

Datenbankentwurf

VO Datenmodellierung

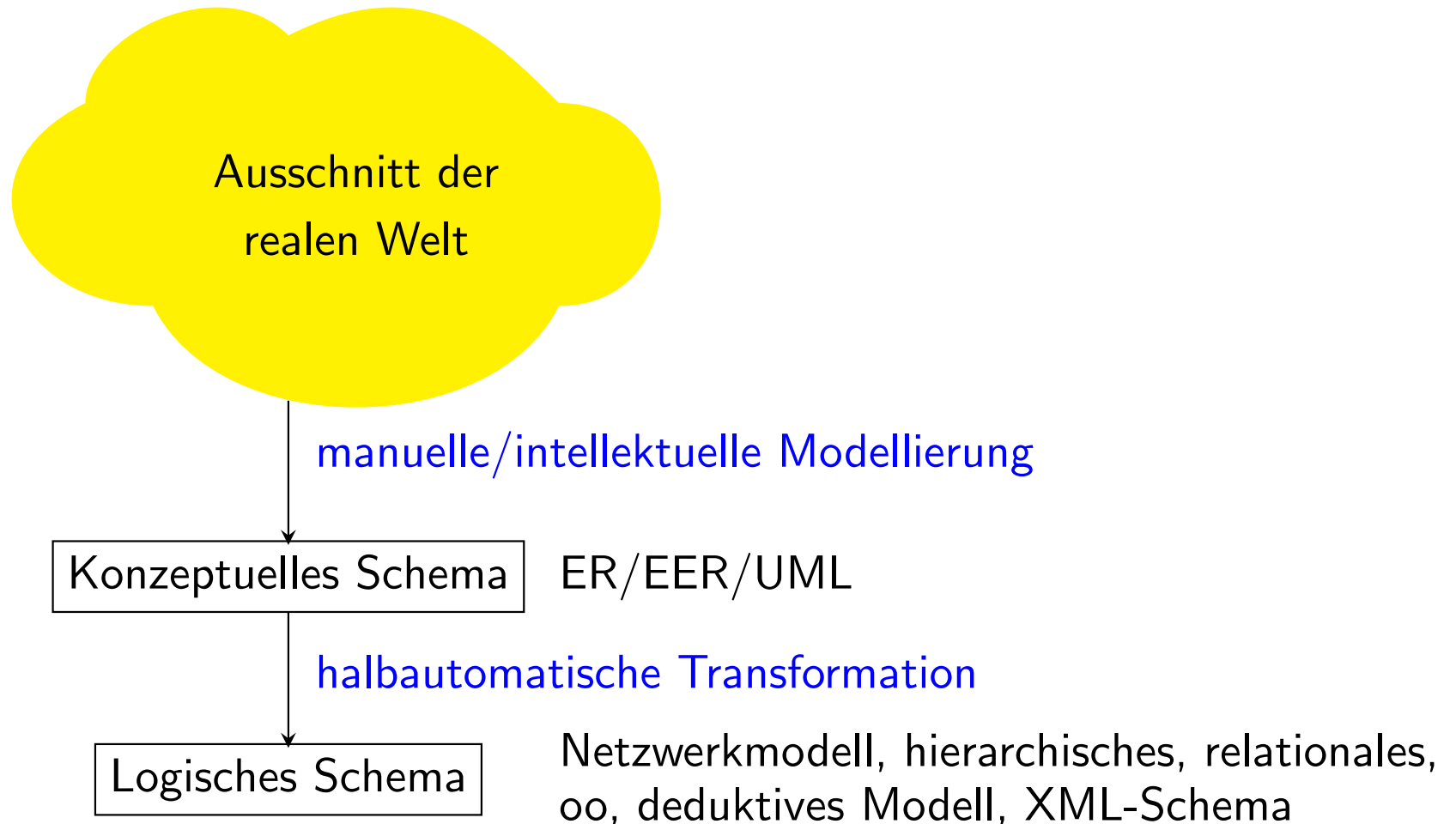
Katrin Seyr

Institut für Informationssysteme
Technische Universität Wien

Überblick

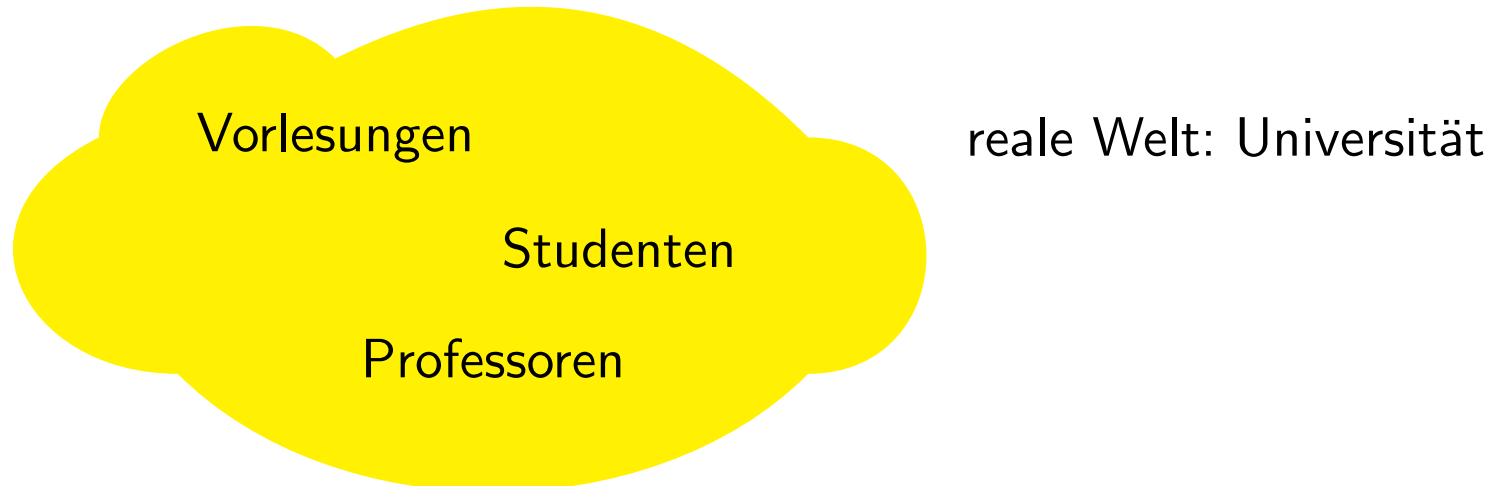
- Wiederholung: Datenmodellierung
- Allgemeine Entwurfsmethodik
- Datenbankentwurfsschritte
 - Anforderungsanalyse
 - Konzeptueller Entwurf
 - Implementationsentwurf (Kap. 3)
 - Physischer Entwurf (VL DBS)

