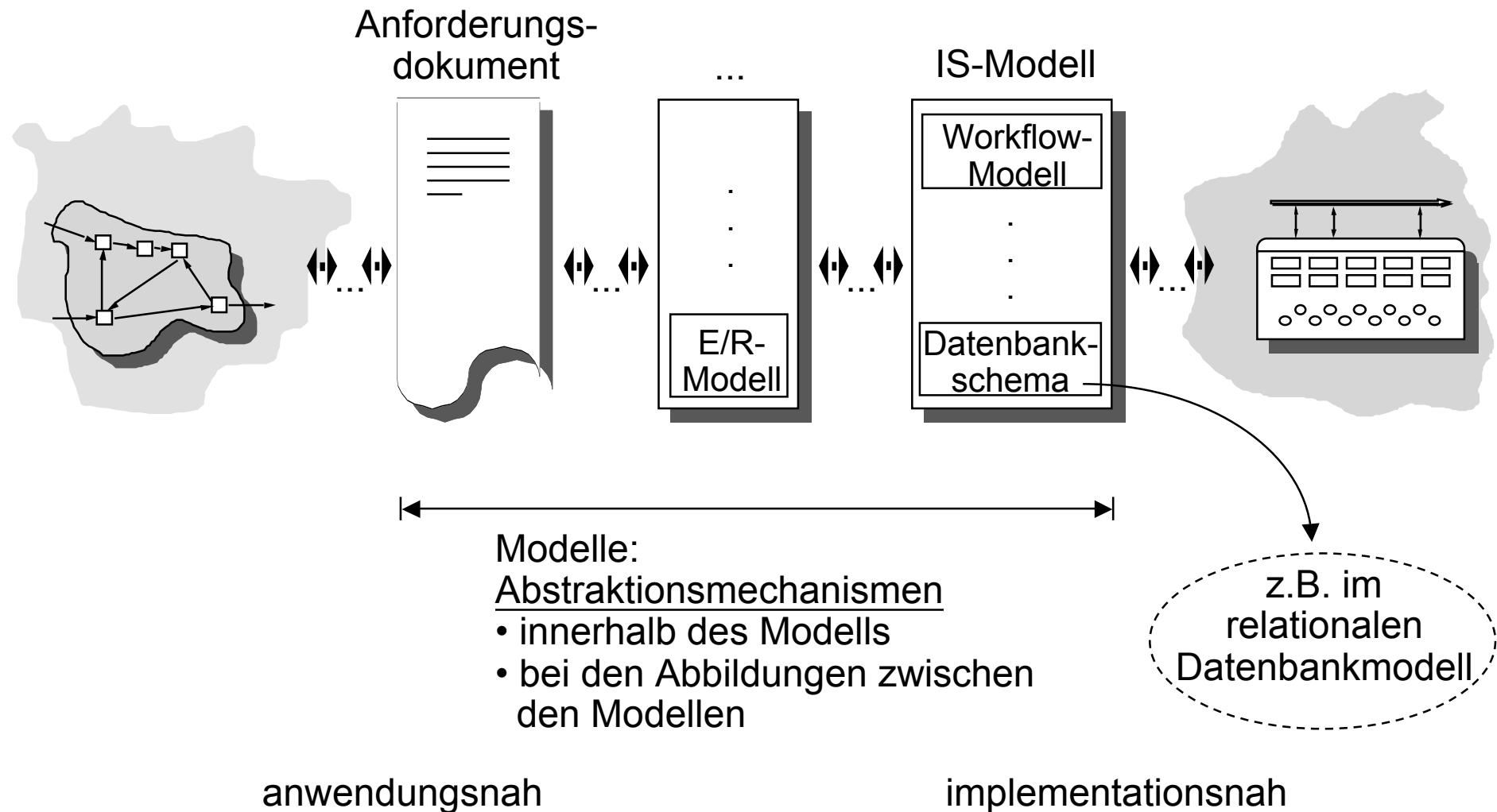


# Modelle und Abstraktion



# Repräsentation "atomarer" Information <sup>(1)</sup>

---

❑ Es wird unterschieden in

- Basiswerte (Literale)
  - Zahlen, Zeichen, Zeichenketten
    - 3.1415; 'Z'; "Otto"
- Zusammengesetzte Werte: heterogene Strukturen (Tupel, Rekords, Strukturen) und homogene Strukturen (Arrays, Listen, Mengen, Multimengen)
  - Tupel: ("Peter", 31) oder record age = 31 name = "Peter" end
  - Mengen: {1, 2, 3, 4}
  - Arrays: [1, 3, 5]
  - Listen <1, 3, 4, 77>

❑ Je nach Wertetyp benötigte Operationen werden vorausgesetzt

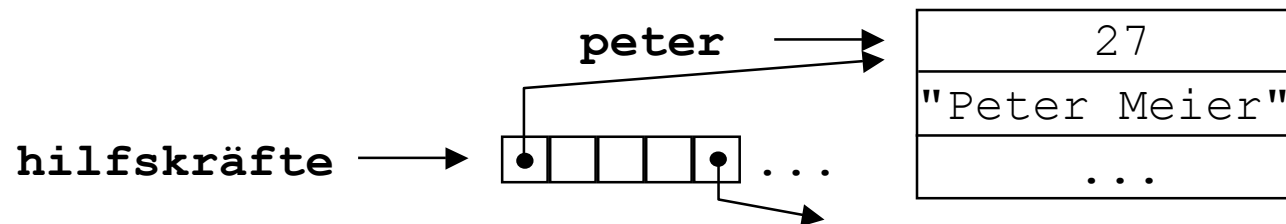
❑ Basiswerte und zusammengesetzte Werte nennen wir auch "Objekte"

