

Informatik ist eine  
Strukturwissenschaft

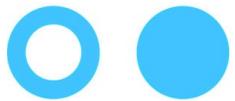
## Teil II

Von Daten und ihren Modellen

Robert Hartmann (SoSe 2024)

basierend auf Folien von  
Prof. Dr. Harm Knolle

Fachbereich Informatik  
Hochschule Bonn-Rhein-Sieg



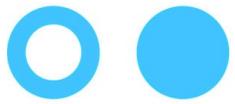
## - Kapitel 3 - Semantische Datenmodelle -

### Inhalt

- 0 - Vorbemerkungen
- Teil I - Von EDV-Anwendungen und Ihren Anforderungen
- 1 - Einführung
- Teil II - Von Daten und ihren Modellen
- 2 - Prozess des Datenbankentwurfs
- 3 - Semantische Datenmodelle
- 4 - Logische Datenmodelle
- 5 - Datenbankmodelle
- 6 - Datenanfrage und Datenänderung
- Teil III - Von Datenbanken und ihren Systemen
- 7 - Datenbanksysteme
- 8 - Speicherstrukturen
- 9 - Ausblick

### Überblick

- ♦ Einführung
- ♦ Entity Relationship Modell (ERM)
- ♦ Erweiterungen des ERM
- ♦ Zeit und Versionen
- ♦ Objekt-orientierte Modelle



## - Semantische Datenmodelle -

### Ziel

- ◆ Umsetzung der "Miniwelt" in eine formale, maschinenlesbare Beschreibung
- ◆ Konzeptionelle Strukturierung / Modellierung / Ordnung von Informationen
- ◆ Vermeidung von Redundanzen und Inkonsistenzen
- ◆ Anwendung der Konzepte semantischer Datenmodelle am Beispiel des Entity-Relationship Modells (ERM)

### Hilfsmittel

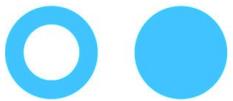
- ◆ Semantische Datenmodelle am Beispiel des ERM

### Inhalt

- ◆ Einführung
- ◆ Entity Relationship Modell (ERM)
- ◆ Erweiterungen des ERM
- ◆ Objekt-orientierte Modelle

### Literatur

- ◆ KeEi15, Kapitel 2 „Datenbankentwurf“
  - Abschnitte 2.5 – 2.12
- ◆ Ku15, Kapitel 2: „Informationsmodellierung“
  - Abschnitte 2.2 – 2.3.4
- ◆ SSH18, Kapitel 4: „Das Entity-Relationship-Modell“



## - Einführung -

### Ausgangssituation

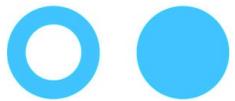
- ♦ Merkmale der Realität bzw. der Miniwelt
  - komplex strukturierte Objekte variabler Größe
  - zahlreiche semantisch unterschiedliche Beziehungen zwischen den Objekten wie
    - Generalisierung / Spezialisierung
    - Aggregation (zusammengesetzte Objekte)
    - Assoziation (Vereinigung von Objekten)
    - Versionen / Alternativen
- ♦ Merkmale kommerzieller Datenbankmodelle
  - Speicherorientiert:
    - die Objekte haben eine feste Länge
  - atomare Felder
    - die Merkmale eines Objekts sind kurz (einige Bytes), haben eine feste Länge und sind nicht weiter unterteilbar

### Folge

- ♦ Gegebenheiten der Realität lassen sich mit den Datenmodellen heutiger Datenbanksysteme nur wenig ausdrucksstark darstellen
- ♦ vielfach sind nur systemnahe (relationale) Beschreibung der Miniwelt möglich
- ♦ Semantik der Realität geht verloren oder ist auf den ersten Blick nicht mehr zu erkennen

### Entwicklung semantischer Datenmodelle

- ♦ zur datenbankunabhängigen formalen Beschreibung der Miniwelt
- ♦ 1976: Entity Relationship Modell (ERM)
- ♦ 1980: Erweitertes ERM
- ♦ 1983: Semantische Datenmodelle
- ♦ 1985: Objekt-orientierte Modelle
- ♦ 1990: Objektrelationale Modelle
- ♦ heute: noch keine "überzeugenden" rein objekt-orientierte Datenbanksysteme



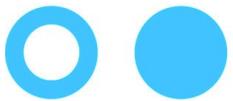
## - Entity Relationship Modell (ERM) -

### Inhalt

- ◆ Einführung
- ◆ Entity Relationship Modell (ERM)
- ◆ Erweiterungen des ERM
- ◆ Zeit und Versionen
- ◆ Objekt-orientierte Modelle

### Überblick

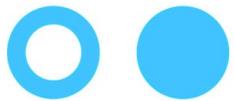
- ◆ Schema der Miniwelt (Diskursbereich)
- ◆ Ziel: Schema in Diagrammform
- ◆ Historie und Notation
- ◆ Integritätsbedingungen
- ◆ Entity
- ◆ Relationship
- ◆ Kardinalität
- ◆ Attribut
- ◆ Abhängigkeiten
- ◆ Abhängigkeiten von Beziehungen



## - Schema der Miniwelt (Diskursbereich) (I) -

### Merkmale der Dinge im Beispiel Lehrbetrieb (Ausschnitt)

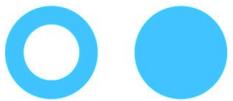
- ♦ Beschrieben werden sollen zudem die **Merkmale** der "Phänomene" bzw. "Dinge"
  - Fachbereiche (**Name**, Dekan, **Studenten**, **Professoren**, ...)
  - Lehrveranstaltungen (**Name**, **Fachbereich**, **SWS**, ...)
  - Professoren (**Name**, **Gehalt**, **Fachbereich**, **Fachgebiet**, ...)
  - Dozenten (**Name**, **Fachgebiet**, ...)
  - Angestellte / Mitarbeiter (**Name**, **Fachbereich**, **Anschrift**, ...)
  - Studenten (**Name**, **Anschrift**, **Haarfarbe**, **Fachbereich**, ...)
  - Bücher (**ISBN**, **Titel**, ...)
  - Prüfungen (**Student**, **Lehrveranstaltung**, **Professor**, **Datum**, **Note**, ...)
  - ...
- ♦ **Gehalt** und **Haarfarbe** gehören nicht zum Diskursbereich des Lehrbetriebs
  - (vielleicht aber zu anderen Anwendungsszenarien)
- ♦ **Beziehungsmerkmale** (Zusammenspiel) zu anderen **eigenständigen Dingen** werden gesondert behandelt



## - Schema der Miniwelt (Diskursbereich) (II) -

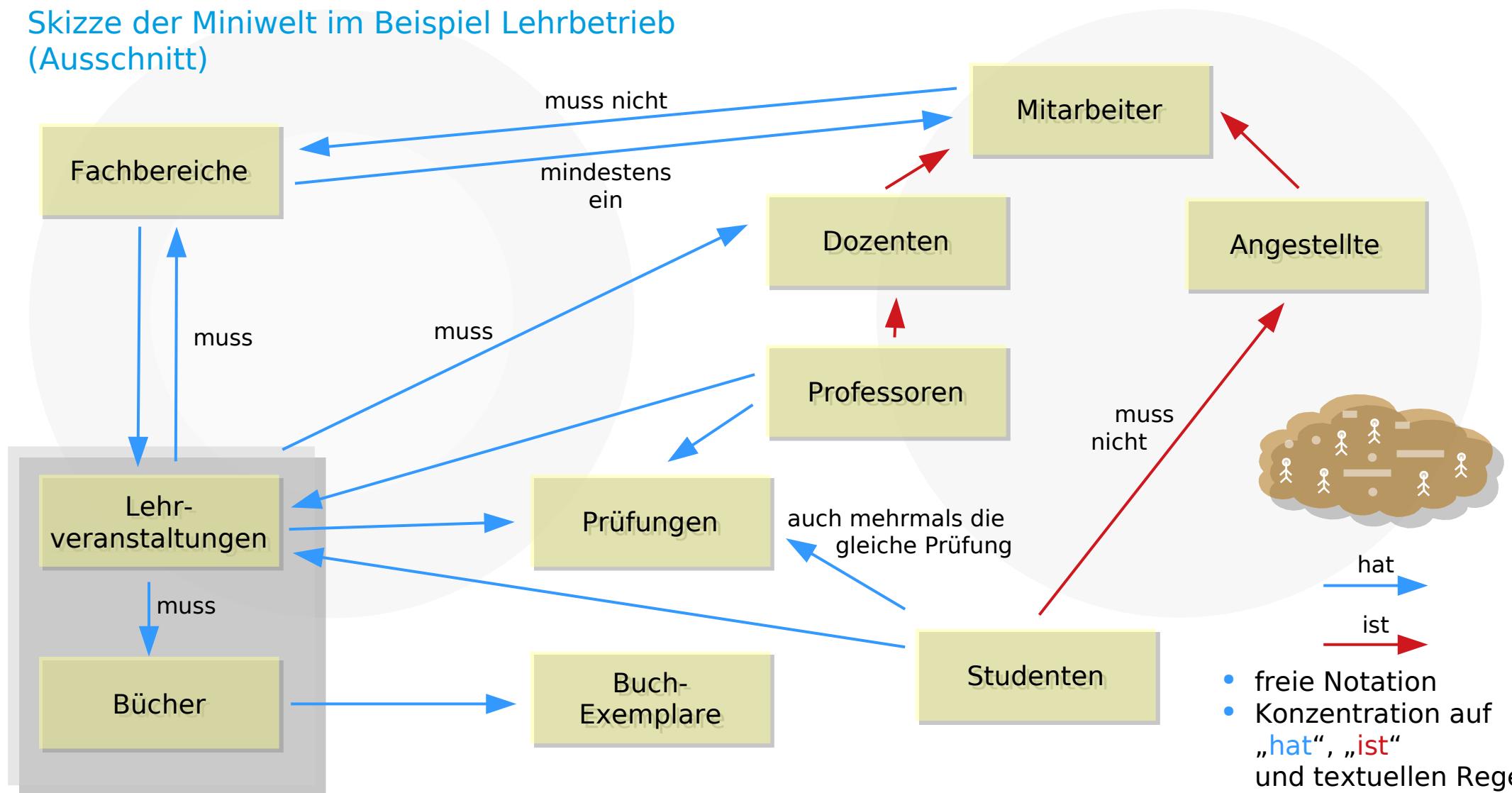
### Zusammenspiel der Dinge im Beispiel Lehrbetrieb (Ausschnitt)