Wiederholung Datenmodellierung

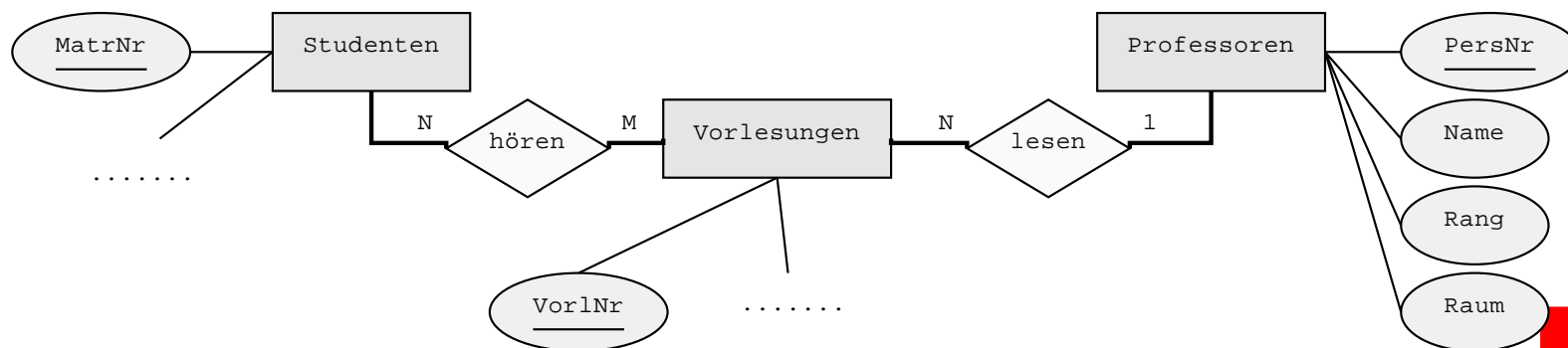


Wiederholung: Bsp Datenmodellierung

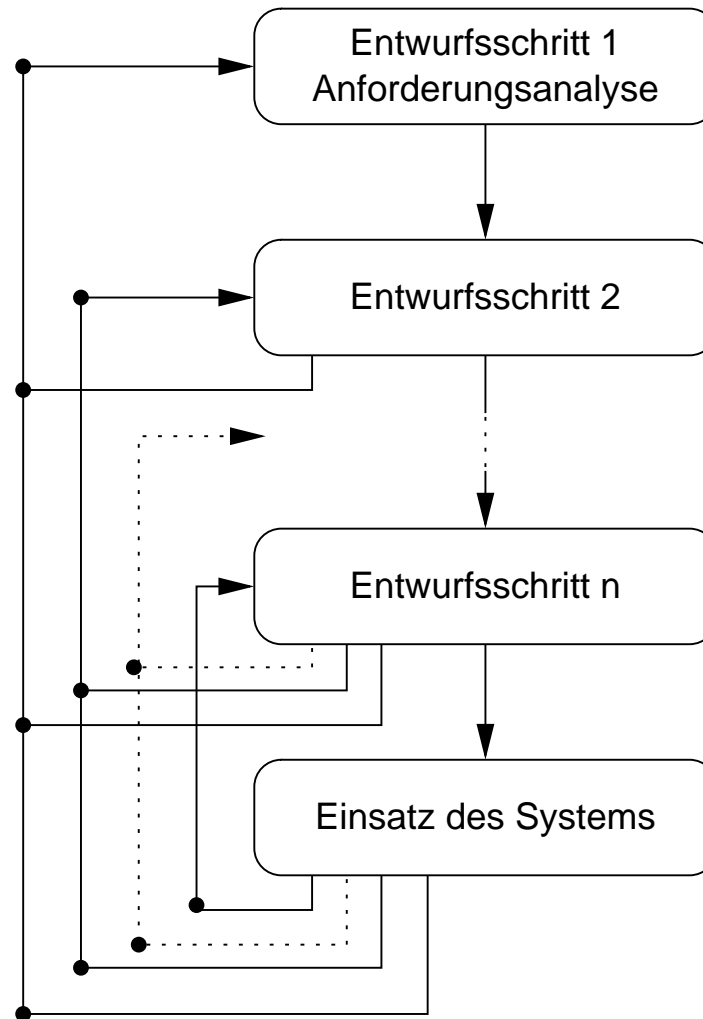
1 Abgrenzung der zu modellierenden Welt



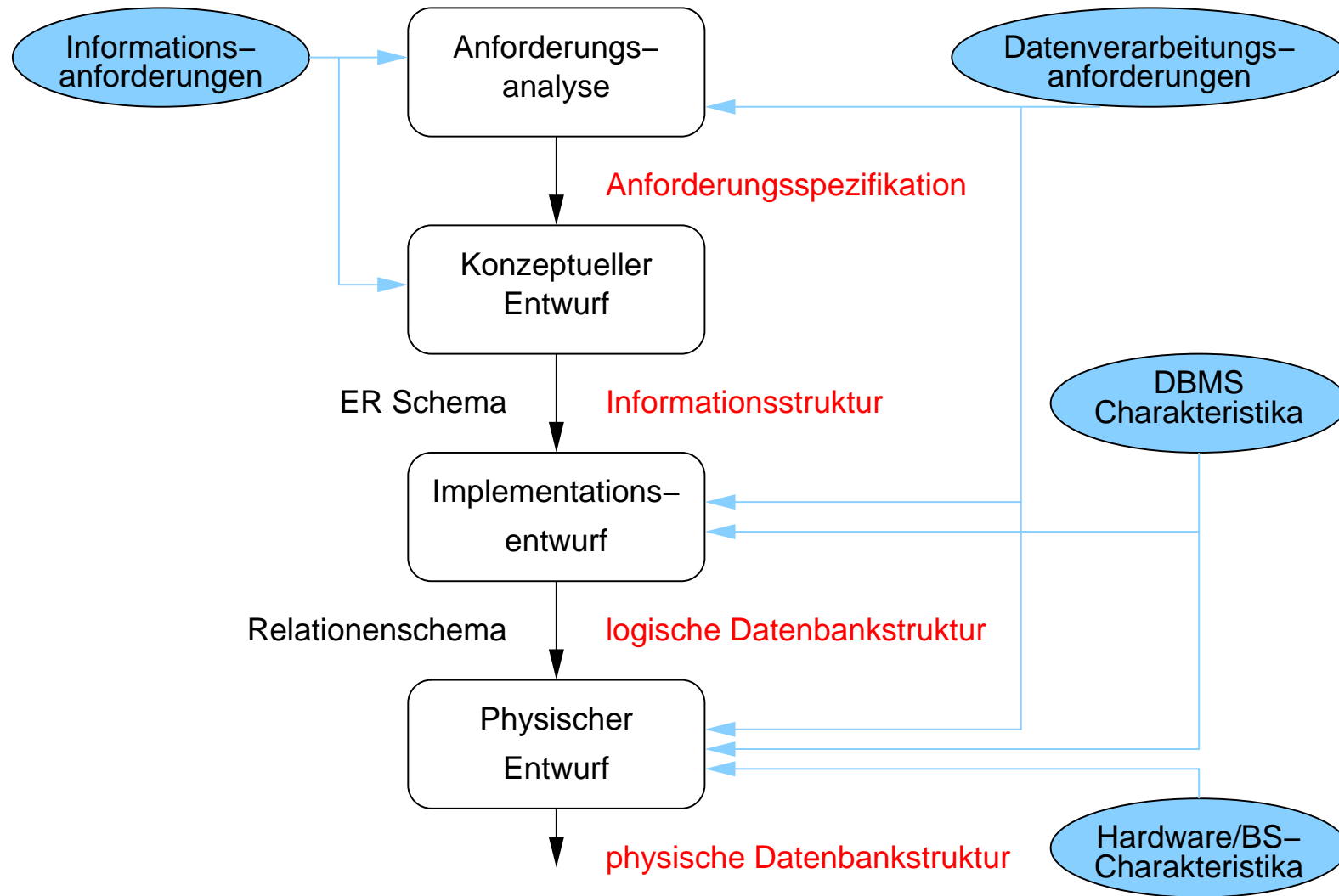
2 Überführung der zu modellierenden Welt in ein Konzeptuelles Schema (in der VL: EER)



Allgemeine Entwurfsmethodik



Datenbankentwurfsschritte



Anforderungsanalyse

- 1 Identifikation von Organisationseinheiten
- 2 Identifikation der zu unterstützenden Aufgaben
- 3 Anforderungs-Sammelplan
- 4 Anforderungs-Sammlung
- 5 Filterung (Prüfung der Information auf Verständlichkeit und Eindeutigkeit)
- 6 Satzklassifikation (Objekte, Beziehungen zwischen Objekten, Operationen, Ereignisse)
- 7 Formalisierung - Erstellung des Pflichtenhefts, aufgebaut wie folgt:
 - Informationsstrukturanforderungen: strukturierte Information über
 - Objekte
 - Attribute
 - Beziehungen
 - Datenverarbeitungsanforderungen: strukturierte Information über Prozessbeschreibungen

Objekt- und Attributbeschreibung

Objekte

- Uni-Angestellte:
 - Anzahl: 1000
 - Attribute:
 - Personalnummer
 - Name
 - Gehalt
 - Rang
- Studenten:
 - Anzahl: 20.000
 - Attribute:
 - Matrikelnummer
 - Name
 - Adresse

