KONZEPTUELLES DATENBANKEN-DESIGN

Batini, Ceri, Navathe, *'Conceptual Database Design'*, The Benjamin/Cummings Pub., 1992

ISBN 0-8053-0244-1

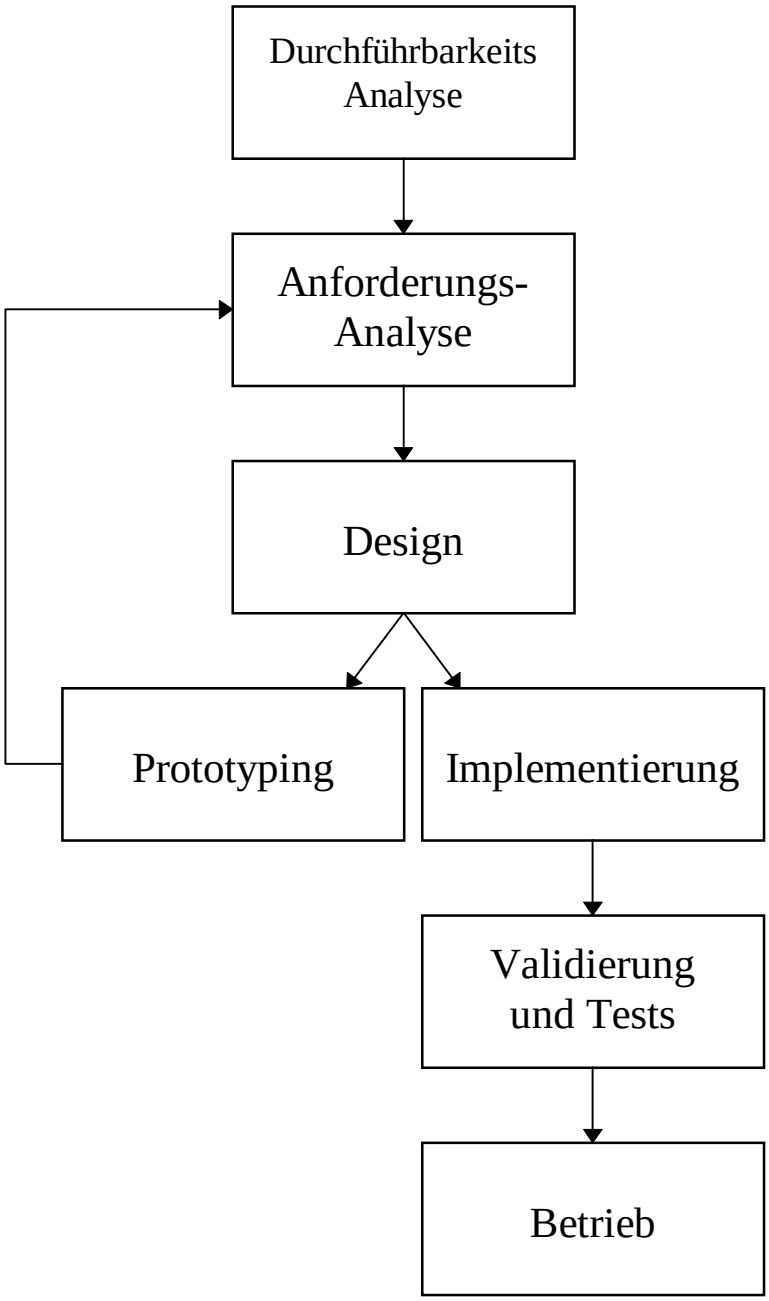
Part I: Kapitel 1 und Kapitel 2

Methode des Datenbanken-Designs

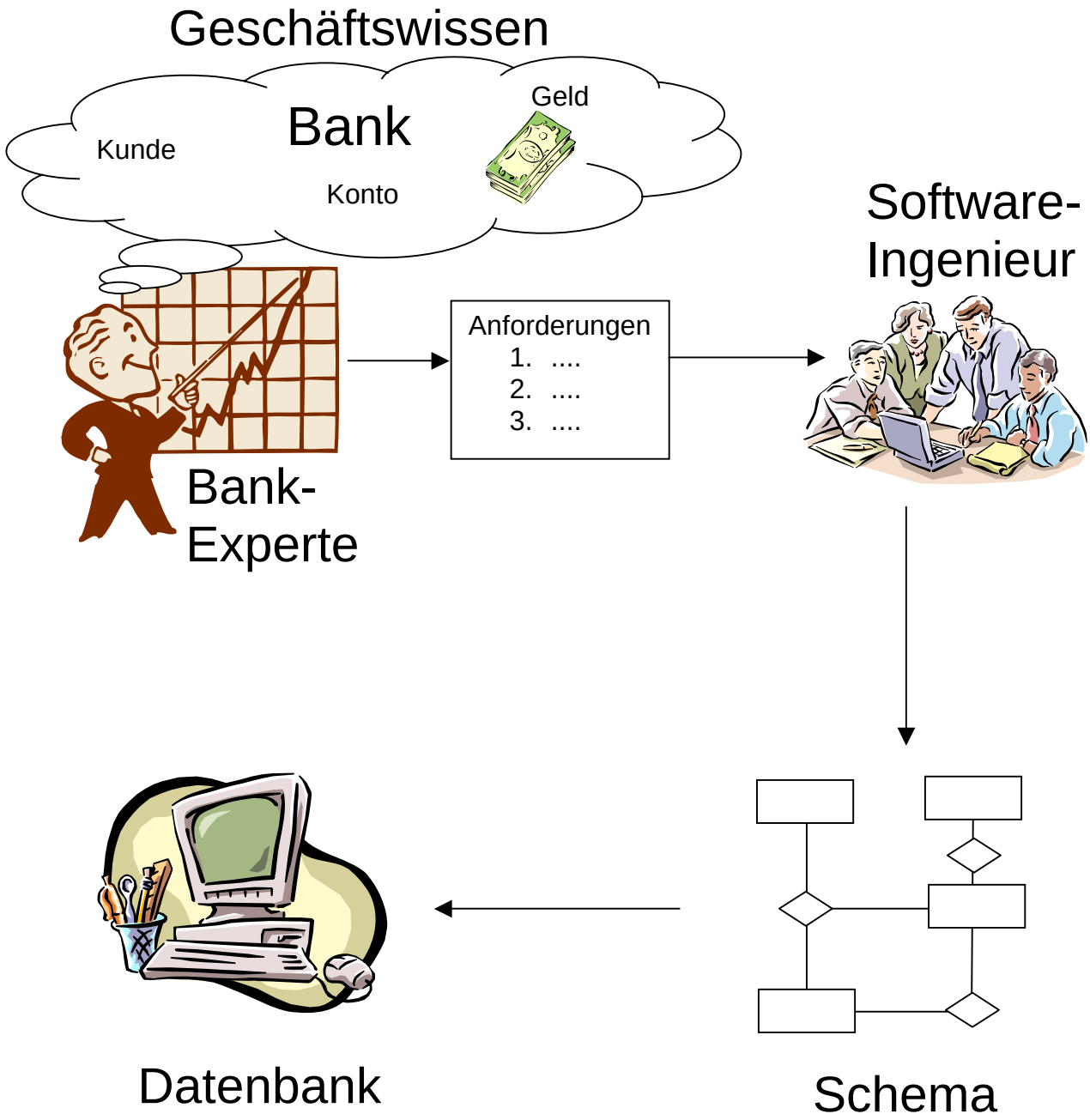
- Eine Folge von Schritten, um eine Menge von Anforderungen in ein logisches und ein physikalisches Schema zu transformieren.

- Eine präzise Definition der Qualität der ausgegeben Ergebnisse.

- Werkzeuge, Techniken und Strategien, um die Transformationen zu unterstützen, mit dem Ziel die erwartete Qualität an Ergebnissen zu bekommen.



Lebenszyklus eines Informationssystems


Geschäftswissen
Bank
Kunde
Konto
Geld
Bank-
Experte
Anforderungen
1. ....
2. ....
3. ....
Software-
Ingenieur
Datenbank
Schema

Beispiel-Anforderungen einer Bank

- Es werden Daten über Kunden, Konten, Buchungen, Filialen, Abteilungen und Mitarbeiter gespeichert.