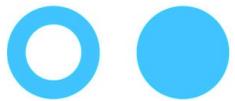
- ♦ Beschrieben werden sollen zudem die Regeln des sinnvollen Zusammenspiels zwischen den "Phänomenen" bzw. "Dingen"
  - Fachbereiche bieten Lehrveranstaltungen an
  - jeder Fachbereich **muss** mindestens einen Mitarbeiter haben
  - Mitarbeiter sind Profs, Angestellte und Studenten
  - Professoren halten Lehrveranstaltungen
  - Studenten belegen Lehrveranstaltungen
  - Professoren prüfen Studenten über Lehrveranstaltungen
  - Studenten können **auch mehrmals** über die gleiche Lehrveranstaltung geprüft werden
- jede Lehrveranstaltung **muss** zu einem Fachbereich gehören, und **muss** von einem Dozenten gehalten werden
- Mitarbeiter **müssen nicht** unbedingt zu Fachbereichen gehörenStudenten **können** gleichzeitig Angestellte sein
- zu **jeder** Lehrveranstaltung gibt es Buchempfehlungen
- Bücher können mehrere Exemplare haben
- ♦ Fragen
  - erfordert die Existenz eines Dings immer eine Beziehung zu anderen Dingen oder kann ein Ding auch alleine existieren?
  - kann ein Ding auch mehrere Beziehungen zu Dingen desselben Typs haben?
  - Ziel: Erkennung und Vermeidung unsinniger Zustände in der späteren Datenbank



## - Schema der Miniwelt (Diskursbereich) (III) -

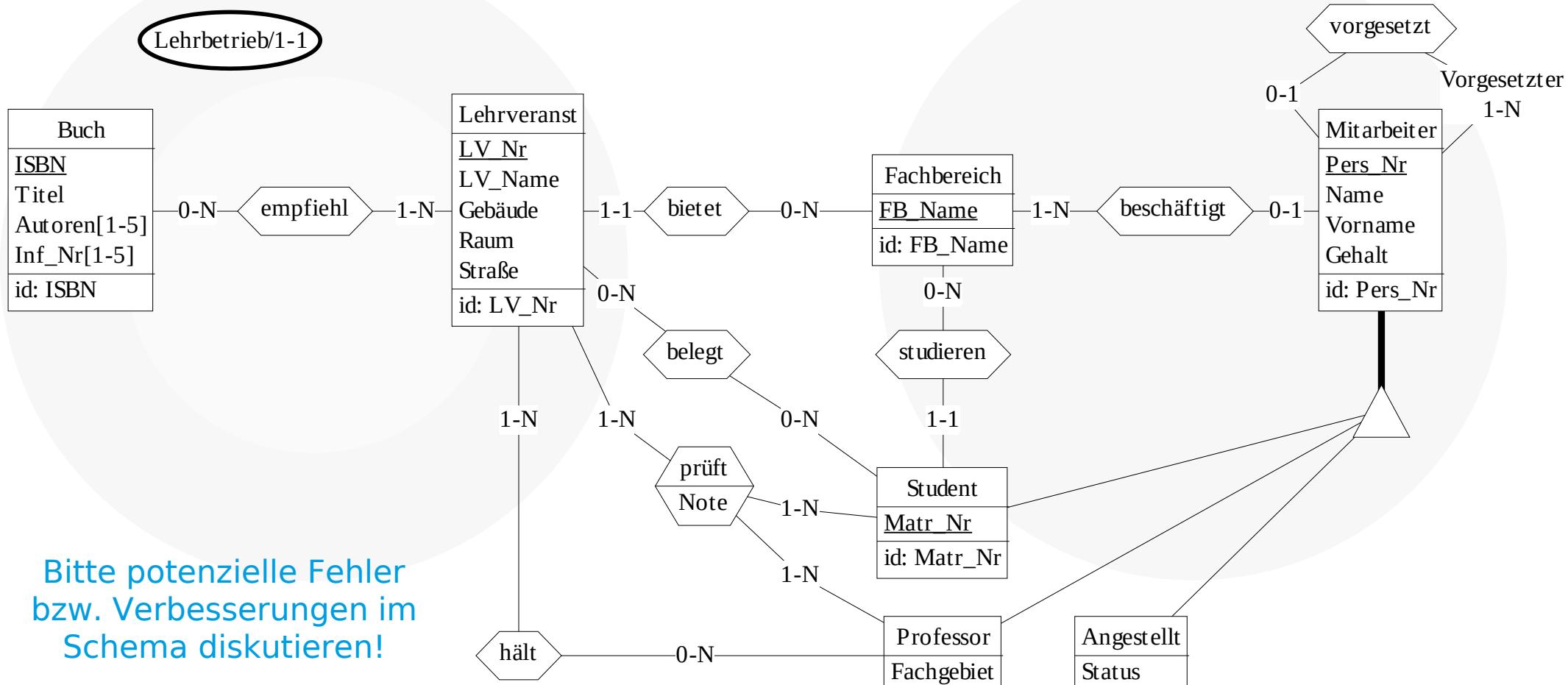
Skizze der Miniwelt im Beispiel Lehrbetrieb  
(Ausschnitt)

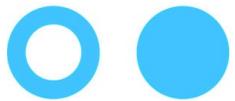




## - Ziel: Schema in Diagrammform -

Erstellt mit dem CASE-Werkzeug DB-MAIN





## - Historie und Notation -

### Historie

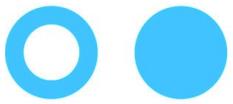
- ◆ 1976 vorgeschlagen von CHEN
- ◆ seit dieser Zeit feste Etablierung im Bereich der Datenmodelle
- ◆ praktisch Standard für die frühe Entwurfsphase der Datenbankentwicklung
- ◆ wird inzwischen von fast allen Werkzeugen des Computer Aided Software Engineering (CASE) unterstützt
- ◆ heute viele Spielarten und Weiterentwicklungen des ERM
- ◆ Problem: zahlreich unterschiedliche (graphische) Notationsformen

### Ergänzung zur CHEN-Notation

- ◆ „min-max-Notation“ von Jean-Raymond Abrial (1974)
- ◆ Erhöhung der Ausdruckskraft bei binären Beziehungen

### Konzepte

- ◆ wenige Konstruktionselemente zur Beschreibung der Gegebenheiten einer Miniwelt
- ◆ Entity (Entität)
  - Duden Fremdwörterbuch: "Dasein im Unterschied zum Wesen eines Dinges"
  - Brockhaus: "die bestimmte Seinverfassung (Wesen) des einzelnen Seienden, auch diese selbst"
- ◆ Relationship (Beziehung)
  - Beziehungen zwischen den Entities
- ◆ Attribut
  - Eigenschaft von Entities oder Beziehungen
- ◆ Integritätsbedingungen
  - Regeln zur späteren Aufnahme von Daten in die Datenbank



## - Integritätsbedingungen -

### Allgemein

- ◆ Verhindern "unsinniger" Datenkonstellationen
- ◆ Verhindern / zurückweisen von Änderungsoperationen an den Werten einer Datenbank, die zu Verletzungen von Integritätsbedingungen führen würden

### Modellinhärente Integritätsbedingungen

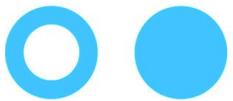
- ◆ durch Konzepte des Datenmodells ausdrückbar
- ◆ Darstellung von zwingenden oder optionalen Beziehungen
- ◆ Darstellung, wie oft ein Entity an einer Beziehung teilnehmen darf
- ◆ Beziehungen erfordern mindestens zwei Entities
- ◆ Eindeutige Identifizierbarkeit von Entities

### Zusätzliche Integritätsbedingungen

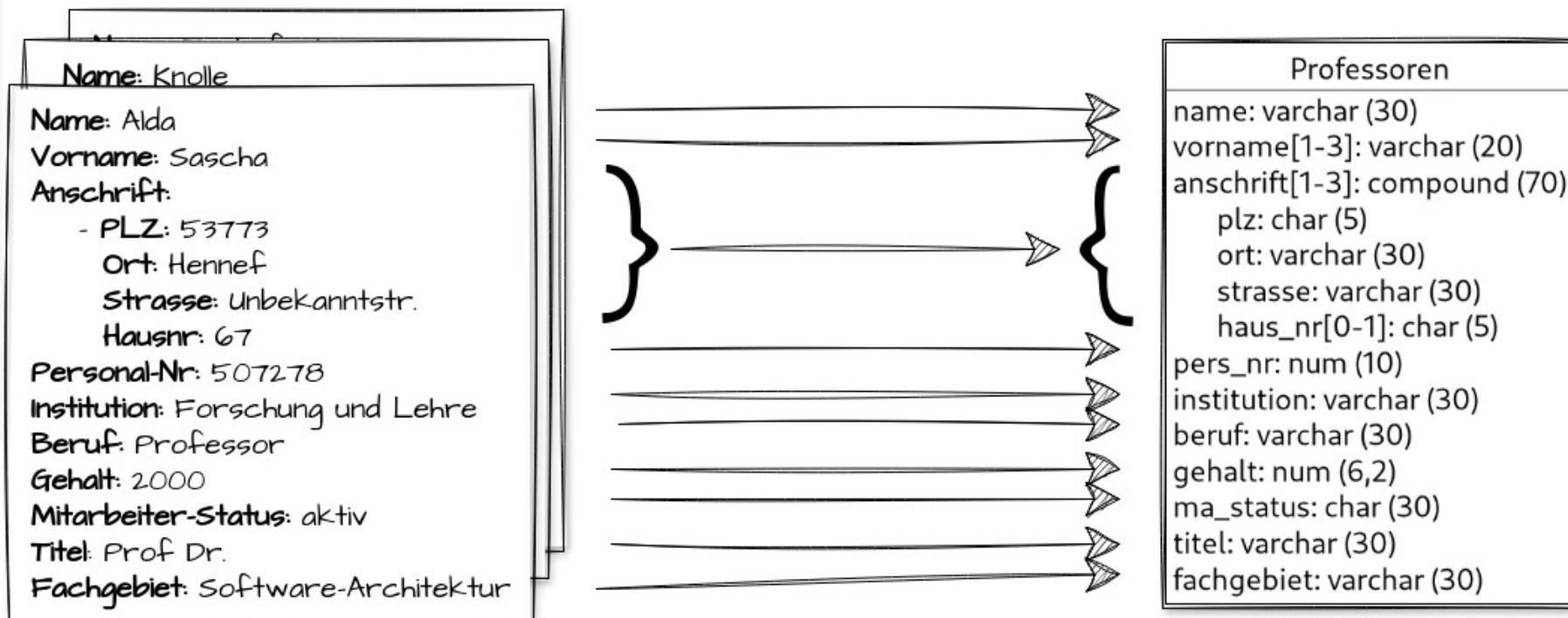
- ◆ nicht durch Konzepte des Datenmodells ausdrückbar
  - Aussagen, welche speziellen Werte eine Entity nicht annehmen darf
  - Aussagen über "entweder / oder"-Abhängigkeiten zwischen Ausprägungen der Entities
  - Mengenspezifische Angaben über Entities
- ◆ müssen gesondert formuliert werden
  - z.B. mit Konstrukten der Prädikatenlogik

