



**Universität  
Zürich** UZH

**Institut für Informatik**

# Informatik II: Modellierung

## Prof. Dr. Martin Glinz

### Kapitel 3

# Datenmodellierung

# Inhalt

---

- 3.1 Grundlagen und Motivation
- 3.2 Entity-Relationship-Modelle
- 3.3 Notation von Datenmodellen
- 3.4 Kardinalitäten
- 3.5 Methodik der Datenmodellierung
- 3.6 Verwendete modelltheoretische Konstrukte

## 3.1 Grundlagen und Motivation

---

Wie kommt Sonja Müller in den Computer



und wie kommen die Fotobücher, die sie im Web bestellt hat, zu ihr?

Welche Daten muss der Buchversand über Sonja Müller und ihre Bestellungen kennen und speichern?



# Grundlagen und Motivation – 2

---

- Betriebliche Daten sind in der Regel
  - langlebig
  - stabil
  - wertvoll
- Ein **Modell der Daten** beschreibt einen zentralen Aspekt eines betrieblichen Informationssystems

# Mögliche Einsatzgebiete für Datenmodelle

---

- **Analyse und Spezifikation** der Daten eines **Problembereichs**
  - Datenmodell **im engeren Sinn**
  - **Abbild** des **Problembereichs**
  - Meist **Vorbild** für Organisation der Daten in einer **Datenbank**
- **Logische Organisation** der Daten einer **Datenbank** (konzeptionelles Schema in der Datenbankterminologie)
- **Physische Organisation** der Daten einer **Datenbank** (internes Schema in der Datenbankterminologie)
- **Grundkonzept** für die **Strukturierung von Daten** in einer **Datenbank** (Datenmodell in der Datenbankterminologie)
- Beschreibung der **Datenarchitektur** eines Systems (Dateien, Datenbanken, Datenverteilung)

# Datenmodelle im engeren Sinn

---

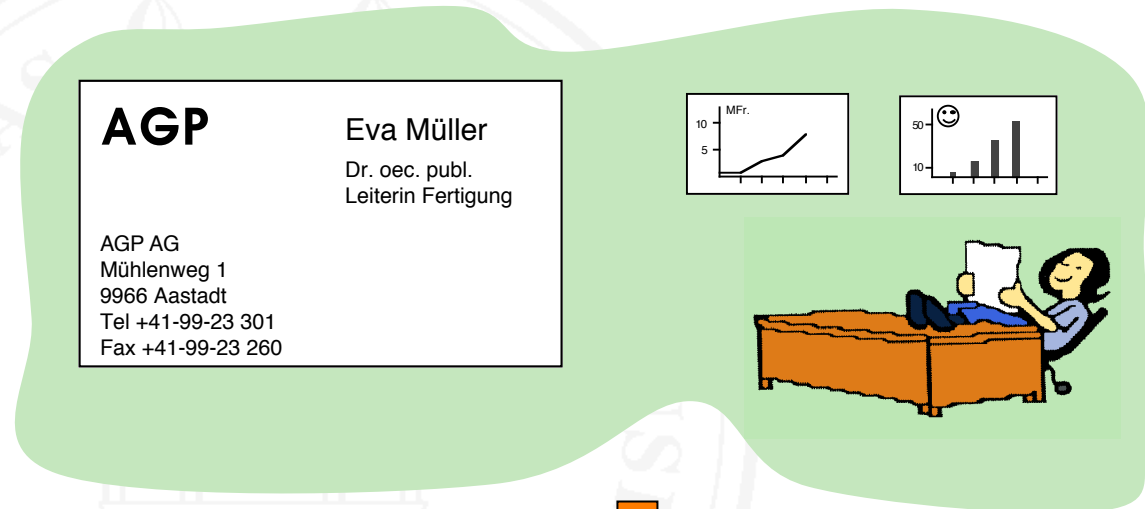
- **Analyse** der Daten eines Problembereichs (Ausschnitt aus der „Realität“)
- **Spezifikation** der Daten, die ein System über diesen Problembereich kennen soll
  - **Welchen Ausschnitt** aus dem Problembereich muss ein geplantes System **kennen** (damit es seine Aufgaben erfüllen kann)?
  - **Wie** soll dieser Ausschnitt **auf Daten** des Systems **abgebildet** werden?
  - **Wo** liegen die **Grenzen** (welche Daten gehören zum System und welche nicht)?
  - Dient ferner als **Vorgabe** für den Entwurf von **Datenbanken**

Klassischer Vertreter der Datenmodellierung im engeren Sinn: **Entity-Relationship-Modell** oder Gegenstands-Beziehungs-Modell (Chen 1976)

## 3.2 Entity-Relationship-Modelle

### Das Abbildungsprinzip

Die interessierenden  
Elemente eines  
Problembereichs ...