- Jeder Kunde hat eine Anschrift.
- Es gibt verschiedene Arten von Konten: Girokonto, Kreditkonto und Anlagekonten.
- Girokonten können nur um einen bestimmten Betrag (Limit, Dispokredit) überzogen werden.
- Kreditkonten haben einen festen Zinssatz.
- Kunden kann ein Kredit- und/oder ein Anlageberater zugeteilt sein.
- Ein Kunde hat seine Konten bei einer bestimmten Filiale.
- Eine Filiale/Abteilung hat Mitarbeiter.
- Eine Filiale/Abteilung hat einen Leiter.
- Jeder Mitarbeiter arbeitet in einer bestimmten Filiale/Abteilung.
- Jeder Mitarbeiter hat eine feste Position mit bestimmten Aufgaben.
- …

Phasen des Datenbanken-Designs

Konzeptuelles Design

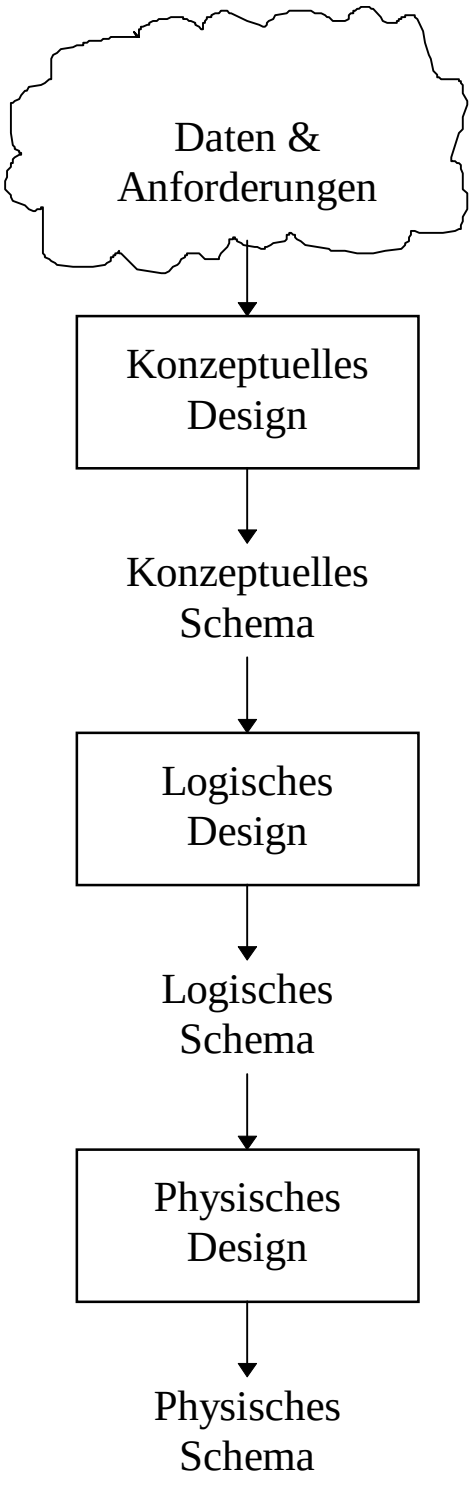
- Eingabe: Anforderungen
- Ausgabe: Eine DBMS unabhängige high-level Repräsentation der Anforderungen, das "konzeptuelle Schema".
- Qualität: Richtigkeit und Lesbarkeit der Repräsentation.

