

Datenbankentwurf

Abstraktionsebenen des Datenbankentwurfs:

1. Konzeptuelle Ebene
2. Implementationsebene
3. Physische Ebene



Kapitel 2

Datenbankentwurf

Abstraktionsebenen des Datenbankentwurfs

1. Konzeptuelle Ebene
1. Implementationsebene
1. Physische Ebene

Objektbeschreibung

● Uni-Angestellte

- Anzahl: 1000
- Attribute

❖ PersonalNummer

- Typ: char
- Länge: 9
- Wertebereich:
0...999.999.99
- Anzahl
Wiederholungen: 0
- Definiertheit: 100%
- Identifizierend: ja

❖ Gehalt

- Typ: dezimal
- Länge: (8,2)
- Anzahl Wiederholung: 0
- Definiertheit: 10%
- Identifizierend: nein

❖ Rang

- Typ: String
- Länge: 4
- Anzahl Wiederholung: 0
- Definiertheit: 100%
- Identifizierend: nein

Beziehungsbeschreibung: *prüfen*

- Beteiligte Objekte:
 - Professor als Prüfer
 - Student als Prüfling
 - Vorlesung als Prüfungsstoff
- Attribute der Beziehung:
 - Datum
 - Uhrzeit
 - Note
- Anzahl: 100 000 pro Jahr

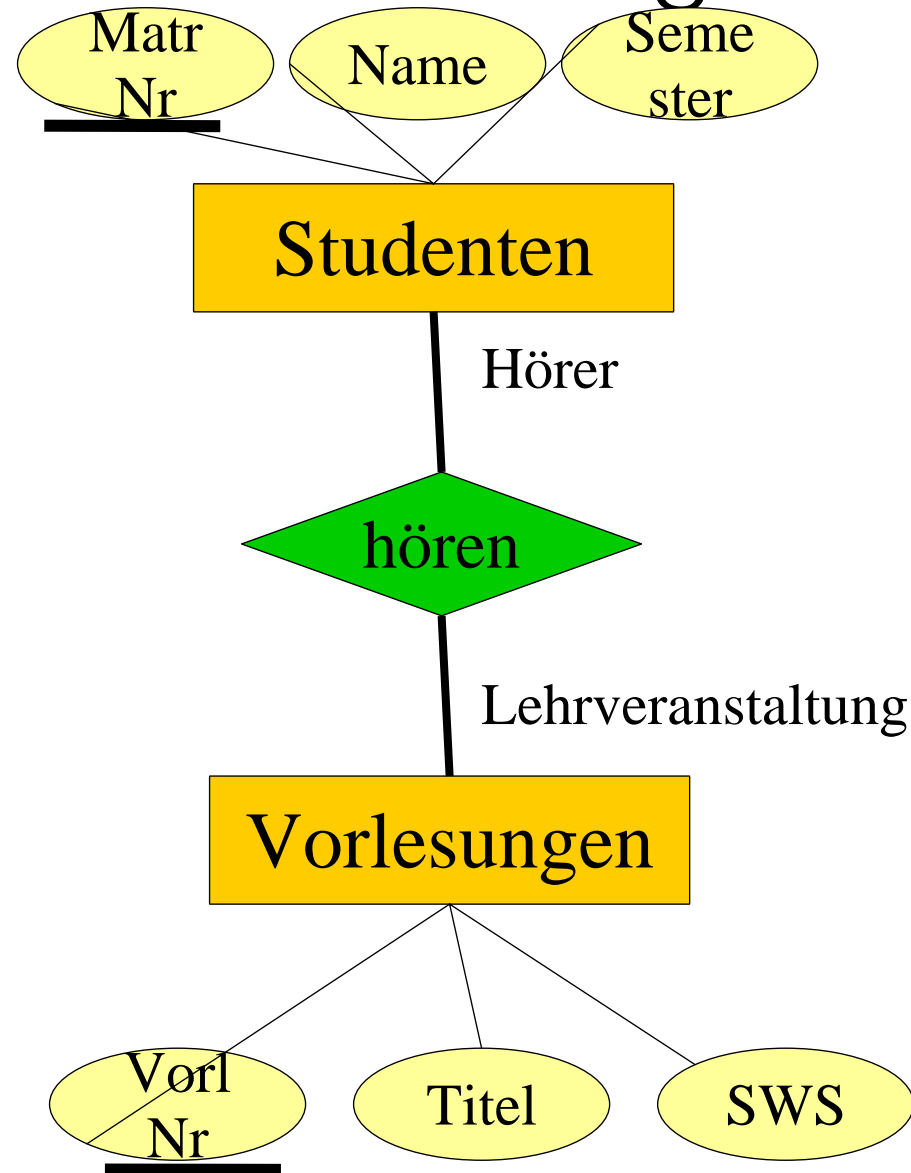
Prozeßbeschreibungen

● **Prozeßbeschreibung:** *Zeugnisausstellung*

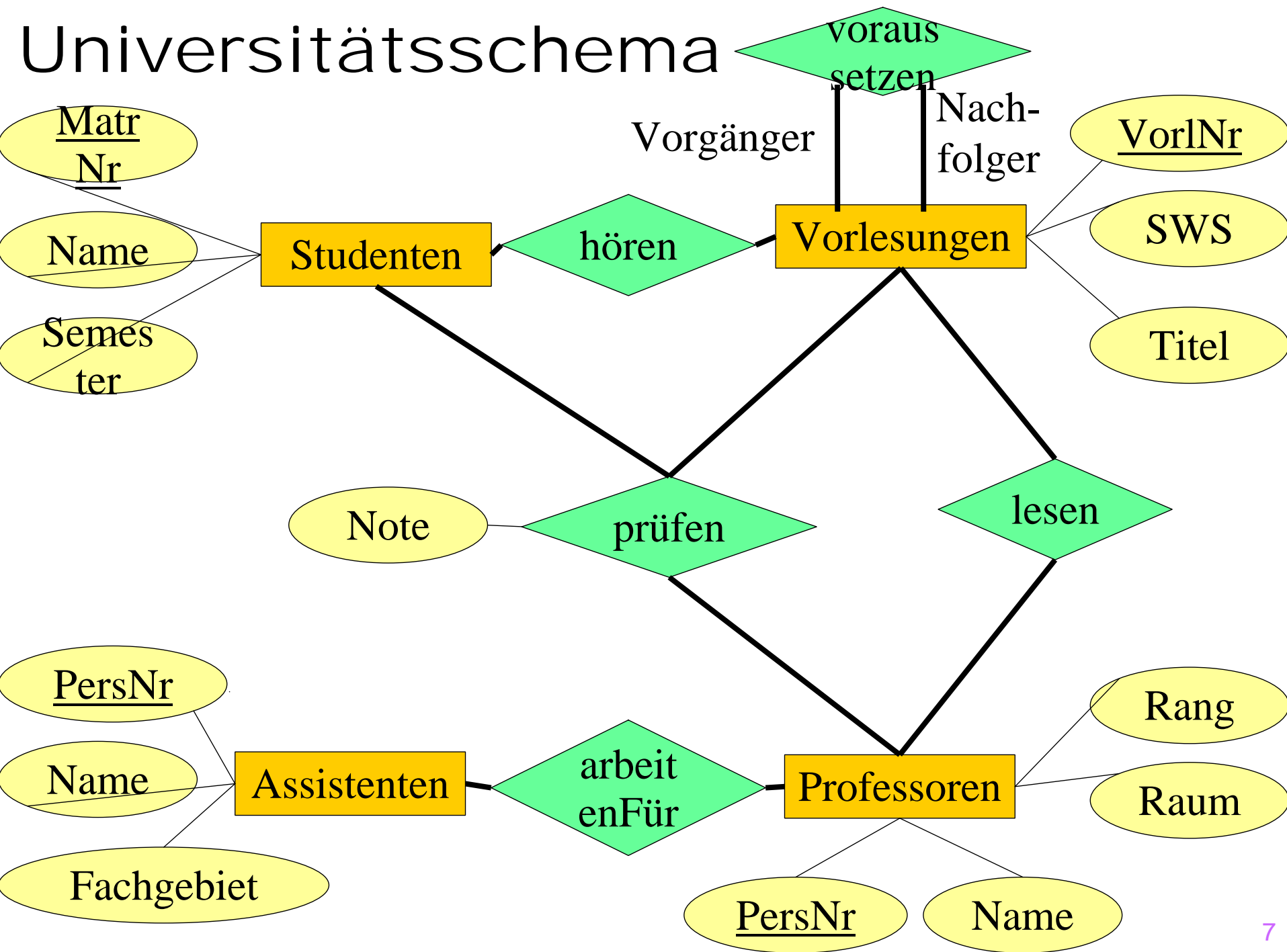
- Häufigkeit: halbjährlich
- benötigte Daten
 - * Prüfungen
 - * Studienordnungen
 - * Studenteninformation
 - * ...
- Priorität: hoch
- Zu verarbeitende Datenmenge
 - * 500 Studenten
 - * 3000 Prüfungen
 - * 10 Studienordnungen

Entity/Relationship-Modellierung

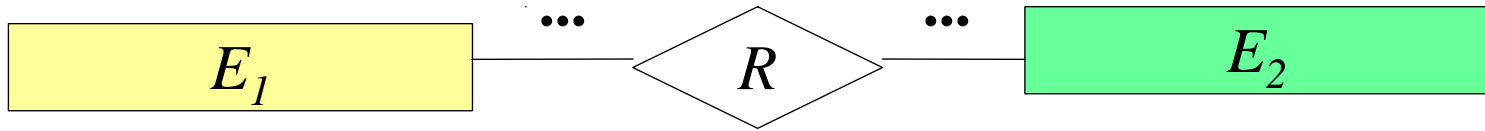
- Entity (Gegenstandstyp)
- Relationship (Beziehungstyp)
- Attribut (Eigenschaft)
- Schlüssel (Identifikation)
- Rolle



