

Inhalt

♦ Grundlagen	
♦ Allgemeines	2
♦ Grundlegende Modellierungsstrukturen (Entity, Relationship, Attribut, Schlüssel, Rollen)	3
♦ Beispiel	8
♦ Charakterisierung von Beziehungstypen	
♦ Allgemeines	9
♦ Funktionalitäten der Beziehungen	10
♦ Funktionalitätsangaben bei n-stelligen Beziehungen	12
♦ Beispiel	14
♦ Die (min,max)-Notation	15
♦ Existenzabhängige Entitytypen	17
♦ Generalisierung	
♦ Allgemeines	18
♦ Beispiele	20
♦ Aggregation	
♦ Allgemeines	22
♦ Beispiele	22

Grundlagen

Allgemeines

Das Entity-Relationship-Modell (ER-Modell) mit der graphischen Notation wurde 1976 in einem wegweisenden Aufsatz von Peter Chen vorgestellt.

Wie der Name schon sagt, modelliert dieses Modell Gegenstände (Entities) und die Beziehungen (Relationships) zwischen diesen.

Die in diesem Kapitel ebenfalls vorgestellten Konzepte der Generalisation und Aggregation, sowie die (*min,max*)-Notation wurden später dem ER-Modell hinzugefügt.

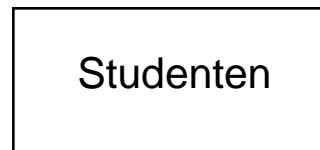
Viele Bücher und Werkzeuge (Editoren) verwenden eine andere graphische Notation. Die Konzepte (Entity, Relationship, Attribut, Schlüssel, Rolle) sind jedoch überall dieselben.

Grundlegende Modellierungsstrukturen [1]

Entity (Gegenstand):

Wohlunterscheidbare physisch oder gedanklich existierende Konzepte der zu modellierenden Welt. Ähnliche Entities (Entities mit denselben Attributen) werden zu Entitytypen (oder Entitätsmengen) abstrahiert, die man dann graphisch als Rechteck darstellt, wobei der Name innerhalb des Rechtecks angegeben wird.

Beisp.:

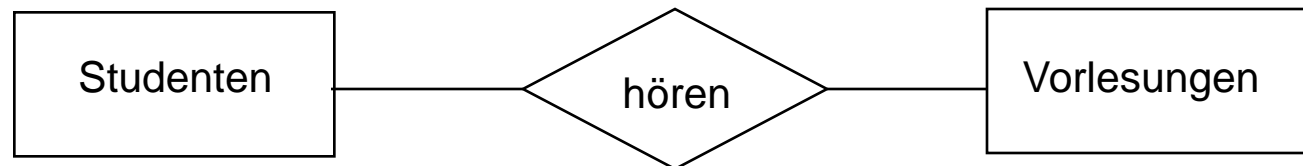


Grundlegende Modellierungsstrukturen [2]

Relationship (Beziehung):

Die Zusammenhänge zwischen den Entities werden analog zu Beziehungstypen zwischen den Entitytypen abstrahiert. Beziehungstypen werden als Rauten mit entsprechender Beschriftung dargestellt. Die Rauten werden dann mit den beteiligten Entitytypen über ungerichtete Kanten verbunden.

Beisp.:

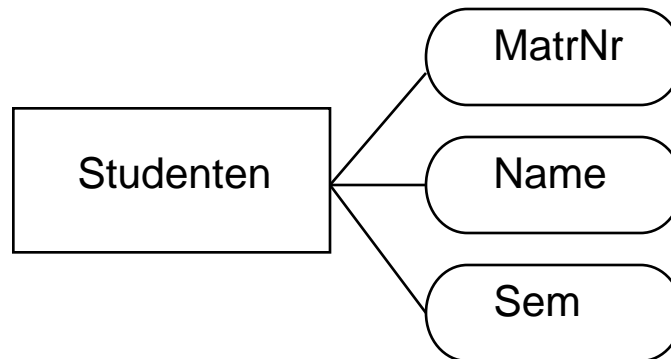


Grundlegende Modellierungsstrukturen [3]

Attribut:

Attribute dienen dazu, Gegenstände bzw. Beziehungen zu charakterisieren. Sie stellen die relevanten Merkmale dar und werden in Form von Kreisen oder Ovalen dargestellt und durch verbindende Linien den Entities und Relationships zugeordnet.

Beisp.:



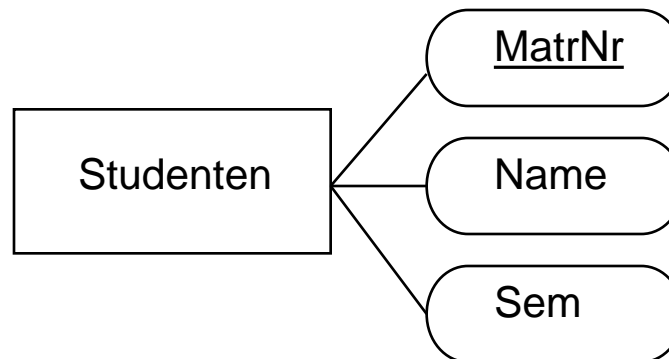
Grundlegende Modellierungsstrukturen [4]

Schlüssel:

Eine minimale Menge von Attributen, deren Werte das zugeordnete Entity eindeutig innerhalb aller Entities eines Typs identifiziert, nennt man Schlüssel. Sehr oft gibt es Attribute, die als Schlüssel ‚künstlich‘ eingebaut werden (z.B. Personalnummer). Schlüsselattribute werden meist durch Unterstreichung gekennzeichnet.

