

Datenbanksysteme

3 Das Entity-Relationship-Modell

Prof. Dr. Gregor Grambow

Hochschule Aalen
Fakultät Elektronik und Informatik

Überblick

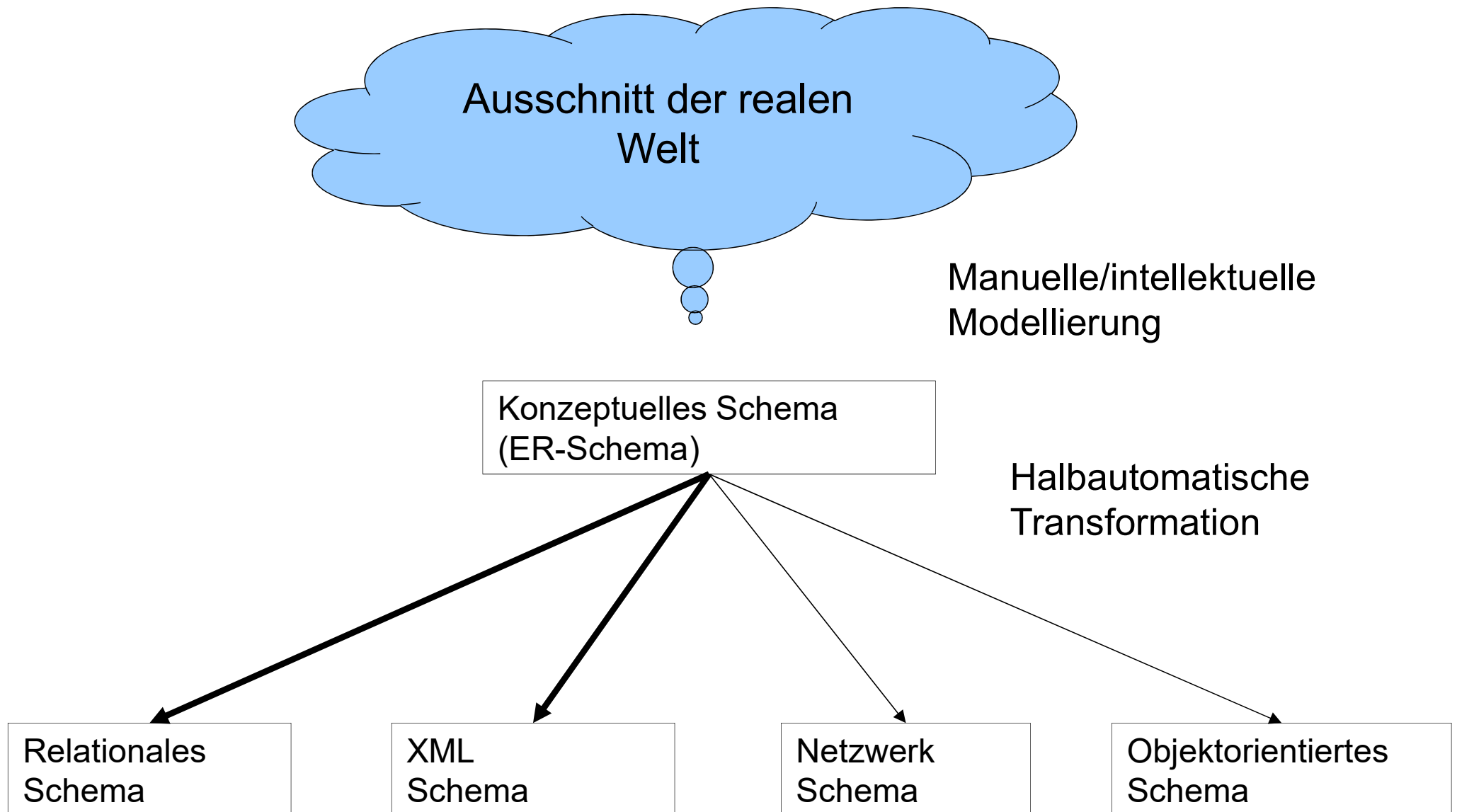
Inhalt

- Eigenschaften
- Datenbankentwurf
- Beispiel

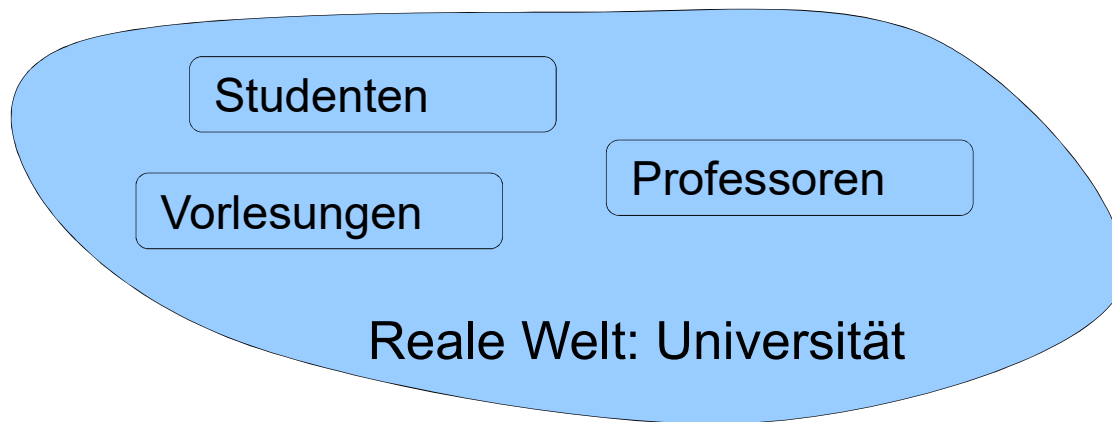
Ziele

- Grundlagen der ER Modellierung kennen

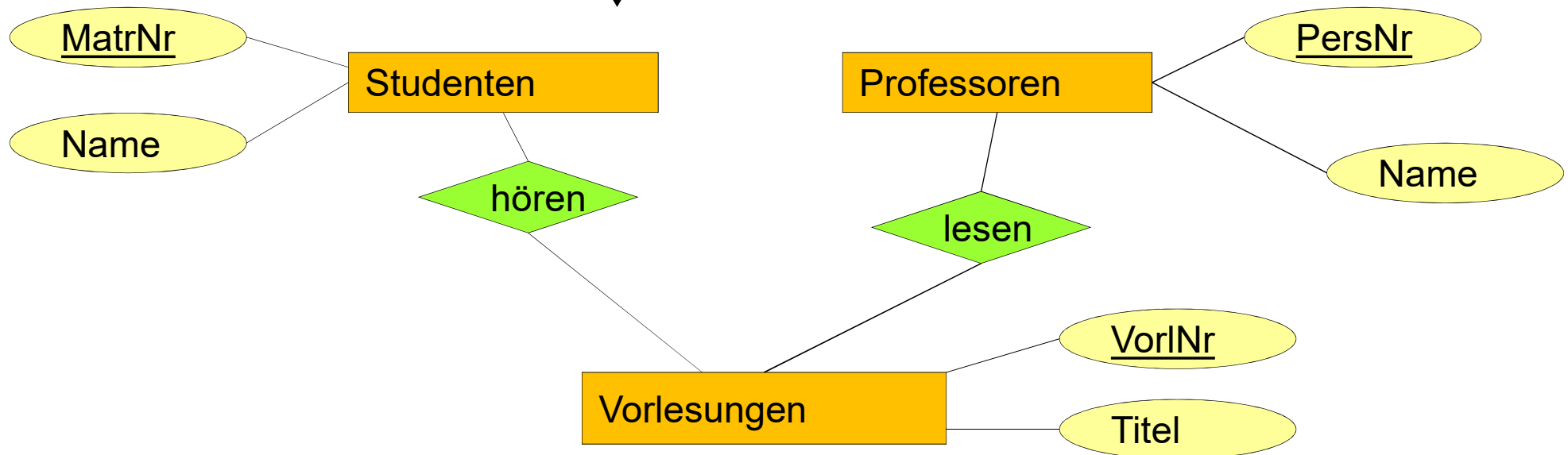
Datenmodellierung



Modellierung einer kleinen Beispielanwendung



Konzeptuelle Modellierung



Ein Datenmodell für die konzeptuelle Ebene

Das Datenmodell hat die Aufgabe einen definierten Ausschnitt der realen Welt so detailliert zu beschreiben, *wie es für Anwendungszweck adäquat ist*.

Den Ausschnitt nennt man *Sachbereich* oder *Domäne* (*Domain*) des Modells.

- entity (Gegenstand)
wohlunterscheidbare physikalische oder gedachte Objekte
- relationship (Beziehung, Relation)
Beziehungen zwischen entities

→ Entity-Relationship-Modell.

Peter Chen: The entity-relationship model - Toward a unified view of data.
ACM Transactions on Database Systems 1, 1976, 9-36.

Beispiele: Entities und Relationships

entities

- * Unternehmen
- * Abteilung
- * Mitarbeiter
- * Person
- * Projekt
- * Lieferant

relationships

- * Anstellung (Mitarbeiter, Abteilung)
- * Lieferung (Lieferant, Projekt, Artikel)
- * Empfehlung (Person, Buch, Thema)

Attribute

- beschreiben Entities
- jedes Attribut a eines Entities hat einen *Wertebereich (value set)* V
- formal: a ist Abbildung $a: E \rightarrow V$
- Element v eines Wertebereichs V : *Wert* $v \in V$
- Schreibweise:
- Wertebereich (domain) eines Attributs a : $dom(a)$

Beschreibung von Entities

- Ein Entity e repräsentiert *genau einen* Gegenstand aus dem Sachbereich.
 - Beschreibung durch Menge $A = \{a_1, \dots, a_n\}$ von *Attributen* (Merkmale)
 - die *Kombination* der *Werte* zu jedem Attribut unterscheidet einen Gegenstand von allen anderen

formal:

- Entity e ist Abbildung $e: \{a_1, \dots, a_n\} \rightarrow \prod_{i=1}^n \text{dom}(a_i)$
- Entity e ist Element von $\text{dom}(a_1) \times \dots \times \text{dom}(a_n)$

Entity-Typ, Entity-Set

- *Entity-Typ* E: Klasse von Entities mit gleichen Attributen
 - Beschreibung durch *Entity-Deklaration*: $E = (A, K)$
 - $A = \{a_1, \dots, a_n\} \rightarrow$ Attributmenge, die dem Entity-Typ E zugeordnet ist.
- Beispiel:
 - Mitarbeiter =
({Personalnr : Zahl, Name : ZchKette, ..., Steuerklasse : {1..6} }, ...)
- *Entity-Set* E^t : Menge der Entities, die zu einem Zeitpunkt t zum Entity-Typ E gehören.

Beispiel: Entities

- Beispiel: Modellierung von Mitarbeitern → Entity-Typ Mitarbeiter
- Mitarbeiter^{1.10.2004} = {Hugo, Anton}
 - Attributmenge = {Personalnr, Name, Vorname, Einstellungsdatum, Stellung, Gehalt, Steuerklasse}
 - Hugo = (21, Storch, Hugo, 19950601, Arbeiter, 2250, 1) →
Personalnr(Hugo) = 21
 - Anton = (3, Frosch, Anton, 19770101, Angestellter, 3200, 3) →
Personalnr(Anton) = 3

Attribut a	Wertebereich – dom(a)	Hugo	Anton
Personalnr	4-stellige Zahl	21	3
Name	Zeichenkette Länge 20	Storch	Frosch
Vorname	Zeichenkette Länge 10	Hugo	Anton
Einstellungsdatum	Datum	19950601	19770101
Stellung	{Arbeiter, Angestellter}	Arbeiter	Angestellter
Gehalt	6-stellige Zahl	2250	3200
Steuerklasse	{1..6}	1	3

Projektion

- Sind nicht alle Attribute $a_i \in A$ eines Wertes $w \in \text{dom}(a_1) \times \dots \times \text{dom}(a_n)$ interessant, so lässt sich der Wert durch die Operation

Projektion

- auf die gewünschten Attribute aus $B \subseteq A$ reduzieren:

Schreibw. 1 Schreibw. 2 Bedeutung

$w|_B$

$w.B$

$(w_b \mid b \in B)$ (jedes $w_b \in \text{dom}(b)$)

- Ist B einelementig, z.B. $B = \{x\}$ ($x \in A$), so gilt:
- $w|\{x\} = w.\{x\} = (wx)$ oder einfach wx

Identifizierende Attributmenge – Superkey

- Eine Menge $I \subseteq A$ von Attributen eines Entity-Typs $E = (A, K)$, die alleine ausreicht, um ein Tupel eindeutig zu identifizieren, heißt
- *Identifizierende Attributmenge* oder *Superkey* von E .