Attribute

- Personalnummer
 - Typ: char
 - Länge: 9
 - Wertebereich: 0...999.999
 - Anzahl Wiederholungen: 0
 - Definiertheit: 100%
 - Identifizierend: ja
- Gehalt
 - Typ: dezimal
 - Länge: (8,2)
 - Anzahl Wiederholung: 0
 - Definiertheit: 90%
 - Identifizierend: nein

Beziehungsbeschreibung

Beziehung: *prüfen*

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

Prozessbeschreibung

Prozess: *Zeugnisausstellung*

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

Wiederholung: Überblick

- Rec: Datenmodellierung
- Allgemeine Entwurfsmethodik
- Datenbankentwurfsschritte
 - Anforderungsanalyse
 - Konzeptueller Entwurf
 - Das ER Modell
 - Sichtenintegration und Konsolidierung

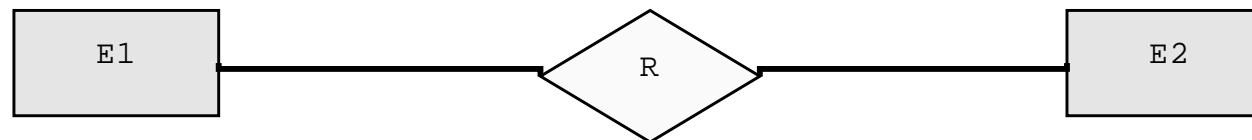
Das ER Modell

- 1 Entities und Beziehungen
- 2 Rollen und Attribute
- 3 Schlüssel
- 4 Funktionalitäten
- 5 (min,max) Notation
- 6 Schwache Entities
- 7 Generalisierung oder Spezialisierung (EER)
- 8 Aggregation (EER)

Entities und Beziehungen

Entities: wohlunterscheidbare Konzepte der zu modellierenden Welt, abstrahiert zu Entitytypen

Beziehungen: verknüpfen mehrere Entities, abstrahiert zu Beziehungstypen



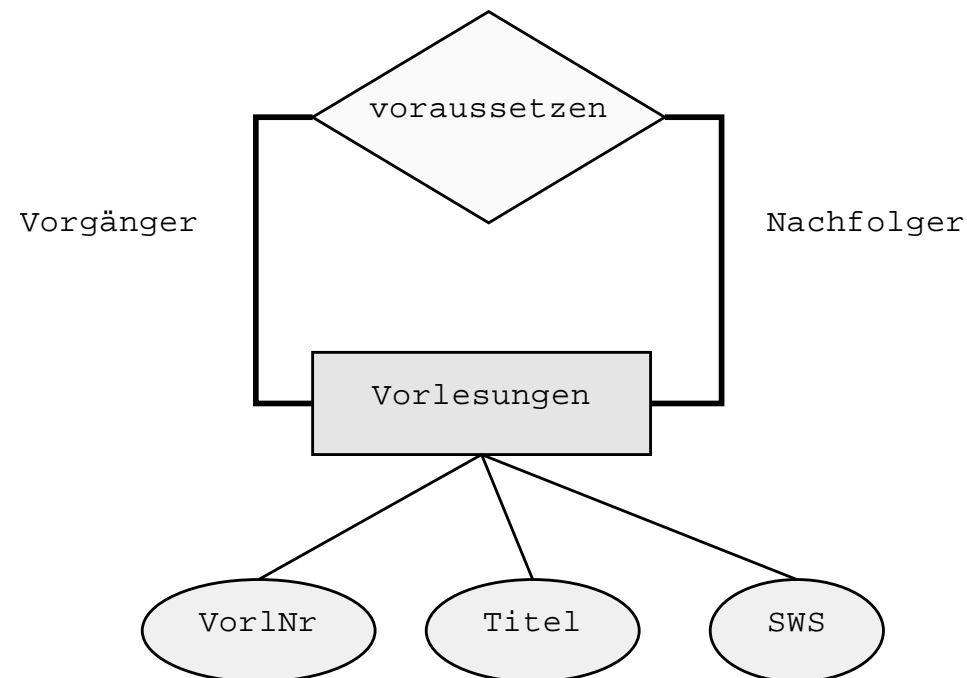
$$R \subseteq E1 \times E2$$



Rollen und Attribute

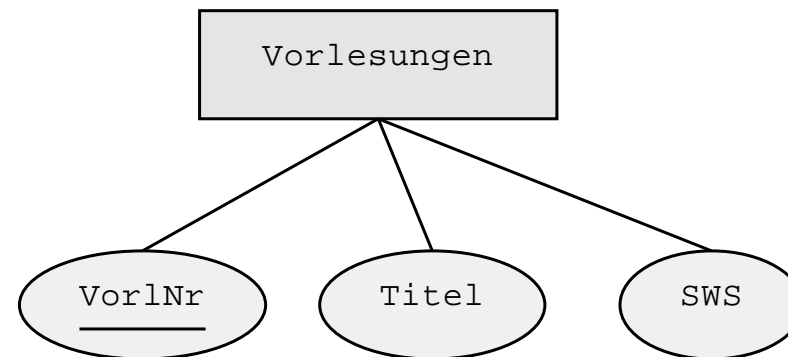
Attribute charakterisieren Entities bzw. Beziehungen.

Rollen können verwendet werden, um zu beschreiben, wie die an einer Beziehung beteiligten Entities sich verhalten.