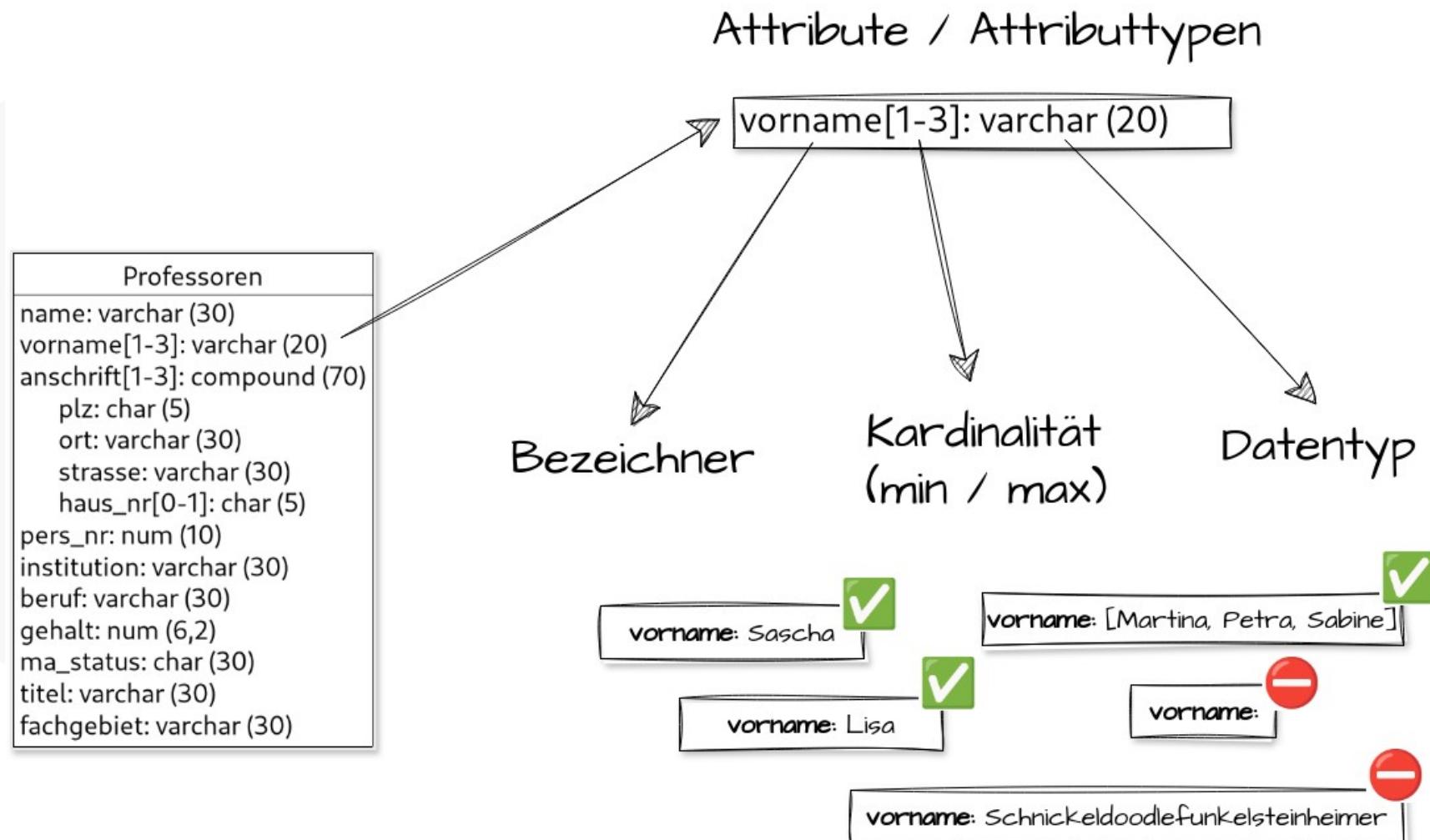
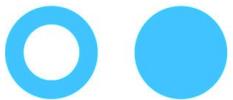
### Folge

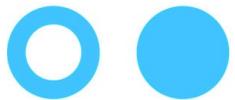
- ◆ Durch die Darstellung der "Miniwelt" mit dem ERM alleine geht Semantik verloren



- ... zum Definieren der Attribute und Wertebereiche
- ... Schablone, in die die Entitäten passen müssen
- ... ähnlich wie eine Java-Klasse







**Zeichenketten:** char(n) / varchar(n)

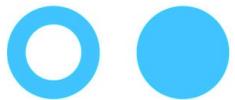
**Datum- und Zeitwerte:** date

**Zahlen:** numeric (n,d)

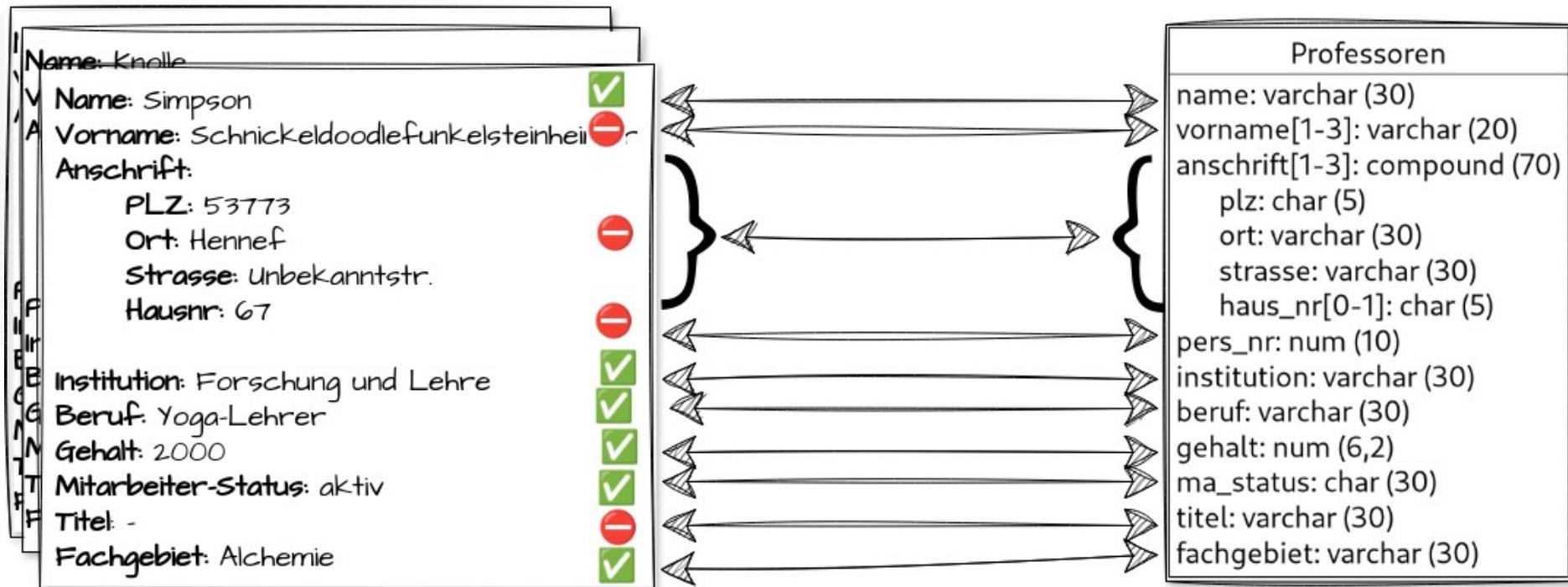
**Wahrheitswerte:** boolean

**Komplexe Werte:** compound

- ... Definieren mögliche Wertebereiche
- ... Festlegung sinnvoll für optimierte Speicherung
- ... Siehe auch Übungsblatt 2, Abschnitt 2.3

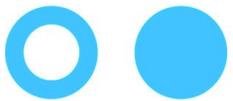


- ... Kardinalitäten müssen passen (min / max)
- ... Werte müssen zu Datentypen passen
- ... Unbekannte Attribute sind nicht erlaubt