Es kann auch mehrere Schlüsselkandidaten geben. Dann wählt man einen dieser **Kandidaten-Schlüssel** als **Primärschlüssel** aus. (Siehe auch beim relationalen Datenmodell)

Beisp.:

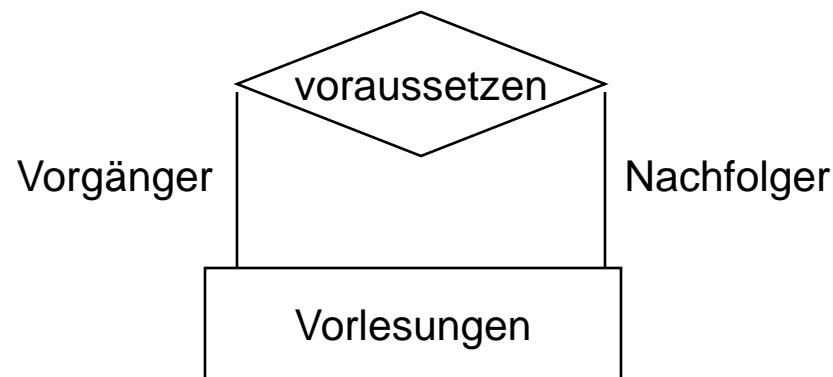


Grundlegende Modellierungsstrukturen [5]

Rollen:

Manchmal ist es notwendig, Beziehungen genauer zu beschreiben. (z.B. bei rekursiven Beziehungen). Sie können je nach Richtung, in die sie ‚gelesen‘ werden eine unterschiedliche Bedeutung haben. Diese Bedeutung (d.h. Rolle, die das betroffene Entity für diese Beziehung einnimmt) wird als Text an die ‚Ausgänge‘ (Kanten) der Beziehungsraute geschrieben.

Beisp.:



Beispiel

Ein konzeptuelles Schema eines Ausschnitts einer Universität

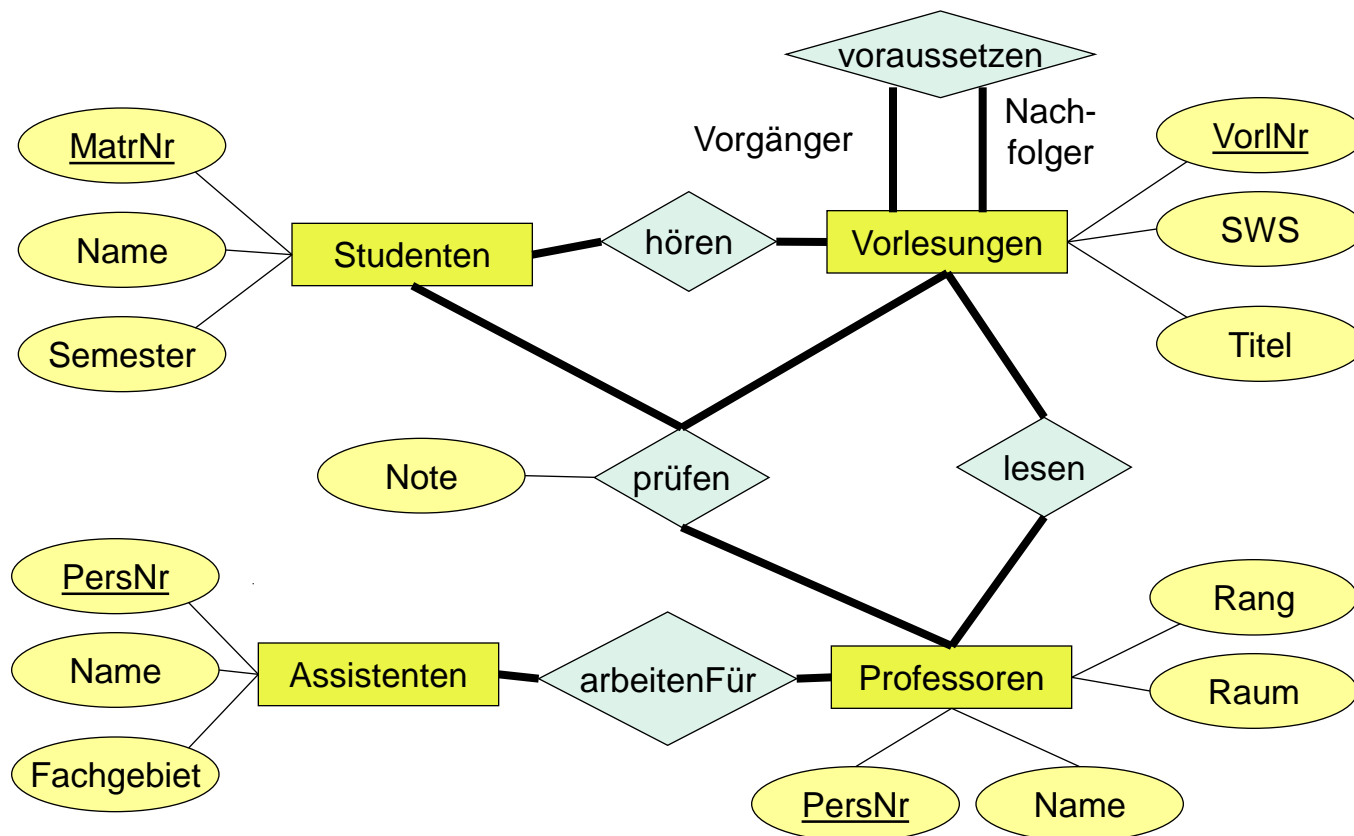


Abb.: Ein konzeptuelles Universitätsschema
(entnommen aus [1])

Charakterisierung von Beziehungstypen

Allgemeines

Ein Beziehungstyp R zwischen den Entitytypen E_1, E_2, \dots, E_n kann als Relation im mathematischen Sinn gesehen werden:

$$R \subseteq E_1 \times E_2 \times \dots \times E_n$$

In diesem Fall bezeichnet man n als den Grad der Beziehung R (meist ist $n = 2$ (binäre Beziehungen)).