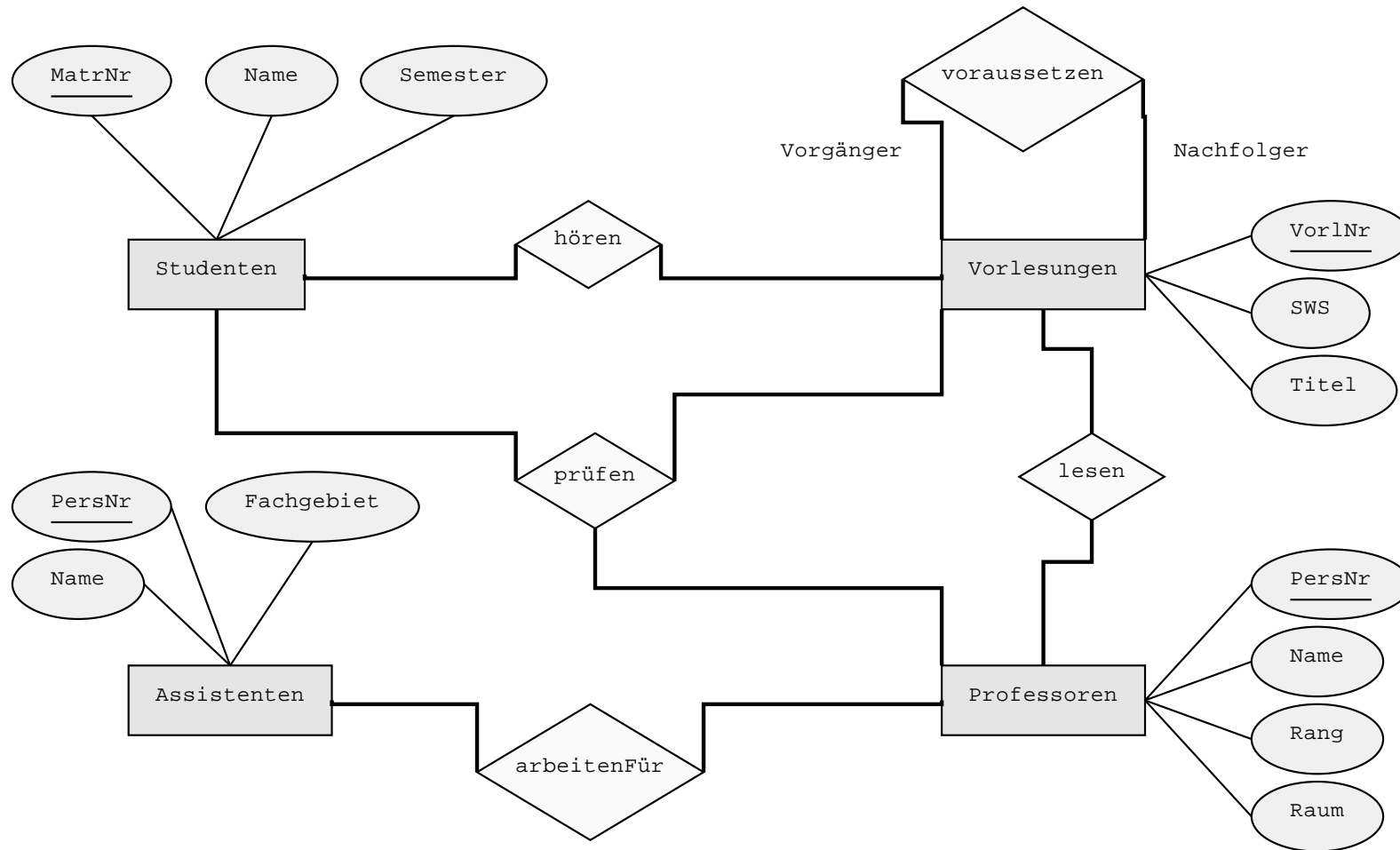


Schlüssel

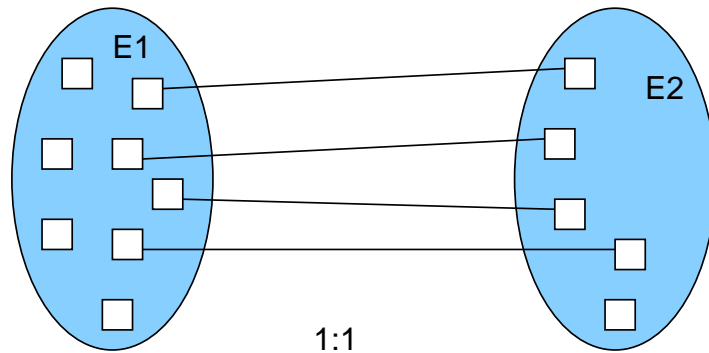
Schlüssel sind eine **minimale** Menge von Attributen, deren Wert eine Entity eindeutig innerhalb aller Entities eines Typs identifizieren.
Bei mehreren Schlüsselkandidaten: wähle Primärschlüssel aus.



Bsp: Universitätsschema

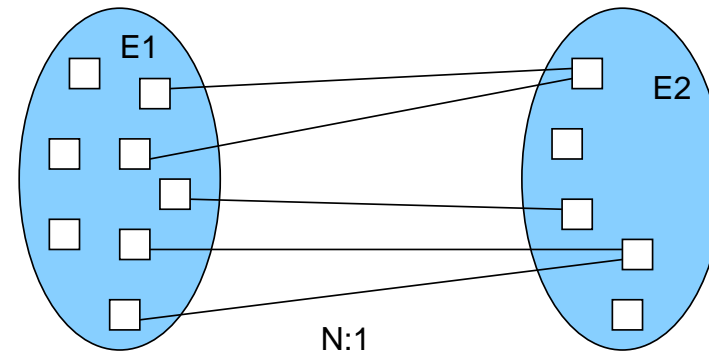


Funktionalitäten



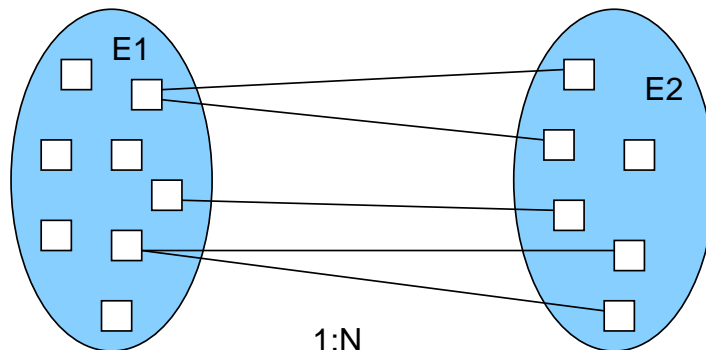
1:1

$$R : E_1 \rightarrow E_2 \text{ bzw. } R^{-1} : E_2 \rightarrow E_1$$



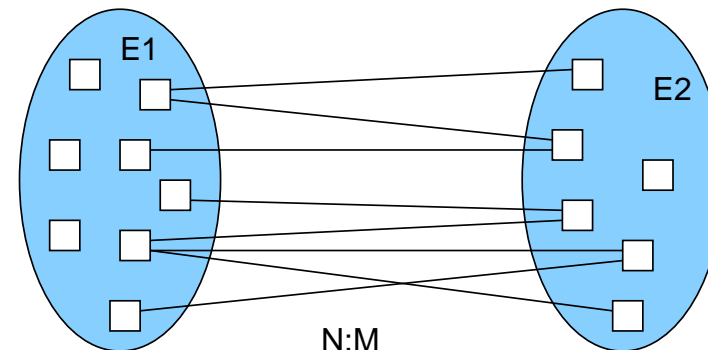
N:1

$$R : E_1 \rightarrow E_2$$



1:N

$$R : E_2 \rightarrow E_1$$



N:M

keine partielle Funktion

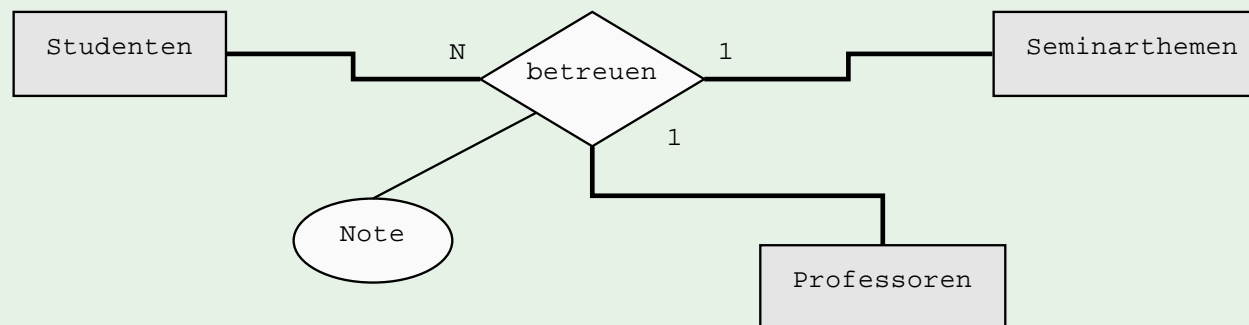
Funktionalitäten bei n -stelligen Beziehungen

Ist R eine Beziehung zwischen mehreren Entities E_1, \dots, E_n , wobei die Funktionalität der Entity $E_k, 1 \leq k \leq n$ mit "1" spezifiziert wird, so wird durch R folgende partielle Funktion vorgegeben:

$$R : E_1 \times E_2 \times \dots \times E_{k-1} \times E_{k+1} \times \dots \times E_n \rightarrow E_k$$

Beispiel

Gegeben ist die ternäre Beziehung *betreuen* zwischen den Entities *Studenten*, *Professoren*, *Seminarthemen*.



betreuen: Professoren \times Studenten \rightarrow Seminarthemen

betreuen: Seminarthemen \times Studenten \rightarrow Professoren

Funktionalitäten bei n -stelligen Beziehungen

Beispiel (ctd.)