... werden **abgebildet** auf

○ **Gegenstände**

○ **Beziehungen**

○ **Attribute**

Name: Müller  
Vorname: Eva  
Geschlecht: weiblich  
...  
P21502

leitet

Name: Fertigung  
Umsatz: 8 MCHF  
Beschäftigte: 52  
...  
CH-PROD

# Entity-Relationship-Modelle: Terminologie – 1

---

- **Gegenstand** (Objekt, Entität, object, entity) – ein individuell erkennbares, von anderen Gegenständen eindeutig unterscheidbares **Element des** untersuchten **Problembereichs**
- **Beziehung** (relationship) – **verknüpft** einen Gegenstand mit einem anderen Gegenstand
- **Attribut** (Eigenschaft, attribute) – ordnet einem Gegenstand einen Wert zu, welcher diesen näher **charakterisiert**



# Entity-Relationship-Modelle: Klassifizierung

---

- Meist interessieren nicht die konkreten Gegenstände und Beziehungen,
- ... sondern die zu Grunde liegenden **Konzepte**
- ⇒ **Klassifizieren** der Daten nach **gemeinsamen Merkmalsarten**,
- ⇒ **Abstrahieren** von den individuellen **Merkmalsausprägungen**

|   |   |  |
|---|---|--|
| Eva Müller, Hans Peter Maier,<br>Anne Weber, ...  | ⇒ | Mitarbeiter(in) mit Attributen<br>Name, Vorname, Geschlecht, ... |
| Fertigung, Einkauf,<br>Finanzen, ...  | ⇒ | Abteilung mit Attributen<br>Name, Umsatz, Beschäftigte, ...      |
| Eva Müller leitet Fertigung<br>Hans Peter Maier leitet Finanzen<br>Anne Weber leitet Einkauf, ... | ⇒ | Mitarbeiter(in)<br>leitet<br>Abteilung                           |
| ...   |   |  |

# Entity-Relationship-Modelle: Terminologie – 2

---

**Gegenstandstyp** (Klasse, Entitätstyp, object type, entity type) – eine eindeutig benannte Einheit, welche eine Menge gleichartiger Gegenstände beschreibt

- Ein Gegenstandstyp wird charakterisiert durch die **Attribute**, welche die Gegenstände dieses Gegenstandstyps haben
- die **Beziehungen**, die von Gegenständen dieses Gegenstandstyps zu Gegenständen anderer Typen bestehen können oder müssen
- Der Beziehungszusammenhang zwischen zwei Gegenstandstypen wird manchmal auch als **Beziehungstyp** bezeichnet, weil er die möglichen Beziehungen zwischen Gegenständen beschreibt
- Auch der Name **Assoziation (association)** ist gebräuchlich
- Weiterführende Konzepte wie schwache Gegenstandstypen und Generalisierung werden hier nicht behandelt

# Pragmatik der Datenmodellierung

---

- Die Klassifizierung hängt vom **Umfeld** und der **Modellierungsabsicht** ab

Beispiele:

(1) Eva Müller wird klassifiziert als

- **Mitarbeiterin** in einem Personalinformationssystem
- **Patientin** in einem Spitalinformationssystem

(2) Die Firma AGP hat mehrere Niederlassungen in verschiedenen Orten. Ort kann modelliert werden als

- **Attribut** von Niederlassung
- **Gegenstandstyp** mit Beziehung zu Niederlassung

(3) In einem Reisebüro werden Reisen gebucht. Die Buchung kann modelliert werden als

- **Beziehung** zwischen Reise und Kunde
- als **Gegenstandstyp** mit Beziehungen zu Reise und Kunde

# Regeln für die Zuordnung und Abgrenzung

---

- **Attribute** modellieren Existenzabhängigkeiten: Der **Wert** jedes Attributs ist **existenzabhängig** von dem Gegenstand oder dem Attributwert, den das Attribut charakterisiert
- **Beziehungen** gibt es **nur zwischen Gegenständen**, nicht zwischen Attributen oder zwischen Gegenstand und Attribut
- Eine **Beziehung**, die **mehrere Attribute** oder gar **eigene Beziehungen** hat, wird meist als **Gegenstandstyp** modelliert

## Aufgabe 3.1

---

Mit welcher Pragmatik wird eine Person modelliert

- a) In einem Informationssystem für die Einwohnerkontrolle einer Gemeinde
- b) Im Personalinformationssystem eines Transportunternehmens
- c) In einem Spital-Informationssystem?