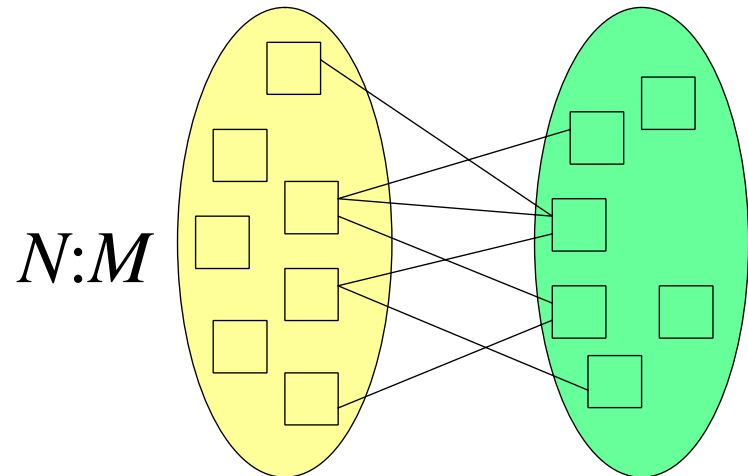
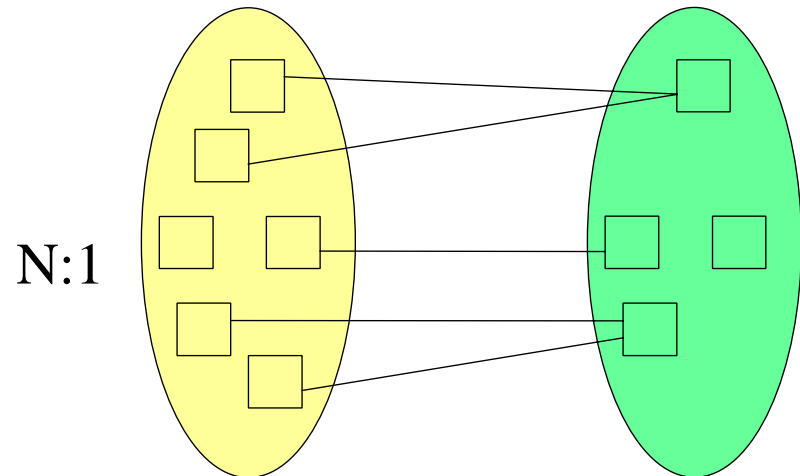
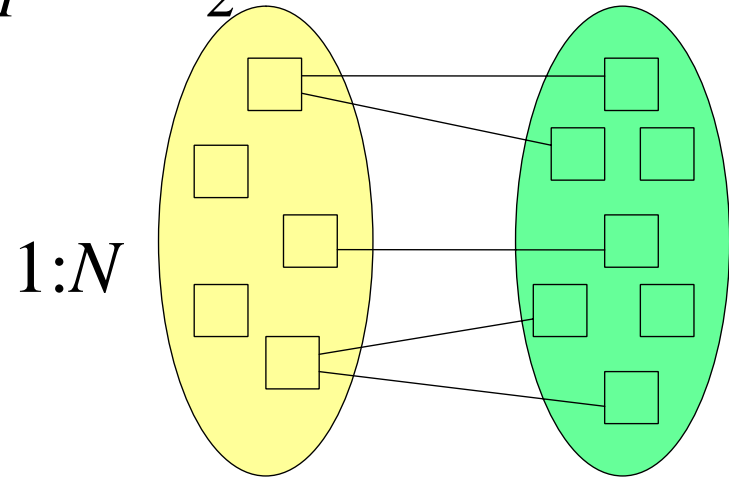
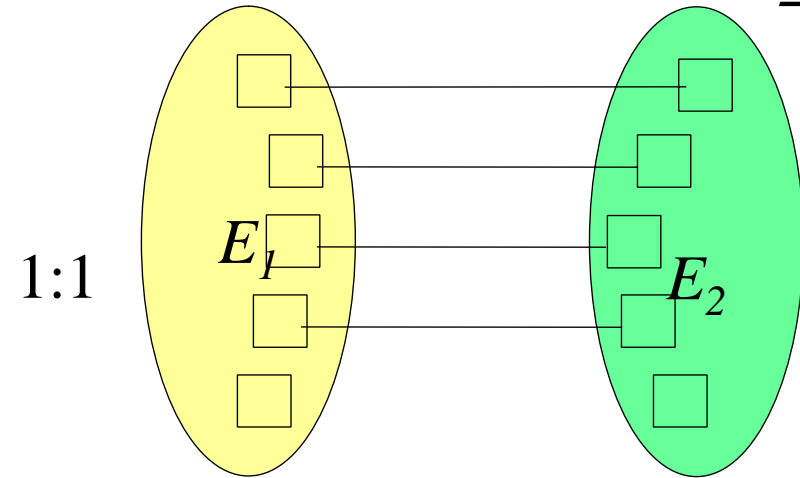
Universitätsschema



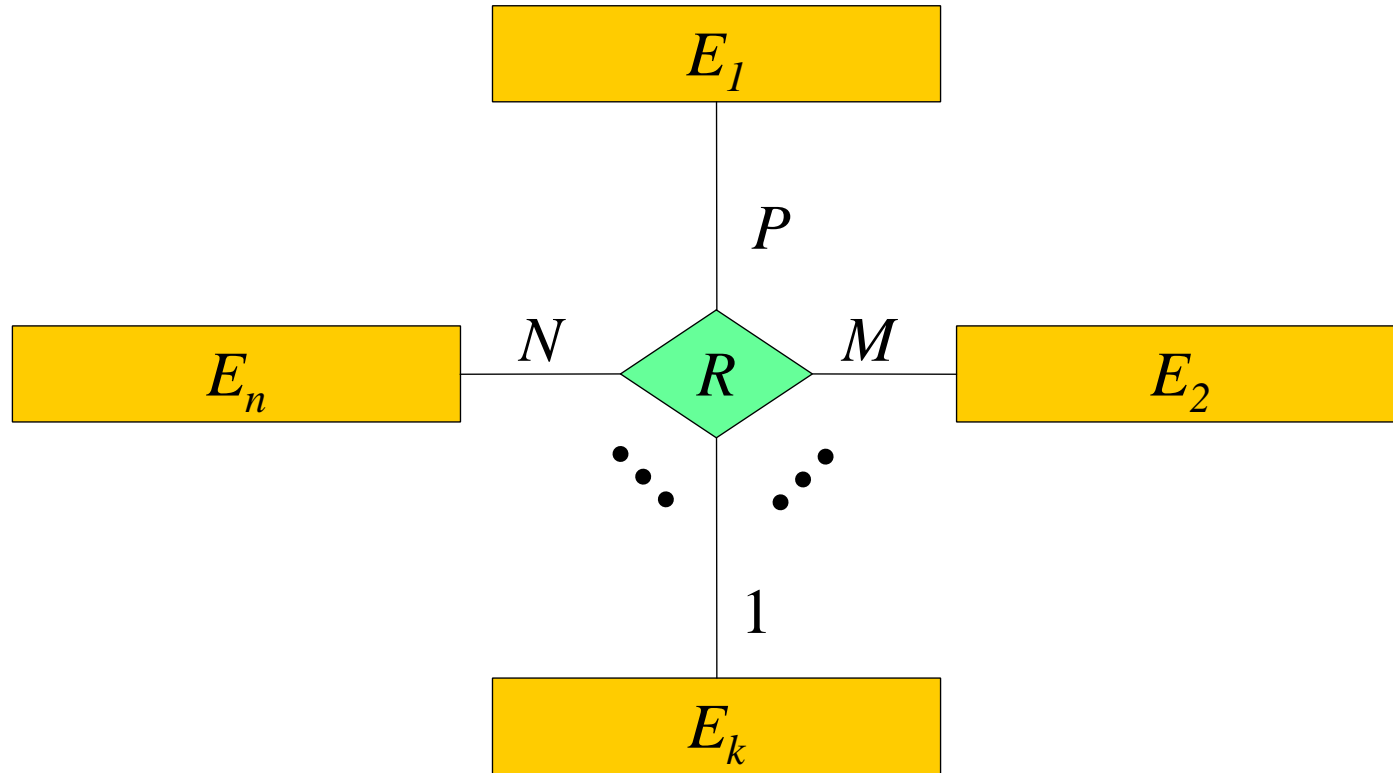
Funktionalitäten



$$R \subseteq E_1 \times E_2$$

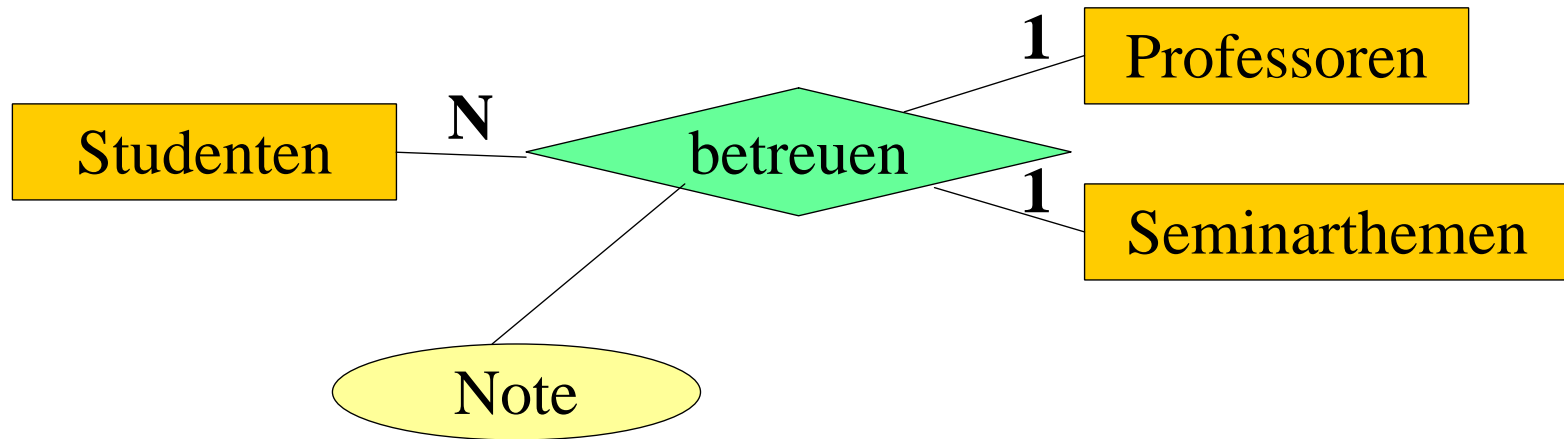


Funktionalitäten bei n -stelligen Beziehungen



$$R : E_1 \times \dots \times E_{k-1} \times E_{k+1} \times \dots \times E_n \rightarrow E_k$$