Ein Element $(e_1, e_2, \dots, e_n) \in R$ nennt man die Instanz des Beziehungstyp R wobei $e_i \in E_i$ für alle $1 \leq i \leq n$ gelten muss.

Nun lässt sich auch der Begriff der Rolle etwas formaler fassen:

Z.B.: $\text{voraussetzen} \subseteq \text{Vorlesungen} \times \text{Vorlesungen}$

Um die einzelnen Instanzen $(v_1, v_2) \in \text{voraussetzen}$ genauer zu charakterisieren wird die jeweilige Rolle benötigt:

(Vorgänger : v_1 , Nachfolger : v_2)

Dadurch ist unmissverständlich festgelegt, dass v_1 der Vorgänger und v_2 der Nachfolger ist.

Charakterisierung von Beziehungstypen

Funktionalitäten der Beziehungen [1]

Man kann Beziehungstypen hinsichtlich ihrer Funktionalität charakterisieren: Ein binärer Beziehungstyp R zwischen den Entitytypen E_1 und E_2 heißt:

1:1-Beziehung, falls jedem Entity e_1 aus E_1 höchstens ein Entity e_2 aus E_2 zugeordnet ist und umgekehrt. (Es kann auch Entities aus E_1 (bzw. E_2) geben, die keinen ‚Partner‘ haben)

1:N-Beziehung, falls jedem Entity e_1 aus E_1 beliebig vielen (also mehreren oder auch keinem) Entities e_2 aus E_2 zugeordnet sein können, aber jedes Entity e_2 aus der Menge E_2 mit maximal einem Entity aus E_1 in Beziehung stehen kann.

N:M-Beziehung, wenn keinerlei Restriktionen gelten müssen.

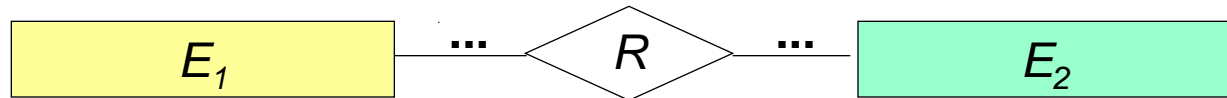
Funktionalitäten werden oft auch als **Kardinalitäten** bezeichnet.

Die binären 1:1 und 1:N - Beziehungen kann man auch als **partielle Funktionen** sehen. Bei der 1:N - Beziehung ist die ‚Richtung‘ bindend, von der ‚N-Seite‘ zur ‚1-Seite‘.

Achtung: Funktionalitäten stellen Integritätsbedingungen dar. Sie müssen in der zu modellierenden Welt immer gelten, nicht nur zum Zeitpunkt der Modellierung!

Charakterisierung von Beziehungstypen

Funktionalitäten der Beziehungen [2]



$$R \subseteq E_1 \times E_2$$

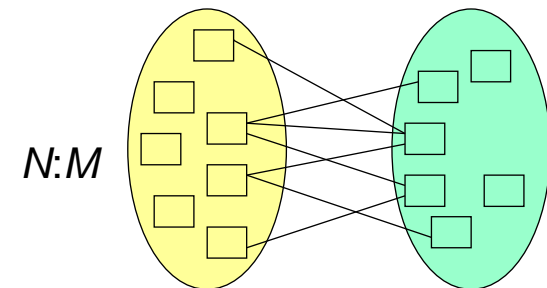
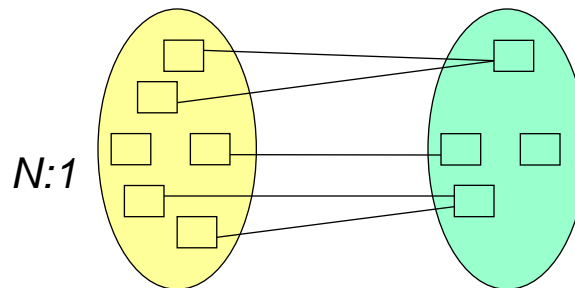
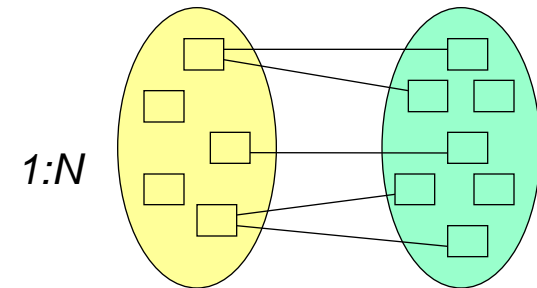
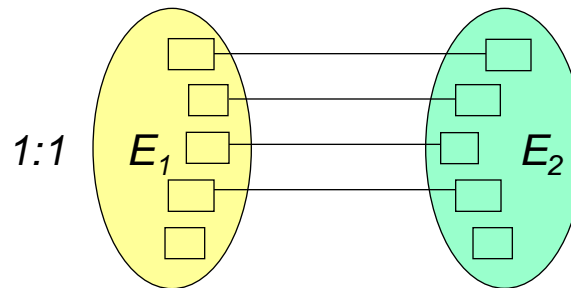