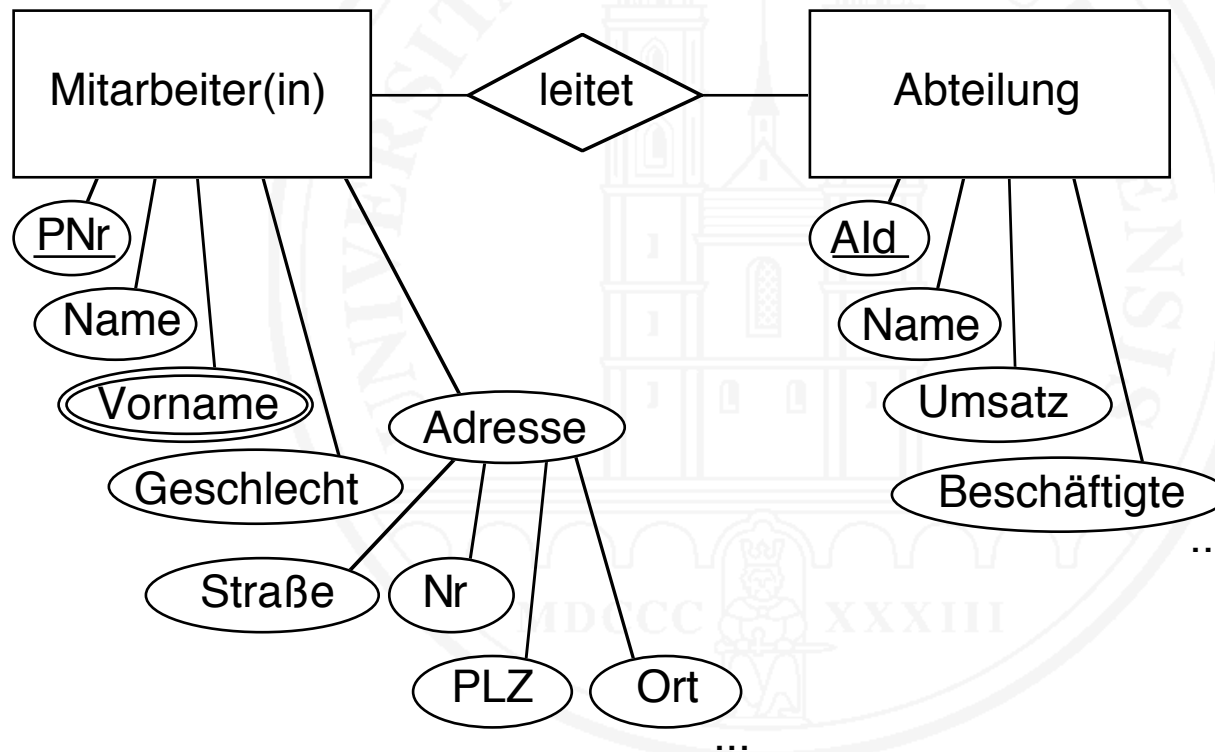
Nennen Sie jeweils einige typische Attribute und Beziehungen, die zu modellieren bzw. wegzulassen sind.

## 3.3 Notation von Datenmodellen

---

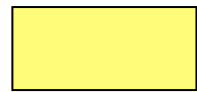
Klassische Notation für Entity-Relationship Modelle:

Das **Entity-Relationship Diagramm**

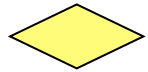


# Notation von Datenmodellen – 2

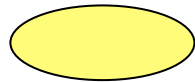
---



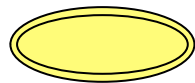
Gegenstandstyp



Beziehung



Attribut



mehrwertiges Attribut



identifizierendes Attribut (Schlüssel)

In großen Modellen:

- Grafische Darstellung der Attribute beansprucht zu viel Platz
- ⇒ **Separate Darstellung mit Text**

# Notation der Attribute

---

- Attribute werden häufig **nicht graphisch**, sondern mit **Text** notiert

Beispiel:

GEGENSTANDSTYP Abteilung

Ald: XX-XXXX

(6 Zeichen mit Trennstrich)

Name: Zeichenkette

Umsatz: MCHF

Beschäftigte: 1..999

- Dabei ist die Angabe von **Wertebereichen (domains)** für die Attributwerte möglich