Logisches Design

- Eingabe: Konzeptuelles Schema.
- Ausgabe: Ein logisches Schema, das dem Datenmodell des DBMS entspricht.
- Qualität: Richtigkeit der Übersetzung, Effizienz des Schemas in Bezug auf Transaktionen.

Physikalisches Design

- Eingabe: Logisches Schema.
- Ausgabe: Ein physikalisches Schema für ein spezielles DBMS.
- Qualität: Performanz.



Datenbezogener Ansatz für das Datenbanken-Design

Konzeptuelles Schema:

Eine Beschreibung der Datenbankstruktur auf höherer Ebene, unabhängig von der Implementierung der DBMS-Software (Datenobjekte, Beziehungen, etc.).

Logisches Schema:

Eine Beschreibung der Datenbankstruktur, die von der DBMS-Software verwendet werden kann (Relationen, Domänen, etc.).

Physisches Schema:

Eine Beschreibung der Implementierung der Datenbank im Sekundärspeicher (Speicherstrukturen und Zugriffsmethoden).

Eigenschaften des konzeptuellen Designs

- Eine große Menge an Abstraktionen, Teil des konzeptuellen Datenmodells.

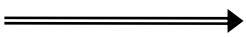
- Verschiedene Design Strategien sind verfügbar für:
  - Schema Design
  - Schema Integration
  - Schema Restrukturierung
  - Schema Pflege

- Saubere Schnittstellen zu
  - Funktionaler Analyse und Design
  - Logischem Design
  - Verteiltem Design

Abstraktionen im konzeptuellen Datenbank-Design

Abstraktion ist ein gedanklicher Prozess, der benutzt wird, um einige Eigenschaften einer Menge von Objekten in den Vordergrund zu bringen. Andere Eigenschaften werden dabei vernachlässigt.

Im konzeptuellen Design werden drei Arten von Abstraktionen benutzt:

- Klassifikation
- Aggregation 
- Generalisierung 

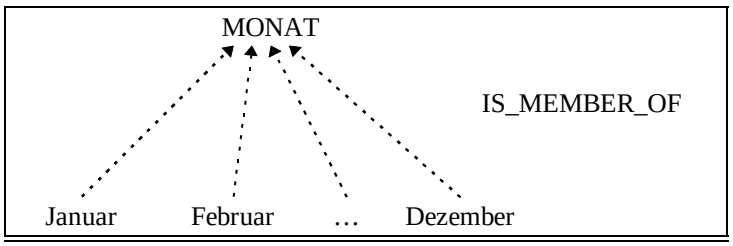
Begriffe:

Objekttypenebene = Objektklassenebene = Schema

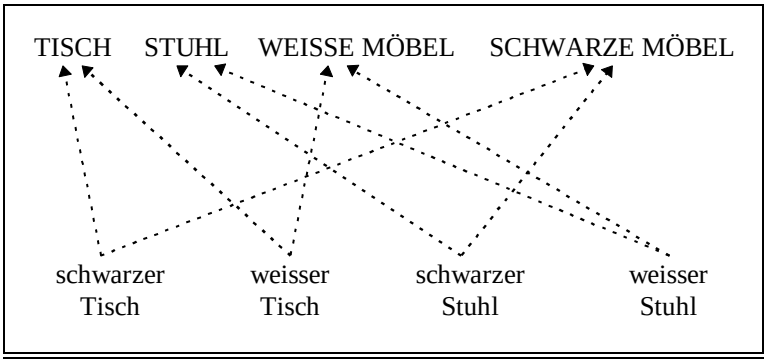
Objektebene = Exemplarebene (Instance-level)

Klassifikation (Classification)

Die **Klassifikation** definiert ein Konzept für eine **Klasse** von Objekten der realen Welt, die durch **gemeinsame Eigenschaften** gekennzeichnet sind.