# Referenzierung zusammengesetzter Werte (2)

---

- ❑ Objektidentität (object identity, OID) bleibt unabhängig vom Zustand erhalten
- ❑ Objekte können mehrfach über die OID referenziert werden (sharing)
- ❑ Der Zustand kann durch destruktive Zuweisung verändert werden.

```
peter.name := "Peter Meier"
```



- ❑ Änderungen des Objektzustands werden unmittelbar auf allen Pfaden sichtbar, z.B. die Änderung des Studentennamen (⇒ Referenzsemantik)

# Datenabstraktionskonzepte z. Informationsstrukturierung

---

**In fast allen Datenbankmodellen findet man Konstrukte für die folgenden Abstraktionskonzepte:**

- ☐ Klassifikation und Instantiierung
- ☐ Generalisierung und Spezialisierung
- ☐ Assoziation
- ☐ Aggregation
- ☐ Identifikation und Schlüssel

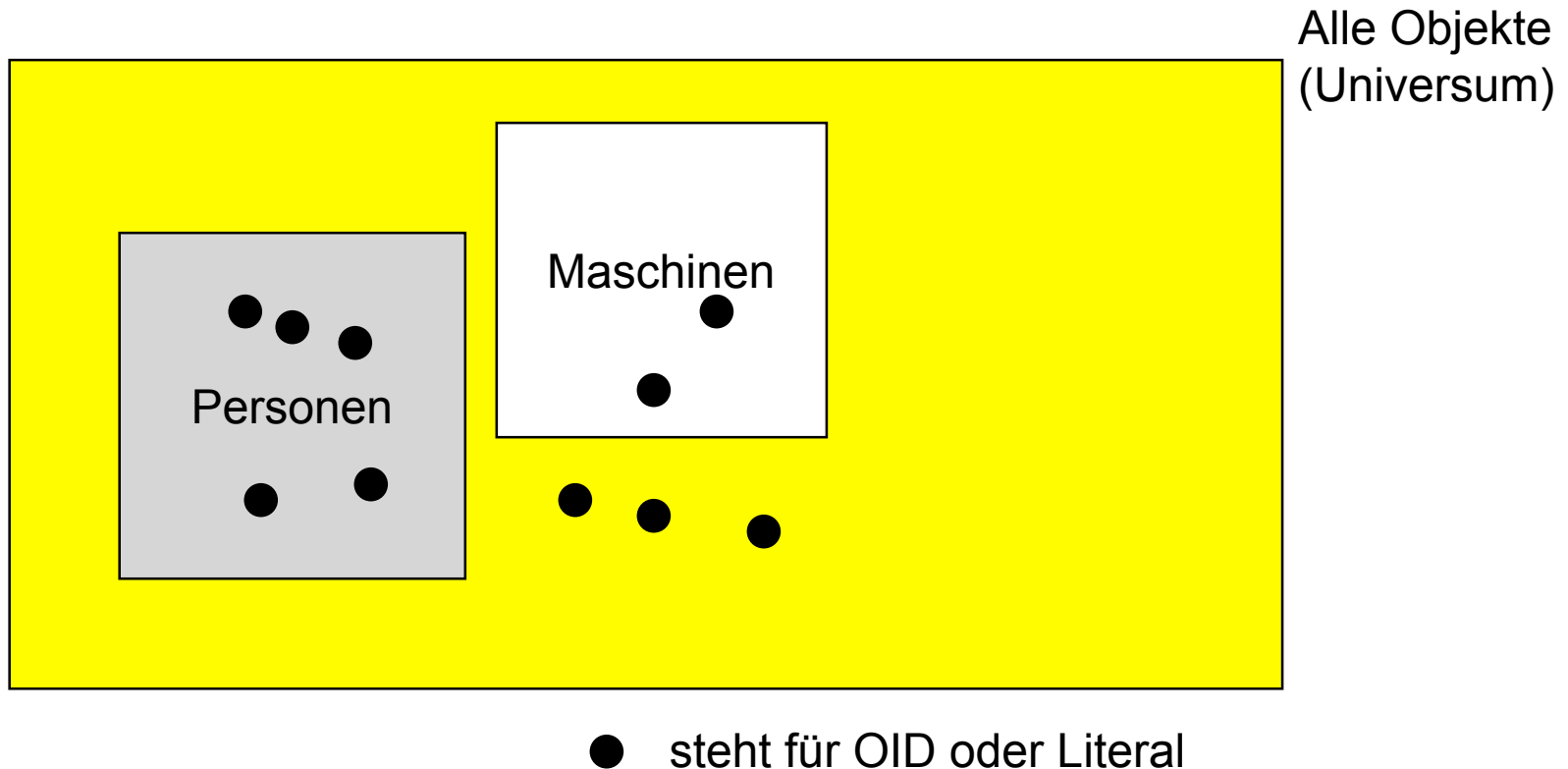
In späteren Kapiteln werden diese Konstrukte beschrieben. Nachfolgend werden die Abstraktionskonzepte anhand einer populären graphischen Notation erklärt.