# Alternative Notationen

---

- Es gibt verschiedene andere Notationen für Entity-Relationship-Modelle
  - Häufig werden Beziehungen statt mit Rauten mit **Linien** dargestellt
  - Allerdings sind so nur **zweistellige** Beziehungen darstellbar
  - Attribute werden in dieser Notation separat mit Text dargestellt

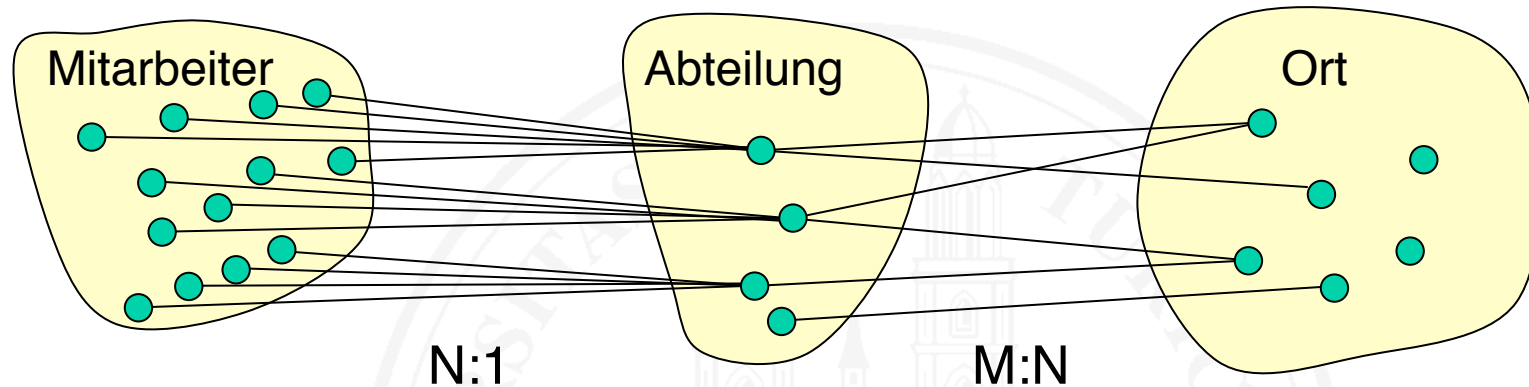


## 3.4 Kardinalitäten

---

- **Kardinalitäten (cardinalities)** sind
  - Beschränkungen der Anzahl der möglichen Beziehungen, die ein Gegenstand zu anderen Gegenständen **höchstens** haben darf oder **mindestens** haben muss
  - Angaben zur **Anzahl der erlaubten Werte** zu einem **Attribut**
- Die **Maximalkardinalitäten** für Beziehungen geben im Wesentlichen an, ob es sich um 1:1, 1:N oder M:N Beziehungen handelt
- Die **Minimalkardinalitäten** spezifizieren zusätzlich, ob die Existenz einer Beziehung zu einem gegebenen Gegenstand optional oder zwingend ist
- **Attributkardinalitäten** dienen zur Beschreibung von optionalen und von mehrwertigen Attributen

# Beispiel: Maximal- und Minimalcardinalitäten



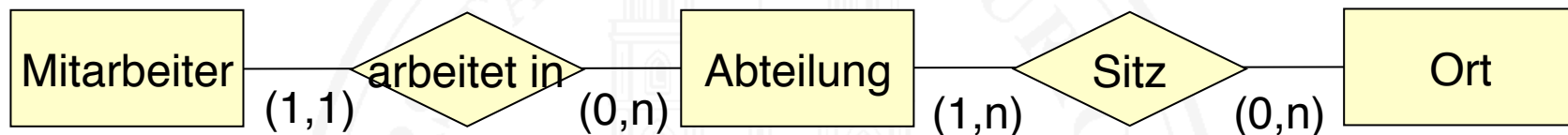
- Das Bild zeigt die Gegenstandstypen **Mitarbeiter**, **Abteilung** und **Ort** mit einigen konkreten Gegenständen und ihren Beziehungen
- Es bestehen folgende **Kardinalitätsbeschränkungen**

|  | min | max |
|--|-----|-----|
| ● Jeder Mitarbeiter arbeitet in genau einer Abteilung: | 1   | 1   |
| ● Abteilungen beschäftigen 0 bis n Mitarbeiter:        | 0   | n   |
| ● Jede Abteilung sitzt in mindestens einem Ort:        | 1   | n   |
| ● In einem Ort können 0 bis n Abteilungen sitzen:      | 0   | n   |