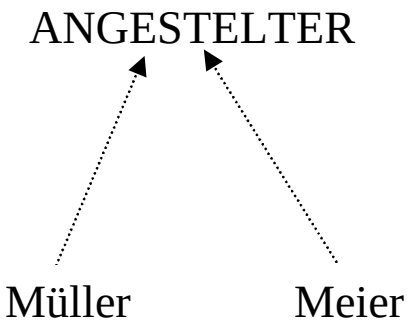


Jedes Objekt der realen Welt kann mehreren Klassen angehören:



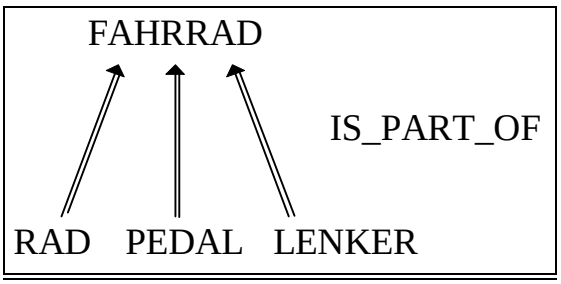
Beispiel 2:

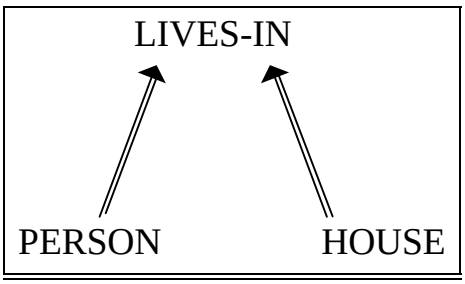
Zusammenfassung von „Angestellter Müller“, „Angestellter Meier“ zur Klasse „Angestellter“.



Aggregation:

Eine **Aggregation** definiert eine **neue Klasse** *K* aus einer Menge *M* von Klassen. Die Objekte aus *K* werden durch Verbindungen von Objekten aus den Klassen, die in *M* enthalten sind, repräsentiert.



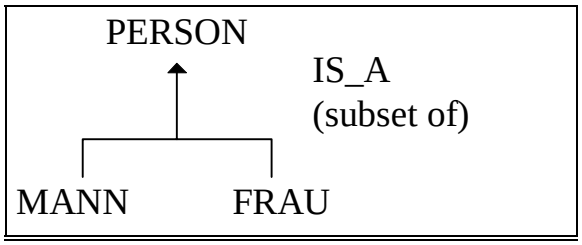


Konventionelle Record-Strukturen basieren nur auf Klassifikation und Aggregation:

- Ein Record ist eine Aggregation von Feldern.
- Ein Feld ist eine Klasse von Werten.

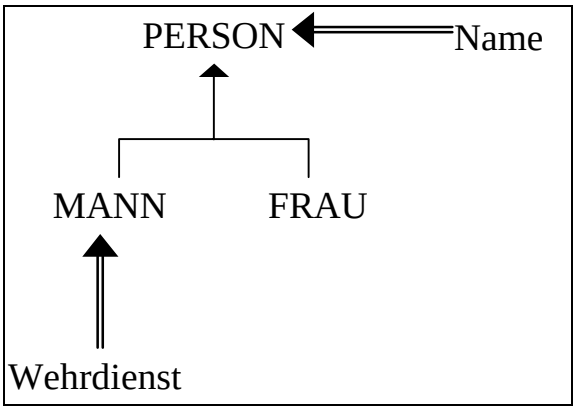
Generalisierung:

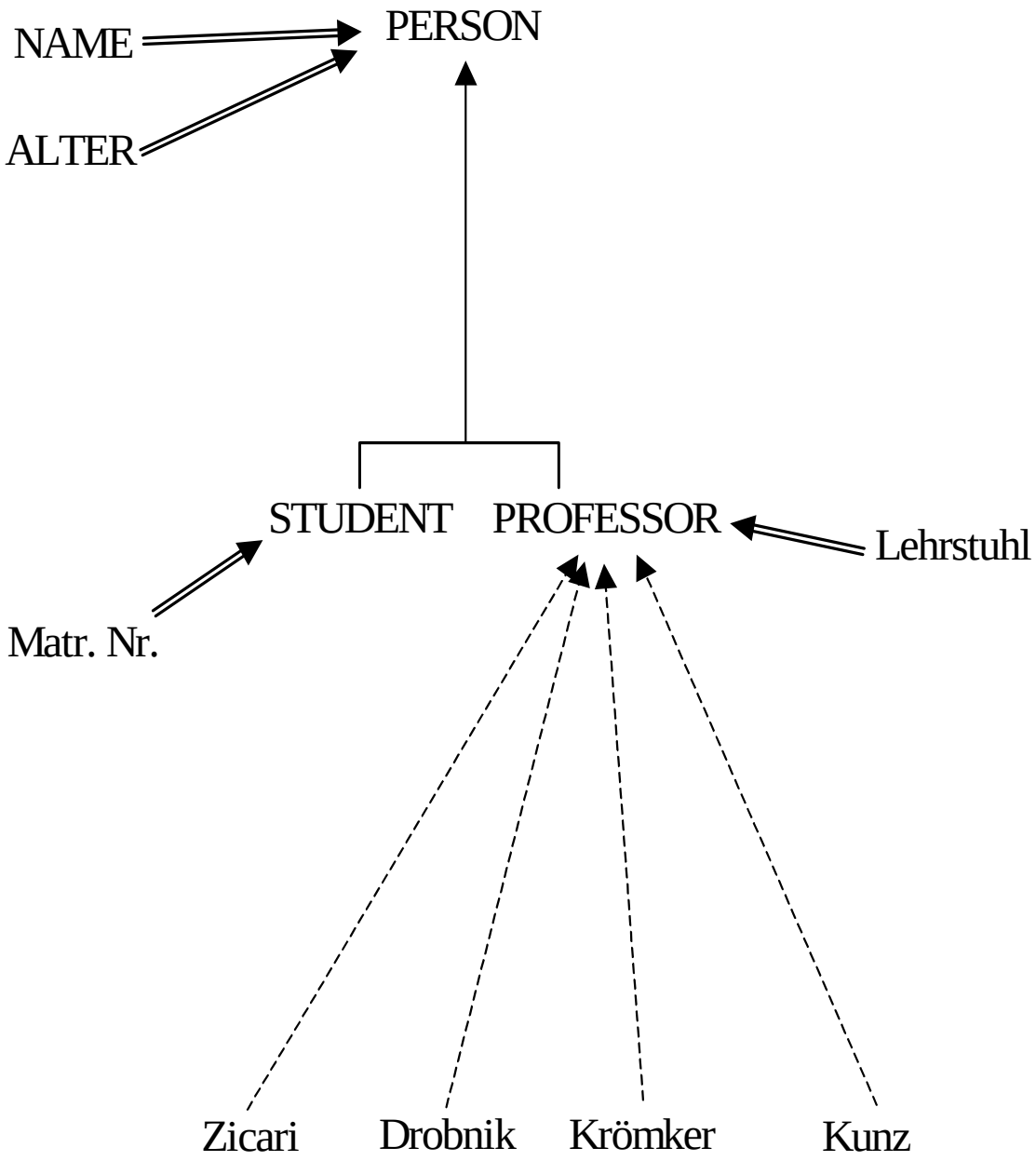
Die **Generalisierung** definiert eine **Untermengenbeziehung** zwischen Elementen von zwei (oder mehr) **Klassen**, indem Gemeinsamkeiten hervorgehoben und Unterschiede vernachlässigt werden.