Schlüssel – Key

- Eine minimale identifizierende Attributmenge von E heißt
Schlüssel oder Key von E
- Ein Entity-Typ kann mehrere Schlüssel haben.
- Ein Schlüssel wird als Primärschlüssel ausgezeichnet.

Beispiel (ohne Angabe der Wertebereiche)

- Mitarbeiter' = ({ P-Nr, Name, Vorname, Einst-Datum, Gehalt, Steuerkl }, ...)
- Schlüssel (Mitarbeiter'): {P-Nr} und { Name, Vorname, Einst-Datum }

Deklaration mit Primärschlüssel:

- Mitarbeiter' =
- ({ P-Nr, Name, Vorname, Einst-Datum, Gehalt, Steuerkl }, {P-Nr})
- *alternativ:*
- Mitarbeiter' =
- { P-Nr, Name, Vorname, Einst-Datum, Gehalt, Steuerkl }

Relationship – Beziehung

Beziehungstyp (Relationship-Typ)

- Beschreibung durch *Relationship-Deklaration*:
- $R = (E_1 \times \dots \times E_n, RA) \quad (RA = \{a_1, \dots, a_m\})$
- Bsp.: Projektmitarbeit = (Mitarbeiter \times Projekt , {Wochenarbeitsleistung : Zahl})

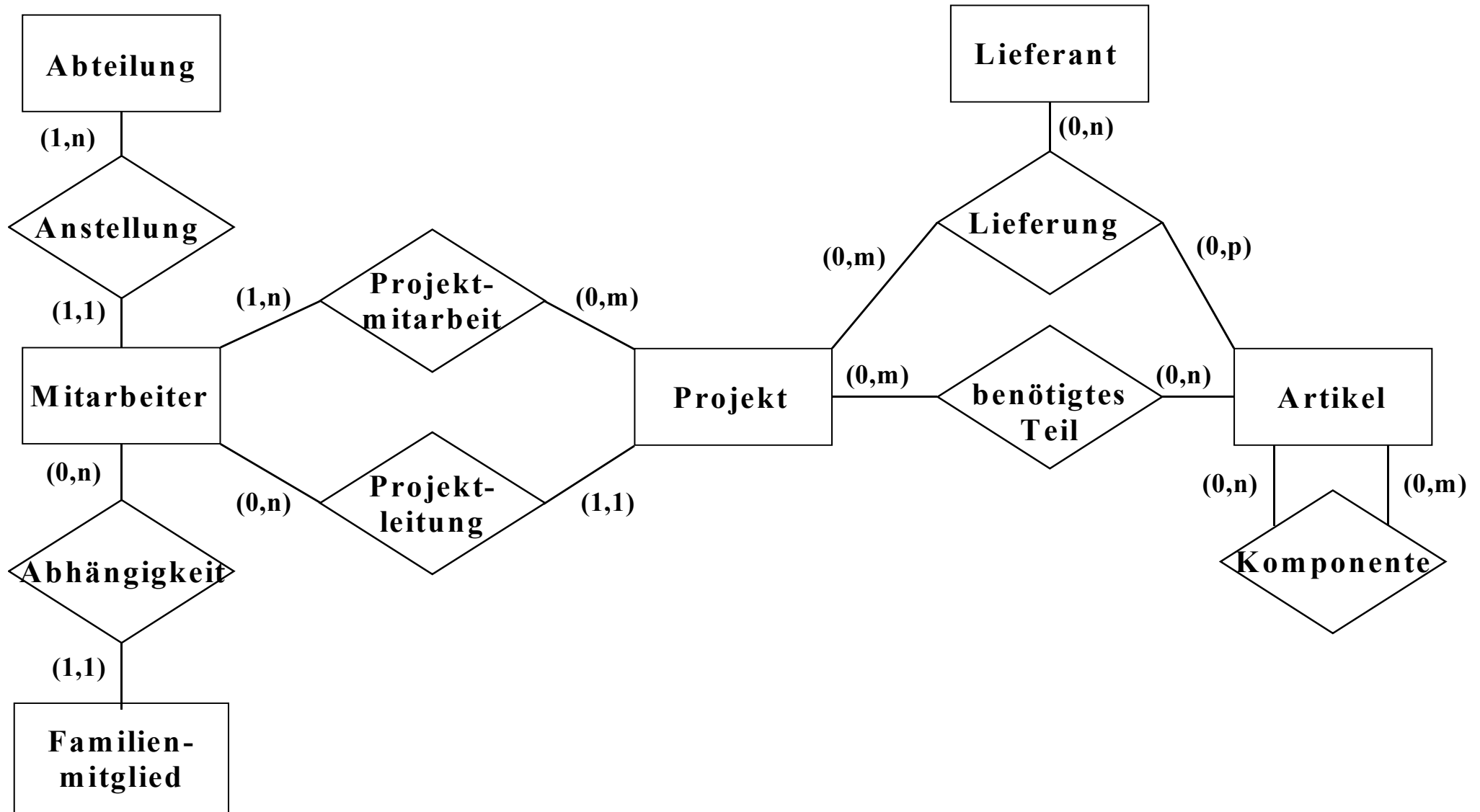
konkrete Beziehung

- $r = (e_1, \dots, e_n, v_1, \dots, v_m) \quad (e_i \in E_i^t, v_j \in \text{dom}(a_j))$
- Bsp.: $r = (\text{Maier}, P3, 10)$
- Einem Beziehungstyp $R = (E_1 \times \dots \times E_n, \{a_1, \dots, a_m\})$ ist zu jedem Zeitpunkt t die Menge R^t der konkreten Beziehungen zugeordnet.

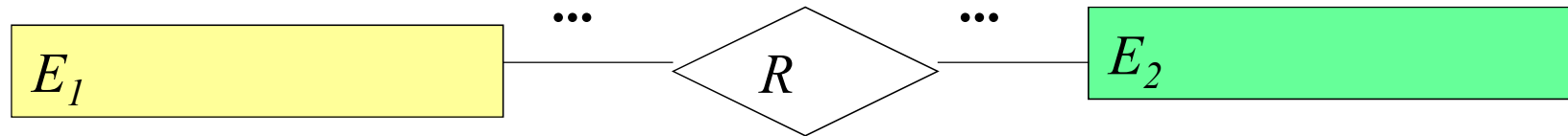
Grafische Darstellung eines ER-Modells

- Knoten:
 - Rechtecke mit Namen: Entity-Typen
 - Ovale mit Namen: Attribute
 - Rauten mit Namen: Abstrakte Beziehungen
- Kanten verbinden:
 - (Rauten für) abstrakte Beziehungen mit (Rechtecken für) Entity-Typen
 - (Ovale für) Attribute mit (Rechtecken für) Entity-Typen

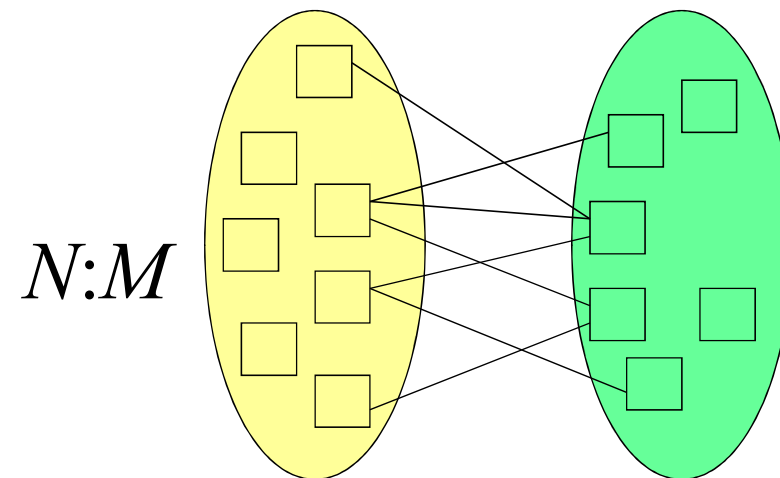
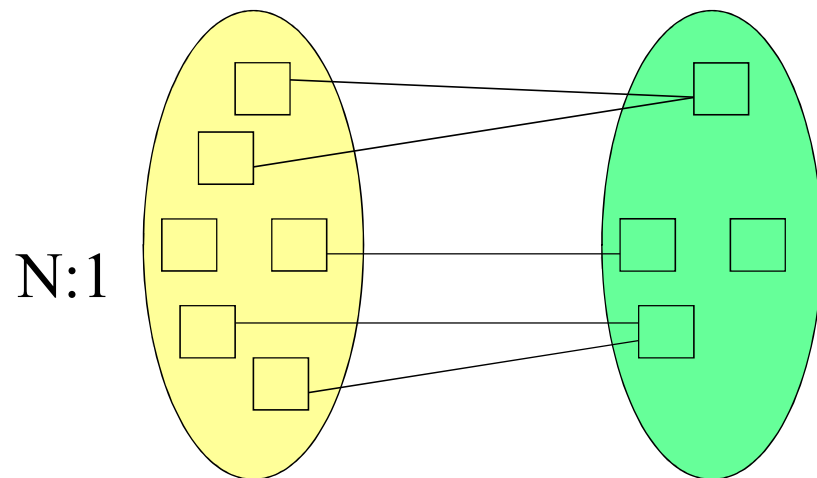
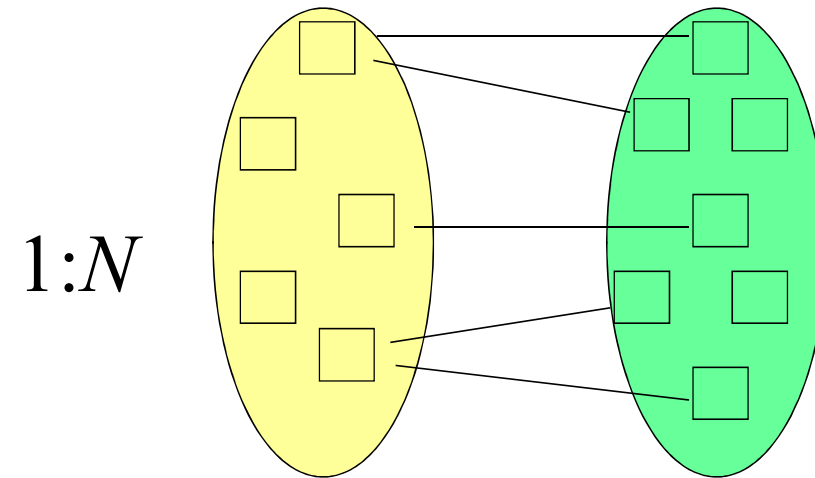
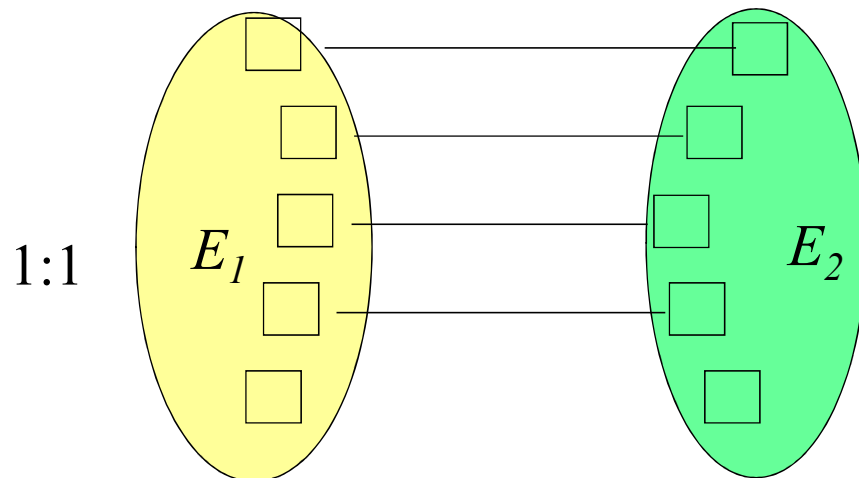
Beispiel grafische Darstellung ER-Modell