# Notation von Kardinalitäten

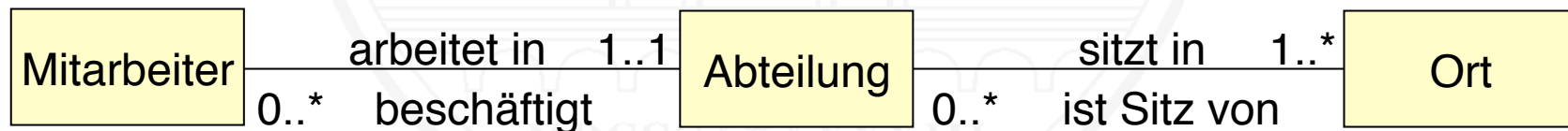
Verschiedene Notationskonventionen:

- Klassisch (u.a. ElMasri und Navathe)



Notation spiegelbildlich zu den Mengenverhältnissen!

- UML (Unified Modeling Language)



- Es gibt eine Vielzahl weiterer Notationen (hier nicht behandelt)

# Beispiel: Attributkardinalitäten

---

GEGENSTANDSTYP Mitarbeiter

PNr: Zahl

Name: Zeichenkette

Vorname [1..5]: Zeichenkette

Geschlecht: [ 'm' | 'w' ]

...

MobilTfPrivat [0..1]: { '0'..'9' | '+' | '-' | '(' | ')' }

...

- **Mindestens einer, höchstens fünf** Vornamen
- Die Angabe der privaten Mobiltelefonnummer ist **optional**

## 3.5 Methodik der Datenmodellierung

---

- **Informationsquellen:** Aufgabenbeschreibungen, Gespräche oder Gesprächsnotizen, Formulare, Beschreibungen vorhandener oder verlangter Datenbestände, Glossare (Begriffslexika), Szenarien, Anwendungsfälle
- **Modellbildung**
  - **Gewinnung:** Informationsquellen erschließen und auswerten
  - **Beschreibung:**
    - Gewonnene Information **klassifizieren, ordnen** und als Datenmodell-Fragmente **darstellen**
    - Fragmente abgleichen / zusammenbauen / vervollständigen
  - Rückkopplung durch ständiges **Validieren/Reflektieren** von Modellfragmenten
- **Mögliche Verfahren:** Objektanalyse, Ereignis-Reaktions-Analyse

# Objektanalyse

---

Prinzip: **Textanalyse** des vorhandenen Materials

- Grammatisches **Subjekt**, grammatische **Objekte** → Kandidaten für Gegenstände, Gegenstandstypen, Attributwerte, Attribute oder Wertebereiche
- **Verben** beschreiben Zusammenhänge oder Handlungen:
  - **Zusammenhänge** → Beziehungen, Attribute
  - **Handlungen** → Funktionalität, Verhalten (wird in Datenmodellen nicht modelliert)
- **Adjektive** präzisieren Aussagen, **schränken** sie **ein** oder sind Kandidaten für **Attributwerte**
- Fragmente **klassifizieren**, **ordnen**, **vervollständigen**

# Objektanalyse – 2

---

- Liefert nur bei **brauchbarem Ausgangsmaterial** brauchbare Ergebnisse
- Das Klassifizieren, Ordnen und Vervollständigen ist die **Hauptarbeit**
- Abgrenzung von Gegenständen/Gegenstandstypen gegen Attribute/Werte:
  - Gegenstände haben eine **Identität**
  - Attributwerte sind **Daten** ohne eigene Identität
  - Attribute von Attributen werden in der Regel **vermieden**: In solchen Situationen Gegenstandstypen und Beziehungen modellieren



# Beispiel zur Objektanalyse

---

Zu erstellen sei ein **Informationssystem für Reisebüros**

Im Glossar (Begriffslexikon) für dieses System findet sich folgende Information:

...

**Buchung** – Kauf eines ↑Arrangements (provisorisch oder fest) durch einen ↑Kunden. Gibt an, welches Arrangement von welchem Kunden gebucht wurde. Enthält ferner Buchungsdatum, Preis, ↑Buchungsstatus und Kurzzeichen des ↑Sachbearbeiters.

...

# Vorgehen

---

**Buchung** - Kauf eines ↑Arrangements (provisorisch oder fest) durch einen ↑Kunden. Gibt an, welches Arrangement von welchem Kunden gebucht wurde. Enthält ferner Buchungsdatum, Preis, ↑Buchungsstatus und Kurzzeichen des Sachbearbeiters.

