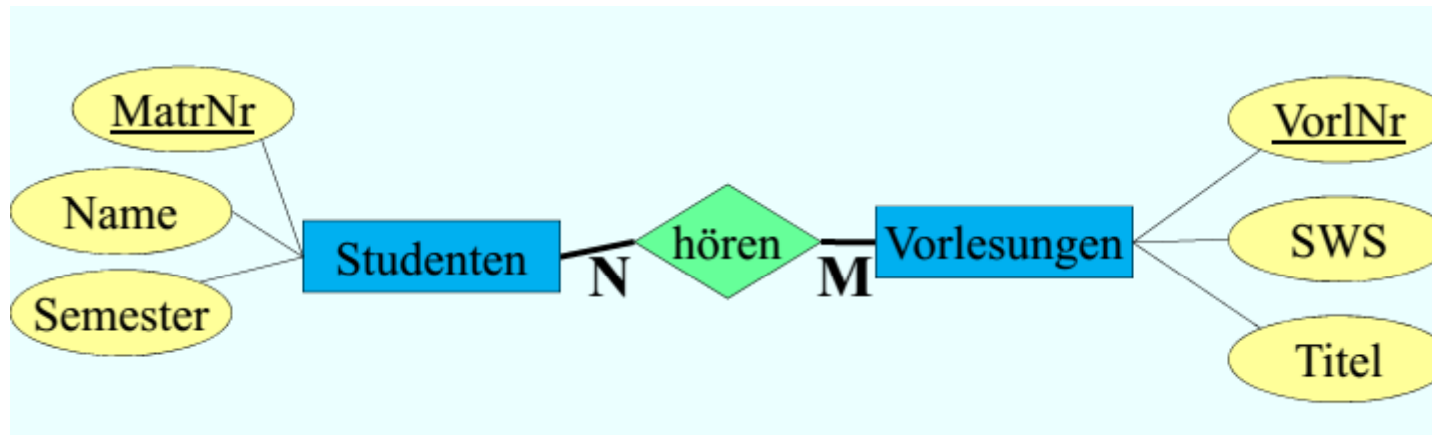


Relationales Datenmodell

N-M Beziehung



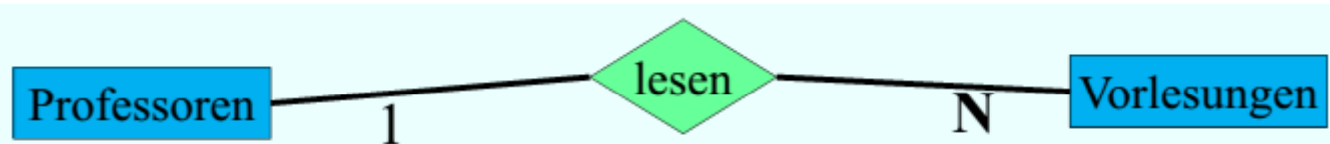
Studenten		hören		Vorlesungen	
<i>MatrNr</i>	...	MatrNr	VorlNr	<i>VorlNr</i>	...
26120	...	26120	5001	5001	...
27550	...	27550	5001	4052	...
		27550	4052		