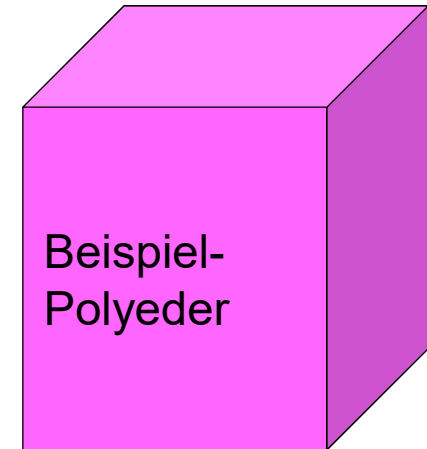
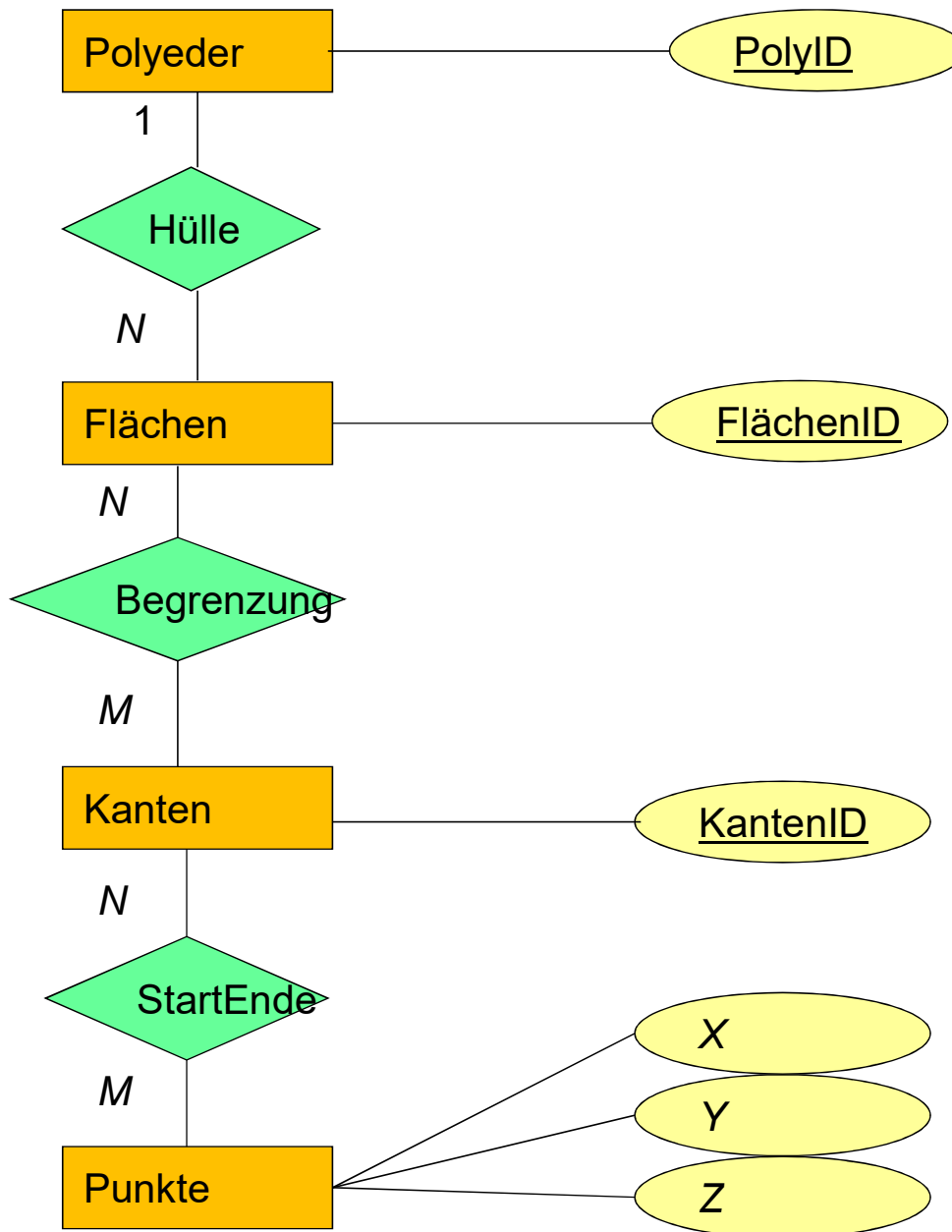
Kardinalitäten (Chen Notation)



$$R \subseteq E_1 \times E_2$$



Beispiel Polyeder



(min, max)-Notation

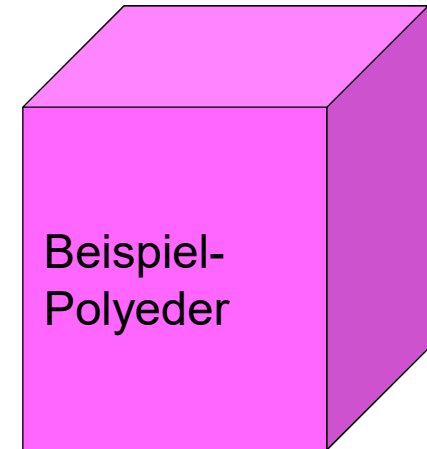
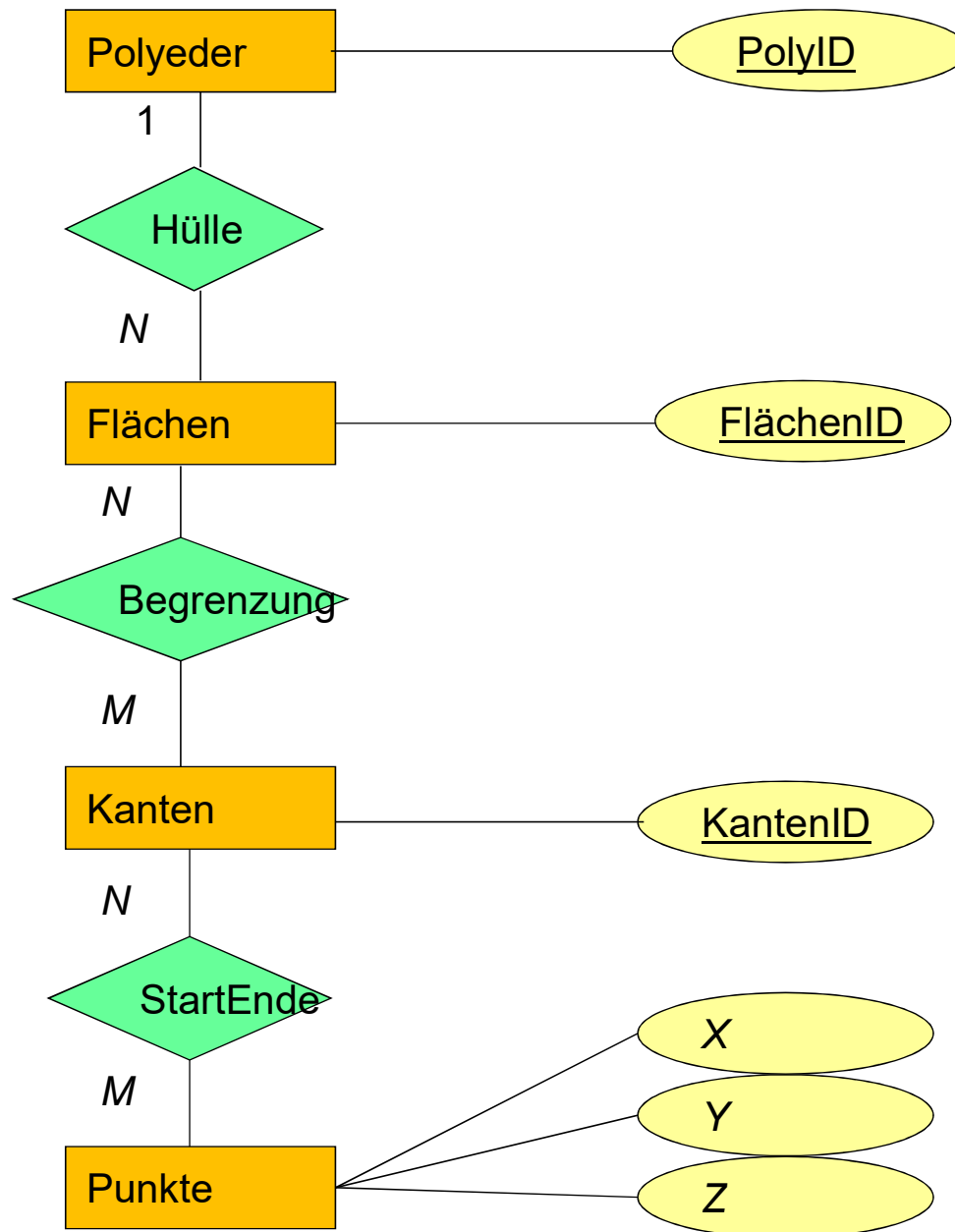
- Kardinalität von Entity-Typen in abstrakter Beziehungen
 $R=(E_1 \times E_2 \times \dots \times E_n)$:
 - minimale Kardinalität von E_x in R :
Mindestanzahl von verschiedenen (konkreten) Beziehungen, in denen jedes Element aus E_x auftreten muss.
 - maximale Kardinalität von E_x in R :
Maximalanzahl von verschiedenen (konkreten) Beziehungen, in denen ein Element aus E_x auftreten darf.
- Notation: (<minimale Kardinalität>, <maximale Kardinalität>)
(Buchstabe \rightarrow “beliebig viel”)