Gegenstandstypen:

Buchung, ~~Kauf~~  
Arrangement  
Kunde

Attribute:

~~Buchungs~~Datum  
Preis  
~~Buchungs~~Status  
~~Kurzzeichen des~~  
Sachbearbeiters  
Verbindlichkeit

Attributwerte:

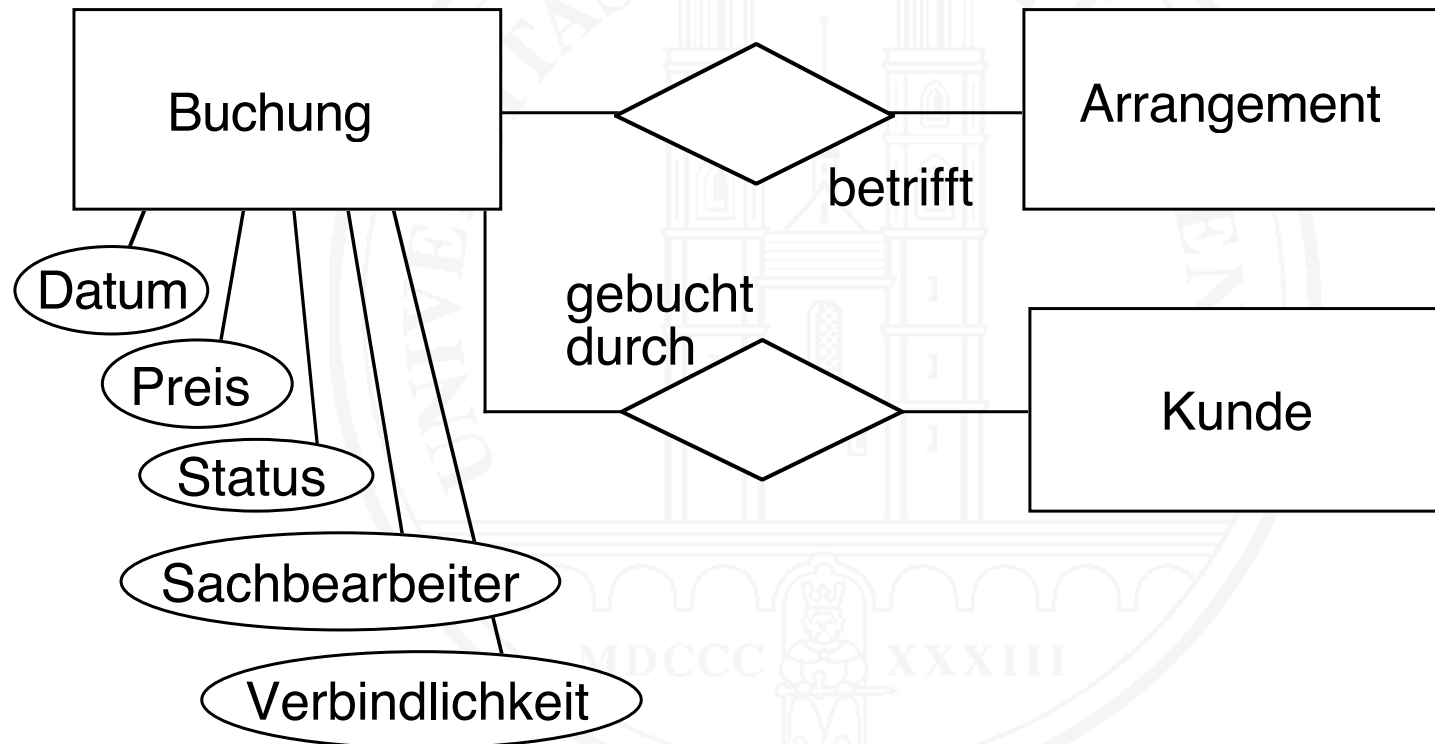
provisorisch  
fest

gebucht wurde: Beziehungen Buchung – Arrangement, Buchung – Kunde  
Enthält: Zuordnung von Buchungsdatum, etc. als Attribute von Buchung

# Beispiel zur Objektanalyse – 2

---

Resultierendes Modellfragment:



# Ereignis-Reaktions-Analyse

---

Prinzip: Analyse von **Systeminteraktionen** mit **Rückschluss auf die benötigten Daten**

- Für jedes externe, auf das Informationssystem einwirkende **Ereignis** die erwartete **Reaktion** des Informationssystems analysieren
  - Welche nicht mit dem Ereignis eintreffenden Daten werden für die Reaktion benötigt?
  - Welche erzeugten Ergebnisse werden nicht sofort, sondern in späteren Schritten benötigt?
- ⇒ Beide Datenarten sind Bestandteile des Datenmodells