Beispiel-Beziehung: *betreuen*



$\text{betreuen} : \text{Professoren} \times \text{Studenten} \rightarrow \text{Seminarthemen}$

$\text{betreuen} : \text{Seminarthemen} \times \text{Studenten} \rightarrow \text{Professoren}$

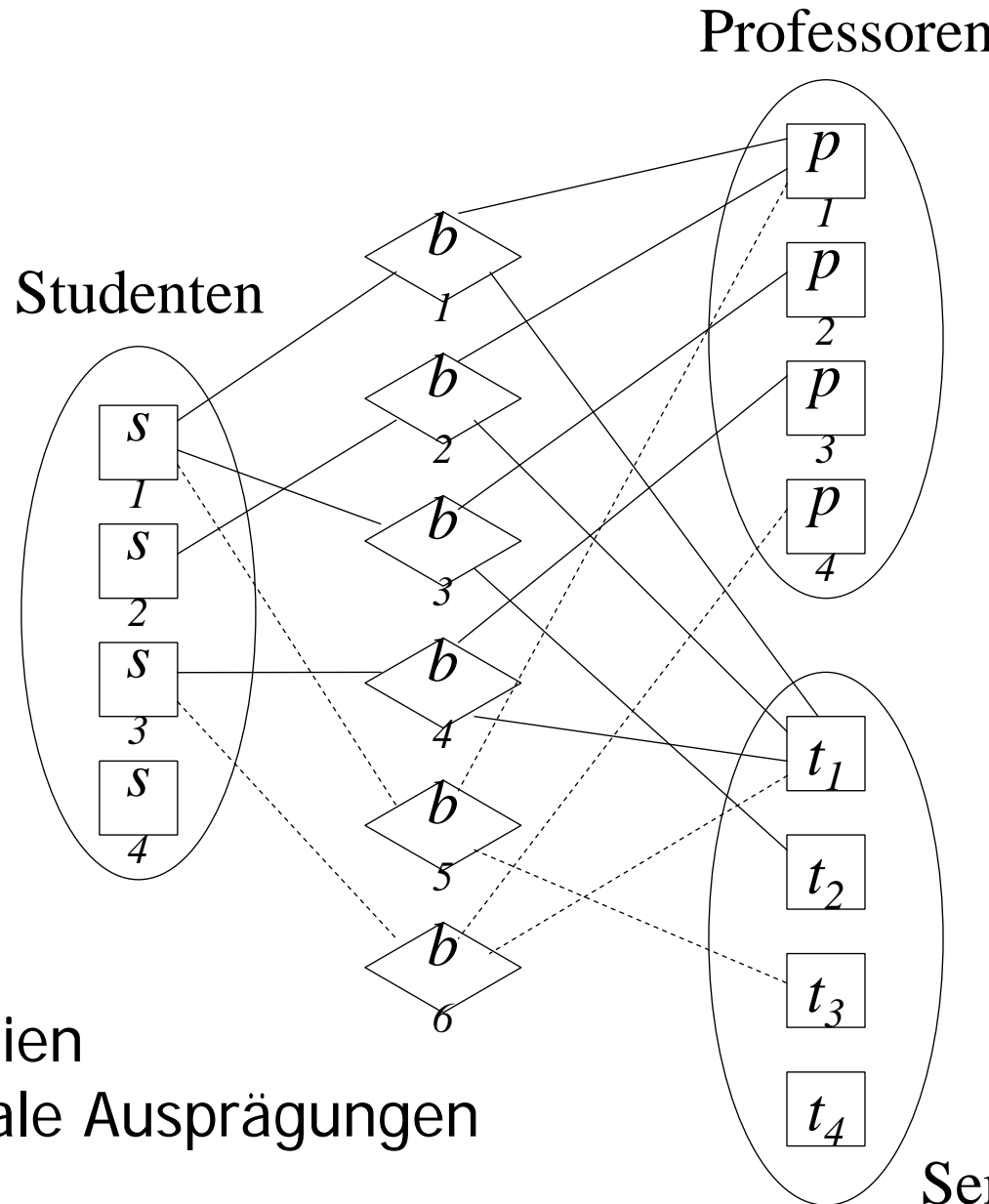
Dadurch erzwungene Konsistenzbedingungen

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

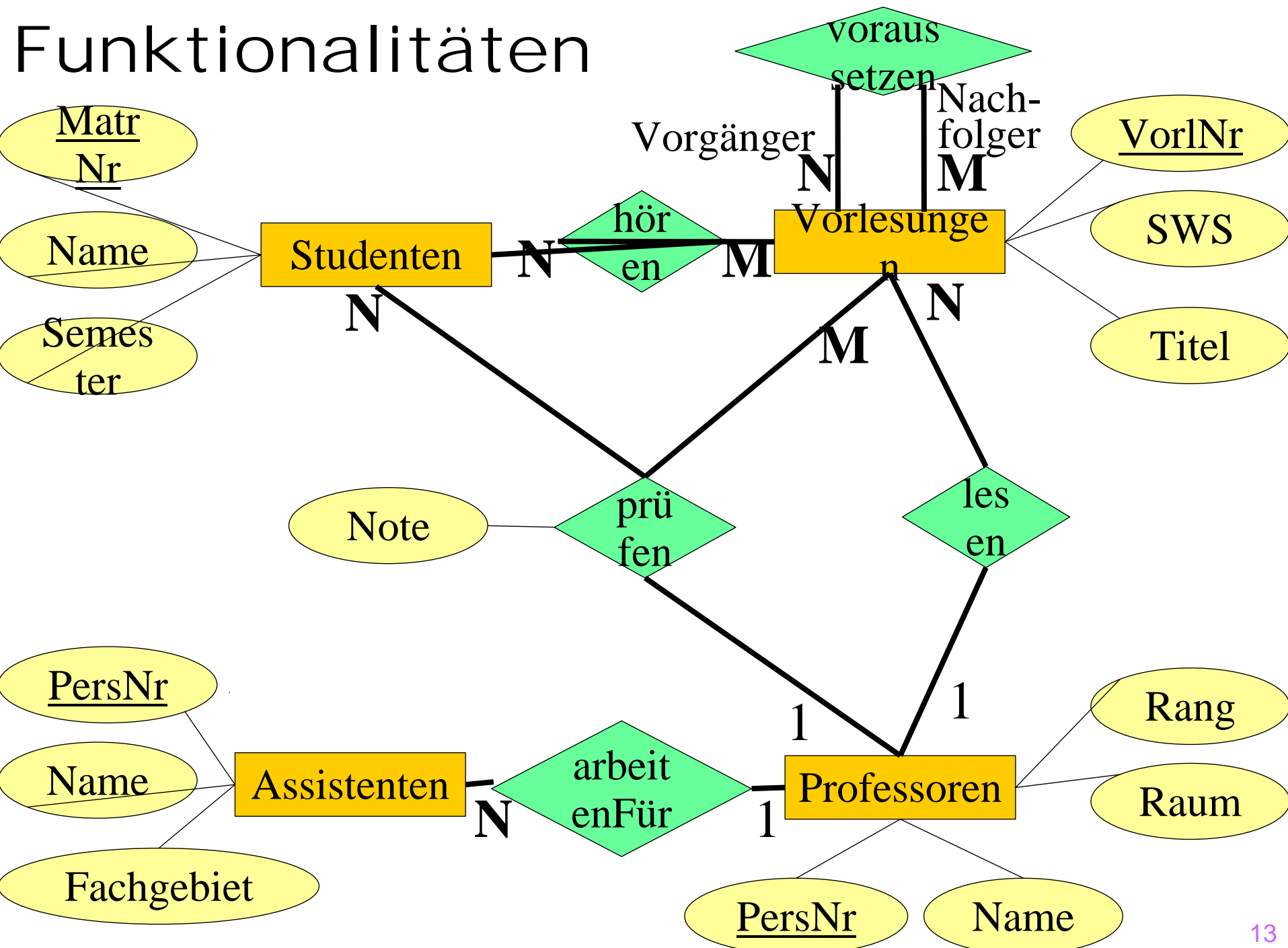
Es sind aber folgende Datenbankzustände nach wie vor möglich:

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

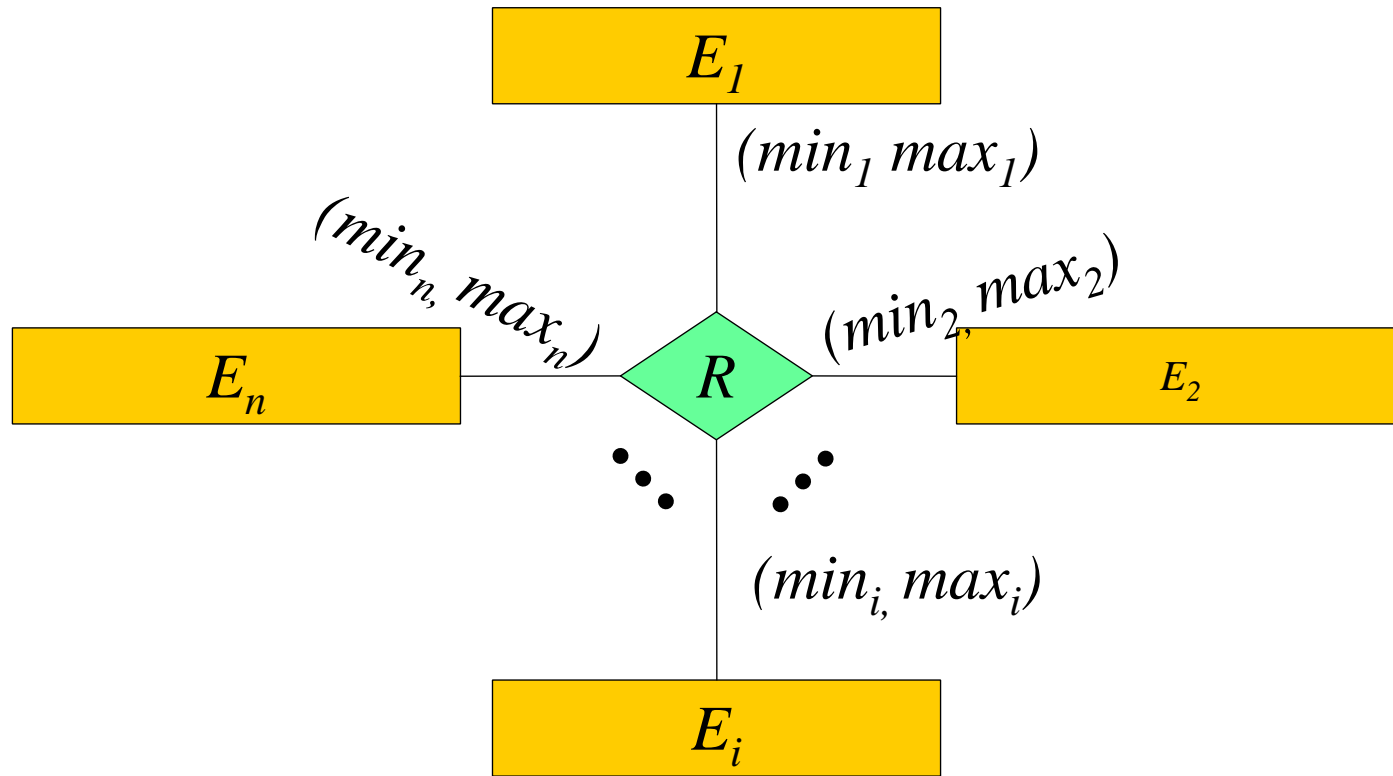
Ausprägung der Beziehung *betreuen*



Funktionalitäten



(min, max)-Notation

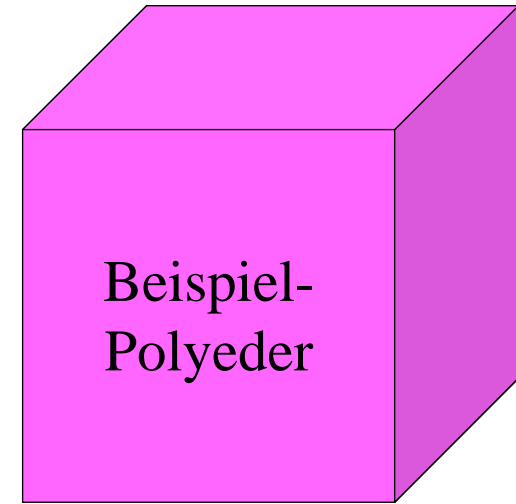
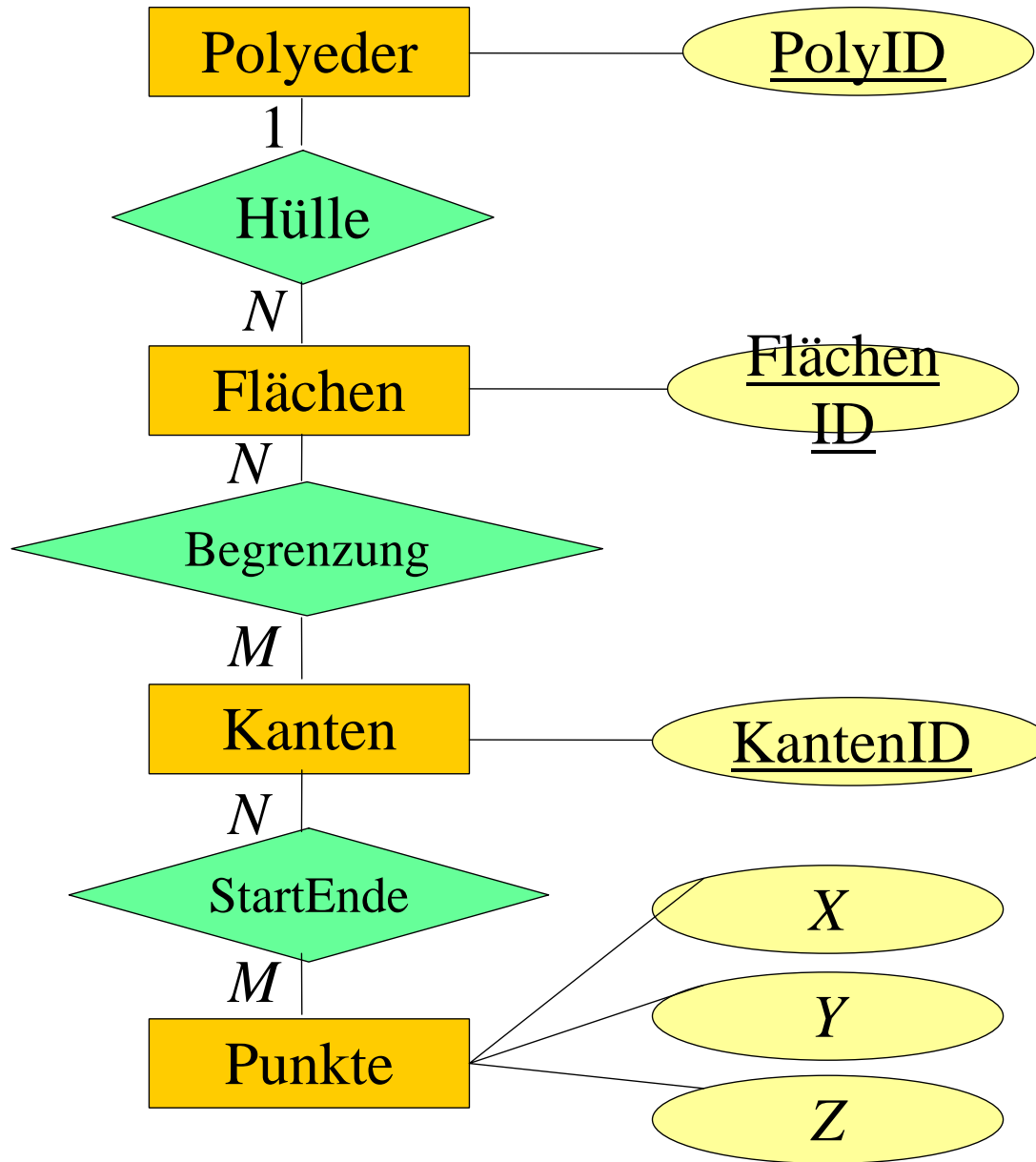