Studenten: {[MatrNr:integer, Name: string, Semester: integer]}

Vorlesungen: {[VorlNr:integer, Titel: string, SWS: integer]}

hören (N:M): {[MatrNr: integer, VorlNr: integer]}

1-N Beziehung



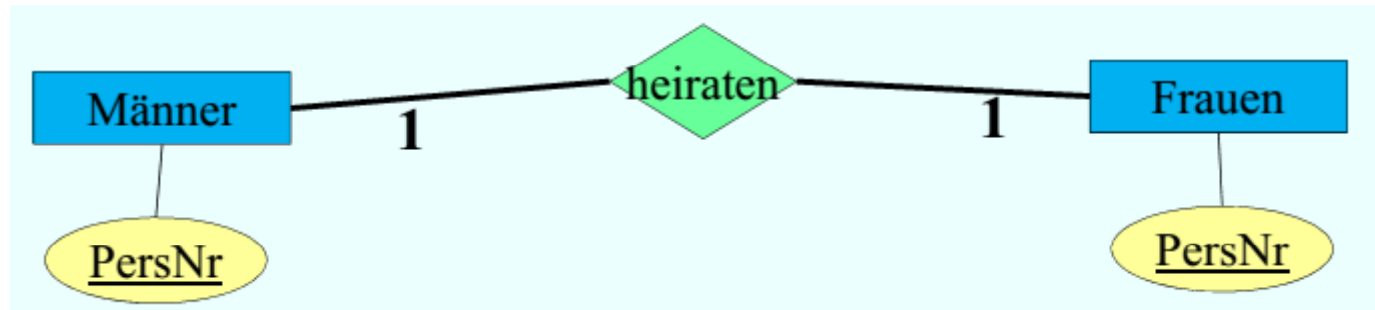
Verfeinerung durch Zusammenfassung

Professoren : {[PersNr, Name, Rang, Raum]}

Vorlesungen : {[VorlNr, Titel, SWS,
gelesenVon]}

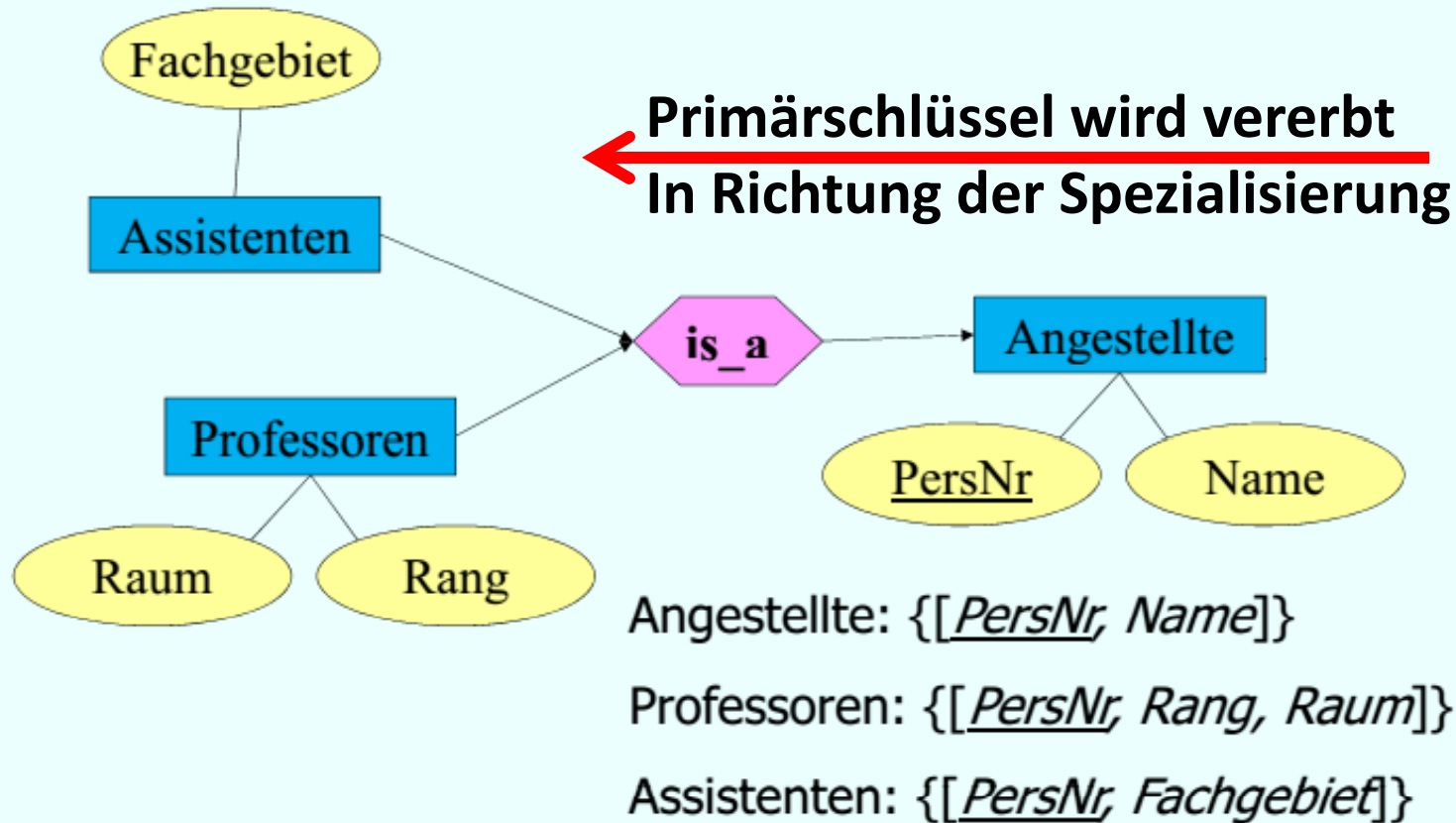
**Fremdschlüssel immer Gegenüber der
1 sonst kommt es zu Anomalien**