(min, max)-Notation

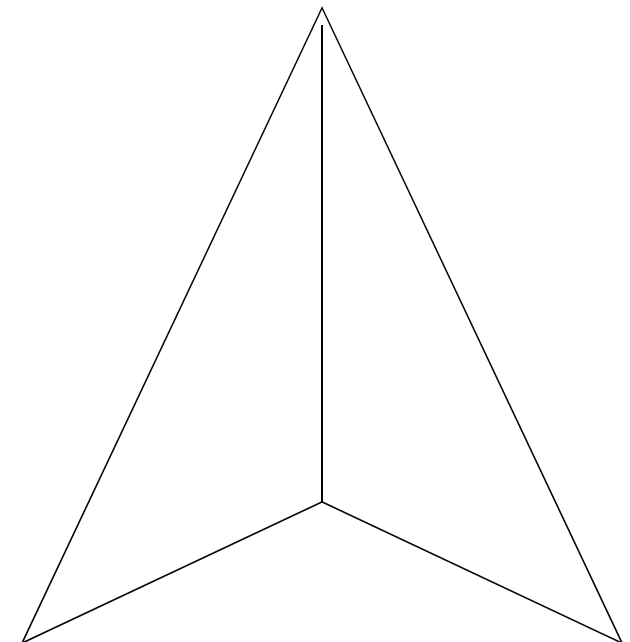
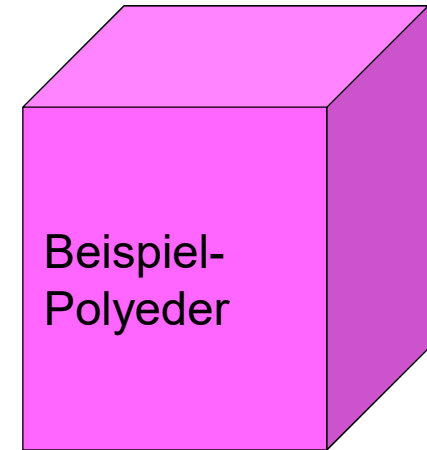
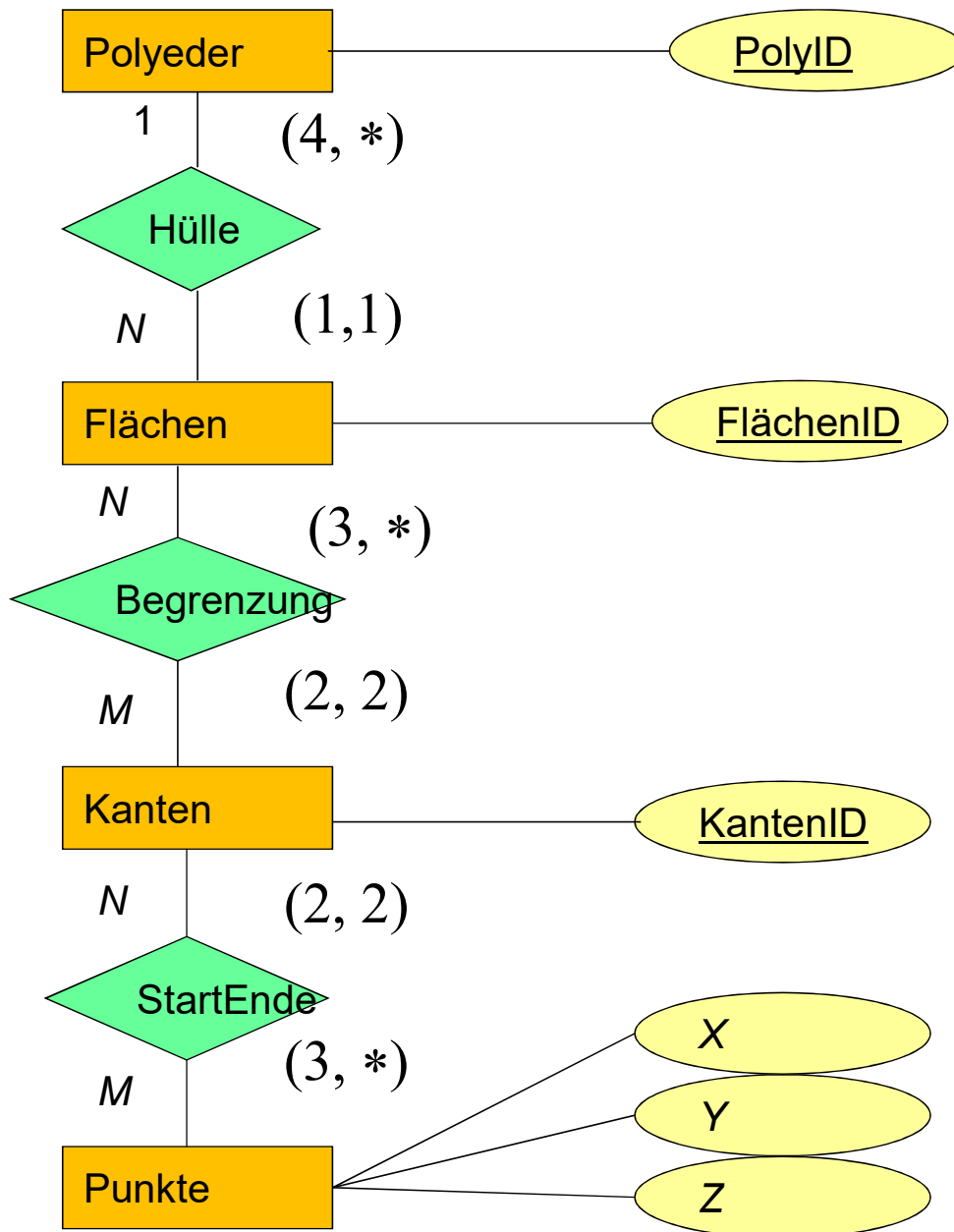
Beispiel: Projektmitarbeit (Mitarbeiter, Projekt)

- Kardinalität von *Mitarbeiter* in *Projektmitarbeit*: (1,n)
jeder *Mitarbeiter* muss in mind. einer Relation *Projektmitarbeit* vorkommen;
d.h. an mindestens einem Projekt mitarbeiten.
- Kardinalität von *Projekt* in *Projektmitarbeit*: (0,n)
ein Projekt kann (muss aber nicht) in einer oder beliebig vielen Relationen *Projektmitarbeit* vorkommen;
d.h. an einem Projekt können beliebig viele Mitarbeiter arbeiten, es gibt aber auch Projekte ohne Mitarbeiter (z.B. abgeschlossene).

Beispiel Polyeder



Beispiel Polyeder



Beispiel ER-Modellierung: Hochschule

- Angestellte
 - Anzahl: 10 000
 - Attribute:
 - PersonalNummer
 - Typ: char
 - Länge: 9
 - Wertebereich:
0...999.999.999
 - Identifizierend: ja
- Gehalt
 - Typ: dezimal
 - Länge: (8,2)
 - Identifizierend: nein
- Rang
 - Typ: char
 - Länge: 4
 - Identifizierend: nein

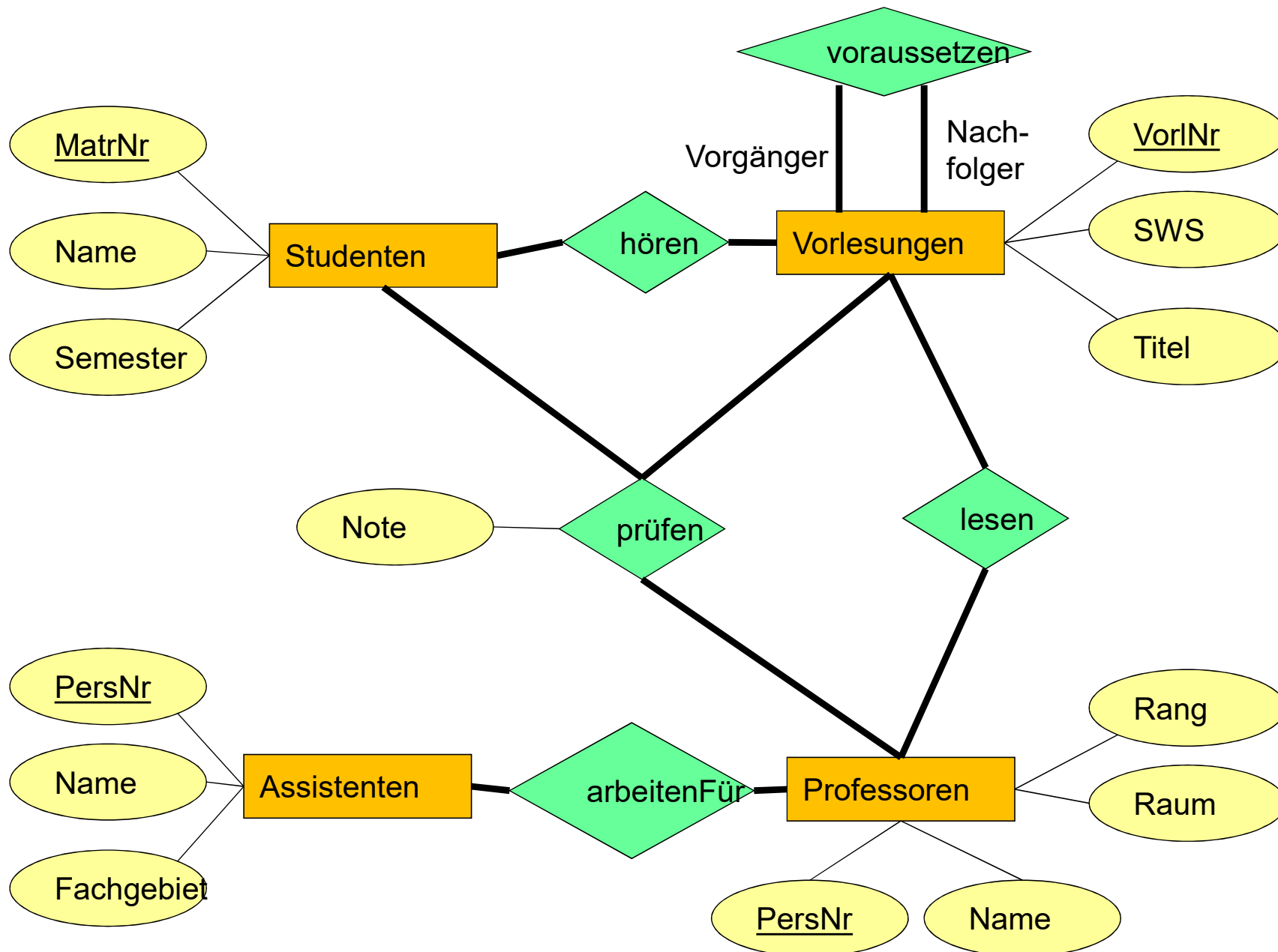
Beispiel ER-Modellierung: Hochschule

- Beziehung: prüfen
- Beteiligte Objekte:
 - Professor als Prüfer
 - Student als Prüfling
 - Vorlesung als Prüfungsstoff
- Attribute der Beziehung:
 - Datum
 - Uhrzeit
 - Note
- Anzahl: 1 000 000 pro Jahr

Beispiel ER-Modellierung: Hochschule

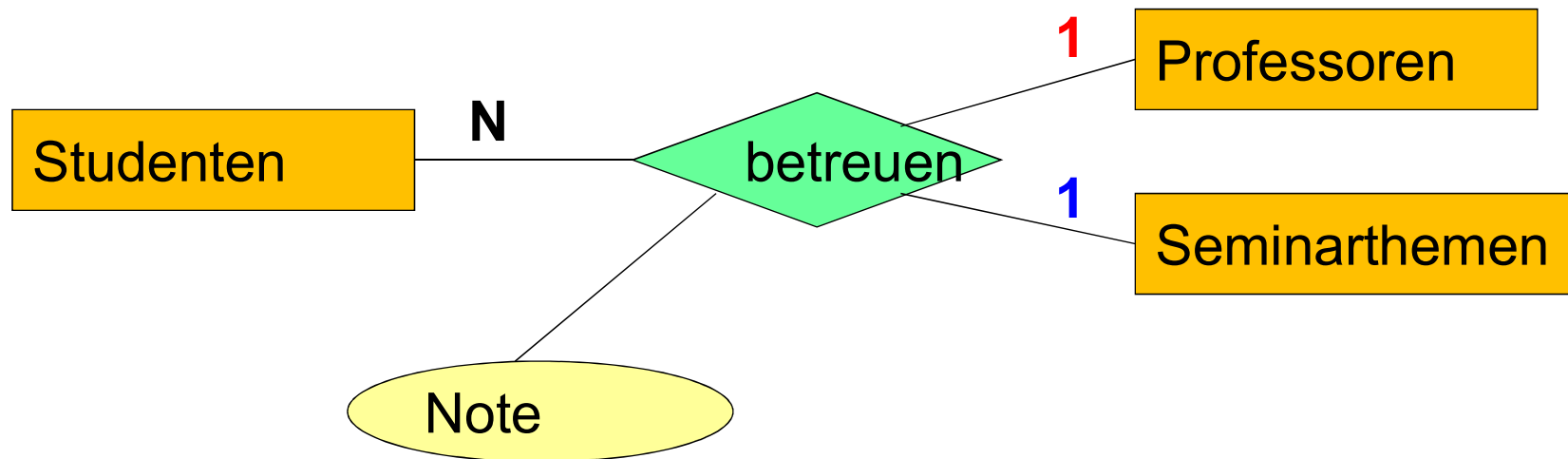
- Prozessbeschreibung: Zeugnisausstellung
 - Häufigkeit: halbjährlich
 - benötigte Daten
 - Prüfungen
 - Studienordnungen
 - Studenteninformation
 - ...
 - Priorität: hoch
 - Zu verarbeitende Datenmenge
 - 5 000 Studenten
 - 100 000 Prüfungen
 - 100 Studienordnungen

Beispiel ER-Modellierung: Hochschule



Beispiel ER-Modellierung: Hochschule

- Beispiel-Beziehung: *betreuen*



betreuen : Professoren x Studenten \rightarrow Seminarthemen

betreuen : Seminarthemen x Studenten \rightarrow Professoren

Beispiel ER-Modellierung: Hochschule

Dadurch erzwungene Konsistenzbedingungen:

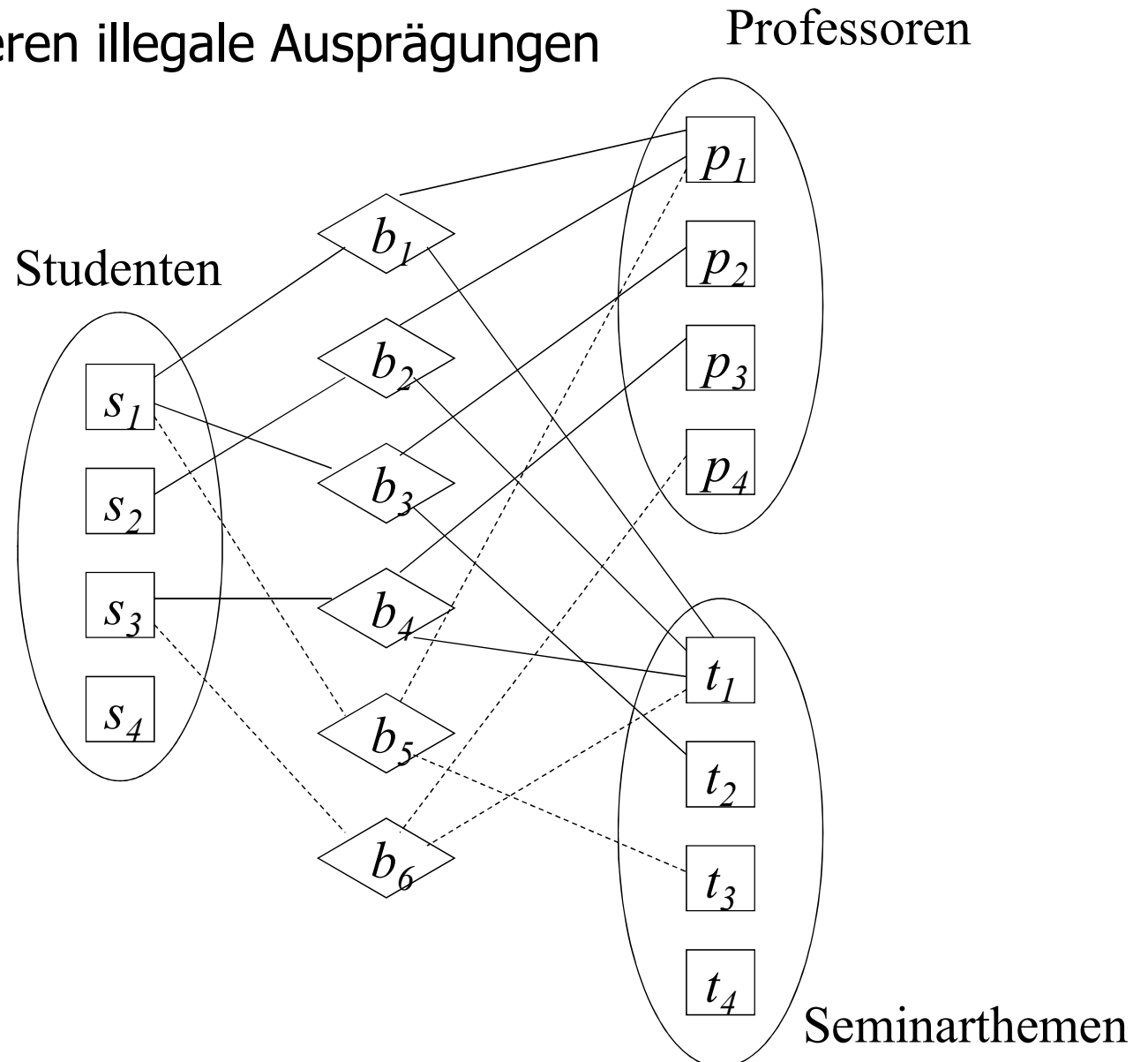
1. Studenten dürfen bei demselben Professor bzw. derselben Professorin nur ein Seminarthema "ableisten" (damit ein breites Spektrum abgedeckt wird).
2. Studenten dürfen dasselbe Seminarthema nur einmal bearbeiten – sie dürfen also nicht bei anderen Professoren ein schon einmal erteiltes Seminarthema nochmals bearbeiten.

Es sind aber folgende Datenbankzustände nach wie vor möglich:

- Professoren können dasselbe Seminarthema „wiederverwenden“ – also dasselbe Thema auch mehreren Studenten erteilen.
- Ein Thema kann von mehreren Professoren vergeben werden – aber an unterschiedliche Studenten.