Generalisierung besitzt die Eigenschaft der **Vererbung** (Inheritance):

Alle Abstraktionen einer generischen Klasse werden von den Subklassen geerbt.




NAME
ALTER
PERSON
STUDENT
PROFESSOR
Lehrstuhl
Matr. Nr.
Zicari
Drobnik
Krömker
Kunz

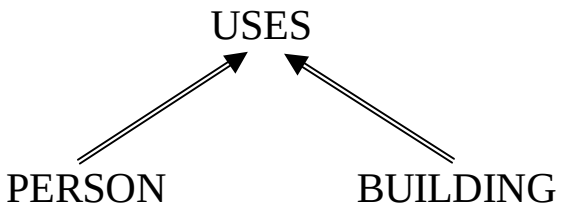
Eigenschaften der Abbildungen zwischen Klassen

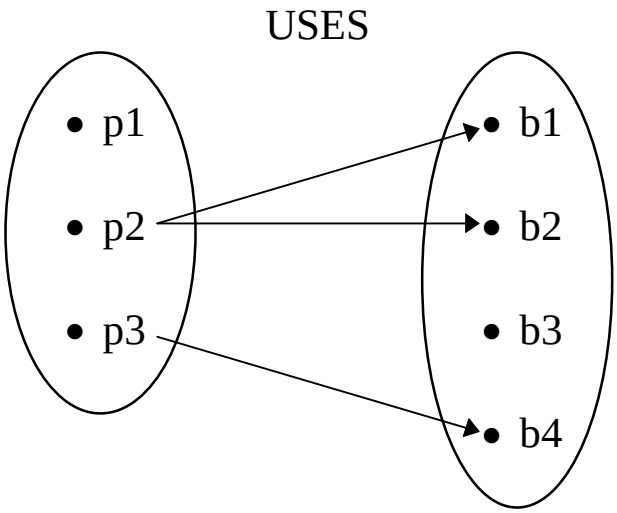
Aggregation und Generalisierung sind Abbildungen zwischen Klassen.

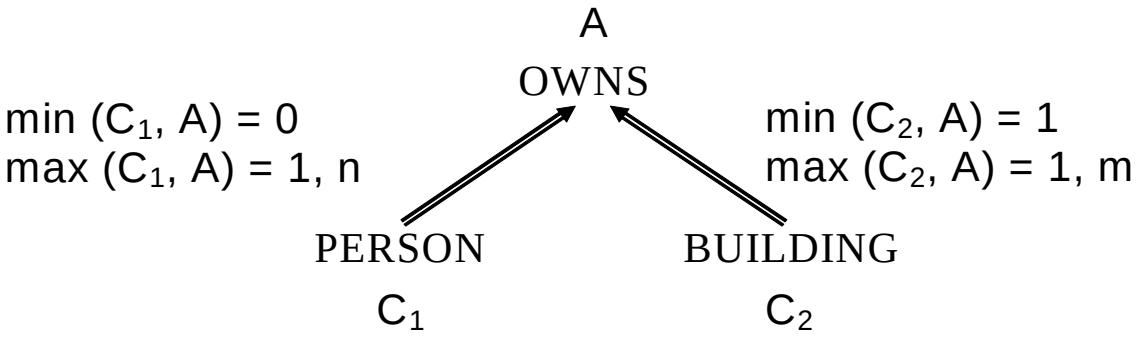
- Binäre Aggregationen
- N-näre Aggregationen
- Generalisierungen