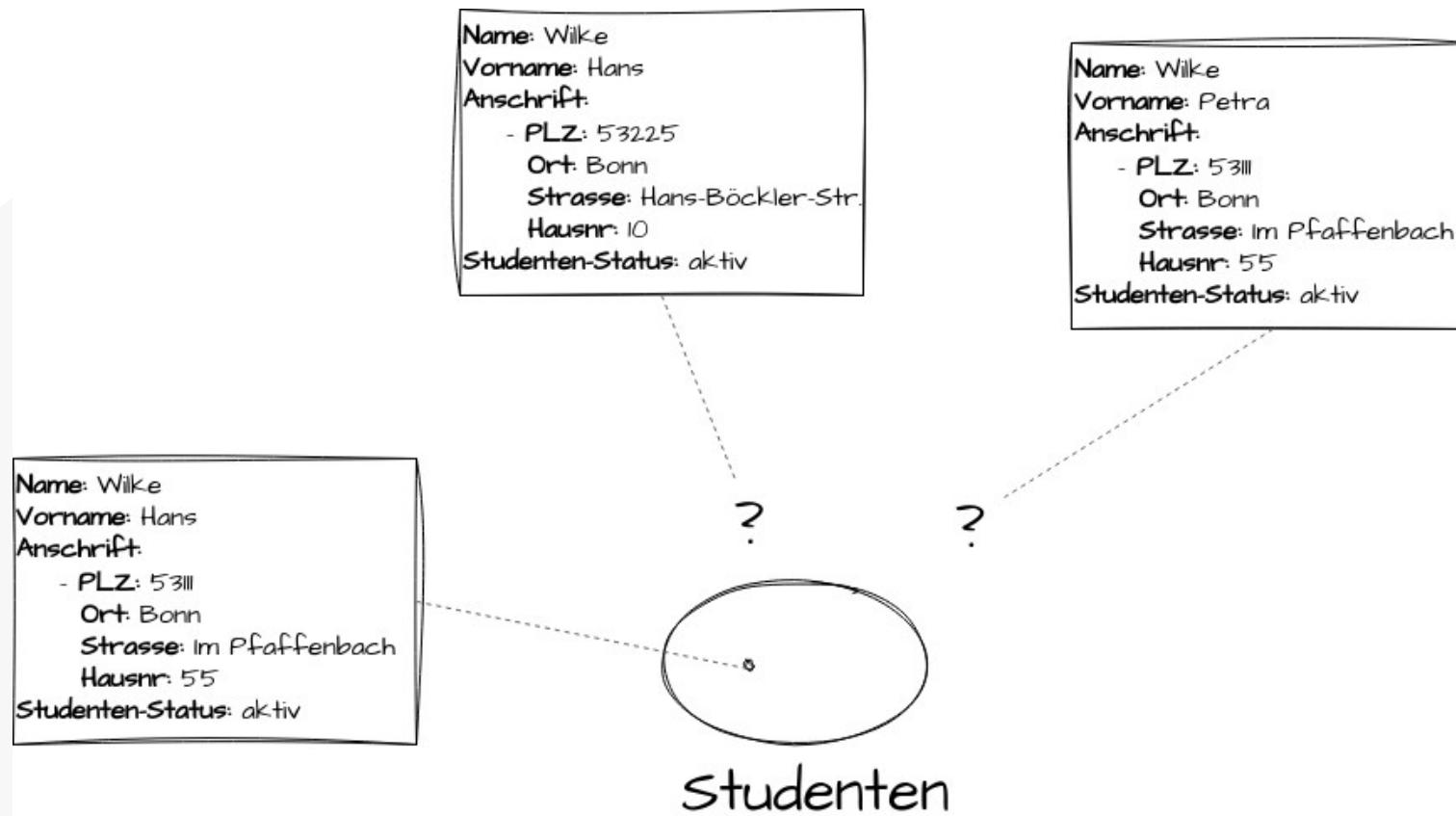


Entitäten

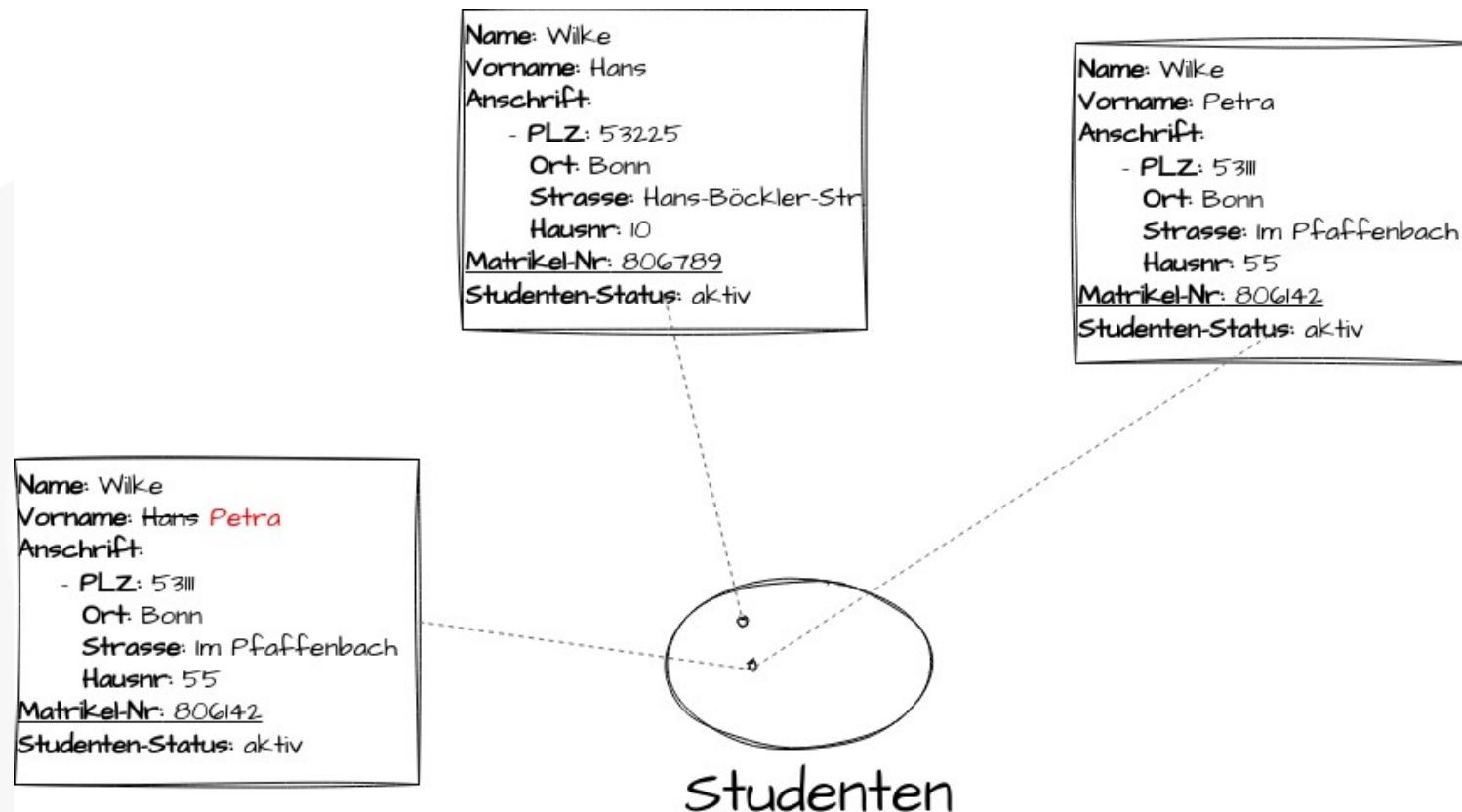
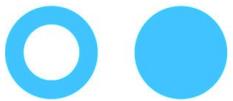
Entitätstyp



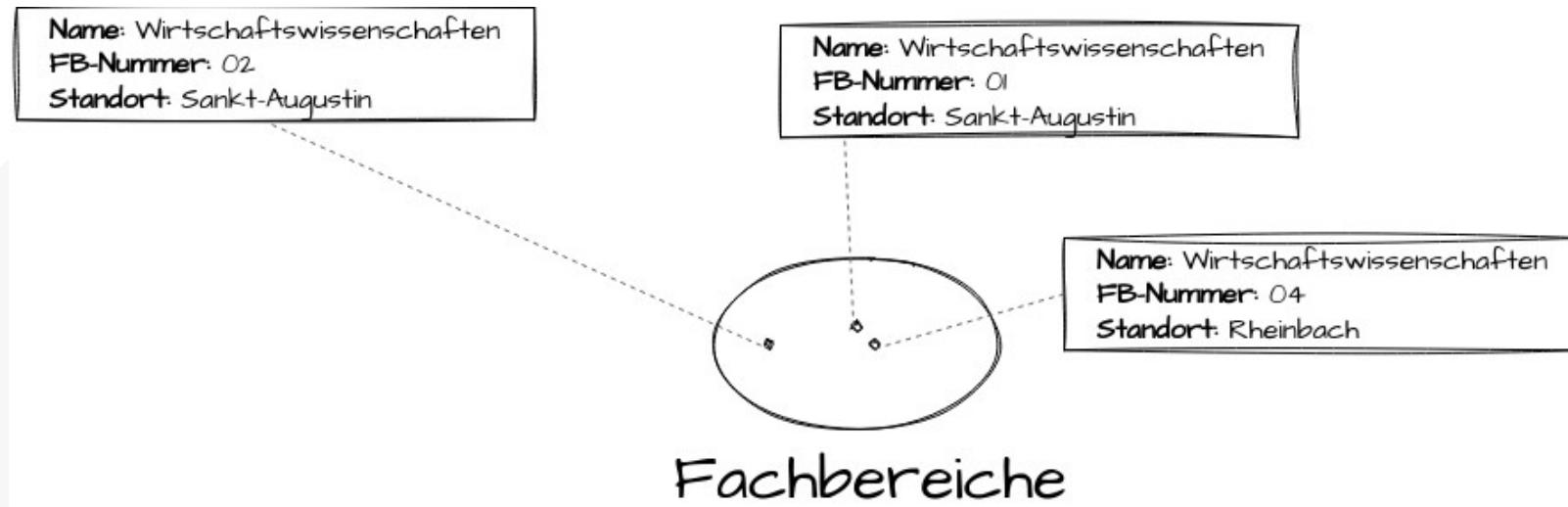
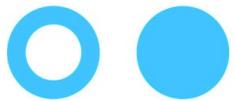
# Wann sind Entitäten gleich?