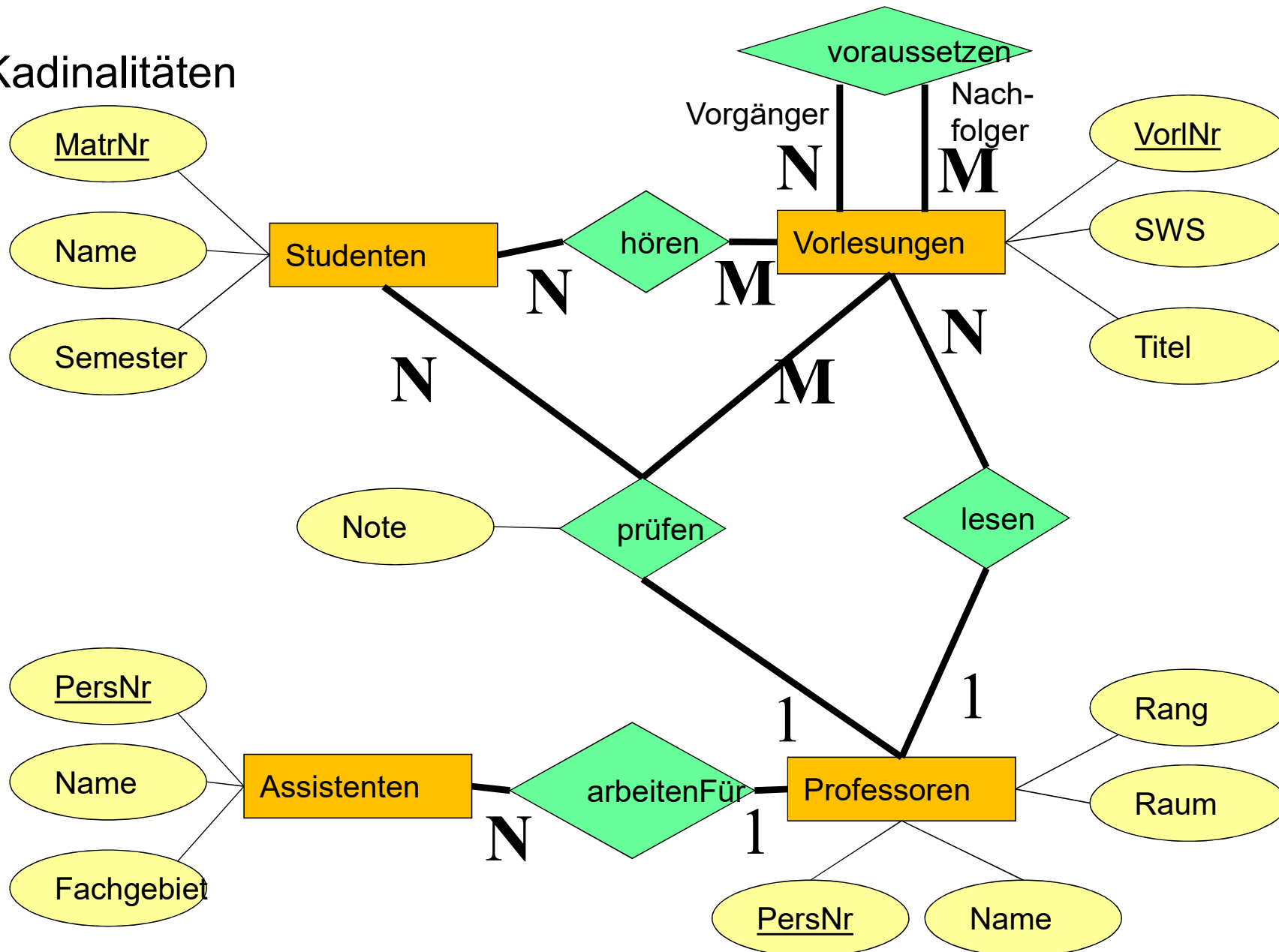
Beispiel ER-Modellierung: Hochschule

- Gestrichelte Linien markieren illegale Ausprägungen



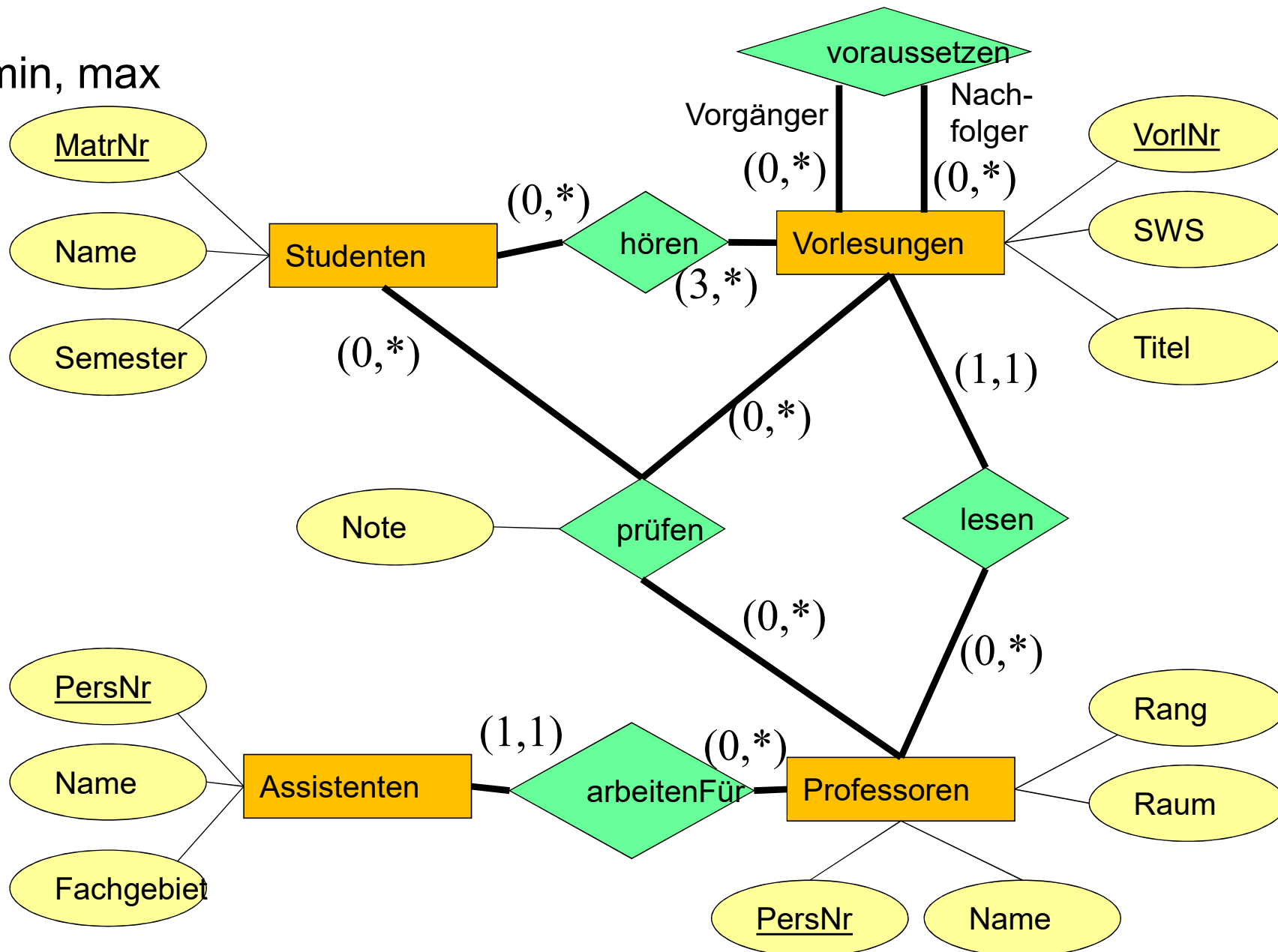
Beispiel ER-Modellierung: Hochschule

- Kardinalitäten

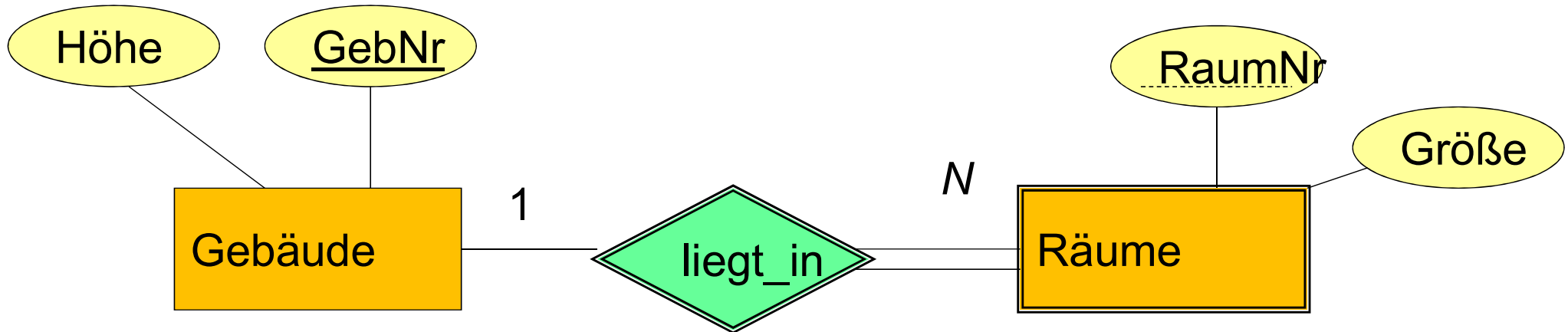


Beispiel ER-Modellierung: Hochschule

- min, max

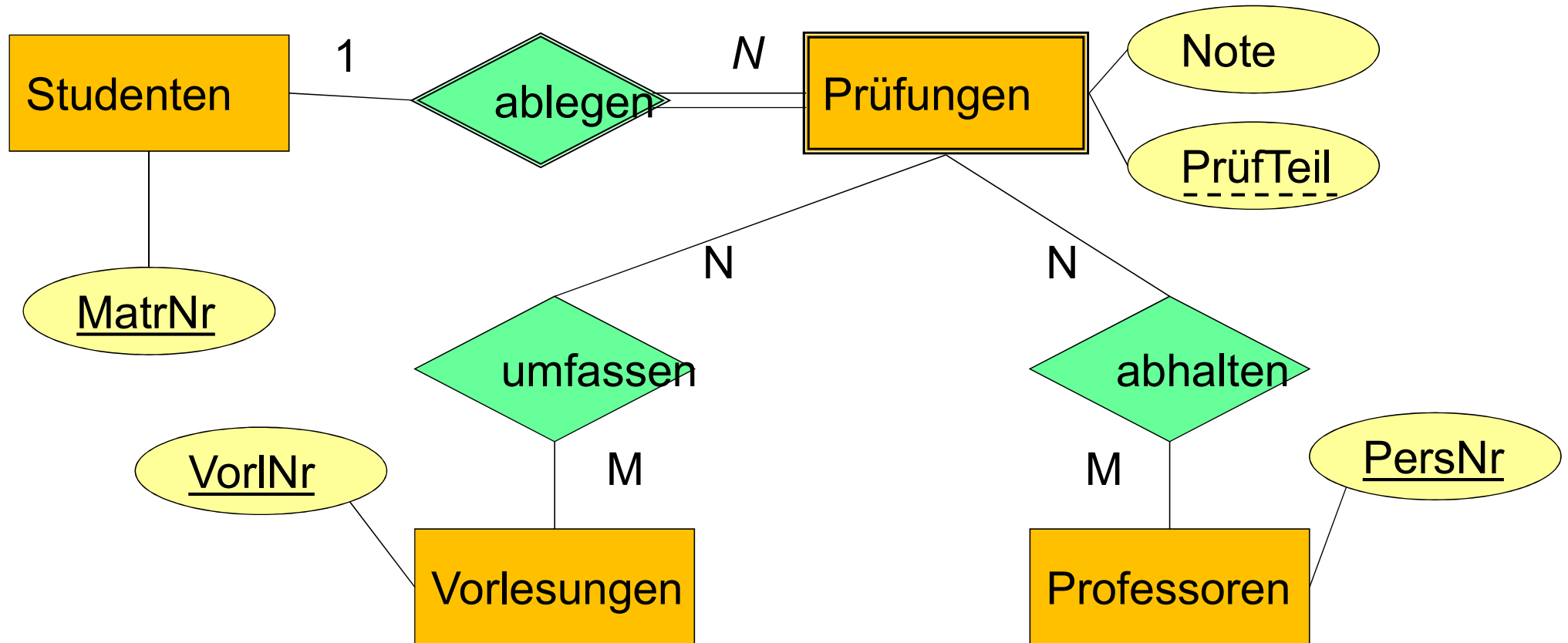


Schwache, existenzabhängige Entities



- Beziehung zwischen "starken" und schwachem Typ ist immer 1:N (oder 1:1 in seltenen Fällen)
- Warum kann das keine N:M-Beziehung sein?
- RaumNr ist nur innerhalb eines Gebäudes eindeutig
- Schlüssel ist: GebNr und RaumNr

Prüfungen als schwacher Entitytyp

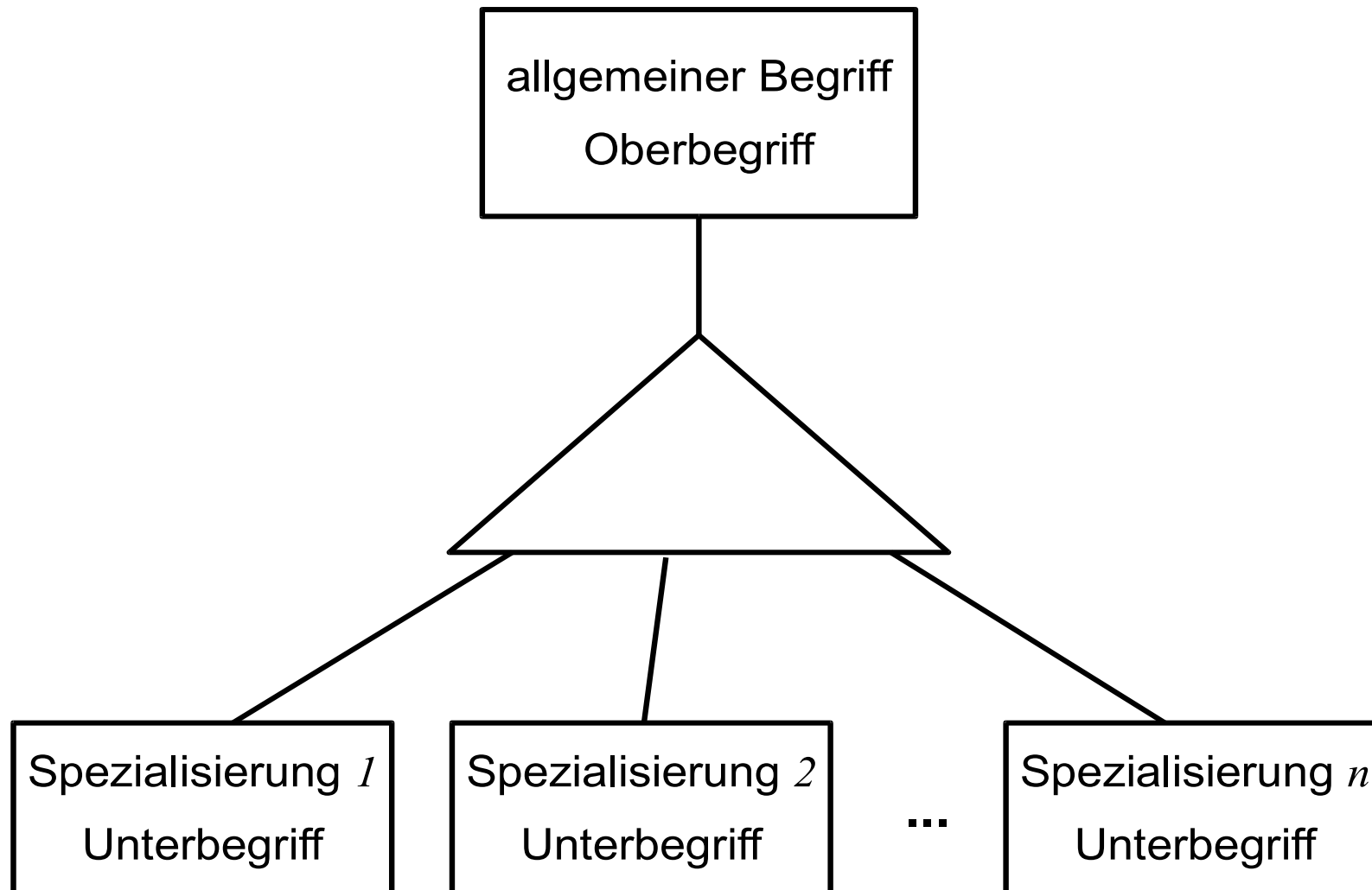


- Mehrere Prüfer in einer Prüfung
- Mehrere Vorlesungen werden in einer Prüfung abgefragt

Besondere Beziehungen: Spezialisierungen - ISA

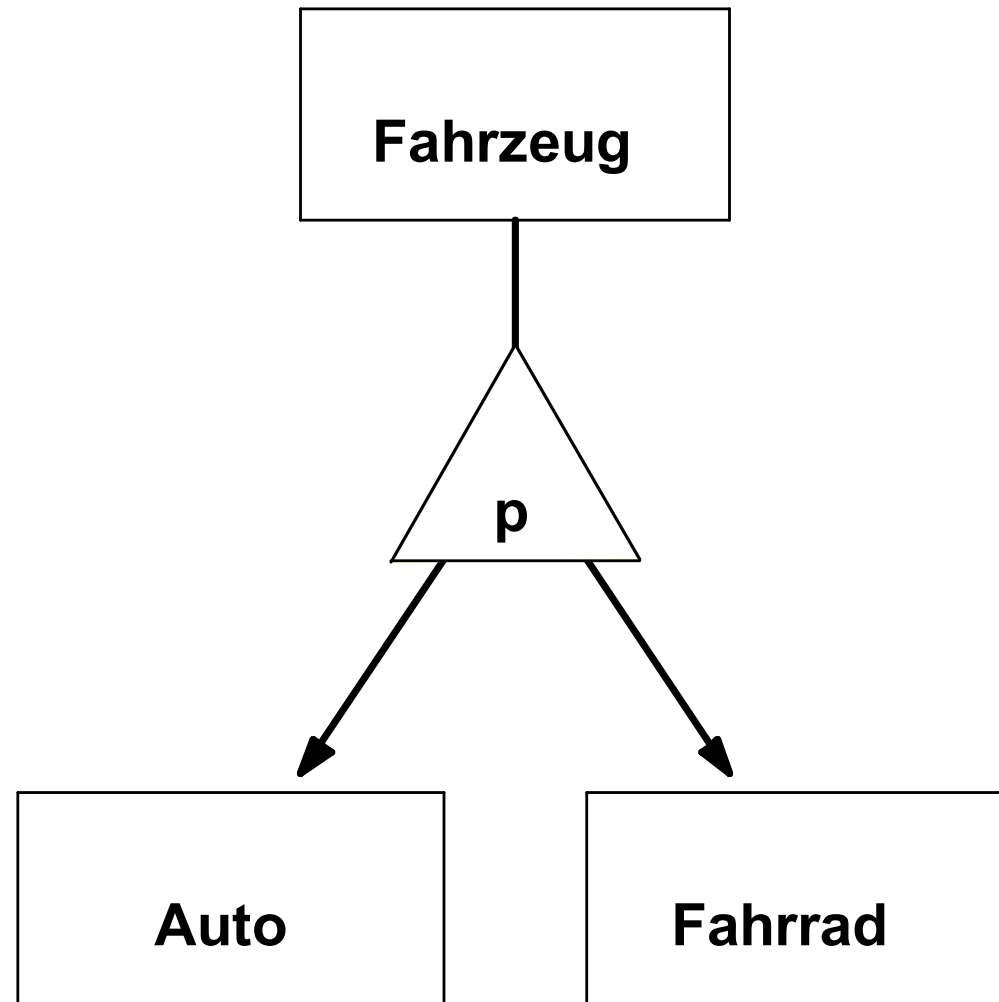
- Spezialisierungen - genauer: Spezialisierungsbeziehungen
 - haben keine Attribute
 - gelten zwischen genau zwei Entity-Typen bzw. Entity-Deklarationen
 - Notation für Entity-Typ E_s spezialisiert Entity-Typ E :
 - $E_s \text{ ISA } E$
 - $E_s \subseteq E$
- Deklaration eines Entity-Typs E_s als Spezialisierung von Entity-Typ E :
 - $E_s(A_s) \text{ ISA } E$
 - A_s ist die Menge der Attribute, die E_s zusätzlich zu den von E geerbten hat
 - E_s erbt auch den Schlüssel von E