**Entity-Relationship-Diagramme** wurden von P.P.S. Chen vorgeschlagen (vgl. P.P.S. Chen. "The Entity Relationship Model - Toward a Unified View of Data. In: ACM Transactions on Database Systems, Vol. 1, No. 1, März 1976, S. 9 ff.) und mehrfach erweitert (➡ extended E/R diagram, EE/R Modell).

# Klassifikation und Instantiierung <sup>(1)</sup>

---

- ❑ Objekte mit ähnlichen Eigenschaften können zu Klassen (Mengen) zusammengefaßt werden.
- ❑ Jedes Objekt ist "Instanz" einer oder mehrerer Klassen.
- ❑ Extension (Menge aller Instanzen einer Klasse)



# Grundlegende Elemente von ER-Diagrammen

---

❑ Objekttyp  (auch Entitätstyp genannt)

❑ Werttyp  (für Basisdatentypen)

❑ Beziehungstyp  (s.u.)

❑ Die Elemente von ER-Diagrammen bilden einen bipartiten Graphen:



❑ Verbindungen zwischen Symbolen der gleichen Typen sind nicht erlaubt.

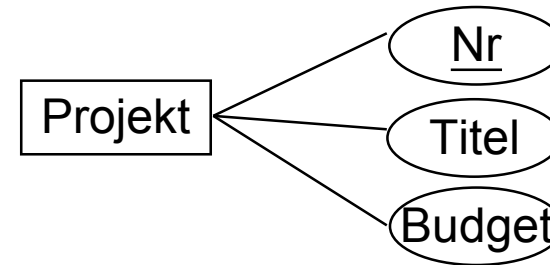
# Objekte/Entitäten und Attribute

---

## Beispiele:

❑ Ein Projekt wird beschrieben durch

- eine Nummer
- einen Titel
- das Budget



❑ Mathematische Bedeutung: Tupel

❑ Ein Tupel kann als Aggregat von Basiswerten aufgefaßt werden.

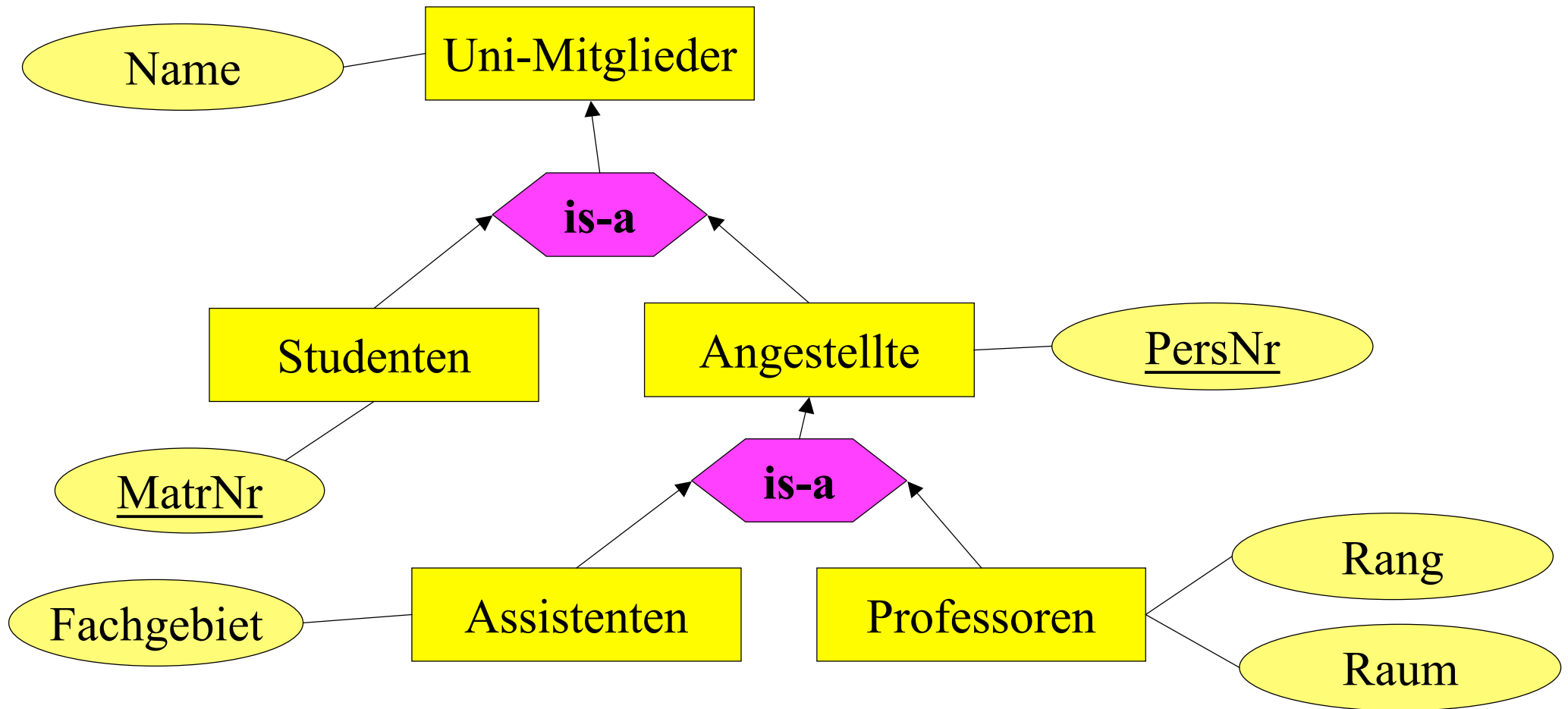
# Generalisierung und Spezialisierung <sup>(1)</sup>