$$R \subseteq E_1 \times \dots \times E_i \times \dots \times E_n$$

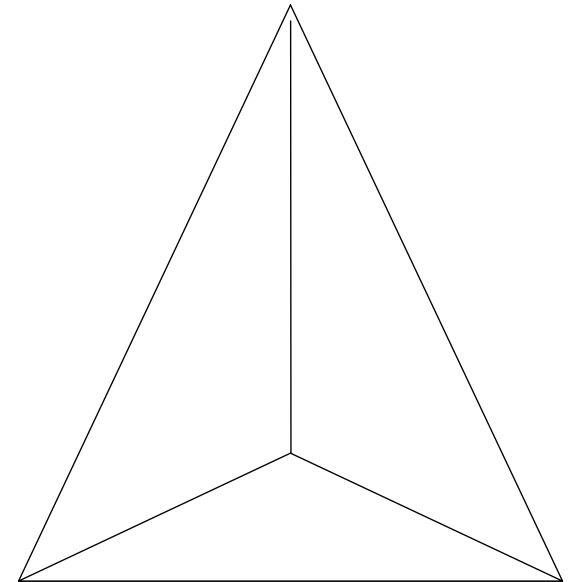
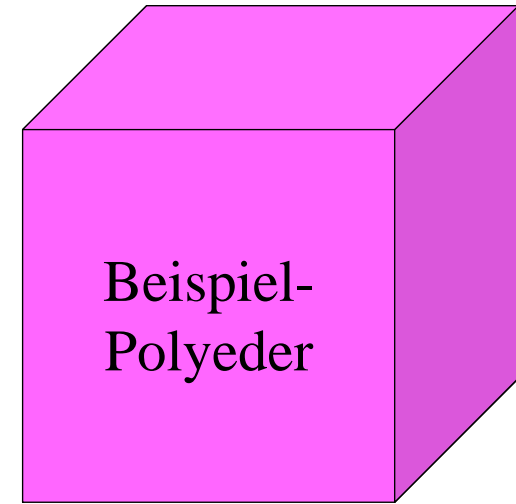
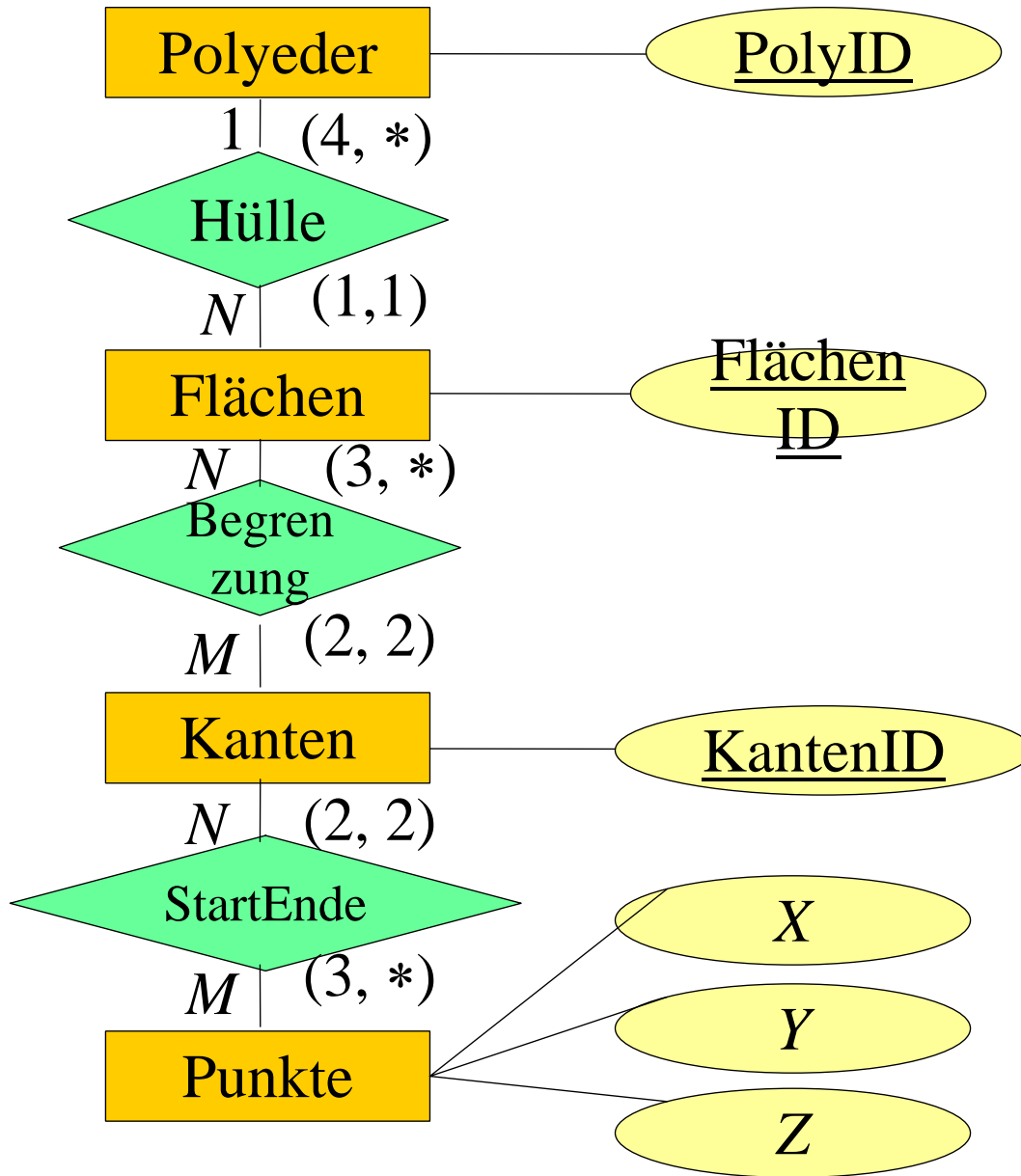
Für jedes $e_j \in E_j$ gibt es

- Mindestens min_j Tupel der Art (\dots, e_j, \dots) und
- Höchstens max_j viele Tupel der Art $(\dots, e_j, \dots) \in R$

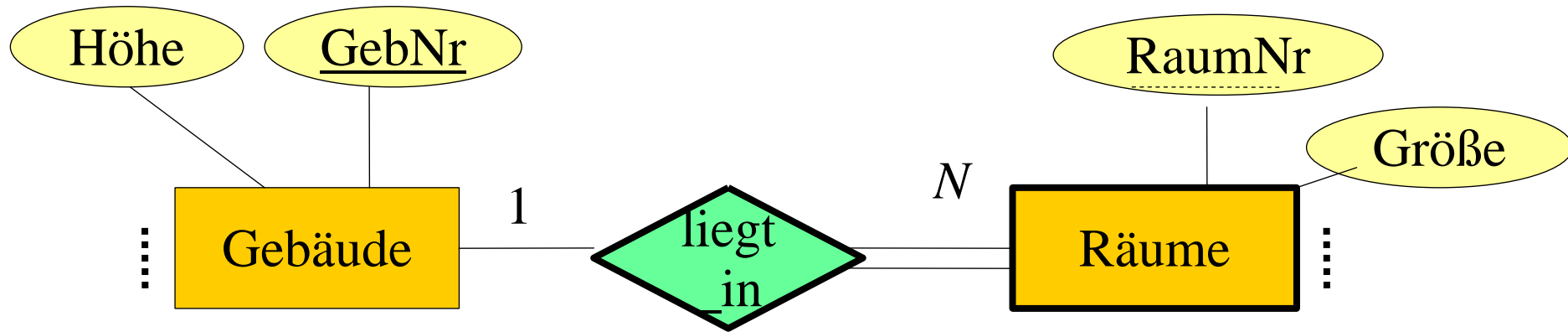
Begrenzungsflächendarstellung



Begrenzungsflächendarstellung

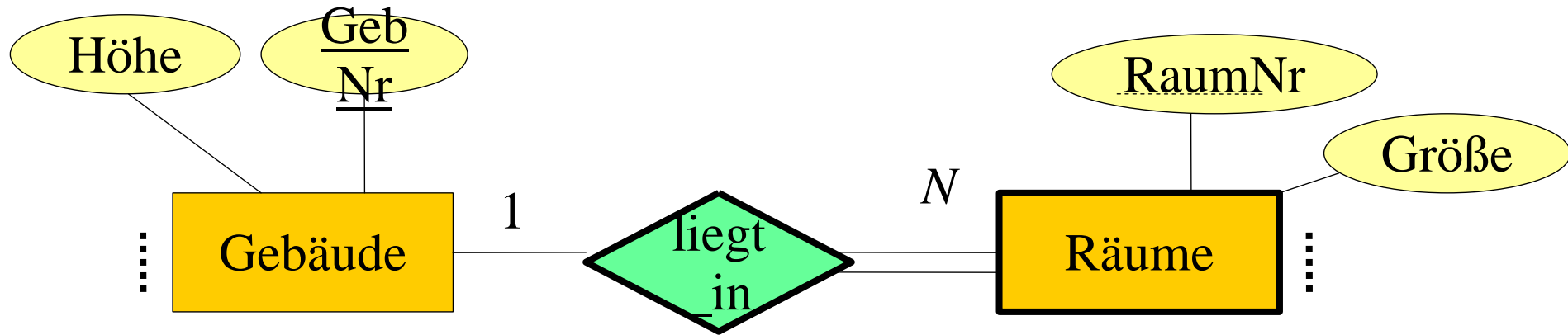


Schwache, existenzabhängige Entities



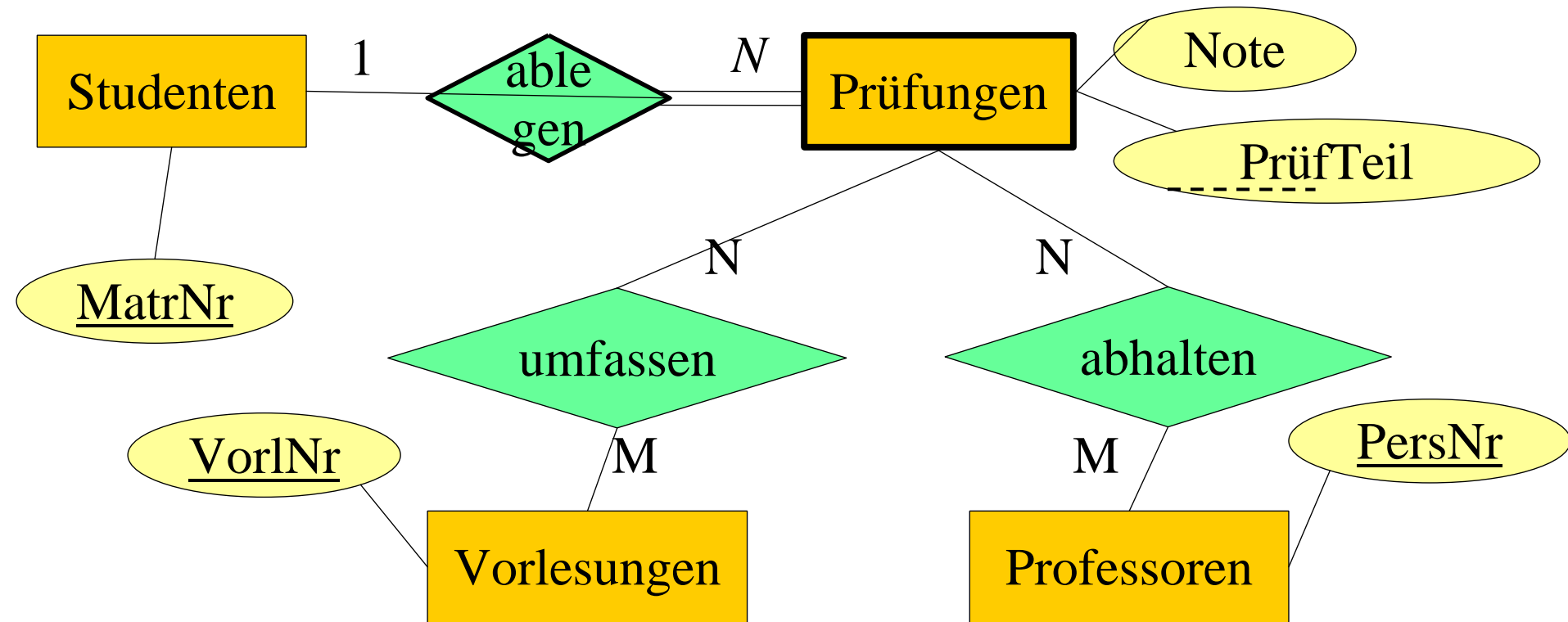
- Schwache Entitäten sind Entitäten, die von einer anderen, übergeordneten Entität abhängig sind.
- Sie sind oft nur in Kombination mit dem Schlüssel der übergeordneten Entität eindeutig identifizierbar.