1-1 Beziehung

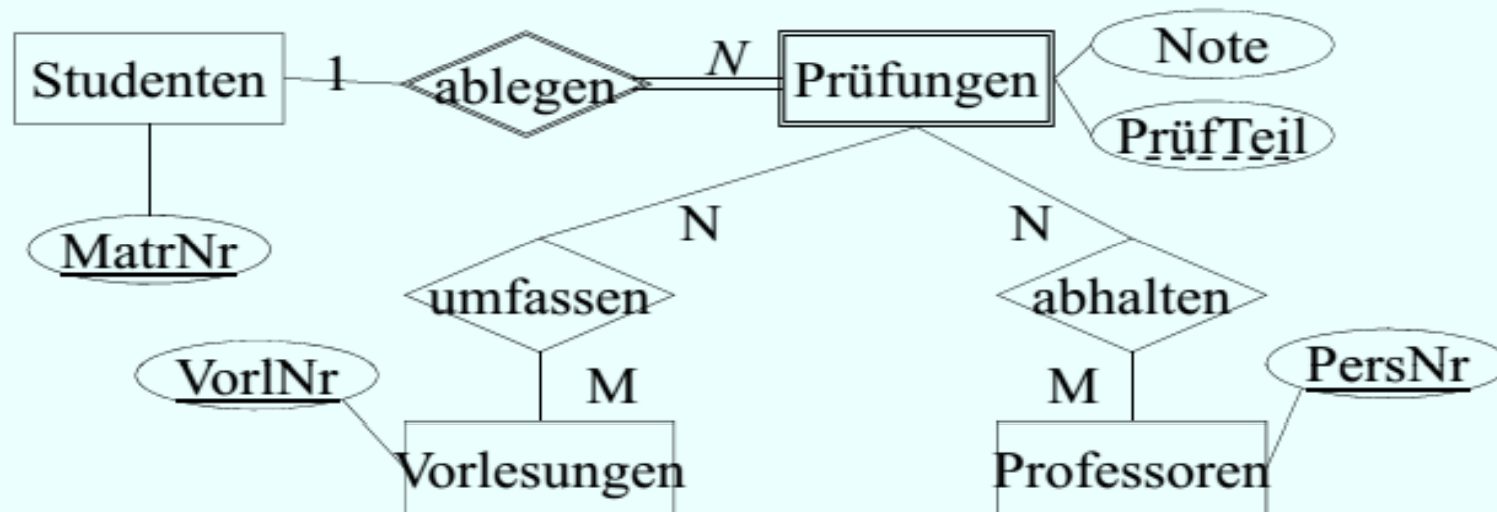


- Genau wie bei 1-N
- Seite wählen, wo die Funktion am wenigste partiell ist (NULL Werte vermeiden)
- Beispiel :
 - Attribut „Präsident von“ nicht bei allen Angestellten einer Uni, sondern bei Universität