---

- ❑ Spezialisierung bezeichnet die Verfeinerung einer Klasse (mehr Information).
- ❑ Generalisierung ist die Vergröberung einer Klasse (weniger Information).
- ❑ Spezielle Klassen (Subklasse) und allgemeine Klassen (Superklasse) bilden eine Subklassenhierarchie (» Subtypisierung, Typhierarchie).
- ❑ Instanzen einer Klasse sind auch Instanzen der Superklasse.
- ❑ Spezialisierungen können disjunkt oder überlappend sein.
- ❑ Subklassen erben die Eigenschaften der Superklasse und fügen evtl. neue hinzu (» Vererbung).
- ❑ Bei Operationen auf Instanzen von Klassen können auch Instanzen von Subklassen verwendet werden (» Subtypisierung).



# Generalisierung und Spezialisierung (2)

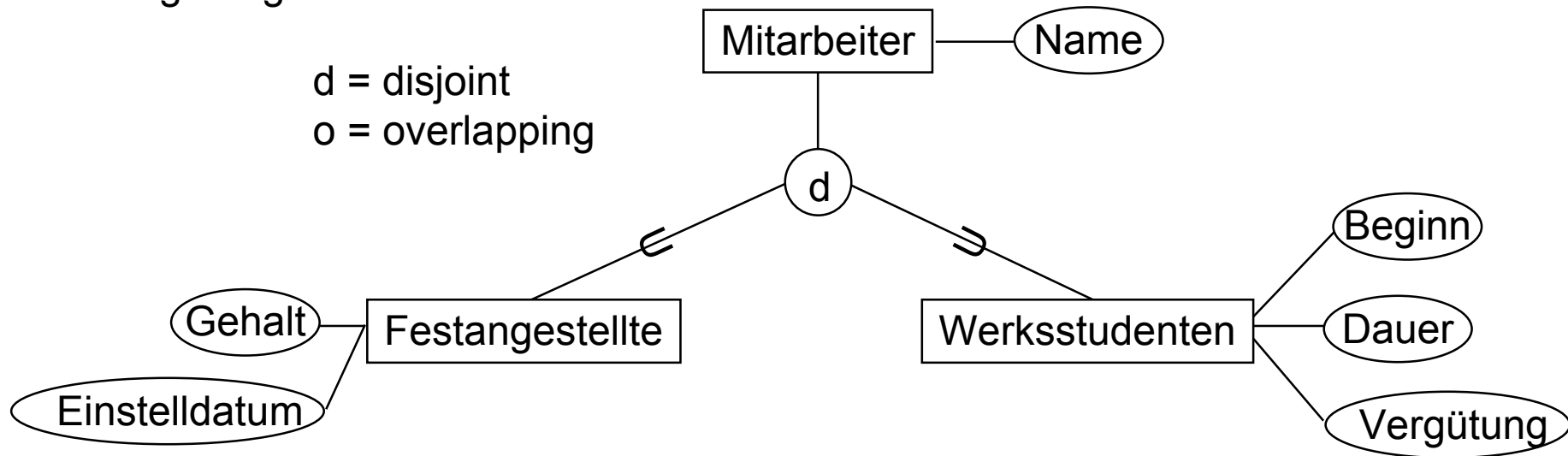


# Generalisierung und Spezialisierung (3)

## Erweitere Entity-Relationship-Diagramme

### Beispiel:

- ❑ Festangestellte und Werksstudenten sind Mitarbeiter. Festangestellte sind keine Werksstudenten
- ❑ Festangestellte haben die zusätzliche Eigenschaften Gehalt und Einstelldatum.
- ❑ Werksstudenten haben die zusätzliche Eigenschaften Beginn, Dauer und Vergütung.