Abb.: Graphische Veranschaulichung von Funktionalitäten binäre Beziehungen
(entnommen aus [1])

Charakterisierung von Beziehungstypen

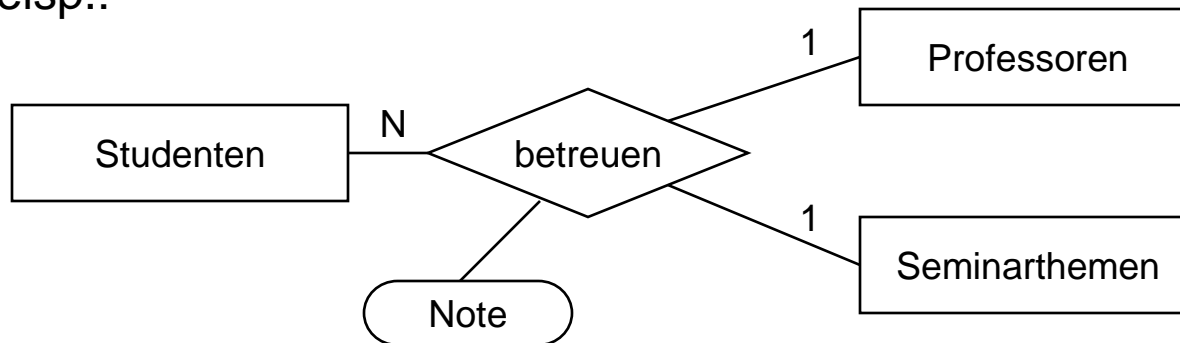
Funktionalitätsangaben bei n-stelligen Beziehungen [1]

Sei R eine Beziehung zwischen den Entitätsmengen E_1, E_2, \dots, E_n .

Die Funktionalität bei der Entitätsmenge E_k ($1 \leq k \leq n$) kann dann mit einer ‚1‘ spezifiziert werden, wenn durch R folgende partielle Funktion vorgegeben wird:

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

Beisp.:



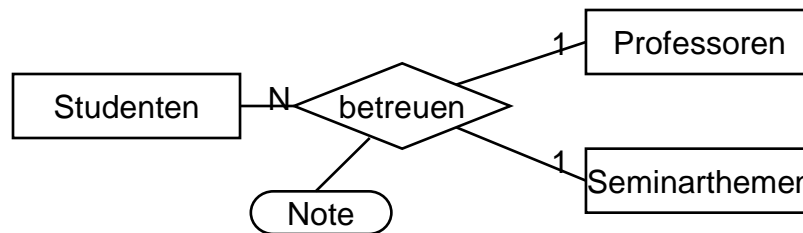
betreuen: Professoren \times Studenten \rightarrow Seminarthemen

betreuen: Seminarthemen \times Studenten \rightarrow Professoren

Charakterisierung von Beziehungstypen

Funktionalitätsangaben bei n-stelligen Beziehungen [2]

Beispiel, Fortsetzung:



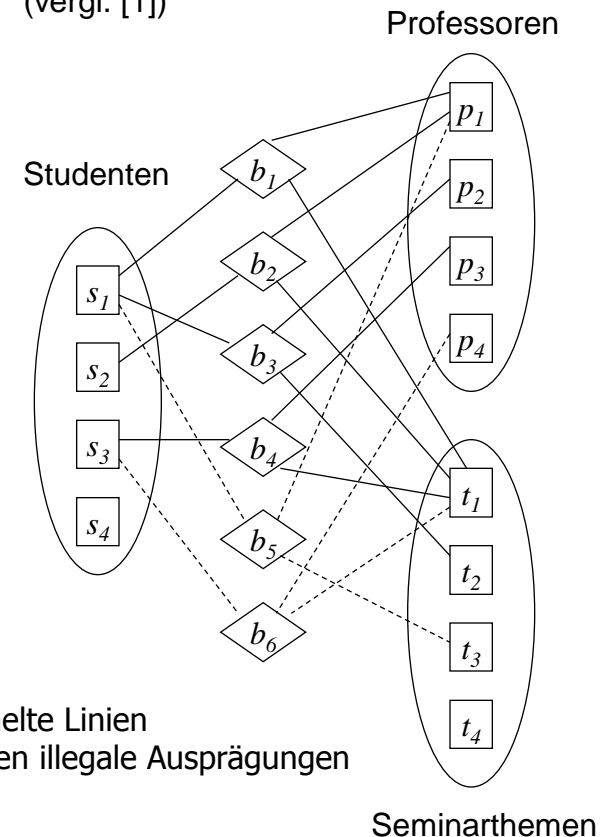
Diese Kardinalitätsangaben legen folgende Integritätsbedingungen fest:

- ◆ Studenten dürfen bei demselben Professor nur ein Seminarthema belegen.
- ◆ Studenten dürfen ein Seminarthema nur einmal bearbeiten.

Es ist aber nach wie vor möglich

- ◆ Professoren können dasselbe Thema mehrmals vergeben.
- ◆ Ein Thema kann von mehreren Professoren vergeben werden.