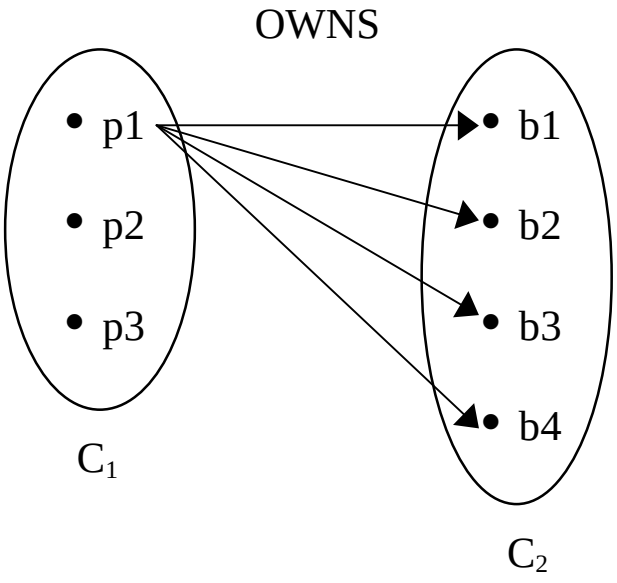
Binäre Aggregationen

Abbildungen zwischen 2 Klassen!






A
OWNS
min (C1, A) = 0
max (C1, A) = 1, n
min (C2, A) = 1
max (C2, A) = 1, m
PERSON
BUILDING
C1
C2


OWNS
p1
p2
p3
b1
b2
b3
b4
C1
C2

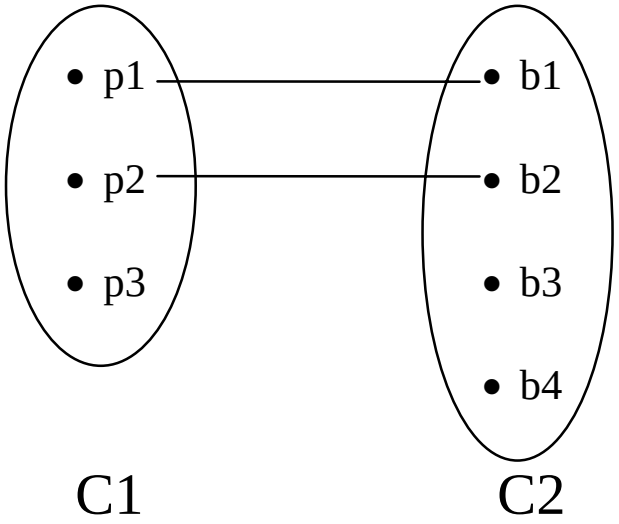
Kardinaltitäten

- Minimale Kardinalität (min-card)
  min-card(C,A): die kleinste Anzahl von Abbildungen A an denen jedes Element von C teilnehmen kann.
  - min-card(C,A) = 0 : optionale Teilnahme
  - min-card(C,A) = 1 : verbindliche Teilnahme

- Maximale Kardinalität (max-card)
  max-card(C,A): die maximale Anzahl von Abbildungen A an denen jedes Element von C teilnehmen kann.
  - max-card(C,A) = 1 : one-mapping
  - max-card(C,A) = m: many-mapping