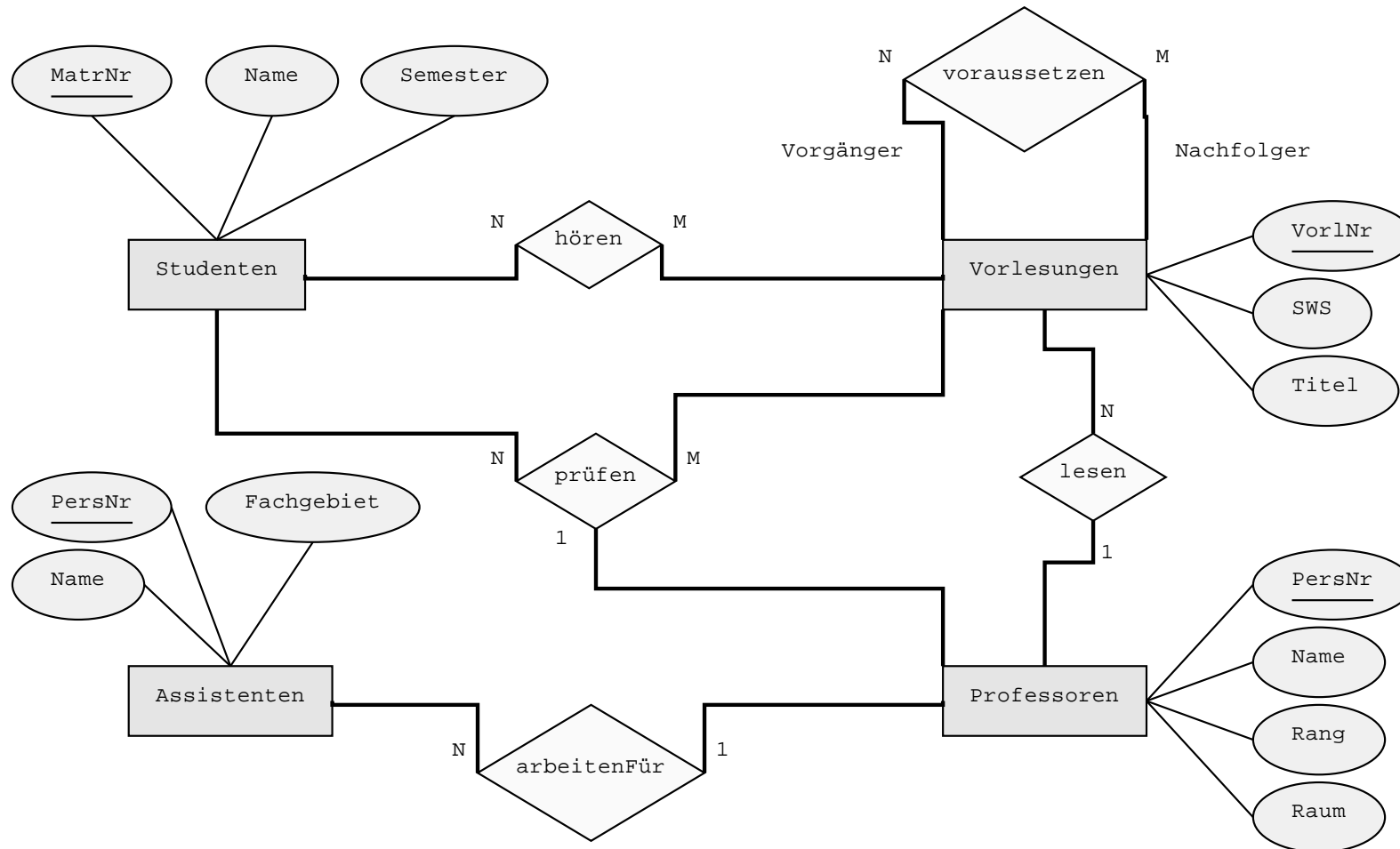
betreuen: Professoren \times Studenten \rightarrow Seminarthemen
bedeutet: Studierende dürfen bei demselben Professor nur ein Seminarthema bearbeiten

betreuen: Seminarthemen \times Studenten \rightarrow Professoren
bedeutet: Studierende dürfen dasselbe Seminarthema nur bei einem Professor bearbeiten (= nicht wiederverwenden)

weiterhin möglich sind:

- Professoren können ein Seminarthema an mehrere Studierende vergeben
- das selbe Seminarthema kann von mehreren Professoren vergeben werden - allerdings nur an verschiedene Studierende (s.o.)

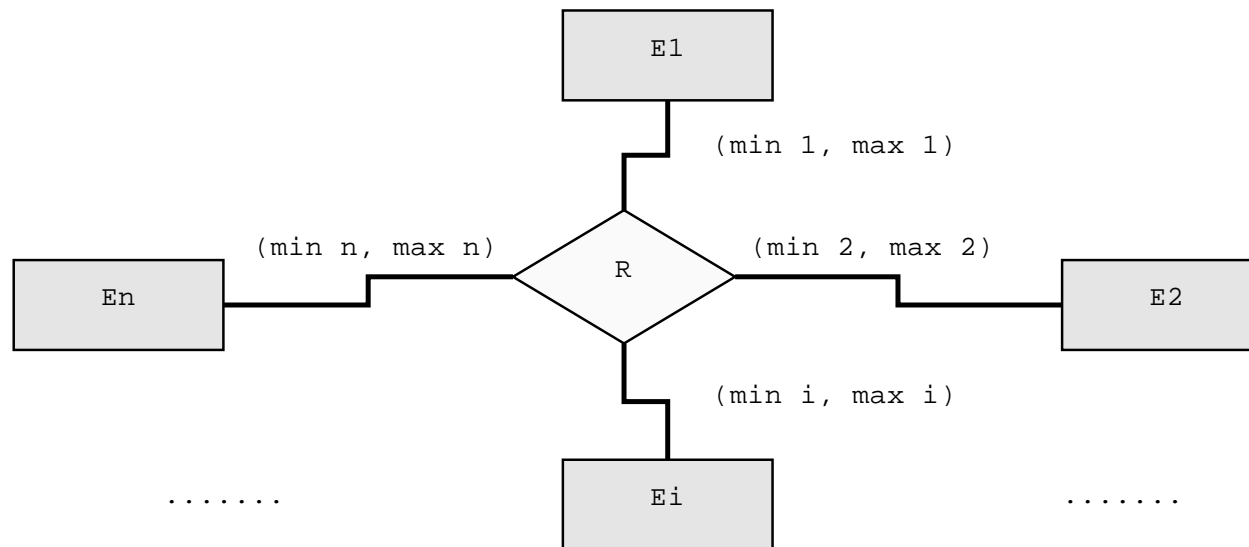
Bsp: Universitätsschema mit Funktionalitäten



(min,max) Notation

Gegeben eine n -stellige Beziehung R zwischen den Entitytypen E_1, E_2, \dots, E_n .

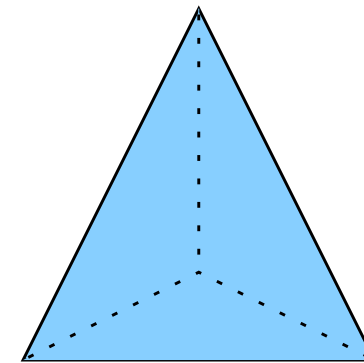
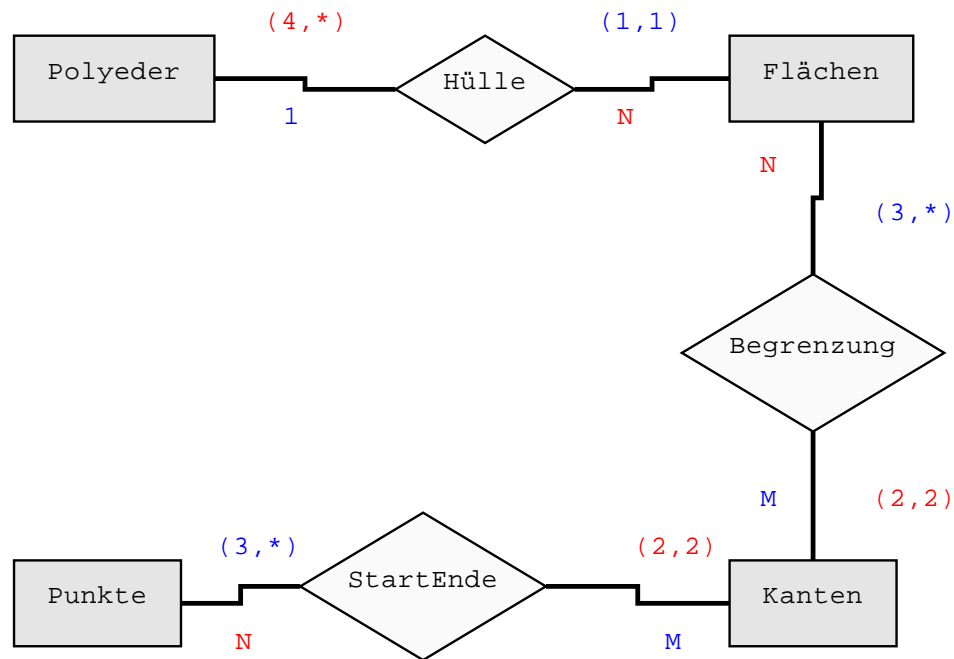
$$R \subseteq E_1 \times \dots \times E_i \times \dots \times E_n$$