# Ereignis-Reaktionsanalyse – 2

---

Ereignisse und erwartete Reaktionen finden sich in

- Szenarien, welche die Interaktion zwischen Benutzern und System beschreiben
- Beschreibungen von Prozessen
  - Arbeitsprozesse bei Informationssystemen
  - Technischen Prozessen bei eingebetteten Systemen
- Beschreibungen von Funktionen und Abläufen

# Beispiel zur Ereignis-Reaktions-Analyse

---

Zu erstellen sei ein **Informationssystem für Reisebüros**

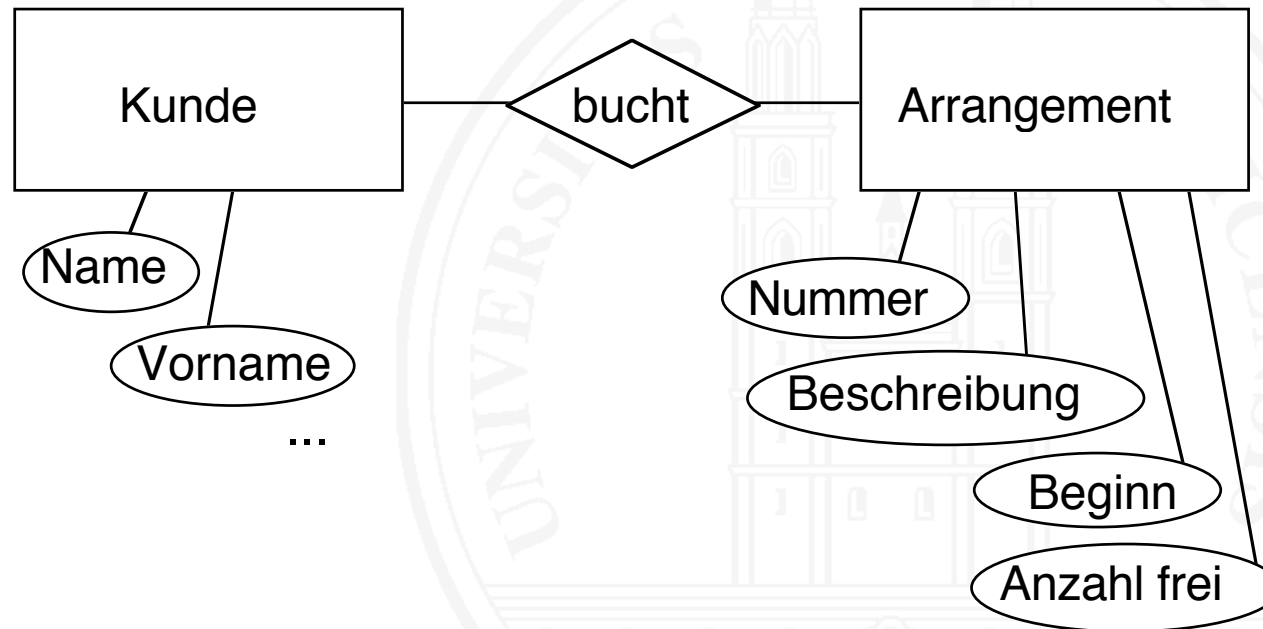
In einem Szenario zu diesem System findet sich folgende Information:

Eine Person will ein Arrangement buchen. Zunächst wird geprüft, ob das gewünschte Arrangement existiert und ob es noch ausreichend freie Plätze hat. Falls ja, so wird das Arrangement gebucht. Hierzu erfasst der Berater die persönlichen Daten des Kunden, sofern dieser nicht bereits in der Kundendatei ist.