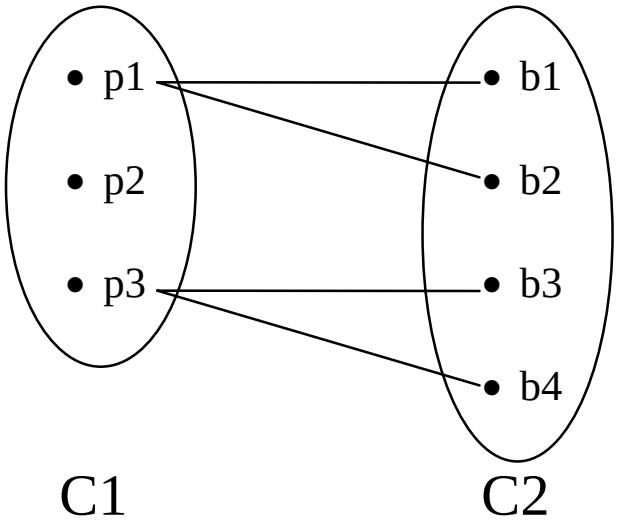
- one-to-one

  max-card(C1,A) = 1 und max-card(C2,A) = 1



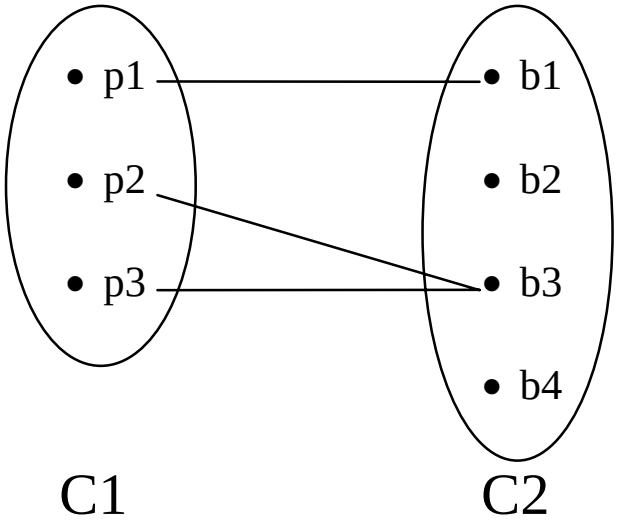
- one-to-many

  max-card(C1,A) = n und max-card(C2,A) = 1



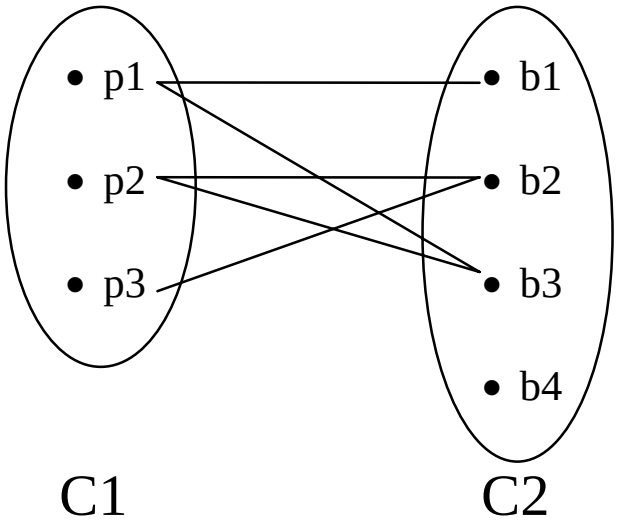
- many-to-one

  max-card(C1,A) = 1 und max-card(C2,A) = n



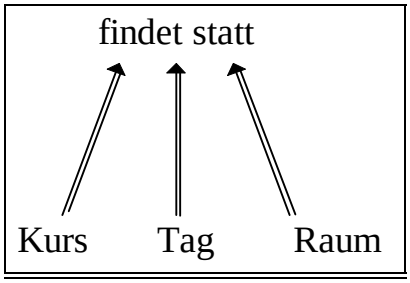
- many-to-many

  max-card(C1,A) = n und max-card(C2,A) = m



N-näre Aggregationen

Abbildungen zwischen **drei oder mehr** Klassen.



Kardinalitäten sind analog zu binären Aggregationen definiert.

Fall 1:

min_card (Kurs, findet_statt) = 1

max_card (Kurs, findet_statt) = 3

Jeder Kurs kann in einer Woche zwischen ein und dreimal stattfinden.

Fall 2:

min_card (Tag, findet_statt) = 0

max_card (Tag, findet_statt) = n

Jeden Tag können beliebig viele Kurse stattfinden.

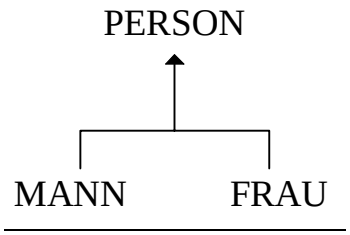
Fall 3:

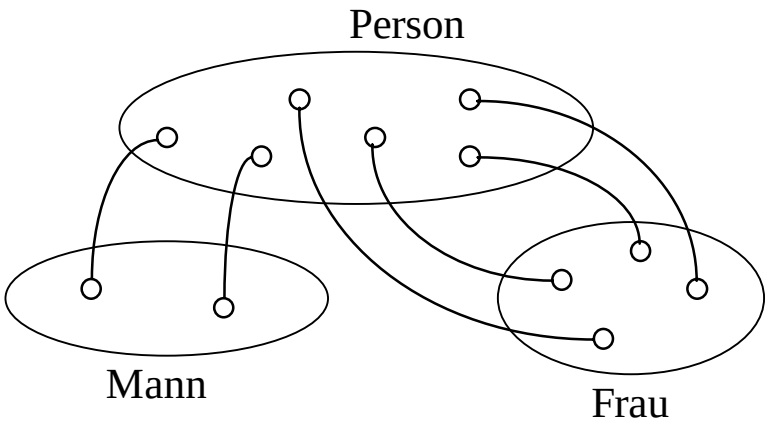
min_card (Raum, findet_statt) = 0

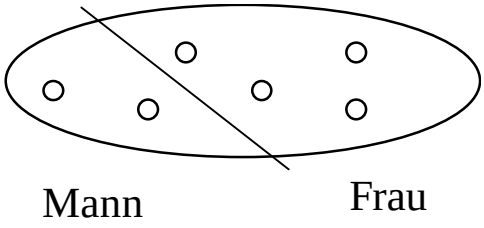
max_card (Raum, findet_statt) = 40

In Jedem Raum können in einer Woche höchstens 40 Kurse stattfinden.

Generalisierungen







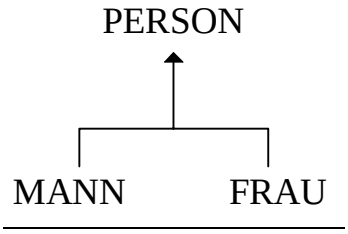
Überdeckung (coverage) von Generalisierungen

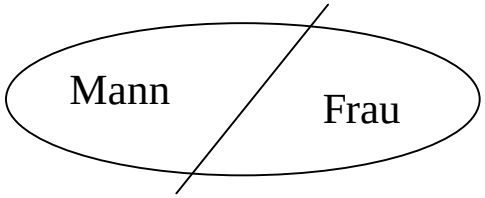
- Die Überdeckung einer Generalisierung ist **vollständig (total),** wenn jedes Element der generischen Klasse zu mindestens einer Unterklasse abgebildet wird.

- Die Überdeckung einer Generalisierung ist **partiell (partial),** wenn es Elemente der generischen Klasse gibt, die nicht zu einer Unterklasse abgebildet werden.