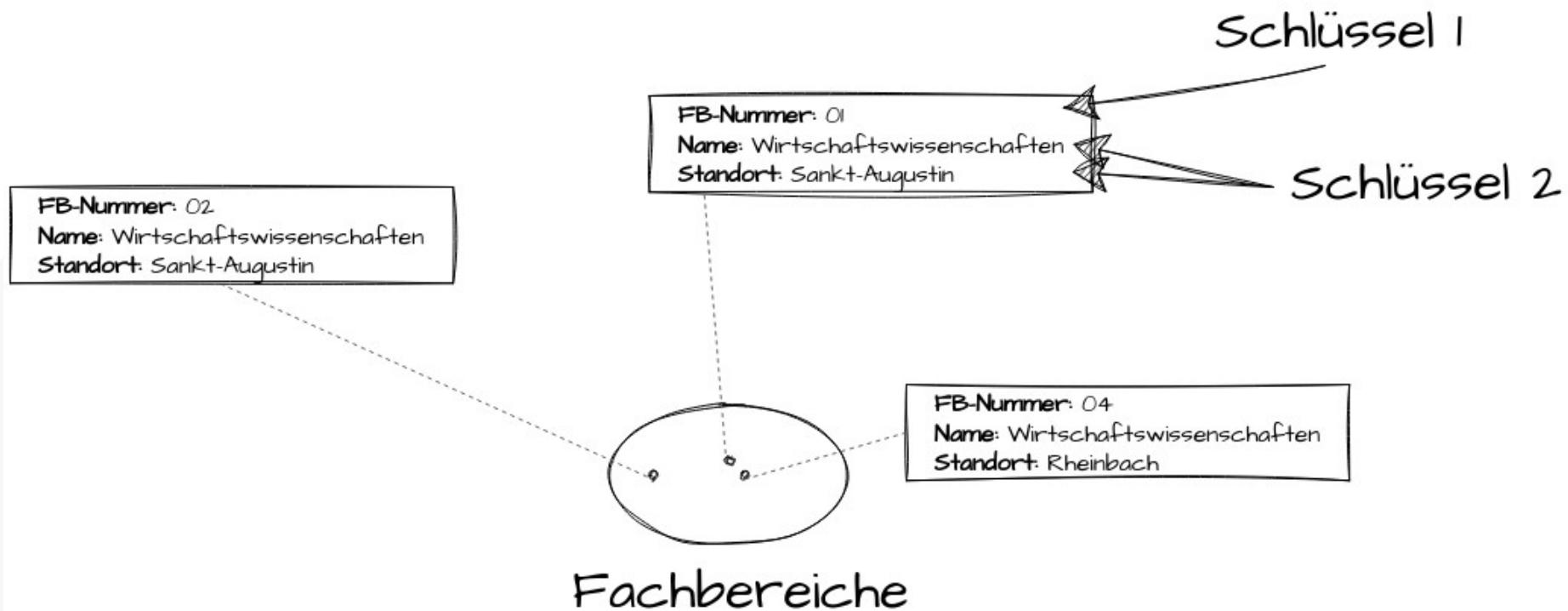
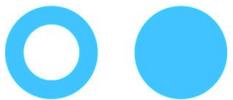
Sind die beiden neuen Wilkes identisch zum bereits gespeicherten Wilke?



Schlüsselattribute sind Attribute, die eine Entität eindeutig identifizieren.

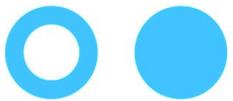


Welche Attribute definieren einen Fachbereich eindeutig?

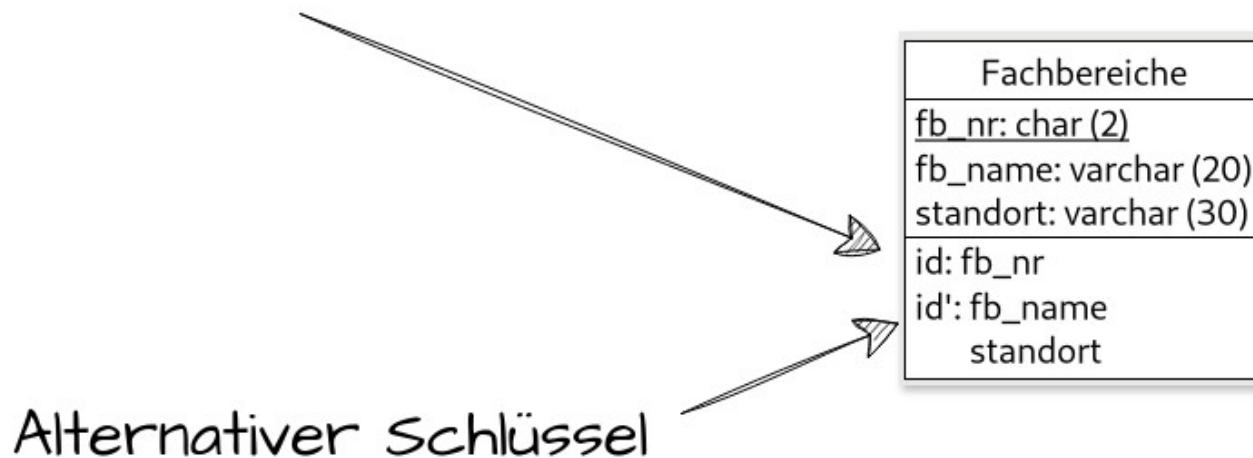


... eine Menge von Attribute, die eine Entität eindeutig identifizieren

... können aus mehreren Attributen bestehen (zusammgesetzte Schlüssel)

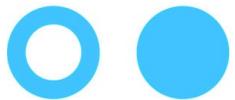


## Primärschlüssel



## Alternativer Schlüssel

... werden unterschieden in den primären und alternative Schlüssel  
... der Primärschlüssel sollte kompakt und effizient sein



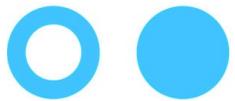
Modulgruppen
bezeichnung: varchar (30)

Studiengaenge
bezeichnung: varchar (30)
kuerzel: varchar (10)

Module
titel: varchar (30)

... fehlende Schlüssel werden im logischen Schema behandelt

... im ER-Diagramm nur semantisch ausdrucksstarke Schlüssel definieren



## - Entity -

### Entity

- ♦ Objekt (Ding bzw. Phänomen) der realen Welt

### Entity-Typ

- ♦ Zusammenfassung von Objekten mit gleichen Charakteristiken (Objekttypen, Dingtypen)

### Semantik

- ♦ Entity-Typen stehen für Mengen von Objekten
- ♦ Diese Mengen können sich schneiden (nicht-disjunkt)

### Bezeichnung

- ♦ Entity a vom Typ E

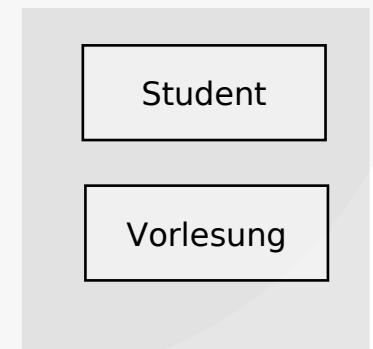
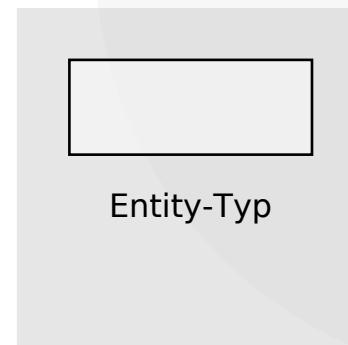
### Beispiele für Entities

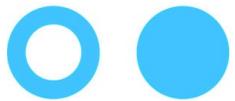
- ♦ Müller, Mathematik, Gebäude B, Prof. Dr. Meier, Informatik

### Beispiele für Entity-Typen

- ♦ Vorlesung, Student, Professor, Gebäude

### Graphische Darstellung





## - Relationship (I) -

### Relationship (Beziehung)

- Beziehungen zwischen zwei oder mehreren Objekten der realen Welt

### Beziehungs-Typ

- Zusammenfassung von Beziehungen mit gleichen Charakteristiken

### Semantik

- Beziehungs-Typen stehen für Mengen von Beziehungen

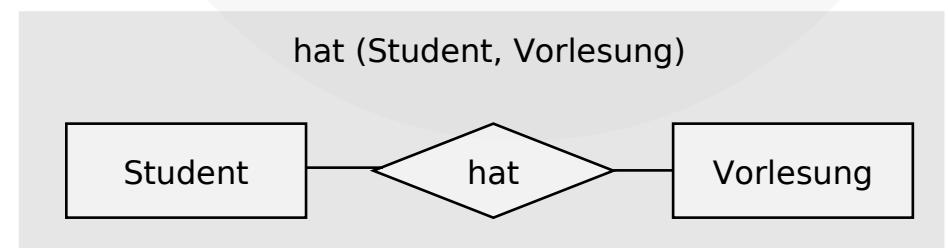
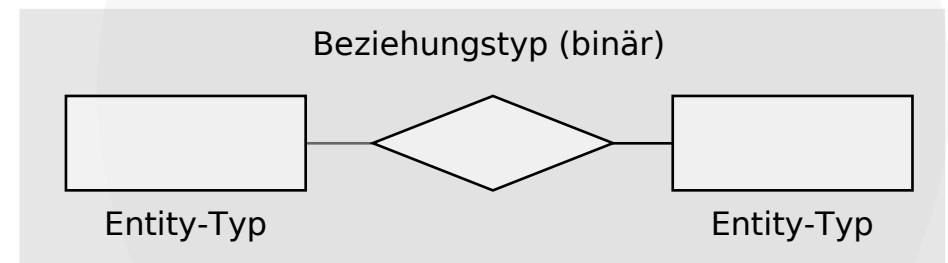
### Bezeichnung