Es gilt: für jedes Tupel $e_i \in E_i$ gibt es

- mindestens min_i Tupel der Art $(\dots, e_i, \dots) \in R$
- maximal max_i Tupel der Art $(\dots, e_i, \dots) \in R$

Bsp: (min,max) Notation



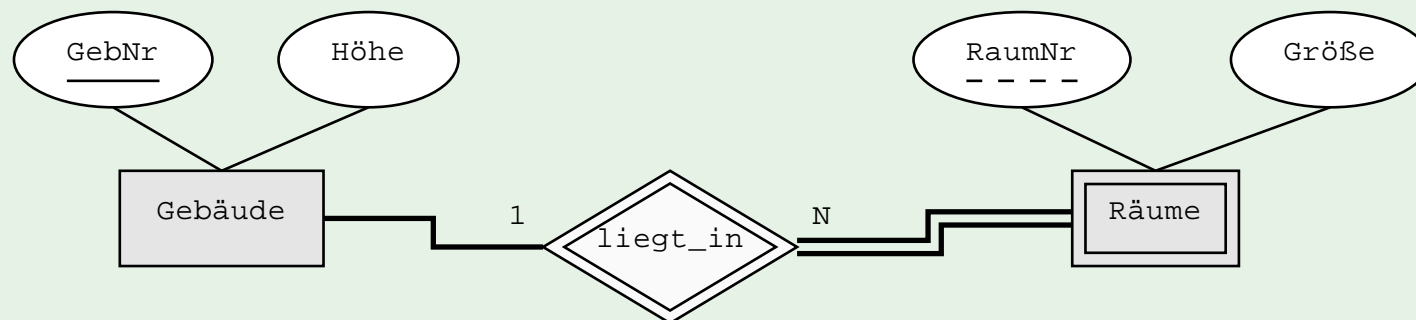
minimales Polyeder

Schwache Entities

Schwache Entities sind Entities, deren Existenz von einer anderen, übergeordneten Entity abhängen und die durch eine Kombination mit dem Schlüssel der übergeordneten Entity identifizierbar sind.

Beispiel

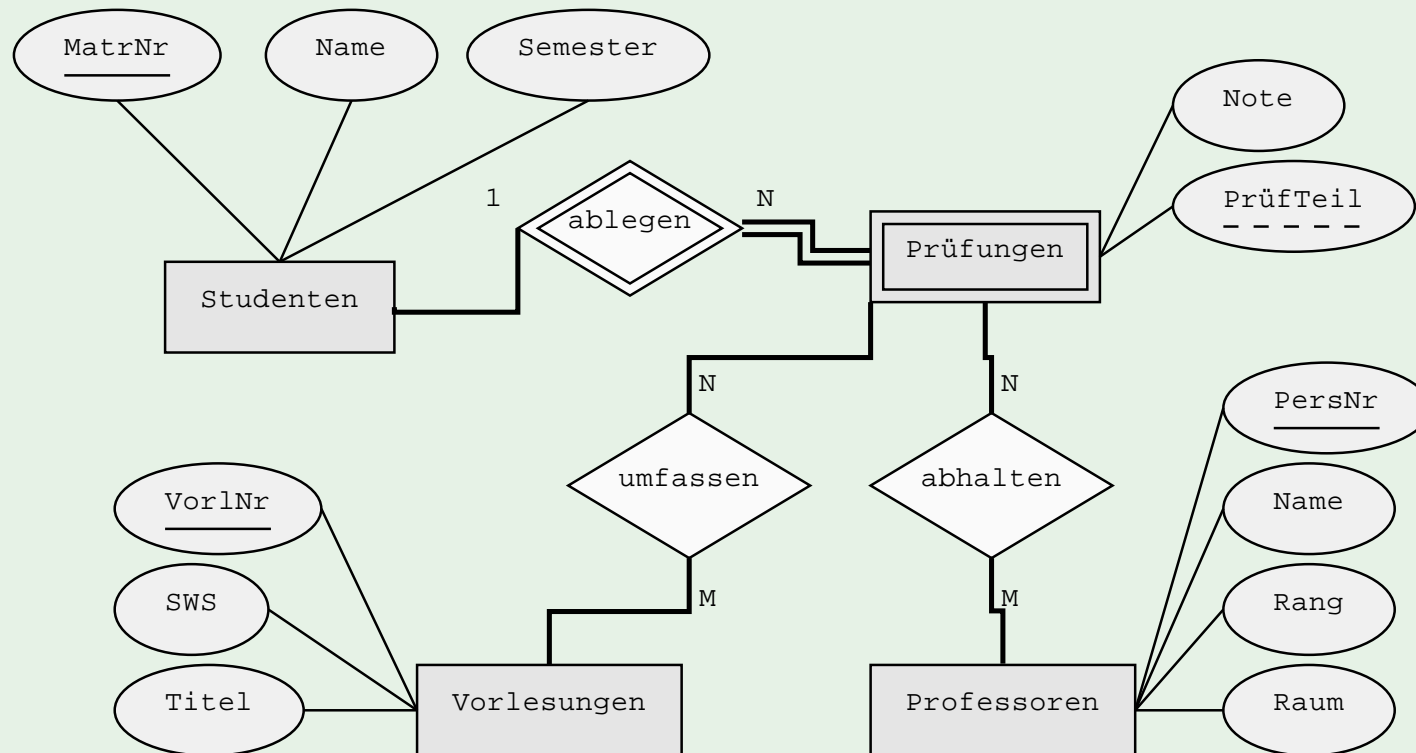
Ist die Nummer eines Raumes nur innerhalb eines Gebäudes eindeutig, so ist der Schlüssel von Räume eine Kombination aus Raum- und Gebäudenummer.



Schwache Entities

Beispiel

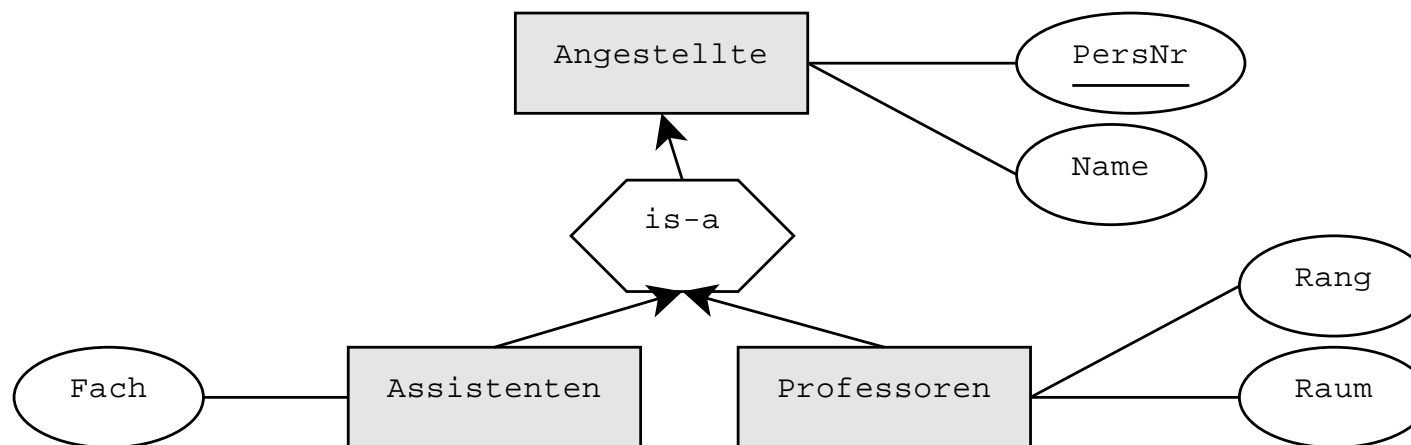
Ein Diplomprüfungsfach besteht aus mehreren Vorlesungen, die alle in Teilprüfungen von den Professoren beurteilt werden. Eine Prüfung selbst ist abhängig vom Studierenden, der die Prüfung ablegt.