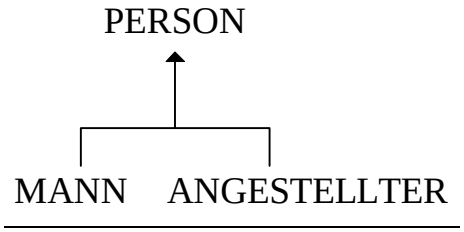
- Die Überdeckung einer Generalisierung ist **disjunkt (exclusive),** wenn jedes Element der generischen Klasse zu höchstens einer Unterklasse abgebildet wird.

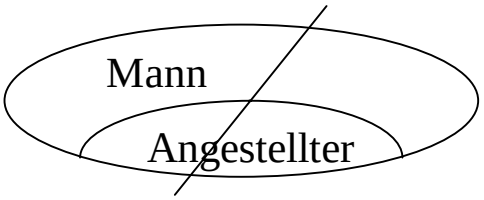
- Die Überdeckung einer Generalisierung ist **überlappend (overlapping),** wenn es Elemente der generischen Klasse gibt, die zu mehreren Unterklassen abgebildet werden.



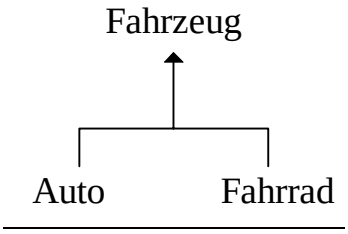


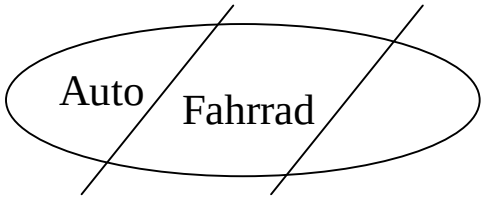
vollständig, disjunkt



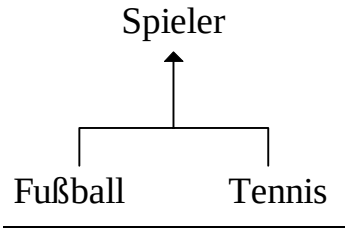


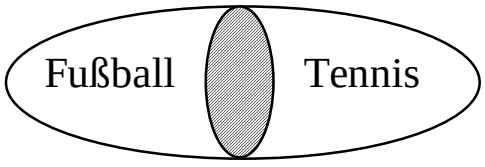
partiell, überlappend





partiell, disjunkt





vollständig, überlappend

Datenmodelle:

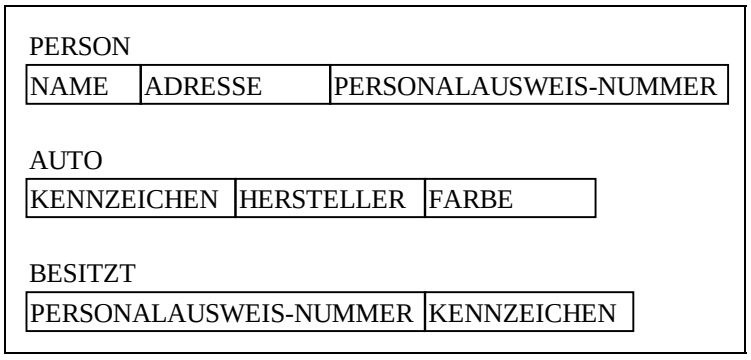
- *Datenmodell*

  Eine Sammlung von Konzepten zur Beschreibung einer Datenmenge mit den Operationen, die diese Datenmenge manipulieren können.

- *Schema*

  eine teilweise Repräsentation der Realität, die durch die Benutzung eines Datenmodells erzeugt wird. Es ist eine statische zeitinvariante Sammlung von sprachlichen oder graphischen Darstellungen, die die Struktur der Daten beschreiben.

Beispiel für ein Schema:



- *Instanz*

  Eine dynamische, zeitvariante Sammlung von Daten, die der durch das Schema definierten Datenstruktur angepasst sind.

PERSON

Willi Müller	Sackgasse 12, 61234 Westheim	387-6713-362
Vera Otto	Parkalle 4, 03241 Norddorf	342-6576-777
Klaus Umweg	Bahnhofstr. 423, 87876 Bad Appel	639-6354-753

AUTO

WH - AD 456	Opel	Weiß
NDF - XX 604	Porsche	Grün
BDA - EF 2	Ford	Gelb
BDA - N 815	VW	Rot

BESITZT

387-6713-362	WH - AD 456
342-6576-777	NDF - XX 604
639-6354-753	BDA - EF 2
639-6354-753	BDA - N 815

Eigenschaften des konzeptuellen Modells

- *Ausdruckskraft* (Expressiveness). Konzeptuelle Modelle unterscheiden sich in der Zahl der unterstützten Modellierungsstrukturen. Generell gilt, dass die Ausdruckskraft eines Modells mit der Zahl der unterstützten Strukturen steigt.

- *Einfachheit* (Simplicity). Ein konzeptuelles Modell muss einfach zu lesen sein. Einfachheit und Ausdruckskraft sind widersprüchliche Ziele.

- *Minimalität* (Minimality). Diese Eigenschaft ist gegeben, wenn kein Konzept durch Komposition anderer Konzepte dargestellt werden kann.

- *Formalität* (Formality). Alle Konzepte eines Modells haben eine einheitliche, präzise und wohldefinierte Interpretation. Formale Konzepte können mathematisch manipuliert werden.