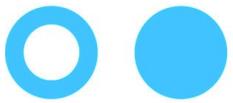
- Beziehung vom Typ R zwischen zwei Entity-Typen E1 und E2
- $R(E1, E2)$

### Beispiele für Beziehungen

- Müller hat Datenbanken
- Meier prüft Mathematik

### Graphische Darstellung





## - Relationship (II) -

### Grad der Beziehung

- Anzahl Objekttypen die an einem Beziehungstyp beteiligt sind

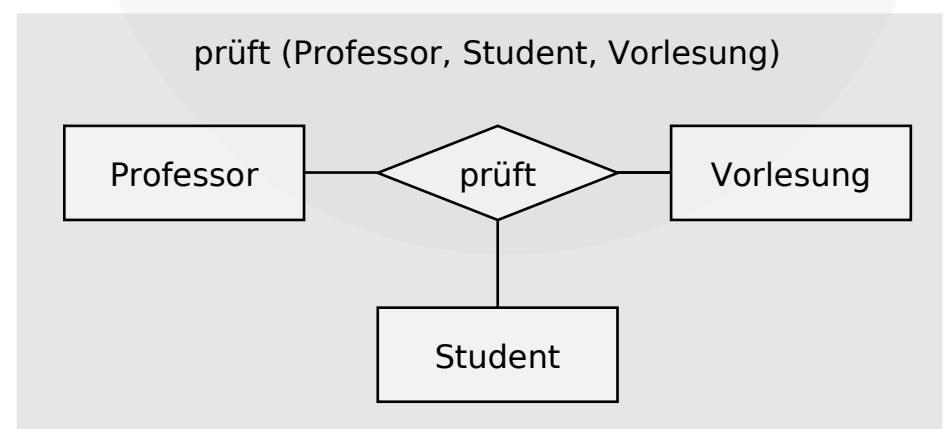
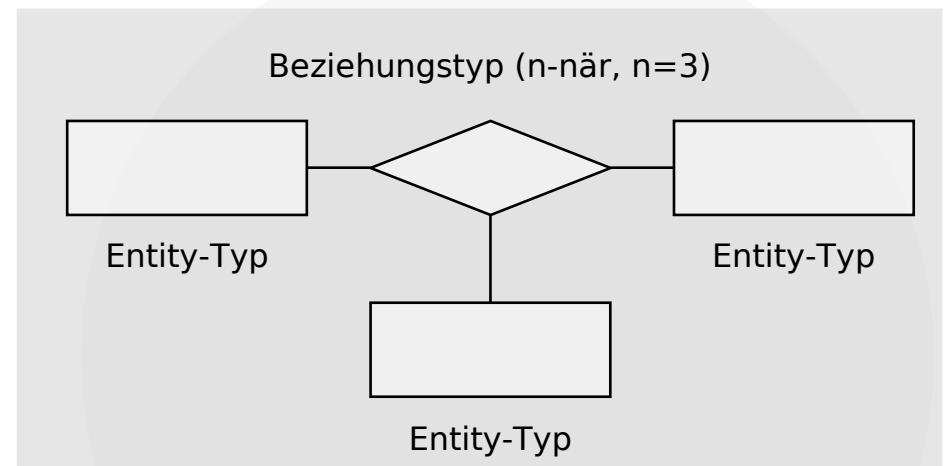
### Semantik

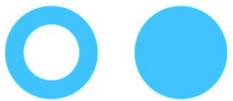
- binärer Beziehungstyp: Beziehungstyp zwischen zwei Entity-Typen
- n-ärer Beziehungstyp: Beziehungstyp zwischen mehr als zwei ( $n > 2$ ) Entity-Typen
- in der Praxis sind n-äre Beziehungstypen mit  $n > 3$  selten

### Bezeichnung

- Beziehung vom Typ  $R$  zwischen  $n$  Entity-Typen  $E_1, E_2, \dots, E_n$
- $R(E_1, E_2, \dots, E_n)$

### Graphische Darstellung





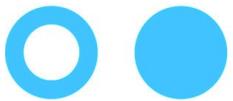
## - Kardinalität -

### Inhalt

- ◆ Einführung
- ◆ Entity Relationship Modell (ERM)
  - Schema der Miniwelt (Diskursbereich)
  - Ziel: Schema in Diagrammform
  - Historie und Notation
  - Integritätsbedingungen
  - Entity
  - Relationship
  - Kardinalität
  - Attribut
  - Abhängigkeiten
- ◆ Erweiterungen des ERM
- ◆ Zeit und Versionen
- ◆ Objekt-orientierte Modelle

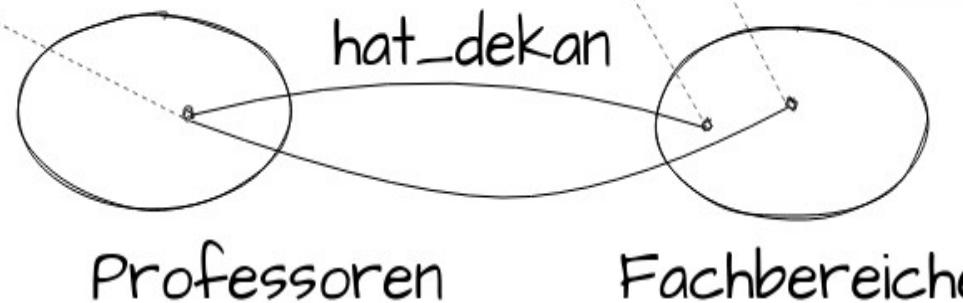
### Überblick

- ◆ Komplexitätsgrad einer Beziehung
- ◆ Binäre 1:1-Beziehung („min-max“-Notation nach Abrial)
- ◆ Binäre 1:N-Beziehung („min-max“-Notation) nach Abrial
- ◆ Binäre N:M-Beziehung („min-max“-Notation) nach Abrial
- ◆ Binäre Muss-Beziehung („min-max“-Notation) nach Abrial
- ◆ Viele unterschiedliche Notationen
- ◆ Beispiel in „min-max“-Notation nach Abrial

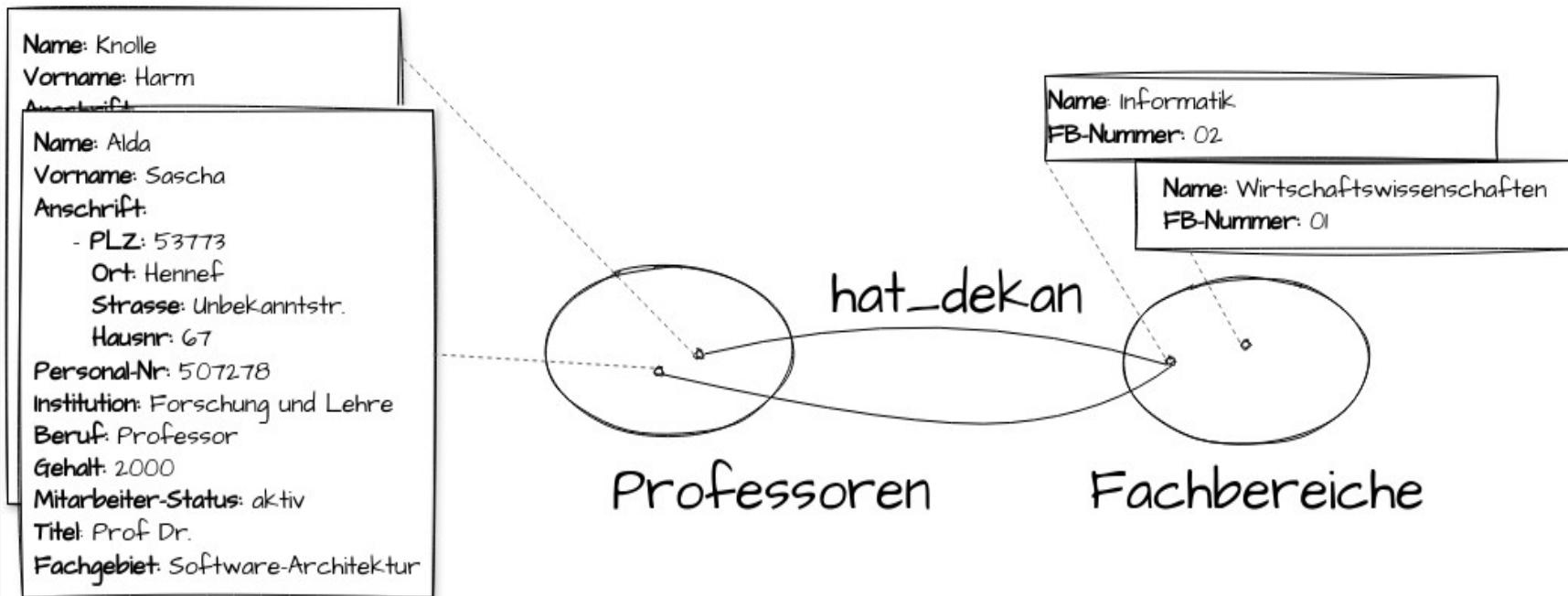
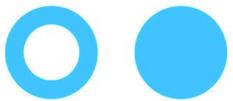


Name: Alda  
Vorname: Sascha  
Anschrift:  
- PLZ: 53773  
Ort: Hennef  
Strasse: Unbekanntstr.  
Haushr: 67  
Personal-Nr: 507278  
Institution: Forschung und Lehre  
Beruf: Professor  
Gehalt: 2000  
Mitarbeiter-Status: aktiv  
Titel: Prof Dr.  
Fachgebiet: Software-Architektur