Spezialisierungen - ISA - allgemein



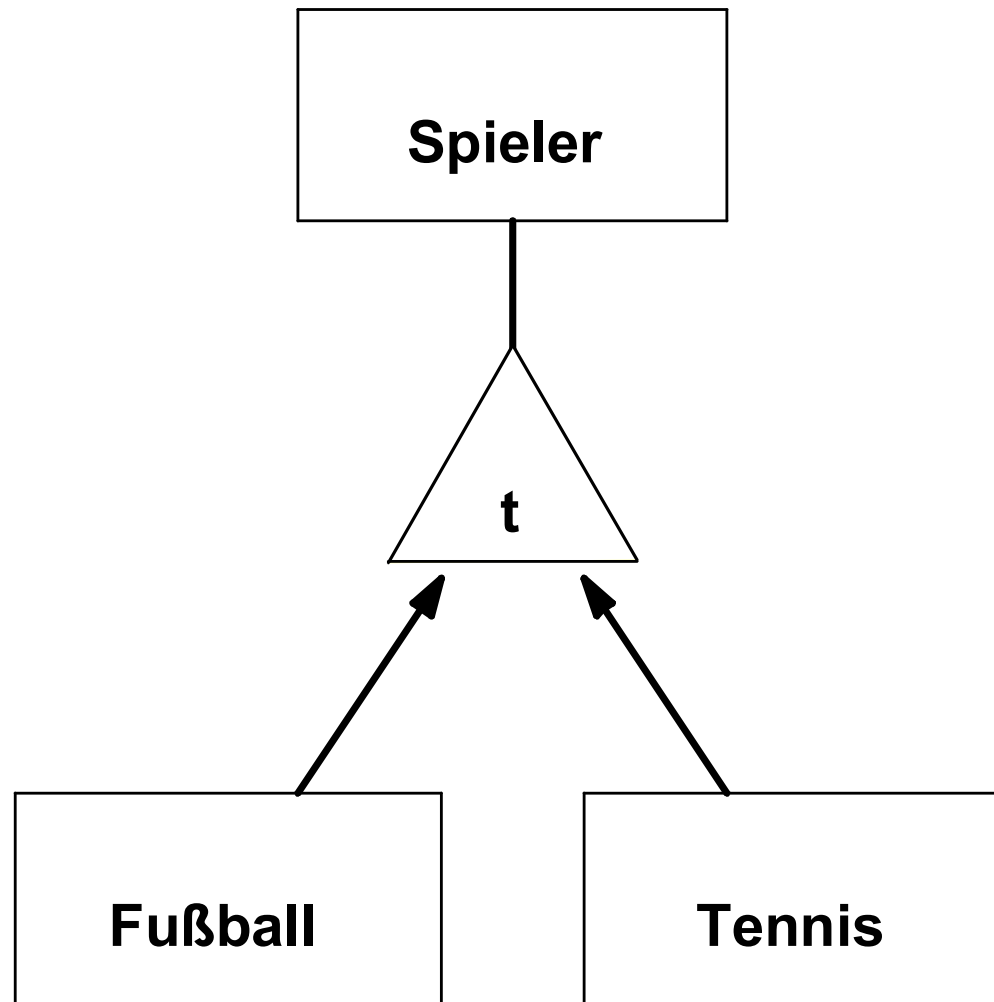
Spezialisierungen - ISA

- Disjunkte Spezialisierungen (nach Vossen)



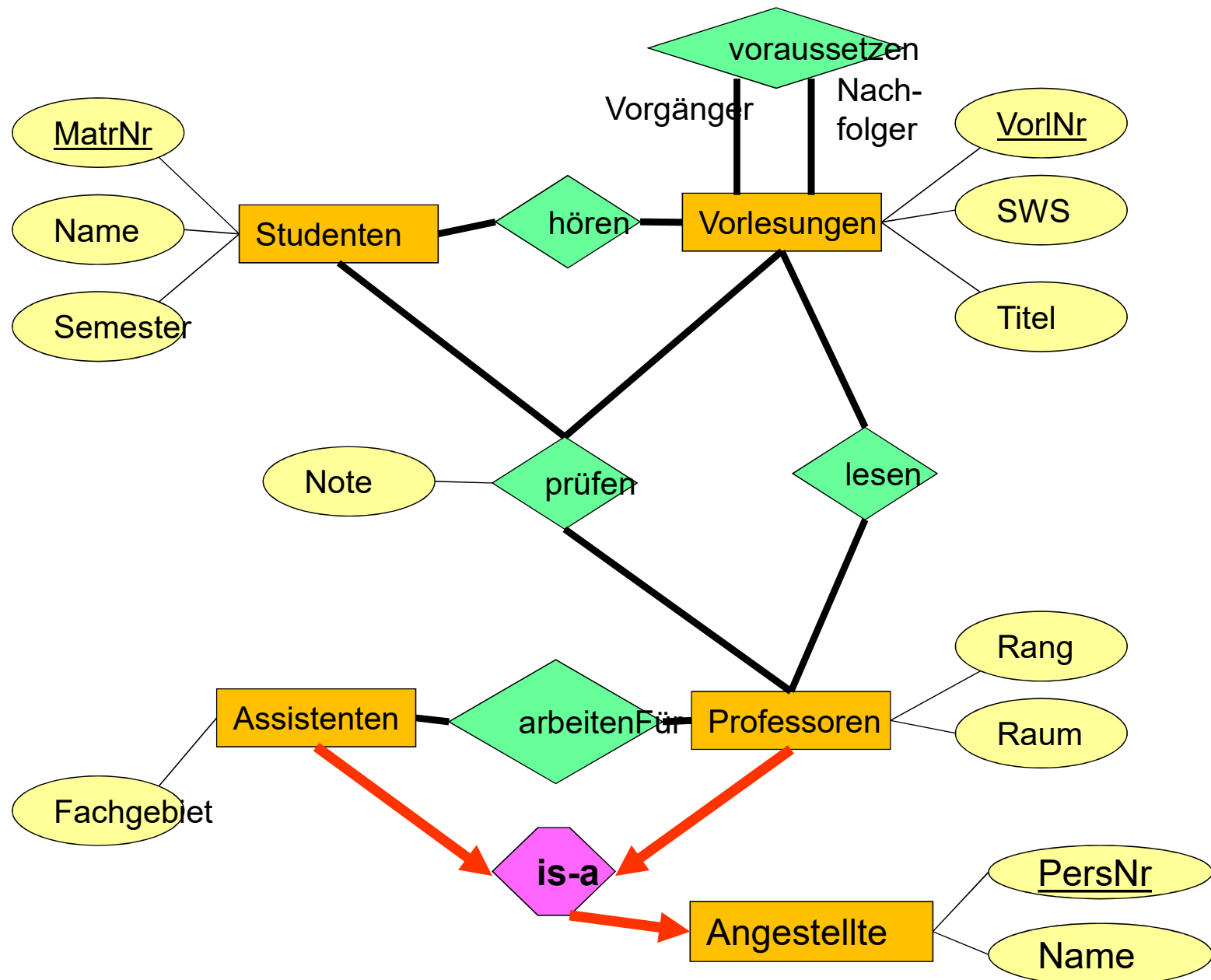
Spezialisierungen - ISA

- Nichtdisjunkte Spezialisierungen (nach Vossen)



Beispiel ER-Modellierung: Hochschule

- ISA



Komplexe Attribute (1)

- mehrwertige Attribute
 - können mehrere gleichartige Werte annehmen bzw. *Mengen als Werte* haben.
Beispiel: Fremdsprachenkenntnisse : Fremdsprache*
 - Entity-Typ Mitarbeiter =
({ Name:ZchKette, ..., Fremdsprachenkenntnisse : Fremdsprache* }, K)
 - Entity Hugo:
Hugo(Fremdsprachenkenntnisse) = { Englisch, Spanisch }
 - Im Diagramm:
 - Attribut mit doppelter Umrandung

Komplexe Attribute (2)

- zusammengesetzte Attribute
 - setzen sich aus mehreren Werten (i.a. verschiedenen Typs) zusammen.
Beispiel: Name : (Vorname, Nachname)
 - Entity-Typ Mitarbeiter =
({ Name : (Vorname : ZchKette, Nachname : ZchKette), ...}, K)
 - Im Diagramm:
 - Ein Attribut, das mit mehreren weiteren Attributen verbunden ist

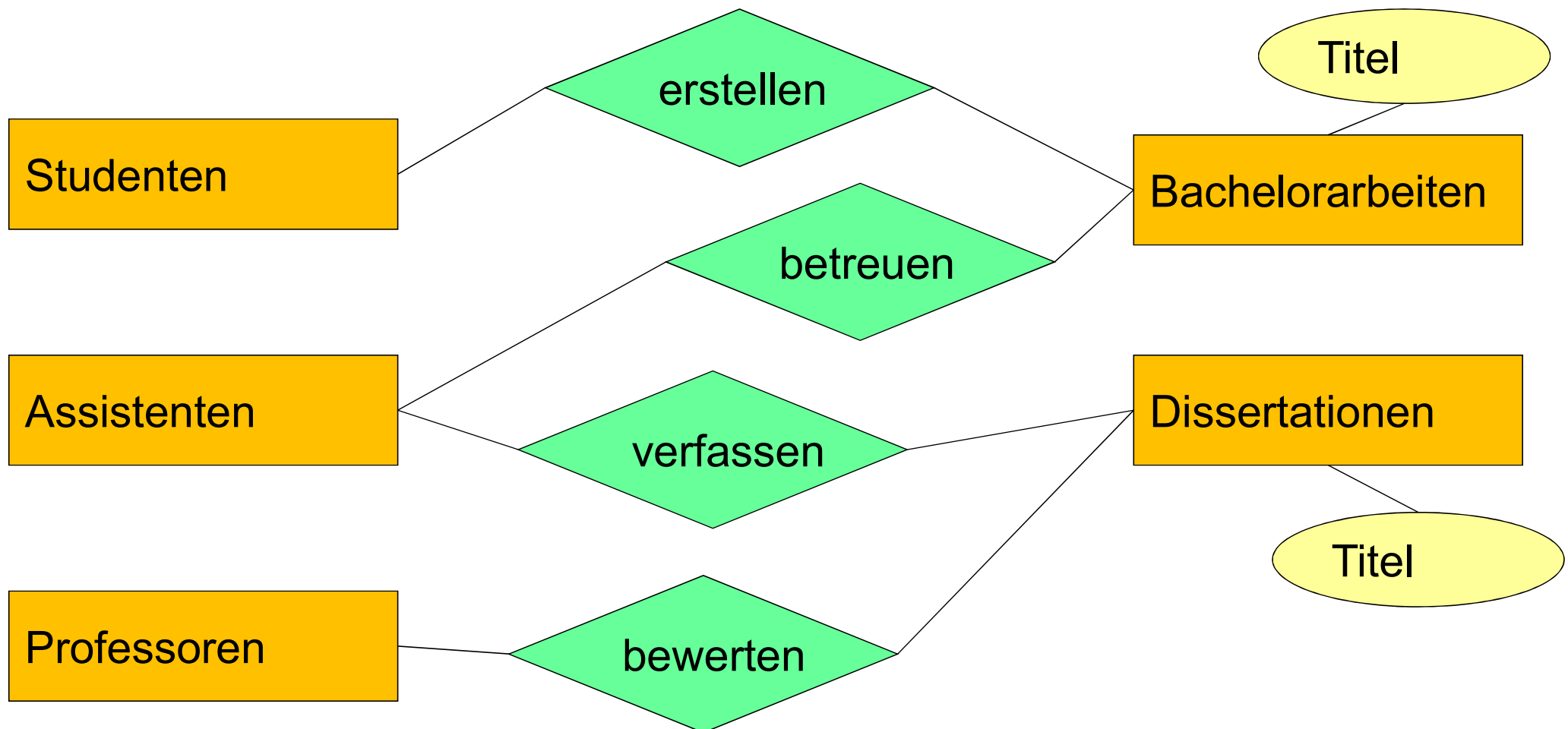
Vorgehen beim Entwurf des konzeptuellen Modells

1. Bestimmung der Entity-Typen
 - Aussagen zu:
 - Name
 - Bedeutung
 - Attribute
 - Wertebereiche und Bedeutung der Attribute
2. Ermittlung welche Beziehungen zwischen den Entity-Typen
 - beobachtbar sind
 - vorstellbar sind
 - durch die vorhandene Verarbeitung gegeben sind
 - aus der geplanten Verarbeitung resultieren werden
3. Ermittlung der Kardinalitäten

Die Entscheidung was Entity(-Typ) und was Beziehung(s-Typ) ist, hängt von der Sichtweise auf das Problem ab!