Schwache, existenzabhängige Entities



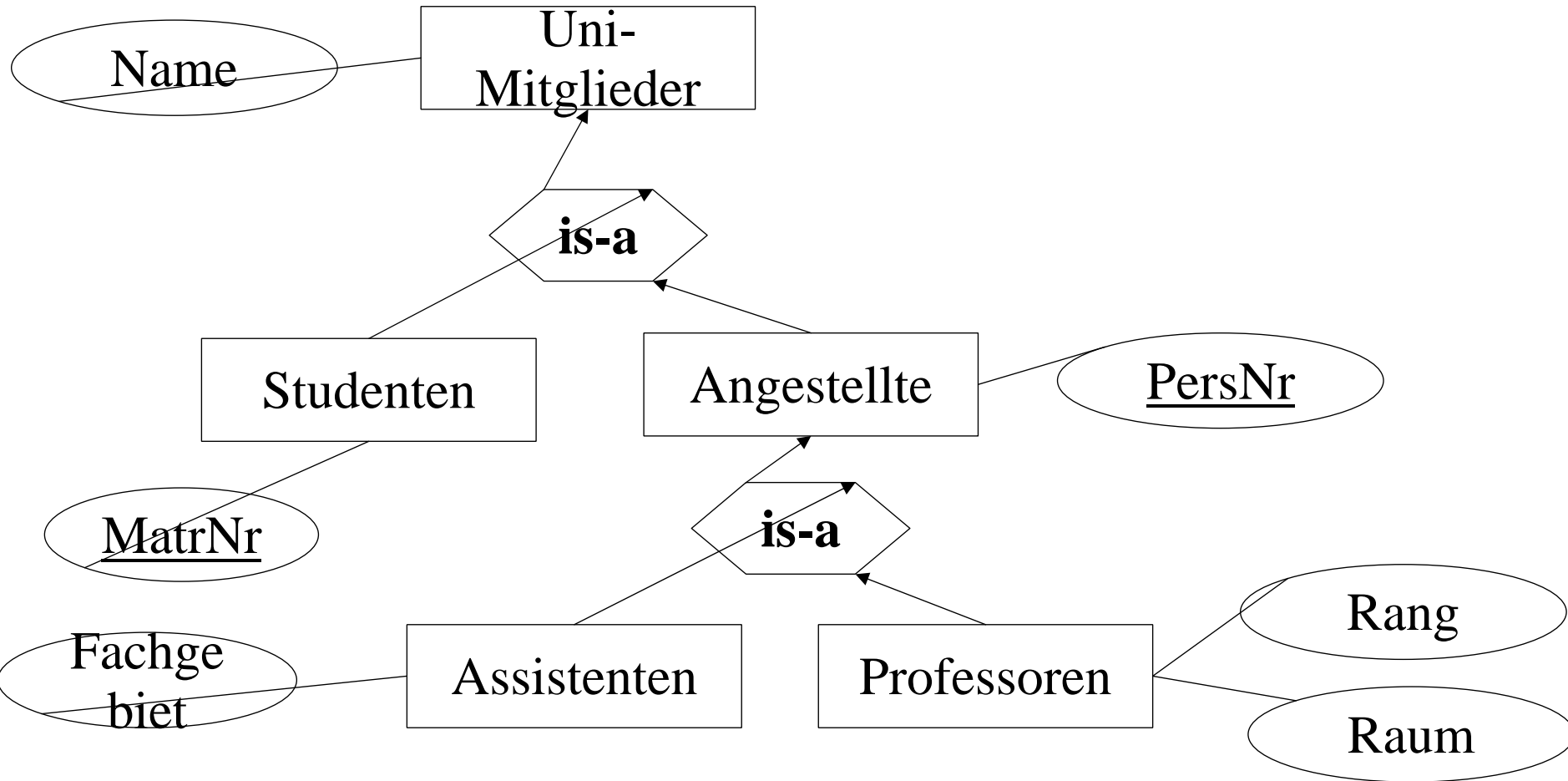
- Beziehung zwischen "starken" und schwachem Typ ist immer 1:N (oder 1:1 in seltenen Fällen)
- Warum kann das keine N:M-Beziehung sein?
- RaumNr ist nur innerhalb eines Gebäudes eindeutig
- Schlüssel ist: GebNr **und** RaumNr

Prüfungen als schwacher Entitytyp



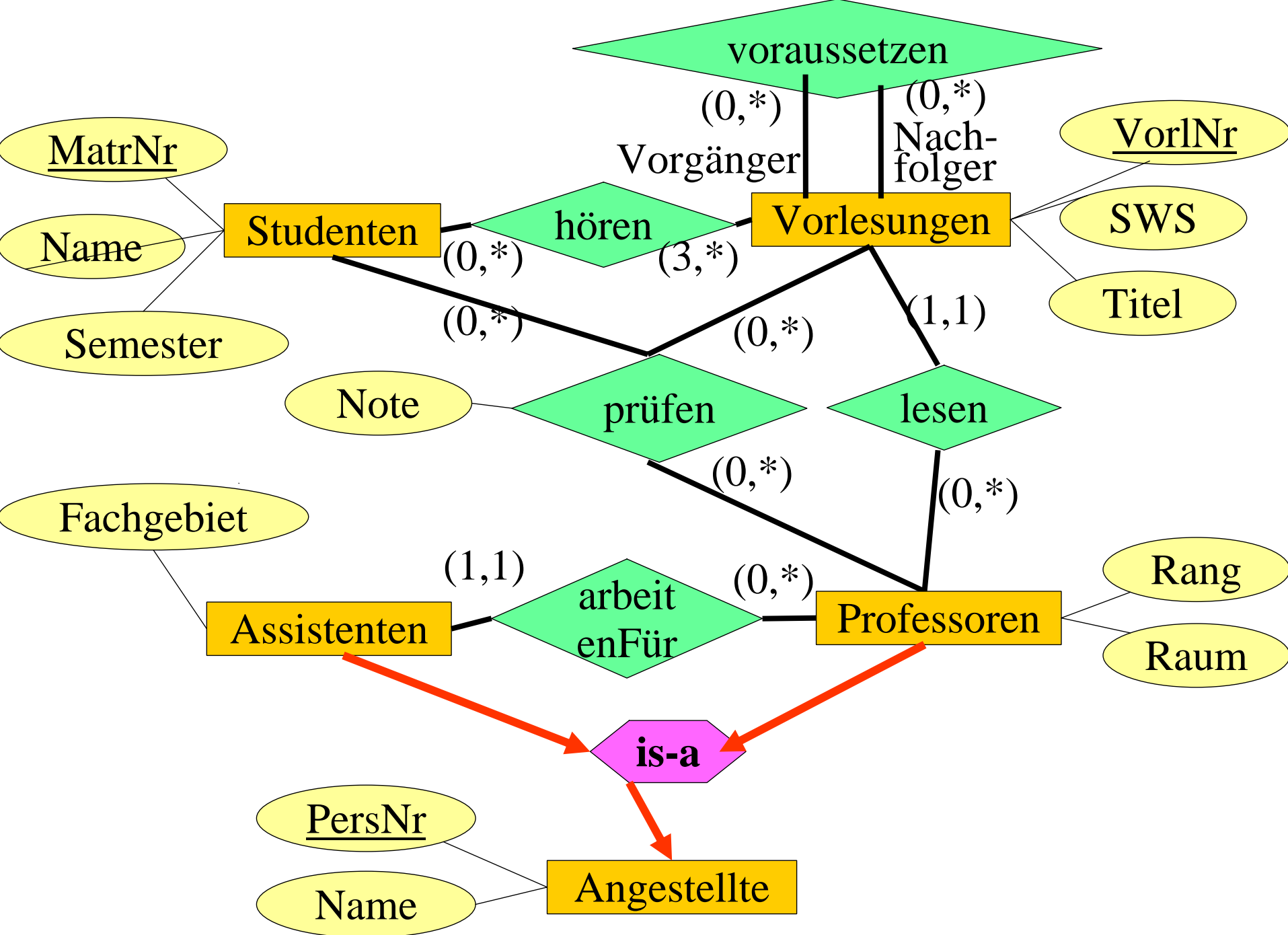
- Mehrere Prüfer in einer Prüfung
- Mehrere Vorlesungen werden in einer Prüfung abgefragt