Mögliche Ausprägungen der Beziehung „betreuen“:
(vergl. [1])



Gestrichelte Linien
markieren illegale Ausprägungen

Charakterisierung von Beziehungstypen

Beispiel

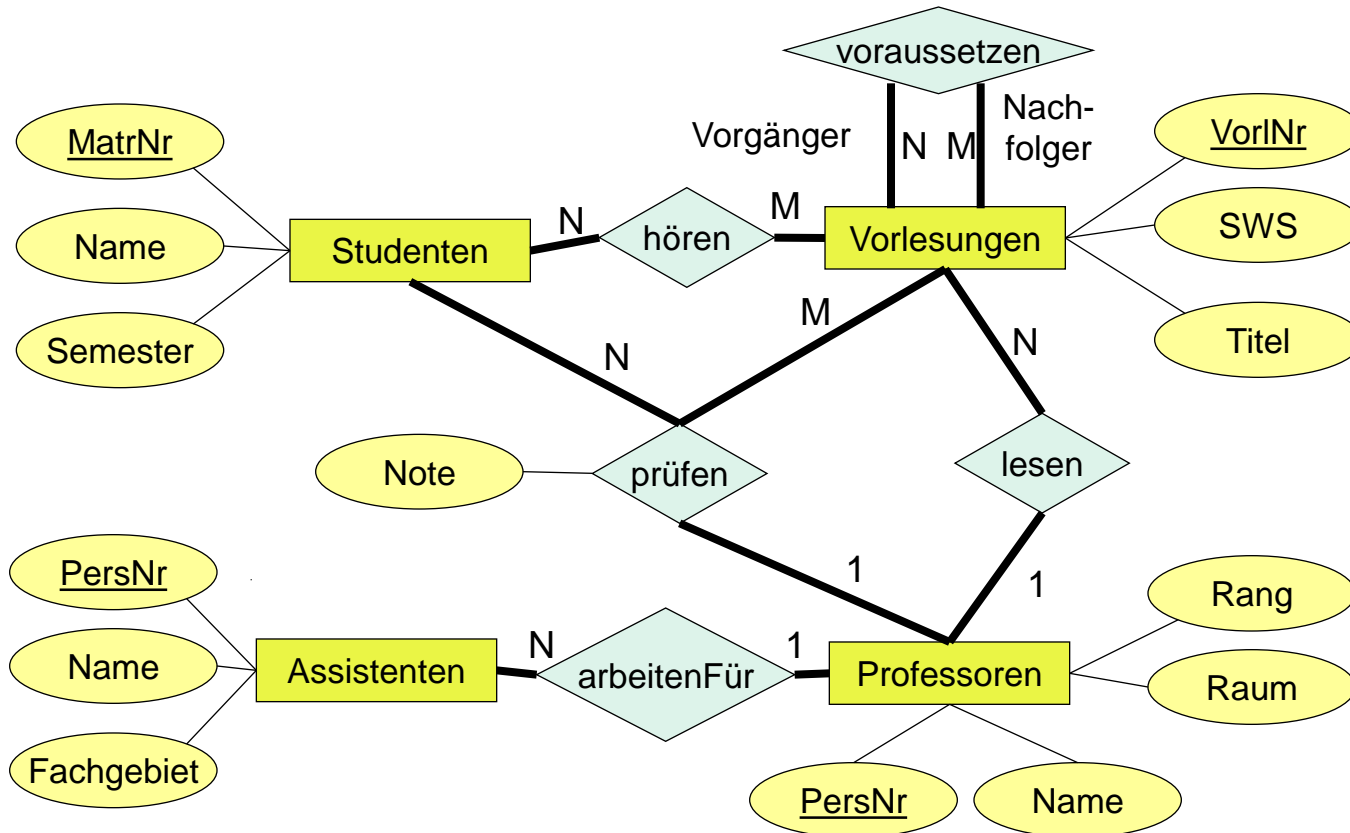


Abb.: Markierung der Funktionalitäten im Universitätsschema
(entnommen aus [1])

Charakterisierung von Beziehungstypen

Die (*min,max*) - Notation [1]

Diese Notation erlaubt es nicht nur die Extremwerte für die Funktionalität anzugeben (die 1:1, 1:N, N:M - Schreibweise unterscheidet nur zwischen „eins“ und „viele“), sondern, wann immer möglich, Ober- und Untergrenzen festzulegen.

Das Zahlenpaar bei der (*min,max*)-Notation besagt, dass jedes Entity dieses Typs mindestens min-mal aber höchstens max-mal in Beziehung steht.

Formal: Sei R eine Relation zwischen den Entitymengen E_1, E_2, \dots, E_n .

$$R \subseteq E_1 \times E_2 \times \dots \times E_n$$

Die Markierung (min_i, max_i) gibt an, dass es für alle $e_i \in E_i$ mindestens min_i und höchstens max_i Tupel der Art (\dots, e_i, \dots) in R gibt.

Sonderfälle:

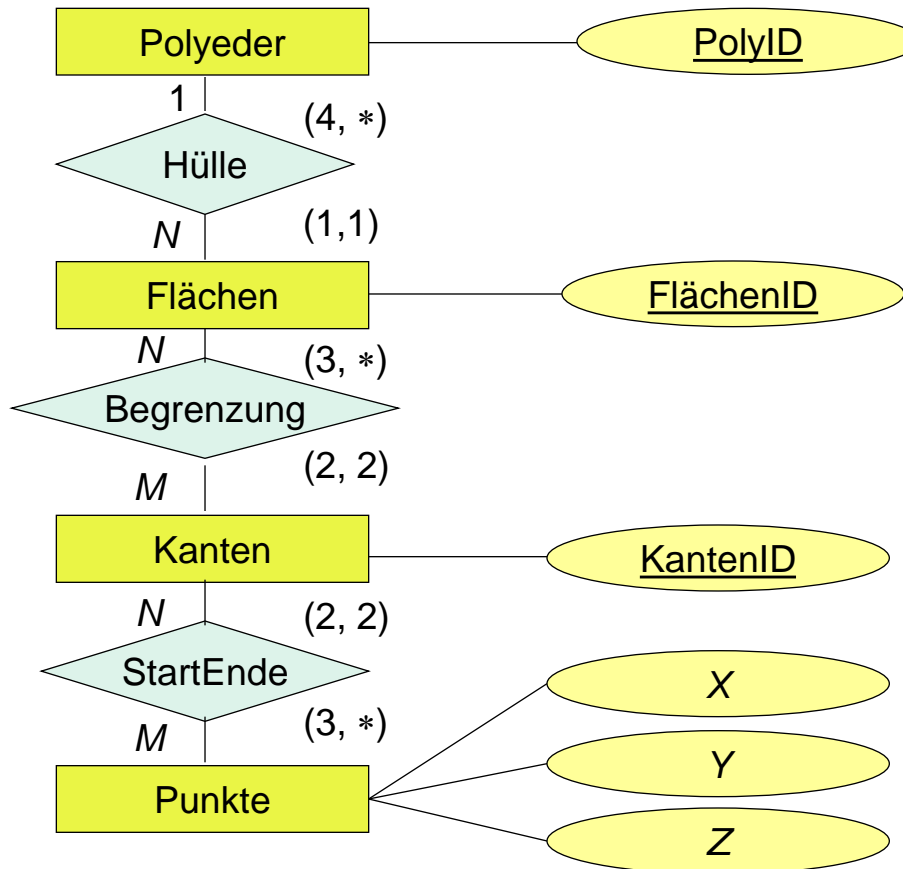
- ♦ Wenn es Entities geben darf, die gar nicht an der Beziehung „teilnehmen“, wird mit *min* 0 angegeben.
- ♦ Wenn ein Entity beliebig oft „teilnehmen“ darf, wird die *max*-Angabe durch * ersetzt.

Die Angabe (0,*) ist somit die Allgemeinste.

Charakterisierung von Beziehungstypen

Die (min,max) - Notation [2]

Beispiel: Konzeptuelles Schema der Begrenzungsflächendarstellung von Poyedern (vergl. [1])



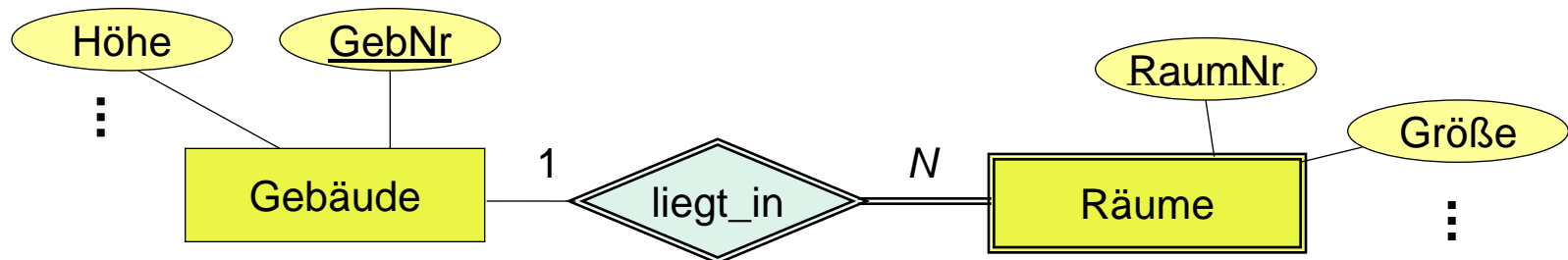
Existenzabhängige Entitytypen

Existenzabhängige Entities (schwache Entities) sind

- ♦ in ihrer Existenz von einem anderen, übergeordneten Entity abhängig und
- ♦ oft nur in Kombination mit dem Schlüssel des übergeordneten Entitys eindeutig identifizierbar.

Schwache Entities werden durch doppelt umrandete Rechtecke repräsentiert. Die Beziehung zu dem übergeordneten Entitytyp und die Kante dazwischen wird ebenfalls durch eine Verdoppelung der Linie markiert.

Beispiel:



Die Beziehung zum übergeordneten Objekt hat im Allgemeinen eine 1:N-Funktionalität, selten eine 1:1.

Generalisierung

Allgemein [1]

Bei der Generalisierung im ER-Modell handelt es sich um eine **Abstraktion auf Typebene** (Die analoge Abstraktion auf Instanzebene bestand ja gerade darin, ähnliche Entities in Form von Entitytypen zu modellieren).

Die Eigenschaften ähnlicher Entitytypen (es sind dies die Attribute und Beziehungen) werden „herausfaktoriert“ und einem gemeinsamen **Obertyp** zugeordnet. Die ähnlichen Entitytypen aus denen die Eigenschaften herausfaktoriert wurden, heißen **Untertypen** (oder Kategorien) des generalisierten Obertyps.

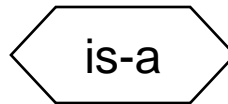
Diejenigen Eigenschaften, die nicht herausfaktorisierbar sind, weil sie nicht allen Untertypen gemein sind, verbleiben im jeweiligen Untertyp. Also stellt ein Untertyp eine **Spezialisierung** des Obertyps dar.

Ein Schlüsselkonzept ist die **Vererbung**. Ein Untertyp erbt sämtliche Eigenschaften des Obertyps.