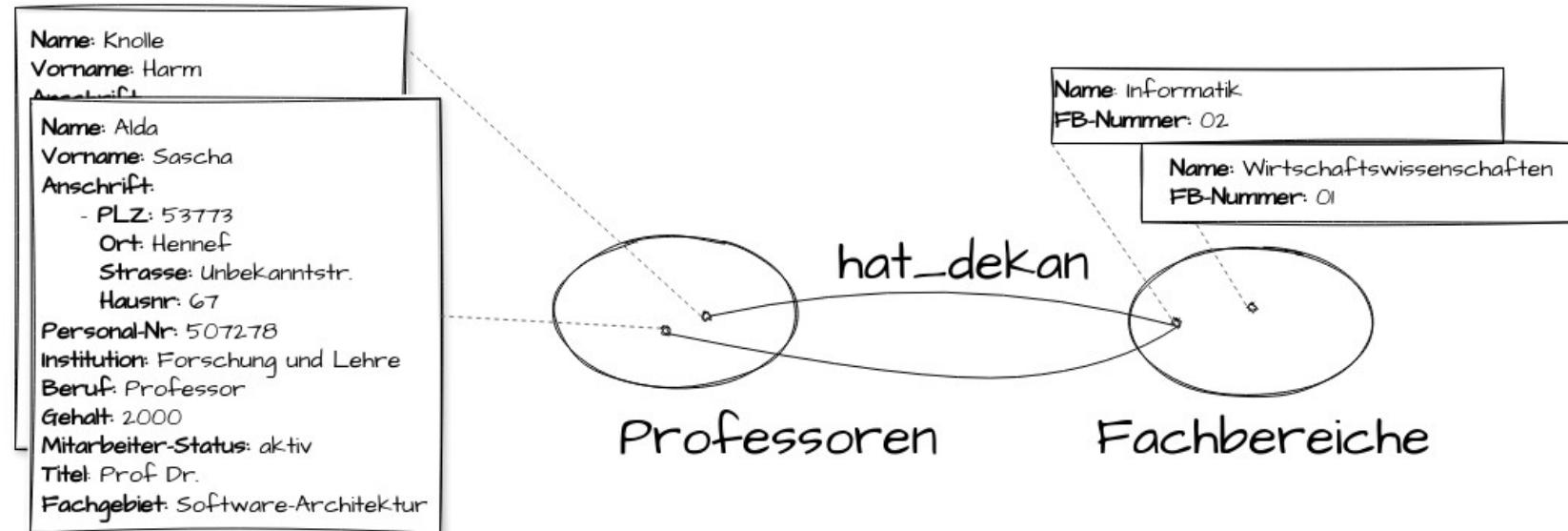
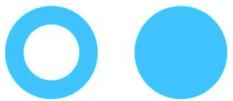
Name: Informatik  
FB-Nummer: 02  
Name: Wirtschaftswissenschaften  
FB-Nummer: 01



... Kann Herr Alda Dekan von zwei Fachbereichen sein?

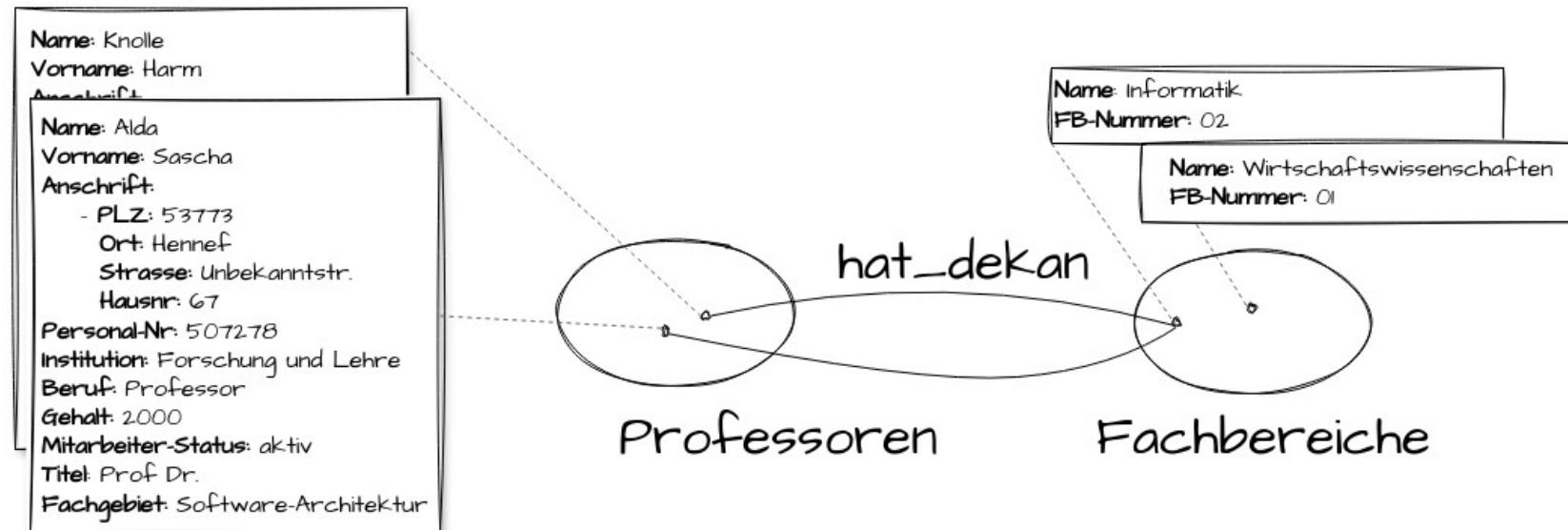
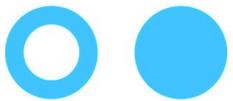


... Kann der Fachbereich Informatik zwei Dekane haben?



Die Kardinalitäten legen fest, wie häufig eine Entität an einer Beziehung Teilnehmen kann bzw. **muss**.

- Wie oft muss ein Professor Dekan eines FB's sein?
- Wie oft kann ein Professor Dekan eines FB's sein?
- Wie viele Dekane muss ein FB haben?
- Wie viele Dekane kann ein FB haben?



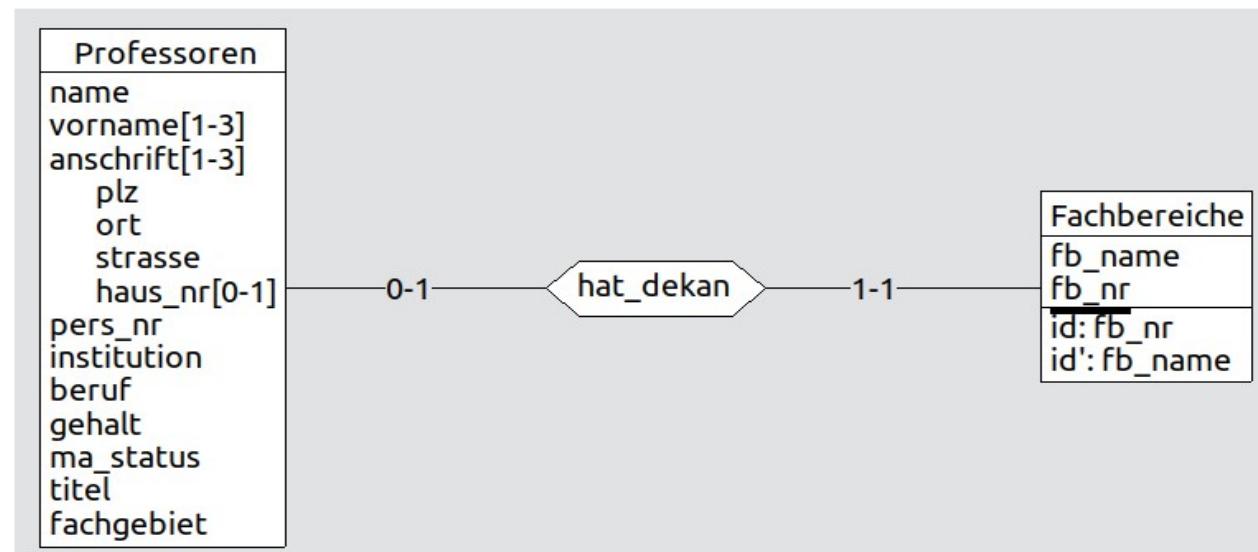
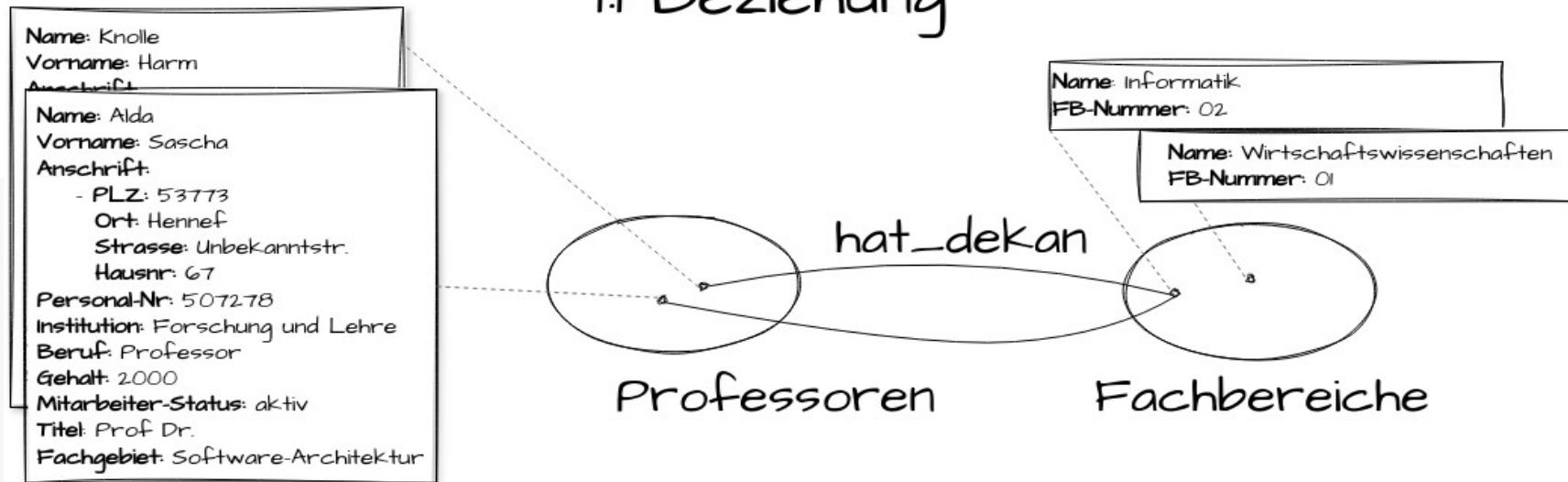
Die Kardinalitäten legen fest, wie häufig eine Entität an einer Beziehung Teilnehmen kann bzw. muss.