Generalisierung

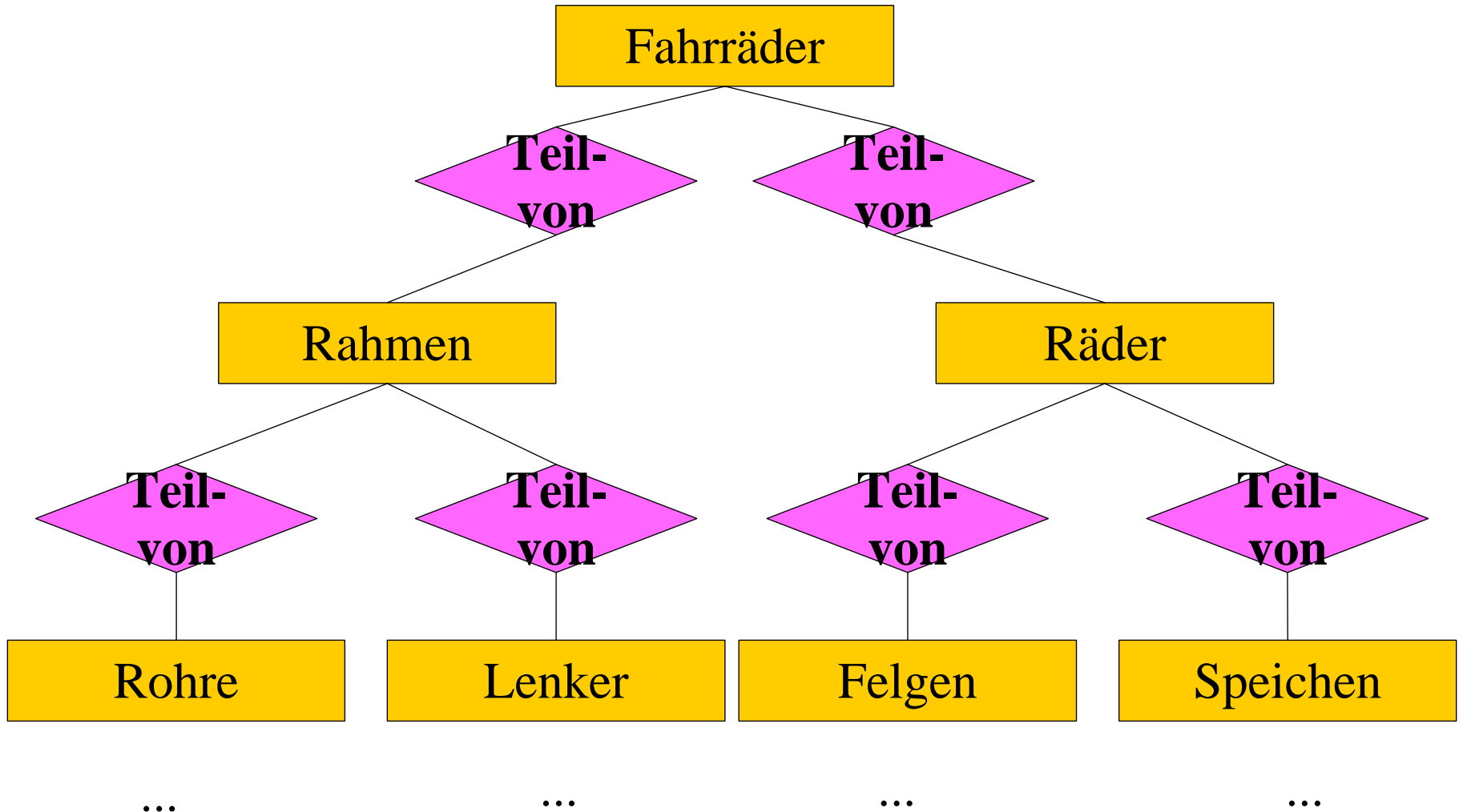


Universitätsschema mit Generalisierung und (min, max)- Markierung

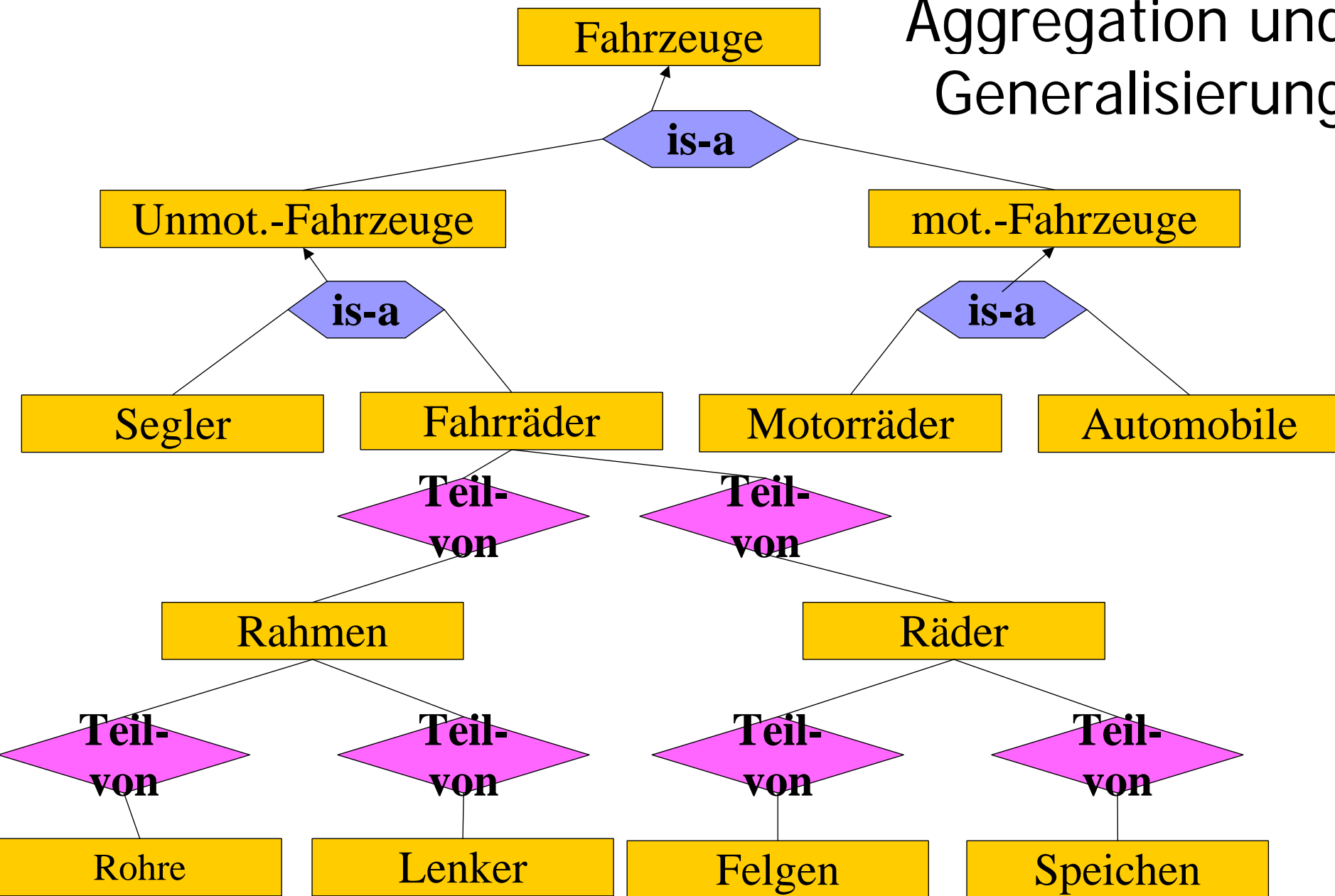
→ Nächste Seite



Aggregation



Aggregation und Generalisierung



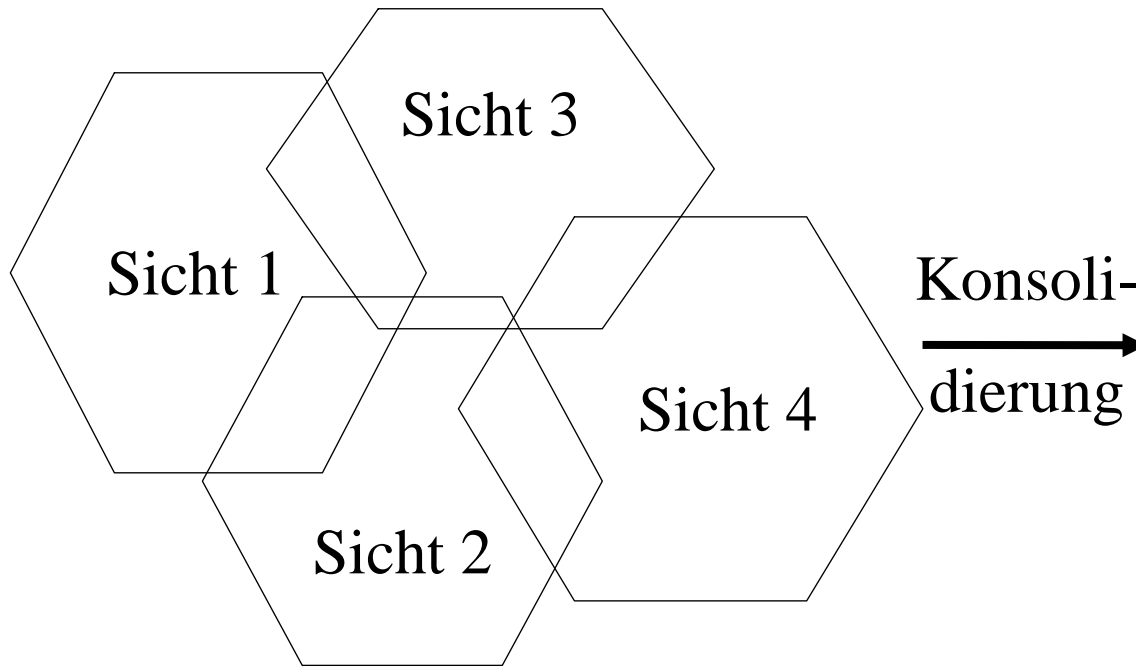
...

...

...

...