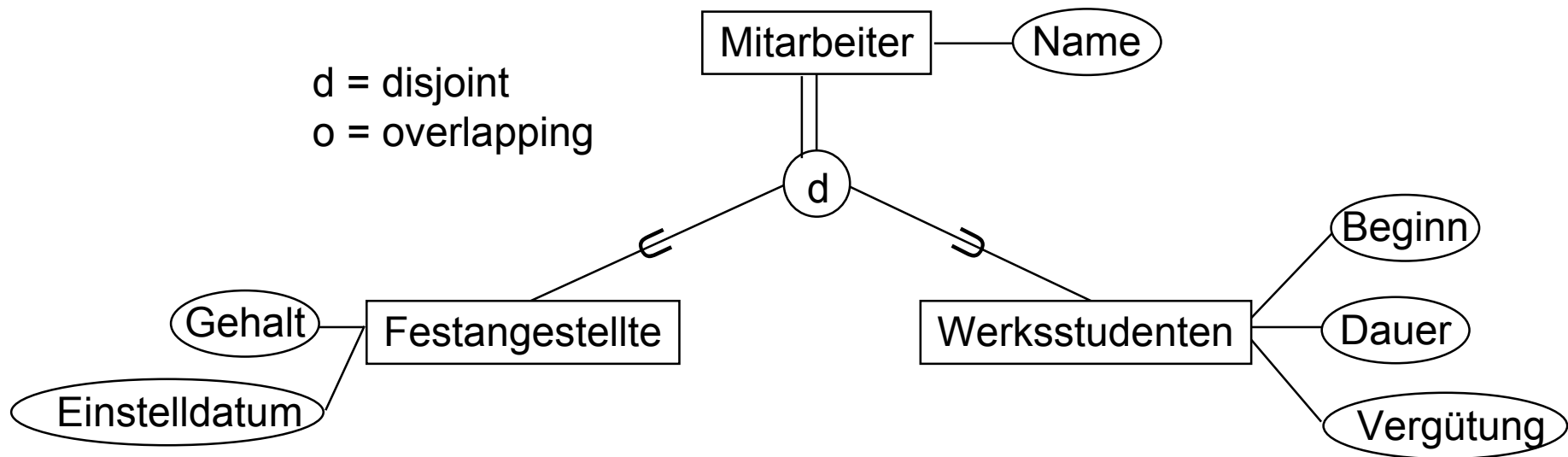


# Generalisierung und Spezialisierung (4)

## Erweitere Entity-Relationship-Diagramme

### Beispiel:

- ❑ Festangestellte und Werksstudenten sind Mitarbeiter. Festangestellte sind keine Werkstudenten
- ❑ Die Menge der Mitarbeiter ist gleich der Vereinigung der Mengen Festangestellte und Werkstudenten, d.h. ein Mitarbeiter ist entweder festangestellt oder Werkstudent



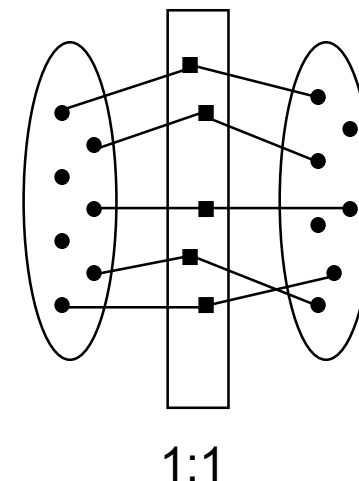
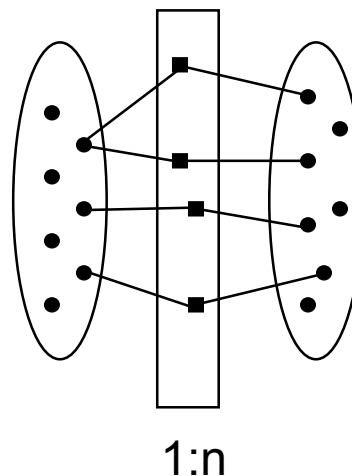
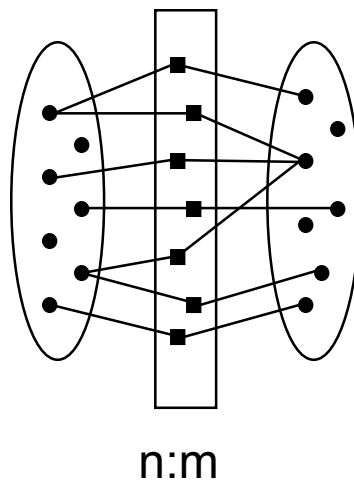
# Assoziation (1)

---

❑ Objekte können miteinander in Beziehung gesetzt (assoziiert) werden:

- Binäre (ternäre, ...) Beziehungen assoziieren zwei (drei, ...) Klassen oder Objekte.
- Allgemein: n-äre Beziehungen zwischen n Klassen oder Objekten, wobei n der Grad der Beziehung ist.
- Beziehungen können dynamisch geknüpft, ausgewertet, gelöst werden.

❑ Kardinalitätsbeschränkungen legen die genaue Zahl oder ein Intervall für die Anzahl der in Beziehung stehenden Instanzen fest.



# Assoziation (2)

---

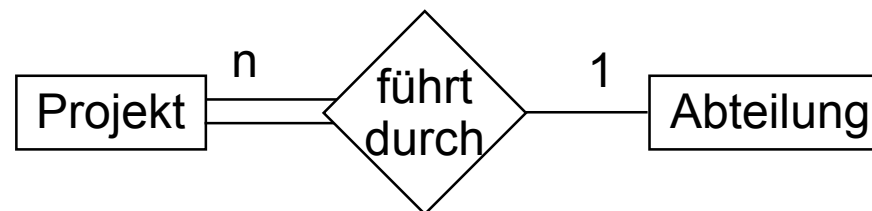
- ❑ **Totale Partizipation:** Jede Instanz einer Klasse muß mit einer Instanz der zweiten Klasse in Beziehung stehen (====).
- ❑ **Partielle Partizipation:** Eine Instanz einer Klasse kann in Beziehung zu einer Instanz der zweiten Klasse stehen (-----) s. Beispiel auf der nächsten Folie).
- ❑ Rollennamen identifizieren die Menge der Instanzen, die mit einer anderen Instanz in Beziehung stehen.
- ❑ Rollen können als abgeleitete Attribute verstanden werden, die die Menge der Instanzen als Attributwerte besitzen.

# Assoziation (3)

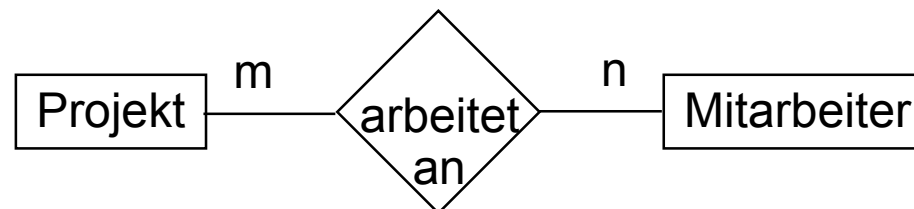
---

## Beispiele:

- ❑ Projekte werden von Abteilungen durchgeführt. Jedes Projekt muß einer Abteilung zugeordnet sein. Eine Abteilung kann mehrere Projekte ausführen.