Relationale Modellierung der Generalisierung



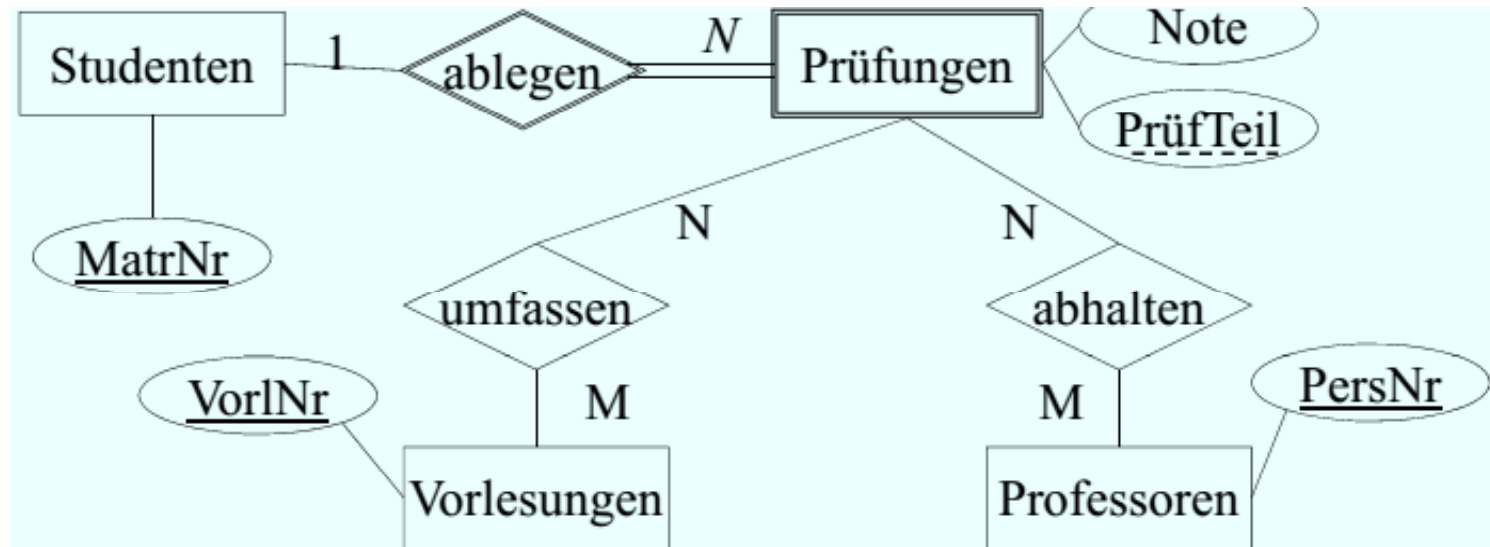
Relationale Modellierung schwacher Entitytypen



Entity-Mengen Studenten, Prüfungen:

Studenten: {[MatrNr: integer, ...]}

Prüfungen: {[MatrNr: integer, PrüfTeil: string, Note: integer]}



- **Prüfung ist Existenzabhängig von Student**
 -> fällt der Student weg, existiert auch die Prüfung nicht mehr
- **PrüfTeil ist nur ein Teilschlüssel**
 -> zur eindeutigen Identifizierung muss auch MatrNr von Student übernommen werden

Prüfungen: {[MatrNr: integer, PrüfTeil: string, Note: integer]}

umfassen: {[MatrNr: integer, PrüfTeil: string, VorlNr: integer]}

abhalten: {[MatrNr: integer, PrüfTeil: string, PersNr: integer]}