Generalisierung

Allgemein [2]

Die Generalisierung wird graphisch als ein Sechseck mit der Text „is-a“ in der Mitte dargestellt:



Auf Instanzebene hat die Generalisierung folgende Auswirkungen: Die Entities eines Untertyps werden implizit auch als Entities (Elemente) des Obertyps betrachtet. Somit ist die Entitymenge des Untertyps eine Teilmenge der Entitymenge des Obertyps. Hinsichtlich der Teilmengensicht sind bei der Generalisierung, bzw. Spezialisierung zwei Fälle von besonderem Interesse:

- ♦ **Disjunkte Spezialisierung:** Die Entitymengen aller Untertypen eines Obertyps sind paarweise disjunkt.
- ♦ **Vollständige Spezialisierung:** Die Entitymenge des Obertyps enthält keine direkten Elemente - sie ergibt sich also aus der Vereinigung der Entitymengen der Untertypen.

Beispiele [1]

Generalisierung der Universitätsmitglieder

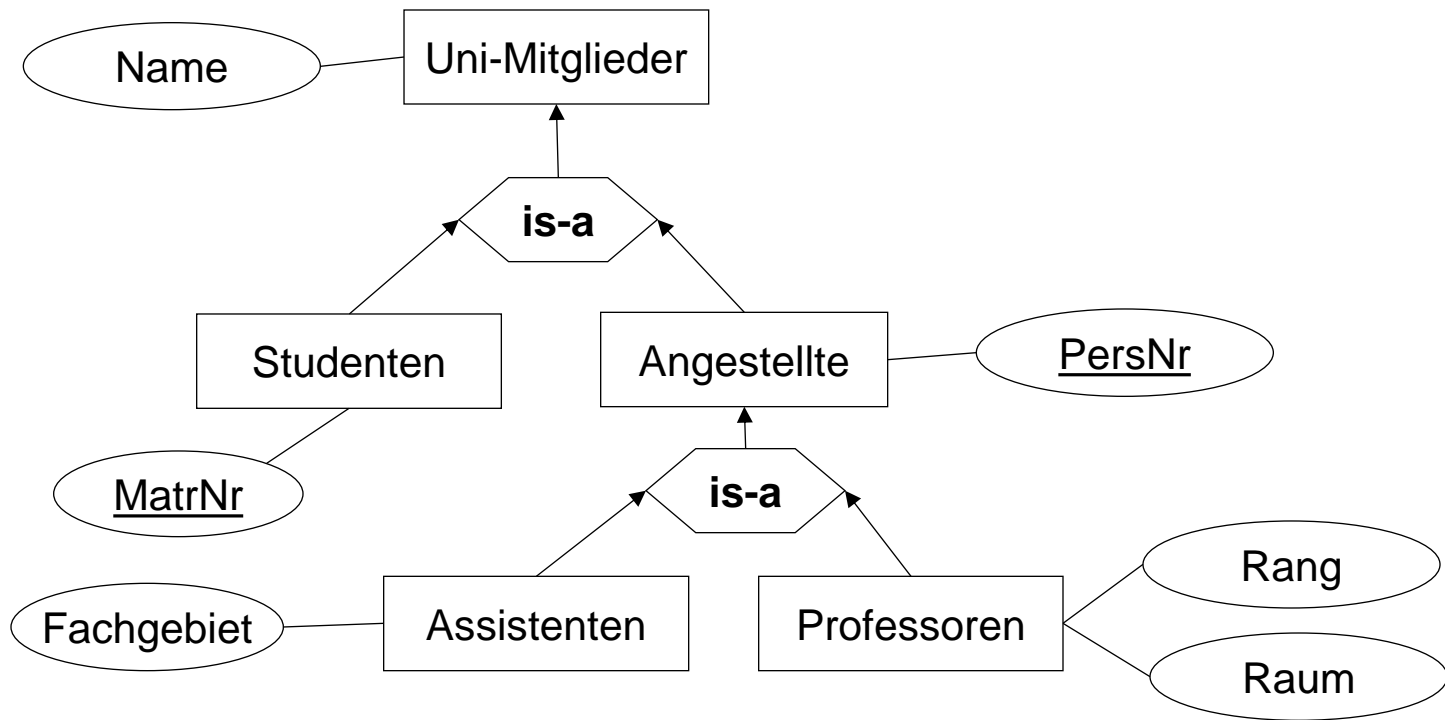
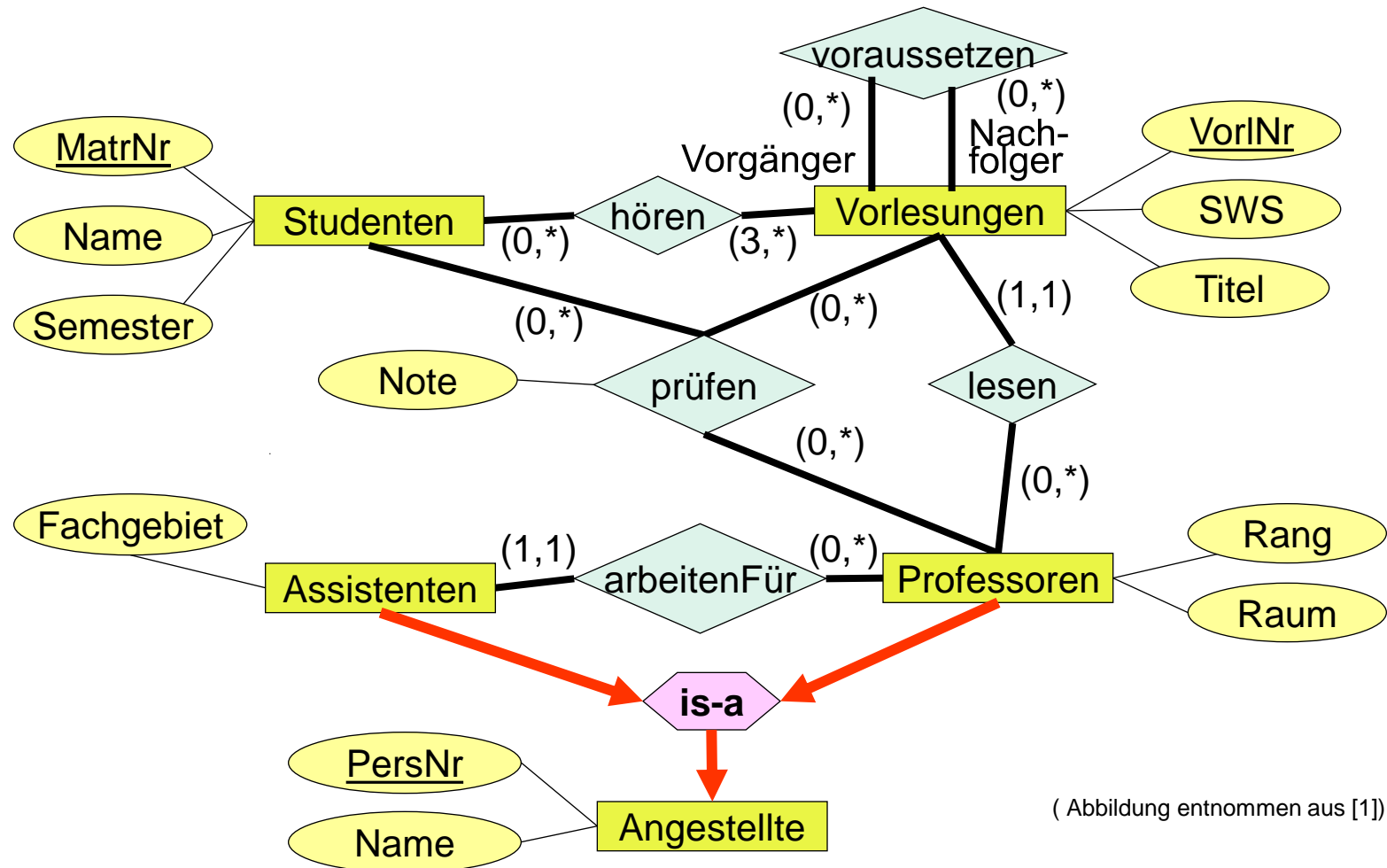