Generalisierung oder Spezialisierung (EER)

Generalisierung wird verwendet, um eine Strukturierung der Entitytypen zu erzielen:

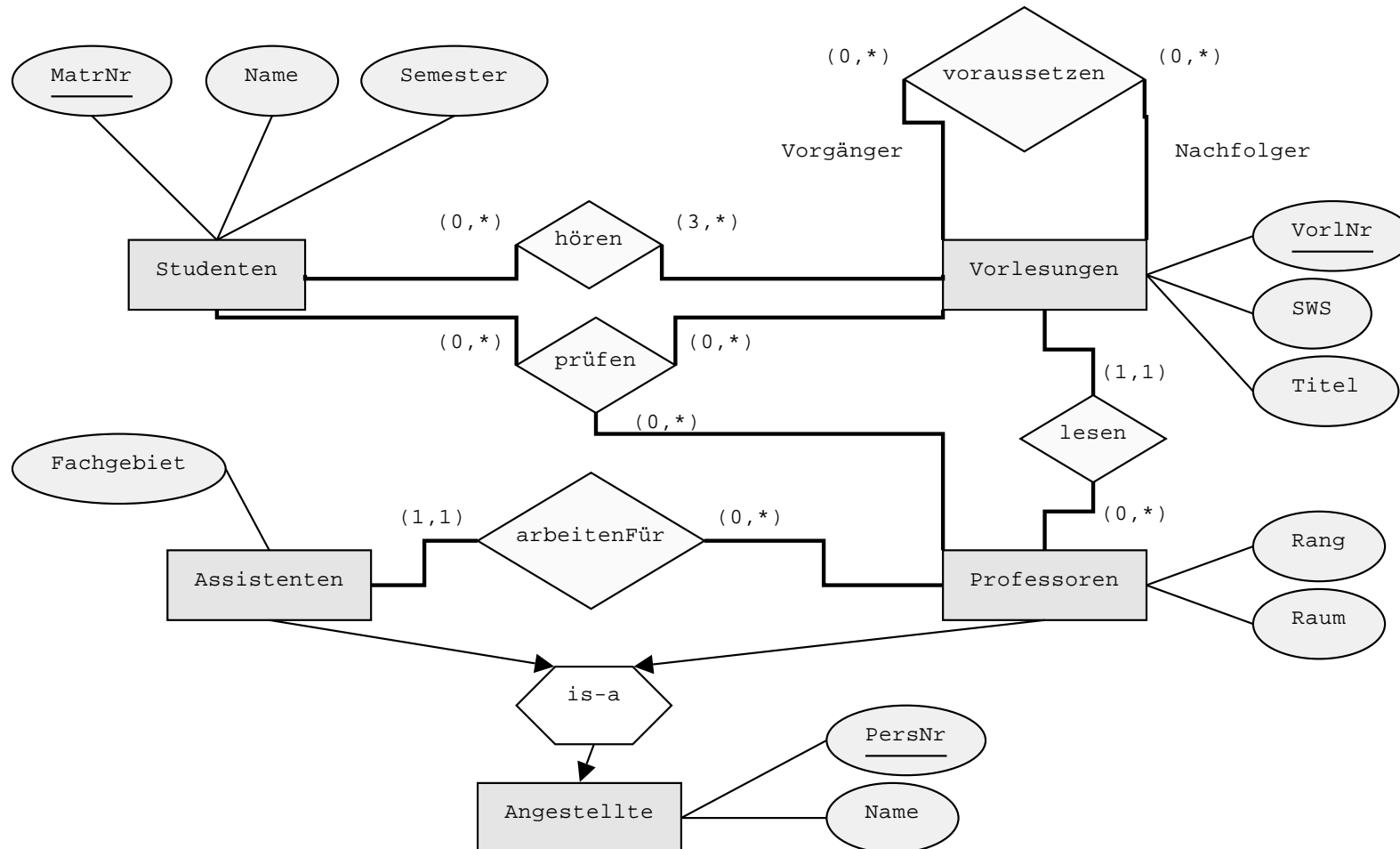
- gemeinsame Eigenschaften von ähnlichen Entitytypen werden einem Obertyp (O) zugeordnet
- unterschiedliche Eigenschaften verbleiben bei den Untertypen (U)



Es gibt:

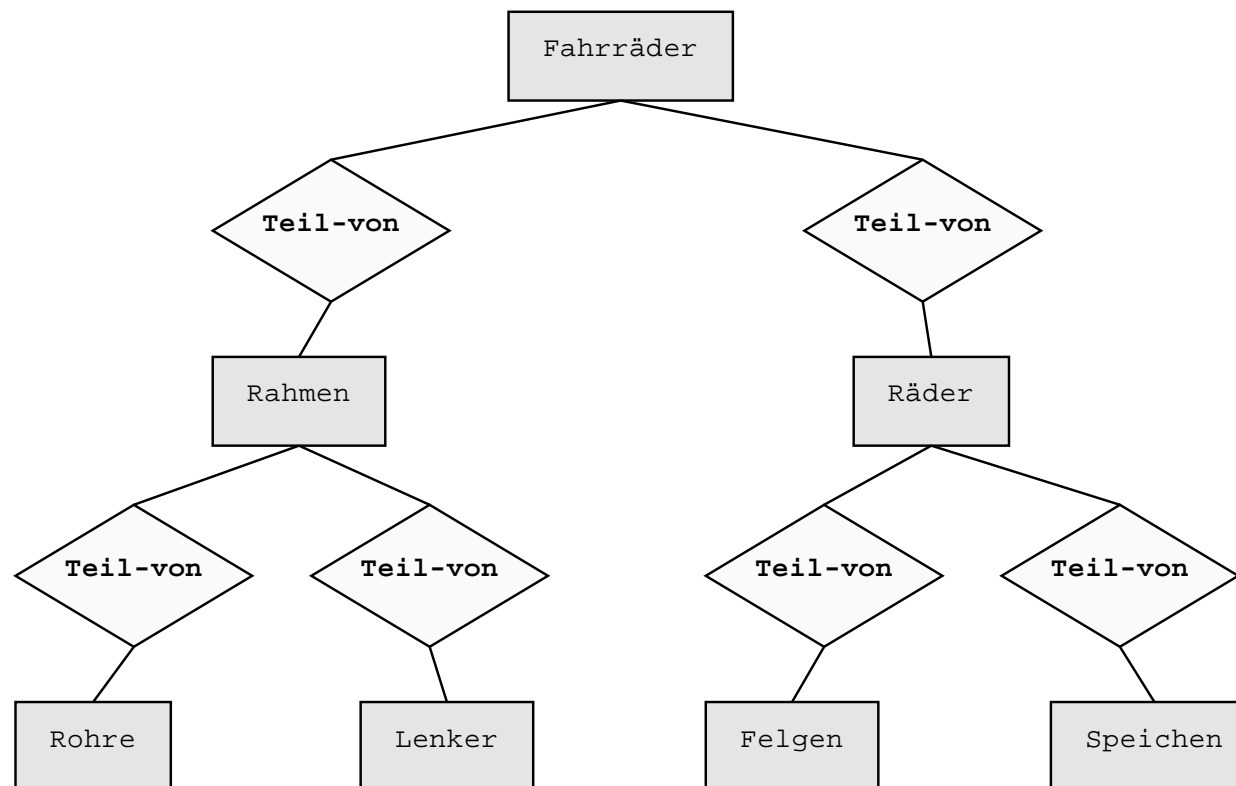
- disjunkte Generalisierung: $U_1 \cap U_2 = \emptyset$
- vollständige Generalisierung: $U_1 \cup U_2 = O$

Bsp: Universitätsschema mit (min, max) Notation und Generalisierung



Aggregation (EER)

Die **Aggregation** ordnet unterschiedliche Entitytypen, die in ihrer Gesamtheit einen strukturierten Objekttyp bilden, einander zu.