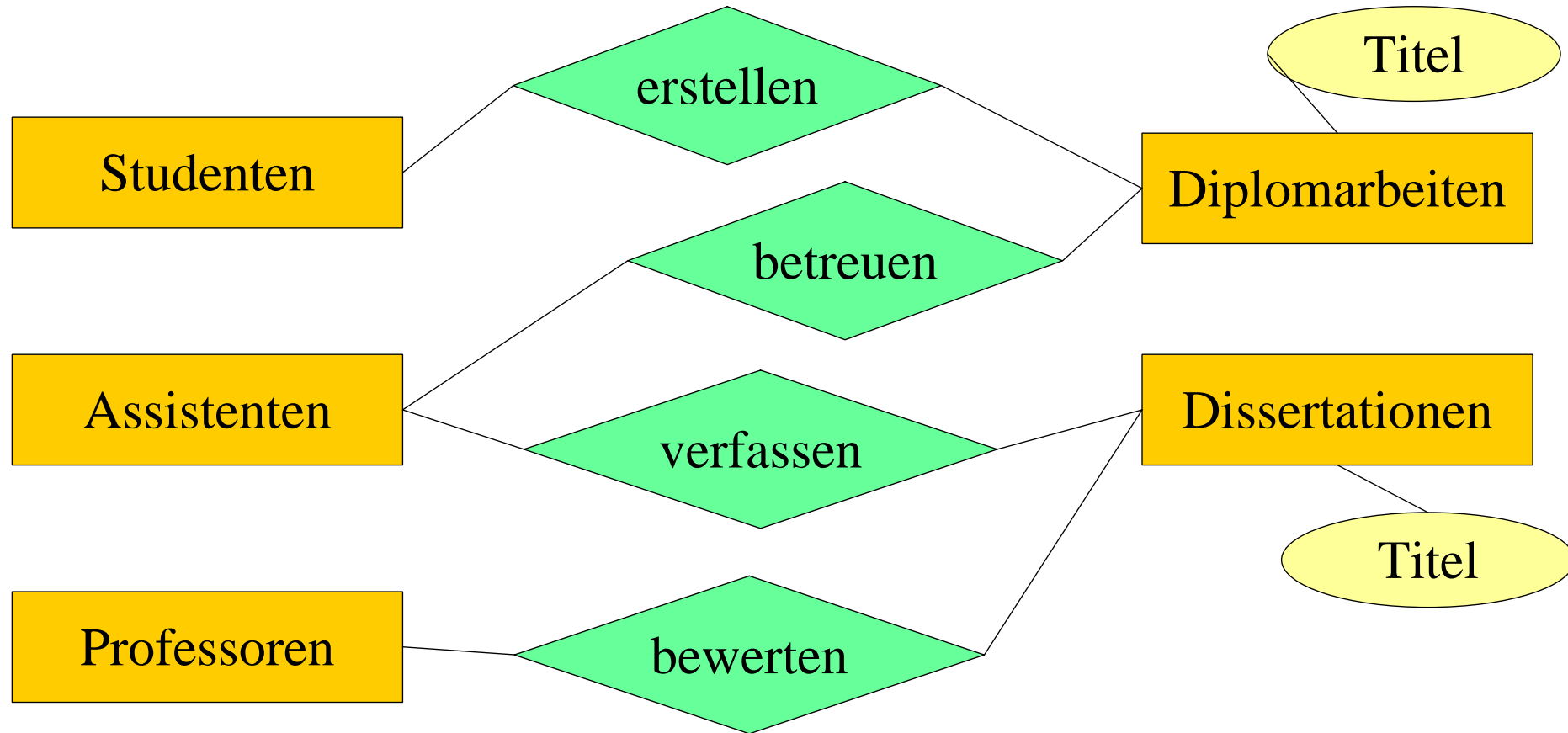
Konsolidierung von Teilschemata oder Sichtenintegration



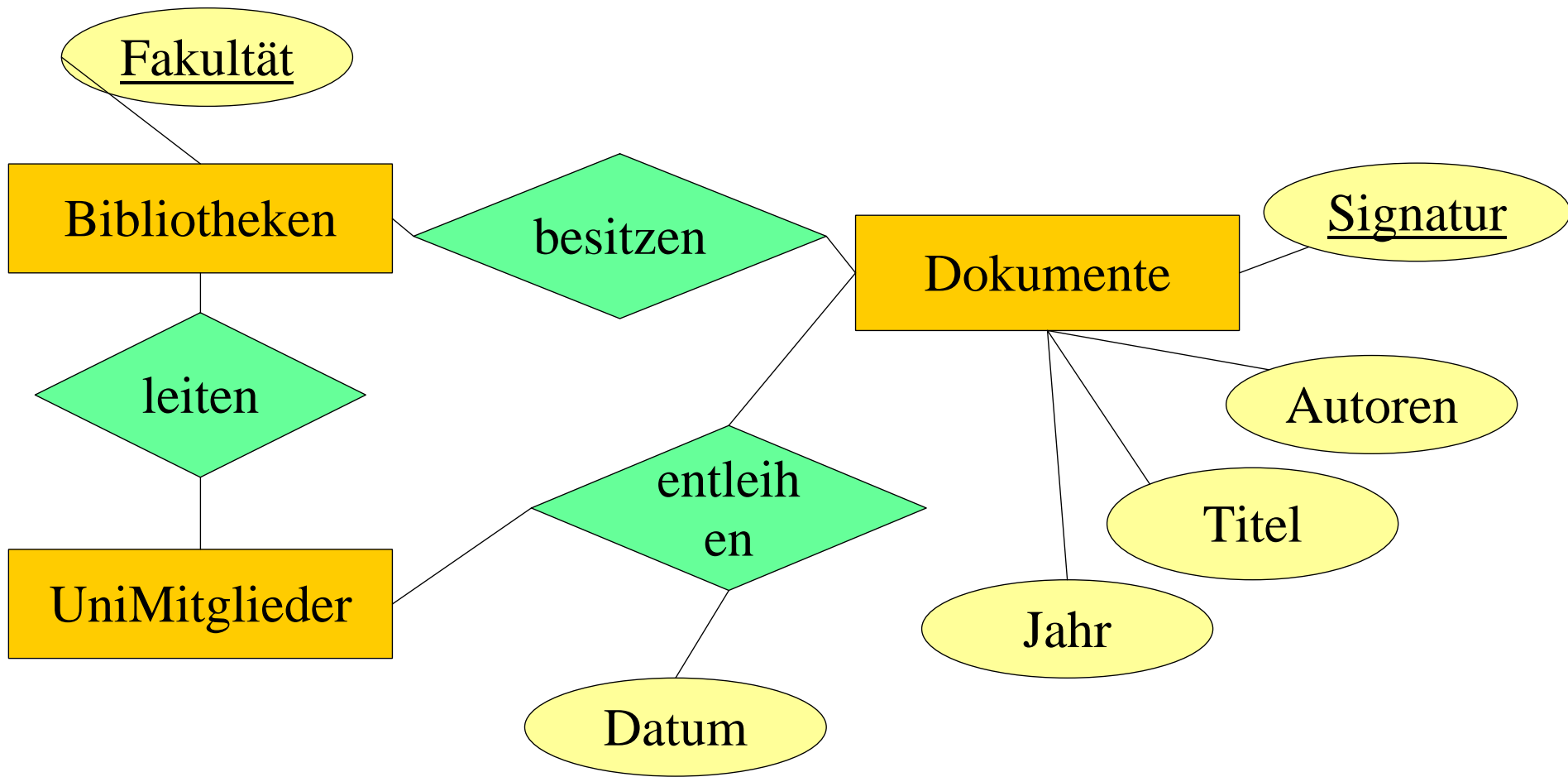
Globales Schema

- **Redundanzfrei**
- **Widerspruchsfrei**
- **Synonyme bereinigt**
- **Homonyme bereinigt**

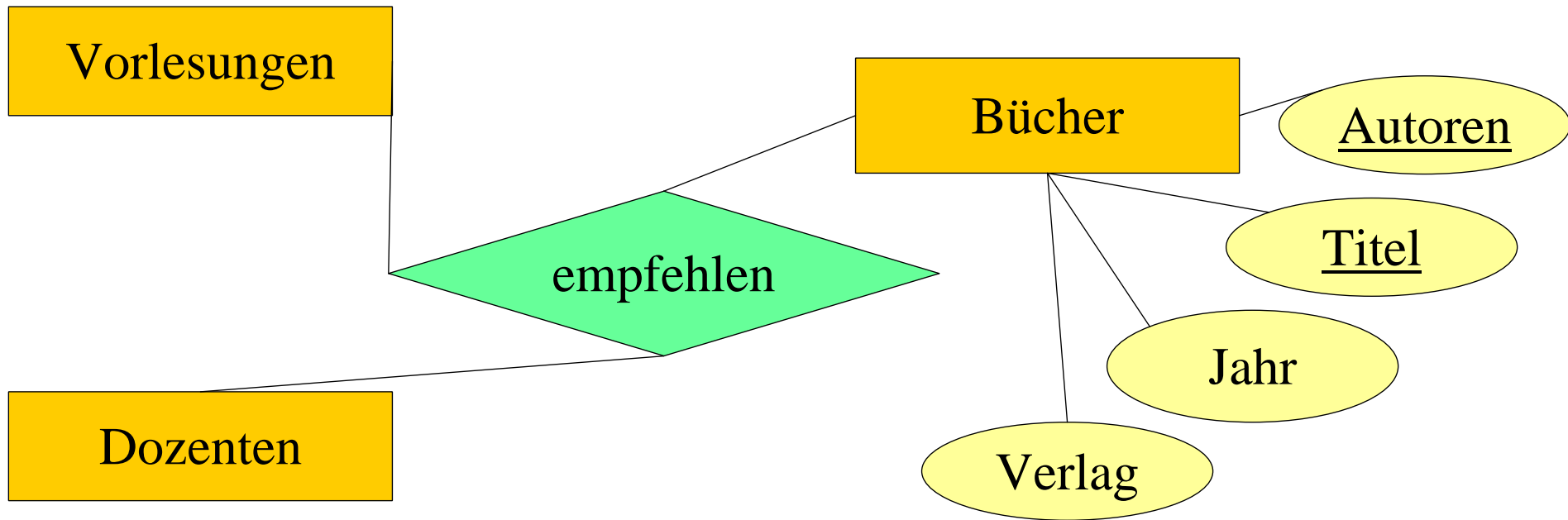
Drei Sichten einer Universitäts-Datenbank



Sicht 1: Erstellung von Dokumenten als Prüfungsleistung



Sicht 2: Bibliotheksverwaltung



Sicht 3: Buchempfehlungen für Vorlesungen

Beobachtungen

- Die Begriffe *Dozenten* und *Professoren* sind synonym verwendet worden.
- Der Entitytyp *UniMitglieder* 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 *UniMitglieder* 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.