Abb.: Generalisierung der Universitätsmitglieder
(entnommen aus [1])

Beispiele [2]

Beispielschema der Universität mit (min,max) -Notation und Generalisierung



(Abbildung entnommen aus [1])

Aggregation

Allgemein

Bei der Aggregation werden Entitytypen, die in ihrer Gesamtheit einen strukturierten Objekttypen bilden, einander zugeordnet. Es gibt also auch hier einen übergeordneten Objekttyp, dem ein oder mehrere Objekttypen untergeordnet sind.

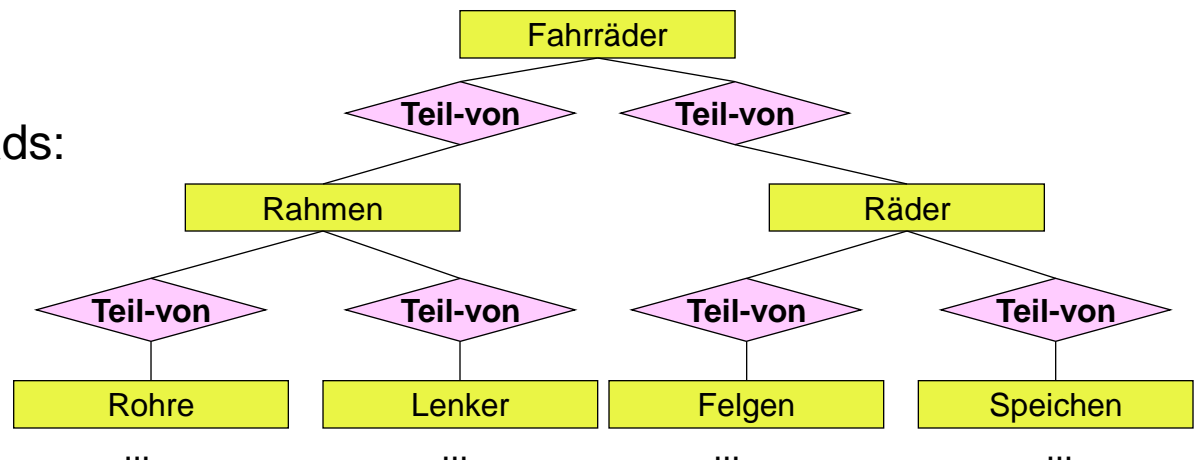
Diese Art von Beziehungen wird als **Teil-von (part-of)** bezeichnet. Graphisch wird sie als eine Raute mit dem Text Teil-von (oder part-of) in der Mitte dargestellt:



Beispiele [1]

Aufbau eines Fahrrads:

(Abbildung entnommen aus [1])



Aggregation

Beispiele [2]

Zusammenspiel von Generalisierung und Aggregation