Sichtenintegration und Konsolidierung

Bei großen Anwendungen: Aufteilung der Anforderungsanalyse in verschiedene Sichten

Beispiel (Universität)

Mögliche unterschiedliche Sichten:

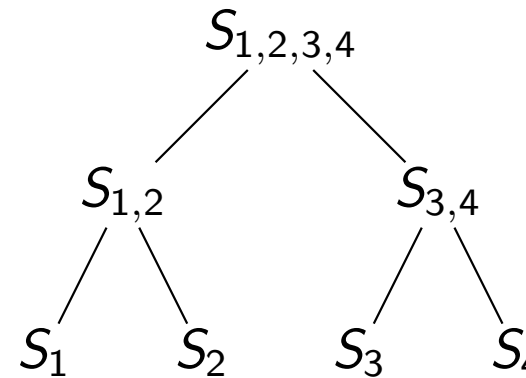
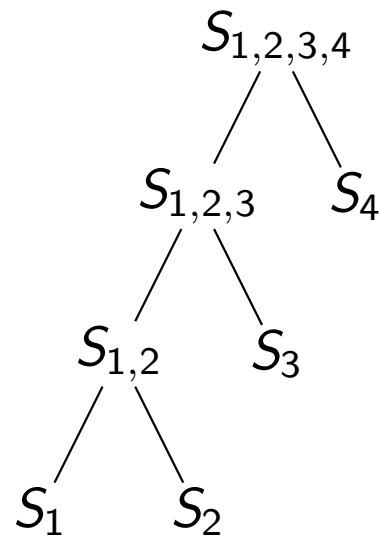
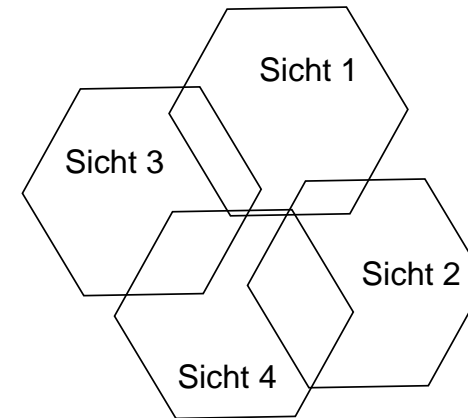
- Lehrendensicht
- Studierendensicht
- Verwaltungssicht
-

Konsolidierung: ein globales Schema wird erstellt, das redundanzfrei, widerspruchsfrei und um Synonyme und Homonyme bereinigt ist.

Konsolidierungsbaum

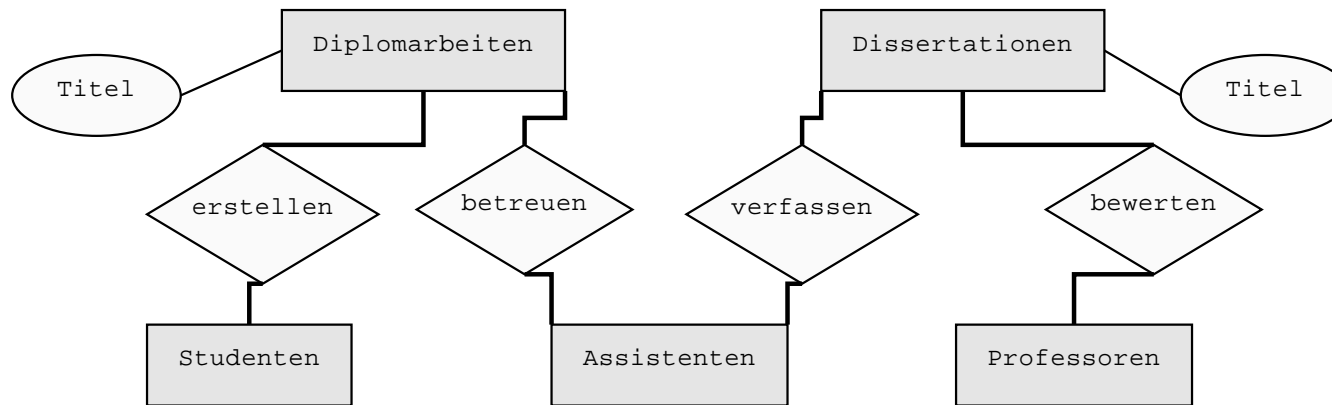
Mögliche Konsolidierungsbäume:

- maximal hoher Konsolidierungsbaum
- minimal hoher Konsolidierungsbaum

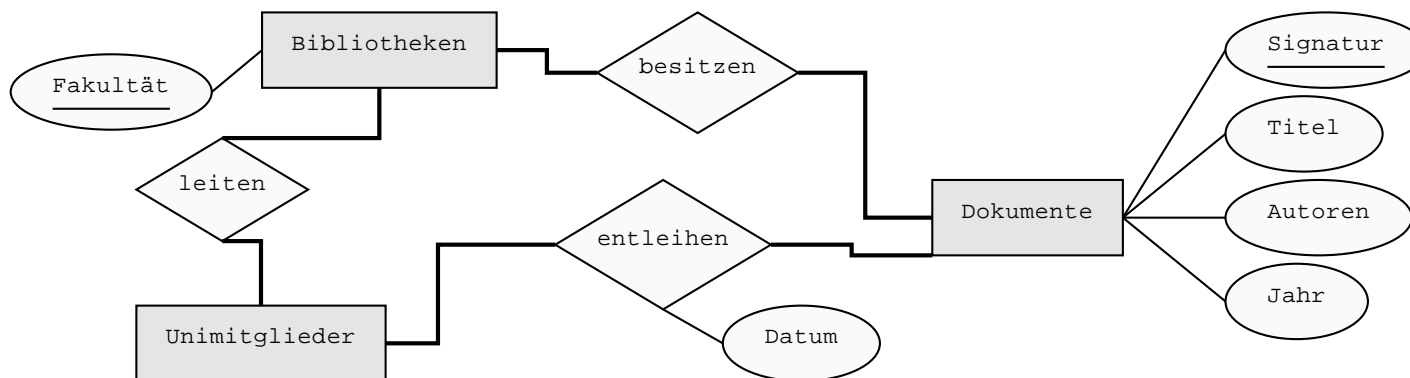


Bsp: Konsolidierung

Sicht 1: Dokumenterstellung als Prüfungsleistung

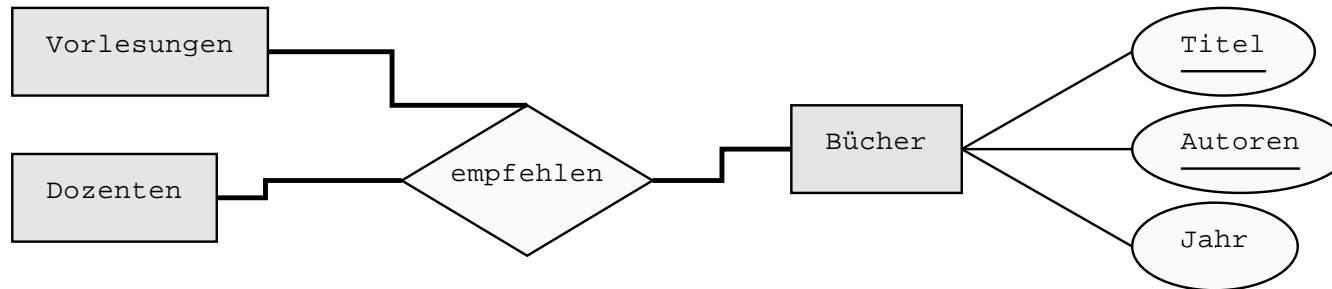


Sicht 2: Bibliotheksverwaltung



Bsp: Konsolidierung

Sicht 3: Buchempfehlung für Vorlesungen



- Dissertationen, Diplomarbeiten und Bücher sind Spezialisierungen von Dokumenten der Bibliotheken.
- Alle Dokumente werden durch die Signatur identifiziert.
- Unimitglieder ist Generalisierung von Studenten, Professoren und Assistenten.
- Synonyme Verwendung von Dozenten und Professoren.
- Alle Diplomarbeiten und Dissertationen werden in Bibliotheken verwaltet.
- Fakultätsbibliothek: geleitet von Angestellten - Revision von leiten bei Spezialisierung von Unimitglieder.
- Die Beziehungen erstellen und verfassen aus Sicht 1 entsprechen den Autoren von Büchern in Sicht 3.

Bsp: Konsolidierung