- ❑ An Projekten arbeiten Mitarbeiter. Ein Mitarbeiter kann an mehreren Projekten arbeiten. Jedes Projekt wird von beliebig vielen Mitarbeitern bearbeitet.



- ❑ Bemerkung: In der Literatur findet man auch andere Beschriftungsregeln.

# Beispiel 1:

---

## **Gegeben:** "Anforderungsdefinition"

An einer Universität werden verschiedene Vorlesungen angeboten, die Teil mehrerer Studienfächer sind.

Diese Vorlesungen werden von genau einem Dozenten gehalten.

Jeder Dozent ist Mitglied genau eines Fachbereiches.

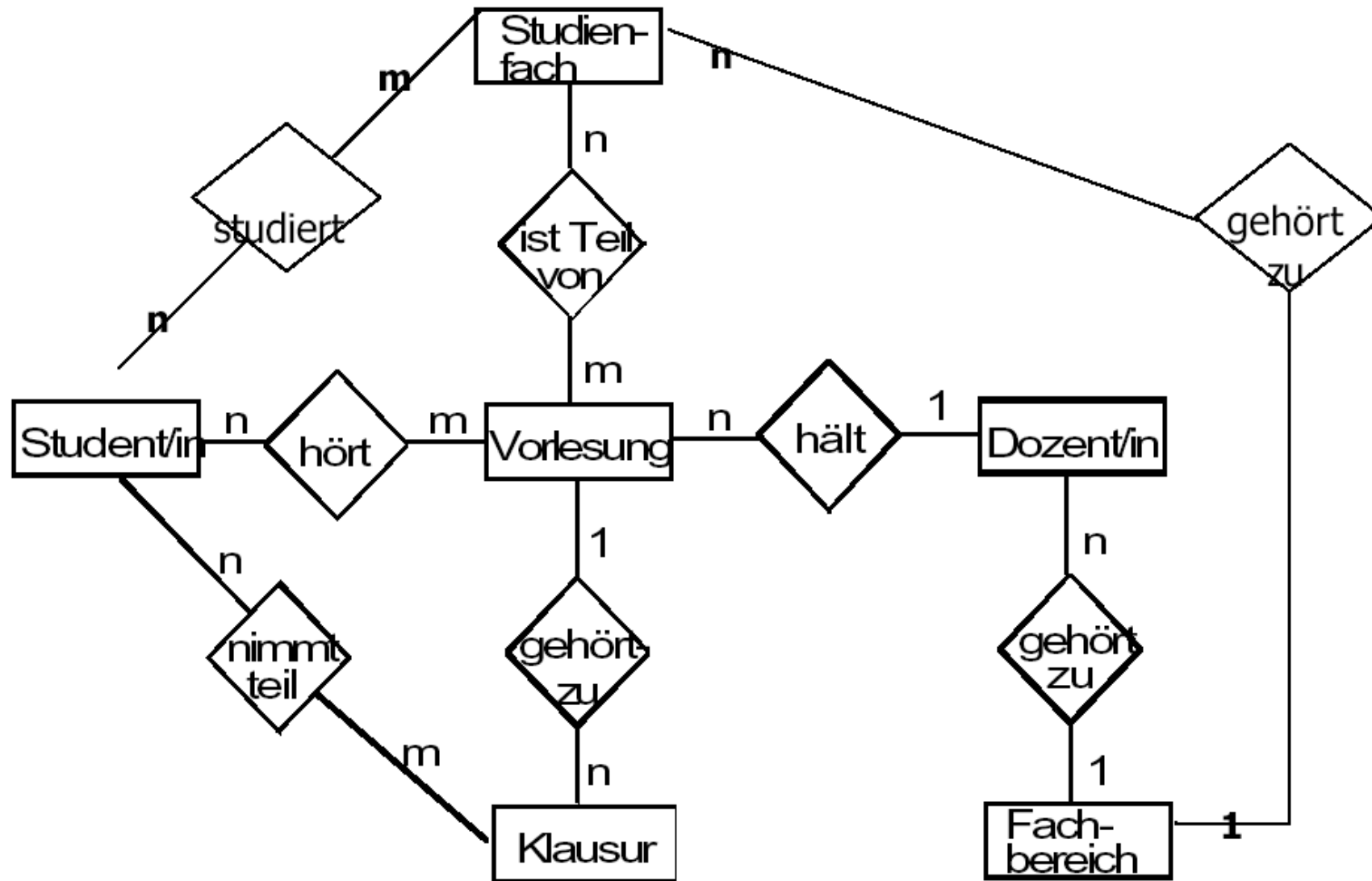
Ein Fachbereich hat mehrere Studienfächer.

Die Vorlesungen werden von Studenten gehört, die jeweils ein oder mehrere Studienfächer belegt haben.

Zu jeder Vorlesung werden mehrere Klausuren angeboten, die von den Studenten geschrieben werden.