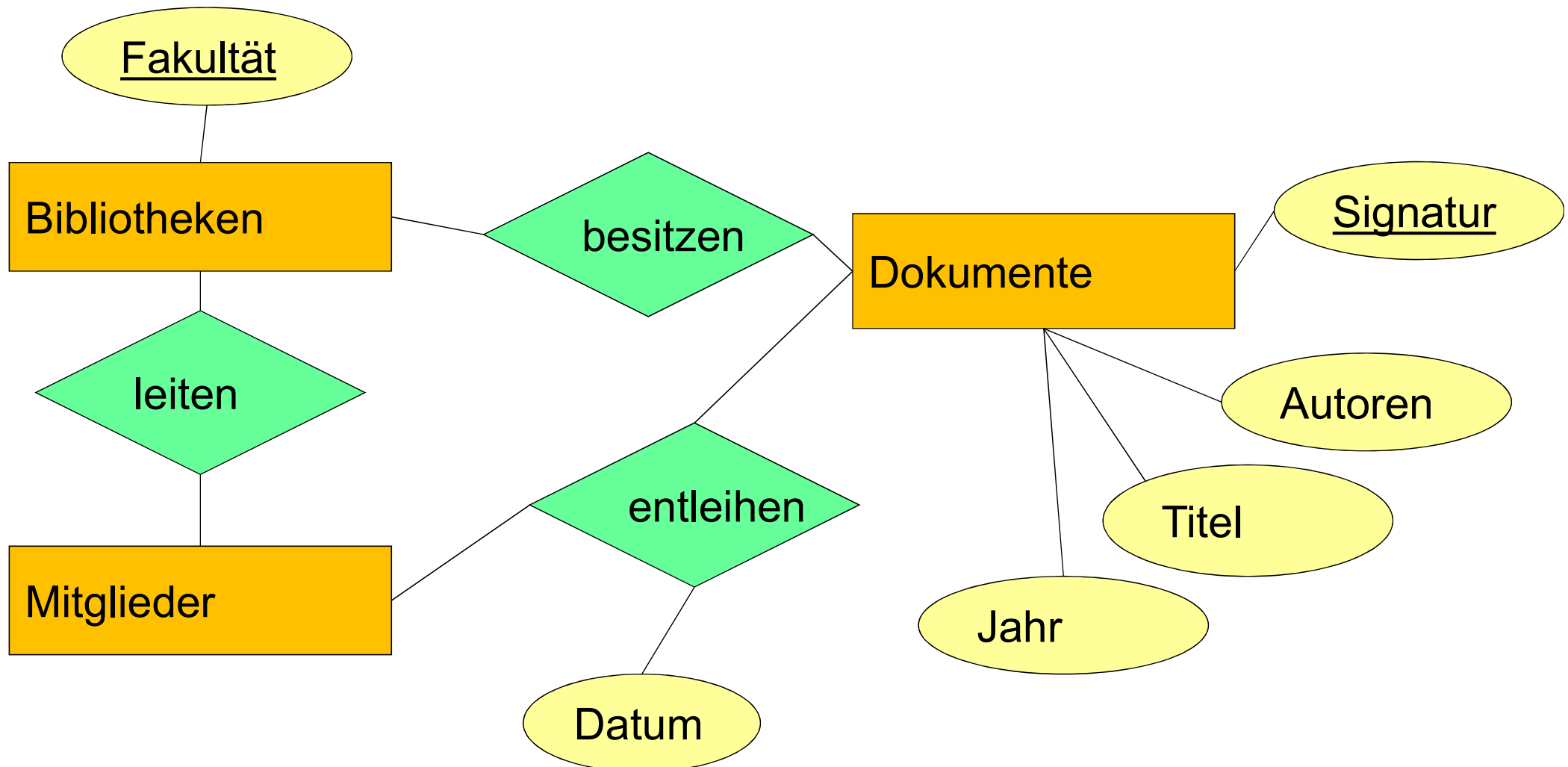
Beispiel ER-Modellierung: Hochschule

- Drei Sichten einer Universitäts-Datenbank
- Sicht 1: Erstellung von Dokumenten als Prüfungsleistung



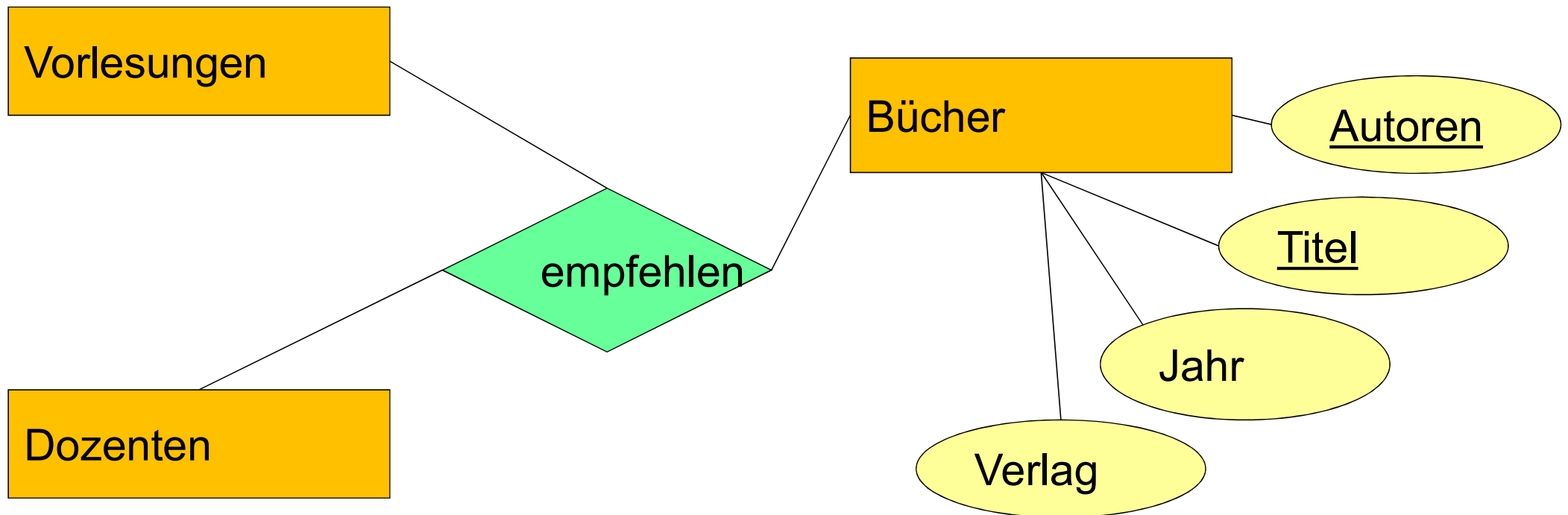
Beispiel ER-Modellierung: Hochschule

- Drei Sichten einer Universitäts-Datenbank
- Sicht 2: Bibliotheksverwaltung



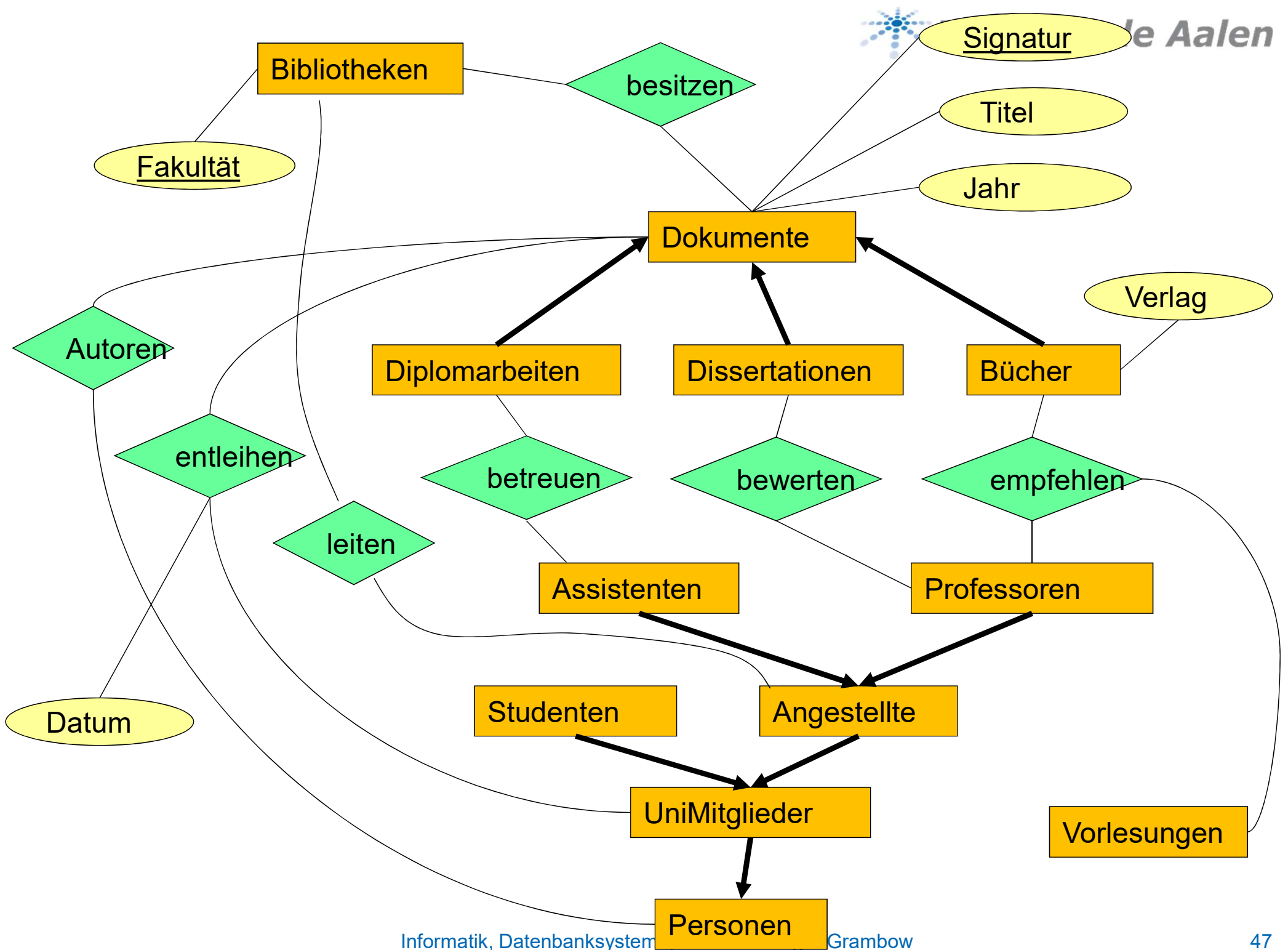
Beispiel ER-Modellierung: Hochschule

- Drei Sichten einer Universitäts-Datenbank
- Sicht 3: Buchempfehlungen für Vorlesungen



Beispiel ER-Modellierung: Hochschule

- Beobachtungen:
 - Die Begriffe Dozenten und Professoren sind synonym verwendet worden.
 - Der Entitytyp Mitglieder ist eine Generalisierung von Studenten, Professoren und Assistenten.
 - Fakultätsbibliotheken werden sicherlich von Angestellten (und nicht von Studenten) geleitet. Insofern ist die in Sicht 2 festgelegte Beziehung leiten revisionsbedürftig, sobald wir im globalen Schema ohnehin eine Spezialisierung von Mitglieder in Studenten und Angestellte vornehmen.
 - Dissertationen, Diplomarbeiten und Bücher sind Spezialisierungen von Dokumenten, die in den Bibliotheken verwaltet werden.
 - Wir können davon ausgehen, dass alle an der Universität erstellten Diplomarbeiten und Dissertationen in Bibliotheken verwaltet werden.
 - Die in Sicht 1 festgelegten Beziehungen erstellen und verfassen modellieren denselben Sachverhalt wie das Attribut Autoren von Büchern in Sicht 3.
 - Alle in einer Bibliothek verwalteten Dokumente werden durch die Signatur identifiziert.



Zusammenfassung

- was ist ein Entity?
- was ist eine Beziehung?
- ER Diagramm
- Kardinalitäten
- Komplexe Attribute: mehrwertige und zusammengesetzte
- Schwache Entities
- Vorgehen beim Entwurf