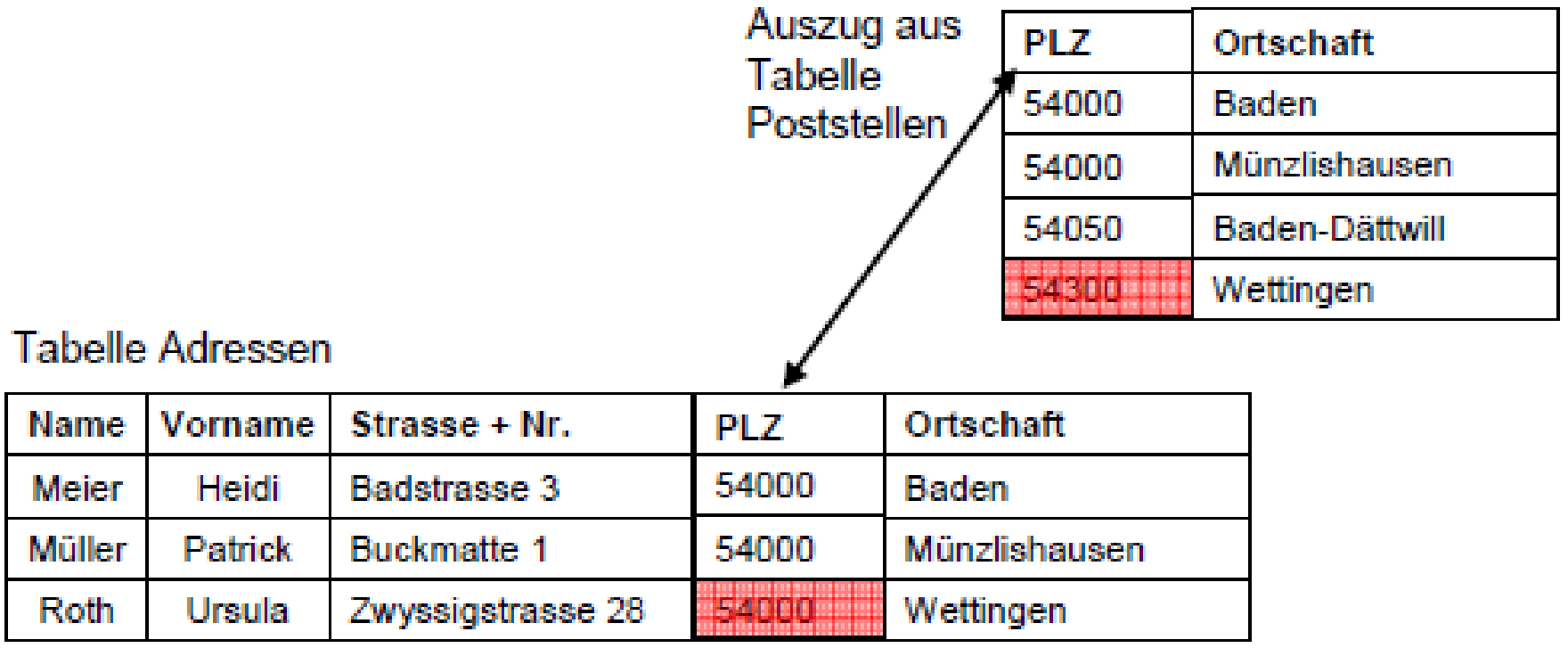
Feld, alle Felder einer Zeile bilden einen Datensatz, der alle Attribute eines Objektes (Entität) umfasst. (vgl. Tupel)

Sammlung gleicher Attribute

Die Spalte KNr. eignet sich dazu, als Primärschlüssel definiert zu werden, weil alle Werte eineindeutig sind, da sie nur einmal vorkommen.

Relationales Datenbankmodell (III)

Enthält ein Datensatz ein Attribut, welches ein Objekt eindeutig charakterisiert, z.B. Kundenummer) kann dieses Objekt als **Primärschlüssel** definiert werden. Wird der Primärschlüssel in einer anderen Tabelle ebenfalls als Attribut verwendet, so spricht man von einem **Fremdschlüssel**. Mit Hilfe dieser Schlüssel können die Tabellen in Beziehungen zueinander gesetzt werden.



Zusammenfassung der ERM und Datenorganisation

Das **ERM** modelliert **statische Datenstrukturen** in einer gegebenen Domäne.

Planung einer Datenbank gliedert sich unter Bezug des ER-Modells in die **5 Schritte**:

1. Entw. des semantischen Modells.
2. Erfassung der Entitätstypen.
3. Zuordnung der Attribute.
4. Analyse der Entitätsbeziehungen.
5. Erstellung des ER-Diagramms

Die Datenorganisation gliedert sich in die **logische** und **physische Datenorganisation**.

Datenbankmanagementsysteme (DBMS) sind eine **Sammlung von Programmen zum Erstellen und Verwalten einer Datenbank**, die gleichzeitigen effizienten Datenzugriff ermöglicht.

Nächste Termine

Nächste Vorlesung:

04.12.2024