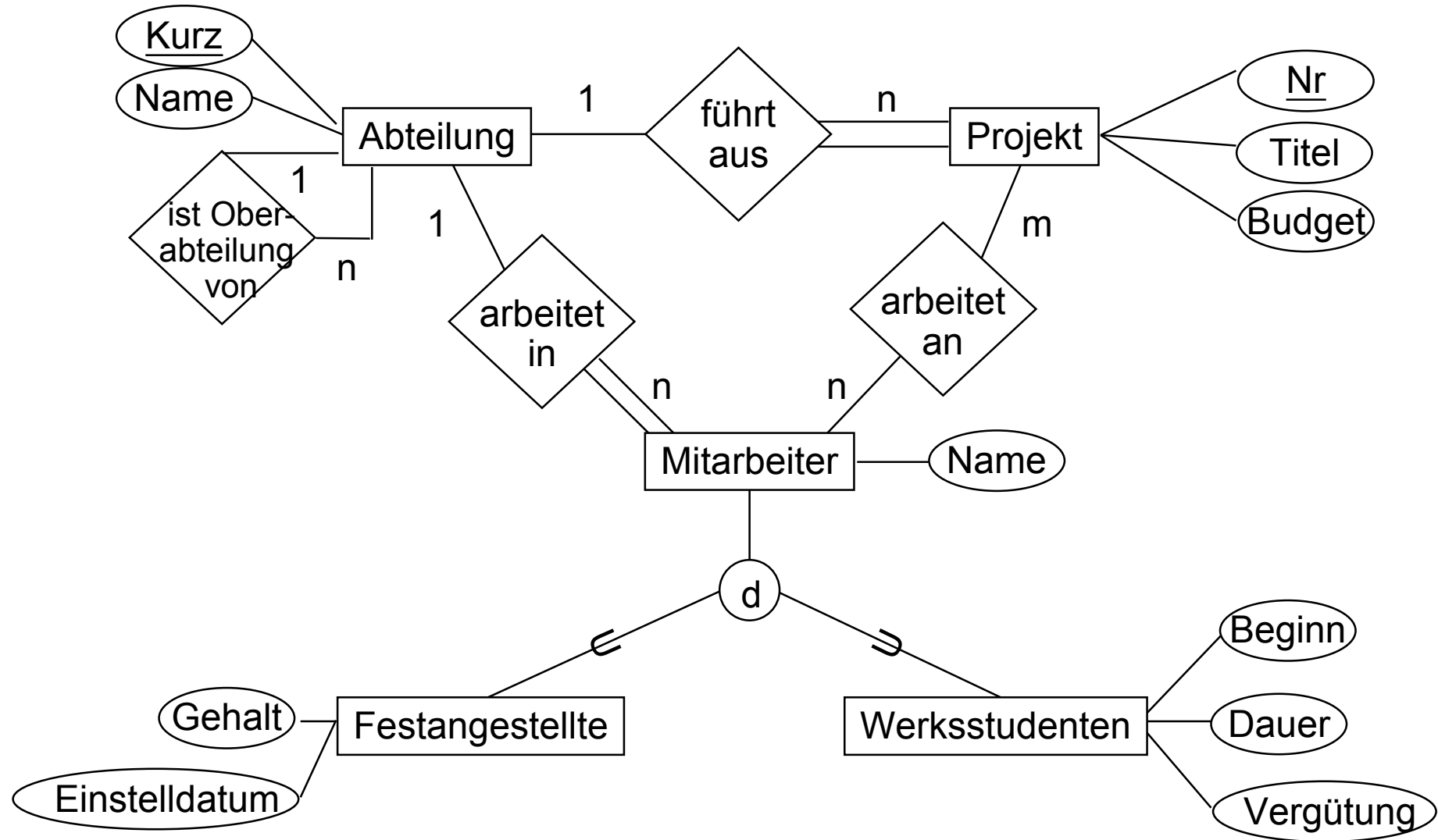
## **Gesucht:** ER-Diagramm

# Mögliche Lösung





# Beispiel 2: ER-Diagramm erläutern



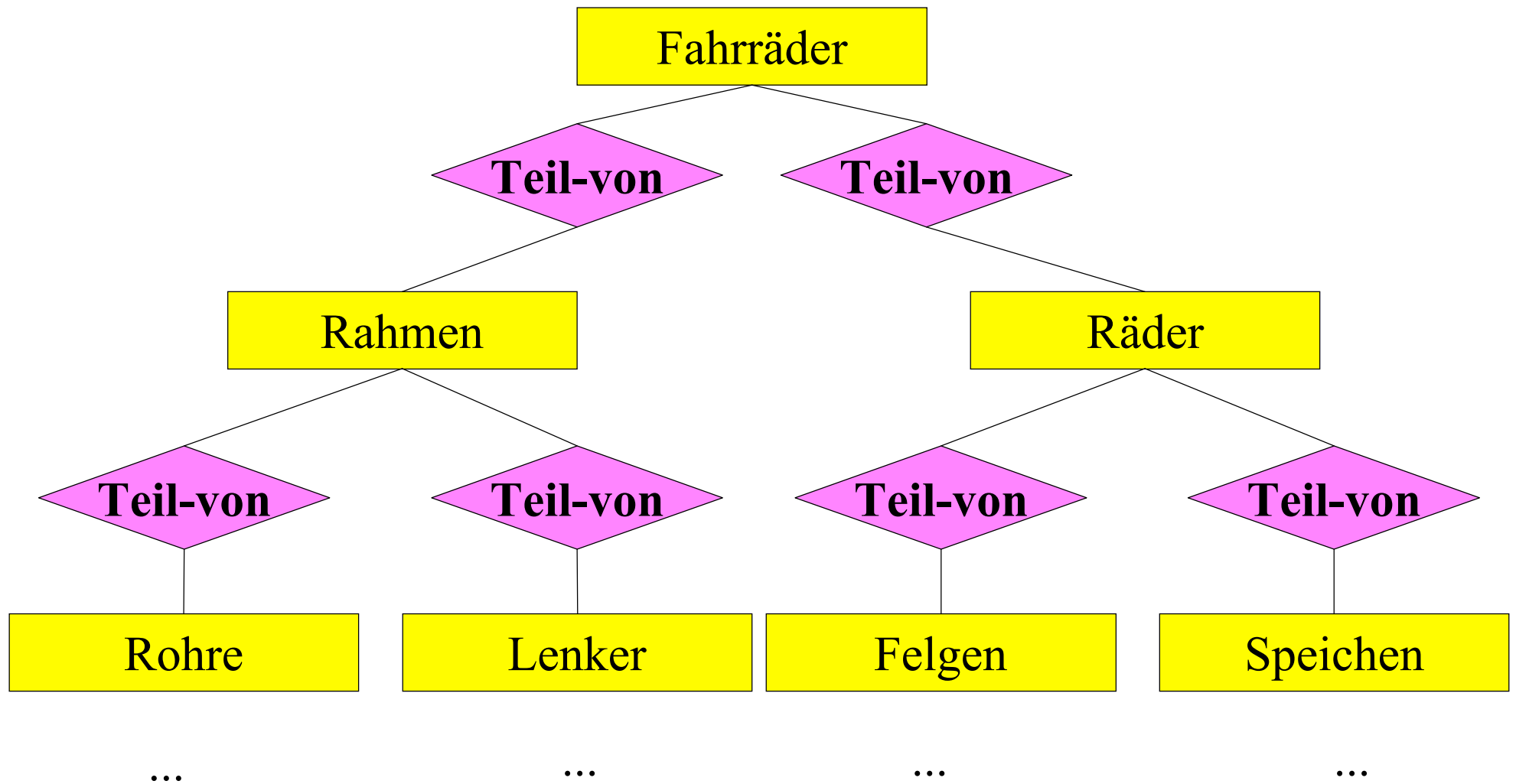
# Aggregation und Dekomposition <sup>(1)</sup>

---

- ❑ Objekte können zu übergeordneten Objekten aggregiert werden:
  - Beziehungen zwischen Komponenten und übergeordnetem Objekt
  - Übergeordnetes Objekt kann wiederum an Beziehungen teilnehmen.
- ❑ Im Vergleich zu "normalen" Assoziation wird die "Aggregation" in der Entity-Relationship-Modellierung nicht besonders unterstützt

# Aggregation

---



# Identifikation und Schlüssel (1)

---

Zur **Identifikation** existieren zwei grundlegende Ansätze in Datenbankmodellen:

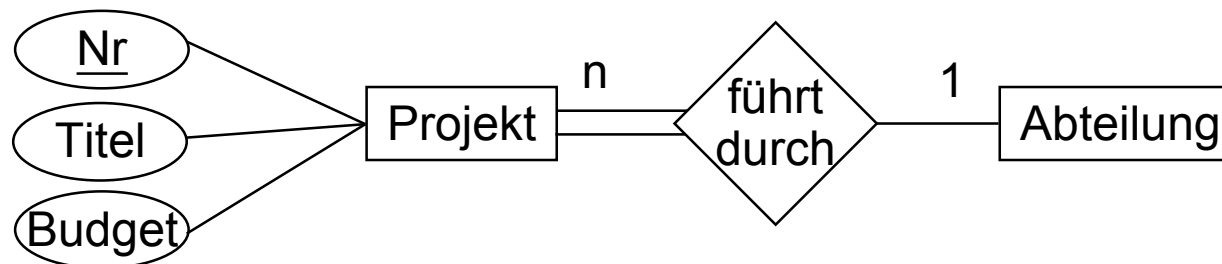
- ❑ **Referentielle Identifikation** bezeichnet direkte Verweise auf Objekte (»»»» Zeiger in Programmiersprachen).
- ❑ **Assoziative Identifikation** verwendet die Werte von Attributen oder Attributkombinationen, um sich eindeutig auf Objekte zu beziehen (»»»» Schlüssel: Ausweisnummer, Fahrgestellnummer, ...).
- ❑ In der Praxis benötigt man häufig beide Formen der Identifikation.

## Schlüssel:

- ❑ Schlüssel sind Attribute oder Attributkombinationen mit innerhalb einer Klasse eindeutigen Werten und eignen sich deshalb zur Identifikation.
- ❑ Es kann mehrere Schlüsselkandidaten geben (Primärschlüssel, Sekundärschlüssel).
- ❑ Schlüssel stellen als Attributwerte Beziehungen zu anderen Objekten her (Fremdschlüssel).
- ❑ Durch Fremdschlüssel referenzierte Objekte müssen existieren (»»»» referentielle Integrität).

# Identifikation und Schlüssel (2)

**Beispiel:** Projekte können durch eine Nummer eindeutig identifiziert werden.



Dabei existieren zwei Möglichkeiten zur Identifikation von Projekten innerhalb der Assoziation "führt durch":

## Referentielle Identifikation

	Projekt	Abteilung
←	•	...
←	•	...
←	•	...
	...	...

## Assoziative Identifikation

Projekt	Abteilung
4711	...
4712	...
4713	...
...	...