# Beispiel zur Ereignis-Reaktions-Analyse – 2

---

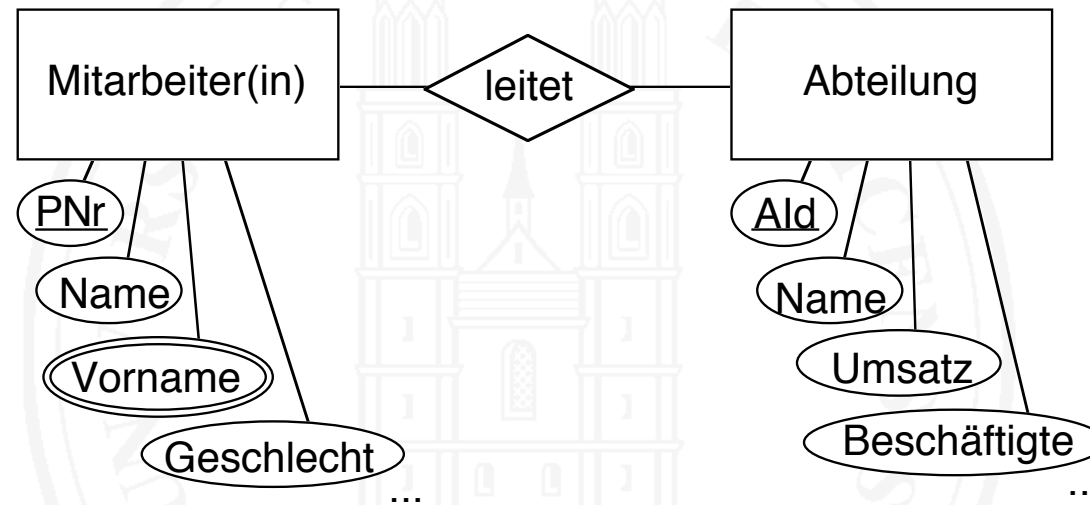
Resultierendes Modellfragment:



**Aufgabe 3.2: Vergleichen Sie die Modellfragmente dieses und des vorausgegangenen Beispiels. Was fällt Ihnen auf?**

## Aufgabe 3.3

Nehmen Sie an, Sie haben MitarbeiterInnen und Abteilungen wie folgt modelliert:



Sie erhalten nun folgende neuen Informationen: „MitarbeiterInnen arbeiten in Projekten mit. Jedes Projekt hat einen Namen. Ferner müssen die Daten von Projektbeginn und Projektende gespeichert werden.“

Analysieren Sie diese zusätzlichen Informationen und stellen Sie das Ergebnis dar, indem Sie das bisherige Modell entsprechend modifizieren.



## Aufgabe 3.4

In der AGP AG gibt es unter anderem folgende Geschäftsregeln:

- Jede Abteilung hat genau eine Leiterin oder einen Leiter.
- Niemand leitet mehr als eine Abteilung.
- Abteilungsleiter müssen mindestens 21 Jahre alt sein.
- Mitarbeitende können in bis zu 6 Projekten beschäftigt sein. Es ist zulässig, dass eine Mitarbeiterin oder ein Mitarbeiter in keinem Projekt involviert ist.
- Auf Projekten arbeiten in der Regel mehrere Personen. Werden Projekte initiiert, kann anfänglich keine Person zugeordnet sein.
- Das Datum des Projektendes muss immer nach dem Datum des Projektbeginns liegen.

a) Modellieren Sie diese Regeln mit Hilfe von Kardinalitäten.

b) Welche Regeln sind so nicht modellierbar?

## 3.6 Verwendete modelltheoretische Konzepte

---

| <b>Datenmodell-Element</b>                  | <b>Modelltheoretisches Konzept</b>          |
|---|---|
| Gegenstand, Gegenstandstyp                  | Individuum                                  |
| Attribut                                    | Attribut                                    |
| Beziehung                                   | Attribut                                    |
| Problembereich                              | Original                                    |
| Abbildung des Problembereichs               | Abbildungsmerkmal                           |
| Abbildung nur der interessierenden Elemente | pragmatisches Merkmal<br>Verkürzungsmerkmal |
| Klassifizierung                             | Verkürzungsmerkmal                          |
| Nicht-Eindeutigkeit                         | pragmatisches Merkmal                       |
| Rechteck, Raute, ...                        | Notation                                    |

# Literatur

---

Batini, C. S. Ceri, S. Navathe (1992). *Conceptual Database Design: An Entity-Relationship-Approach*. Redwood City: Benjamin/Cummings.

Boman, M., J.A. Bubenko, P. Johannesson, B. Wangler (1997). *Conceptual Modelling*. London: Prentice Hall.

Chen, P.P. (1976). The Entity-Relationship Model - Toward a Unified View of Data. *ACM Transactions on Database Systems* 1. 9-36.

Elmasri, R., S.B. Navathe (2003). *Fundamentals of Database Systems*, 4th edition. New York : Pearson/ Addison Wesley.

Oestereich, B. (2004). *Objektorientierte Softwareentwicklung: Analyse und Design mit der UML 2.0*, 6. Auflage. München: Oldenbourg.

Object Management Group (2010). *Unified Modeling Language: Superstructure*, version 2.3. OMG document formal/2010-05-05. <http://www.omg.org/spec/UML/2.3/Superstructure/PDF>