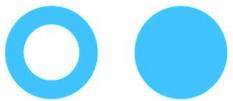
- Wie oft muss ein Professor Dekan eines FB's sein? -> 0
- Wie oft kann ein Professor Dekan eines FB's sein? -> 1
- Wie viele Dekane muss ein FB haben? -> 1
- Wie viele Dekane kann ein FB haben? -> 1

0-1

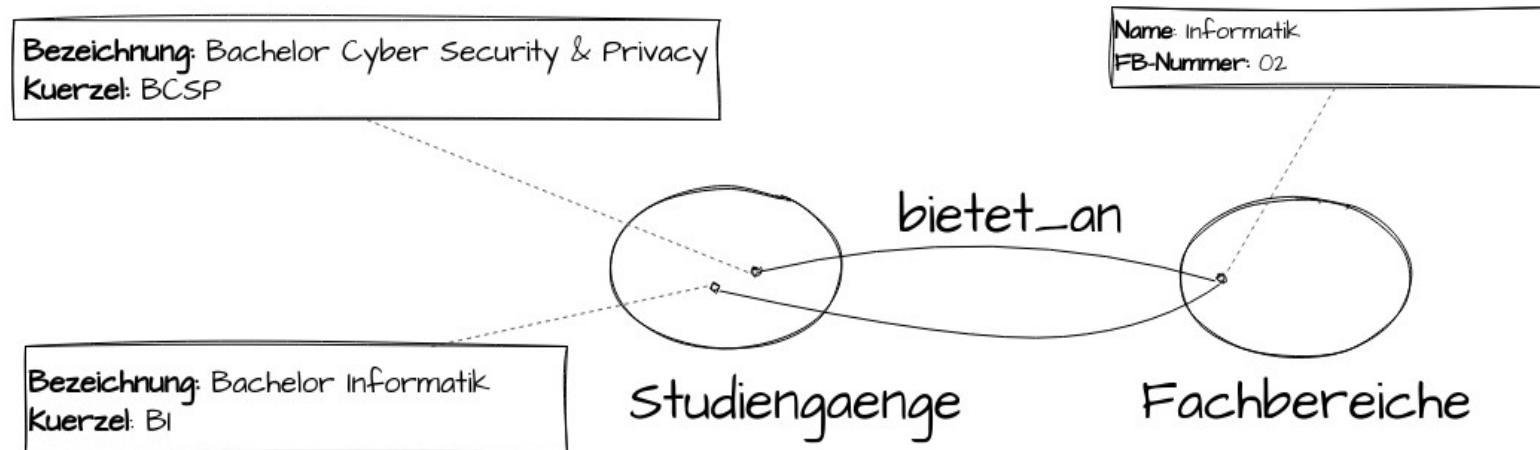
1

## I: Beziehung

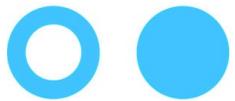




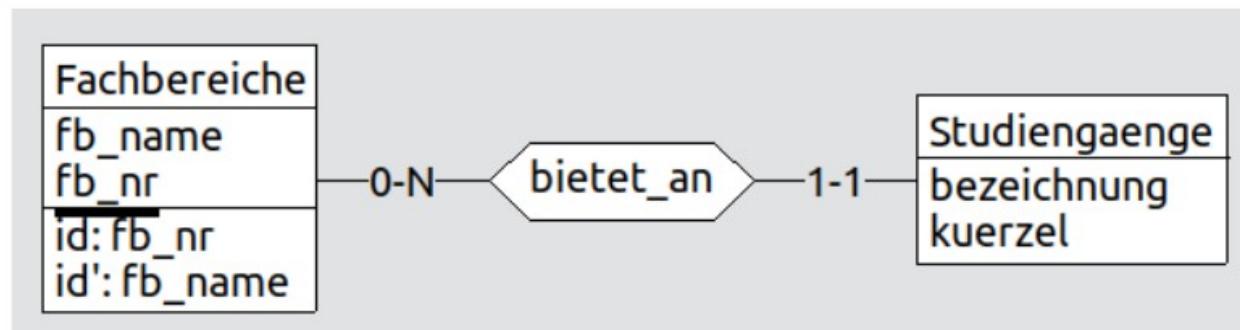
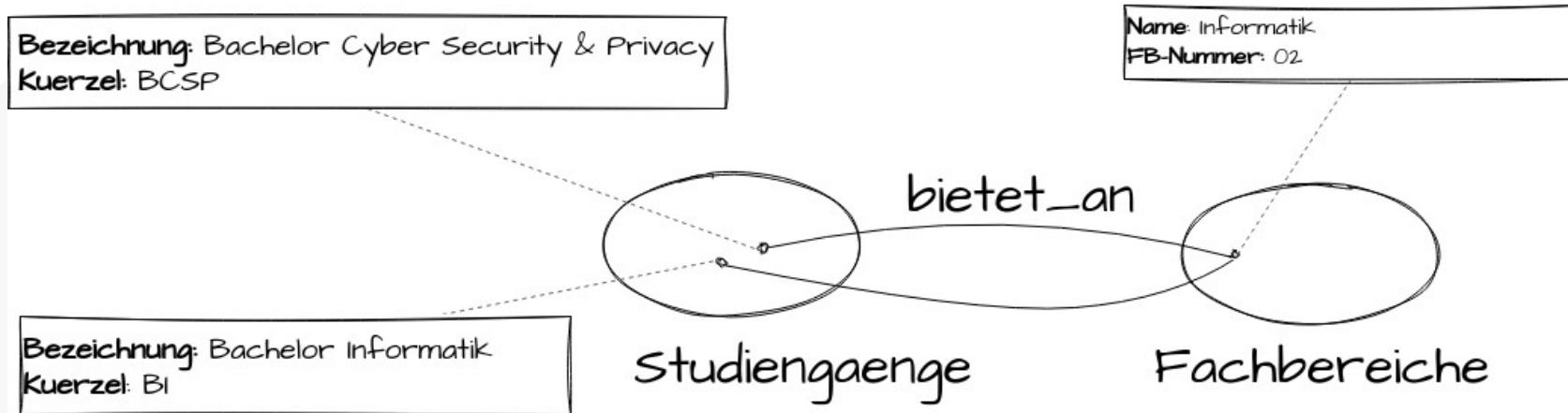
## 1:N Beziehung

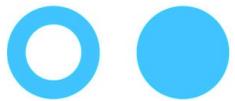


- Wie viele Studiengänge muss ein FB anbieten? -> 0
- Wie viele Studiengänge kann ein FB anbieten? -> N
- Wie vielen Fachbereichen muss ein Studiengang zugeordnet werden? -> 1
- Wie vielen Fachbereichen kann ein Studiengang zugeordnet werden? -> 1

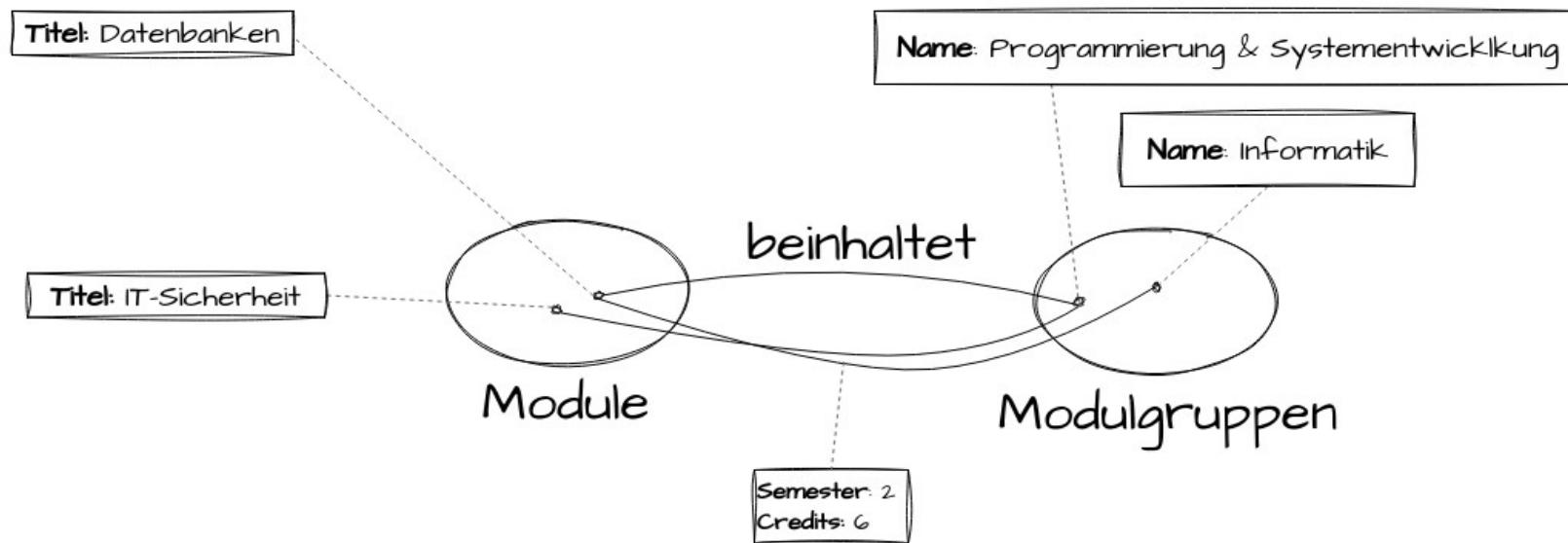


## 1:N Beziehung

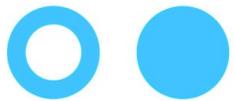




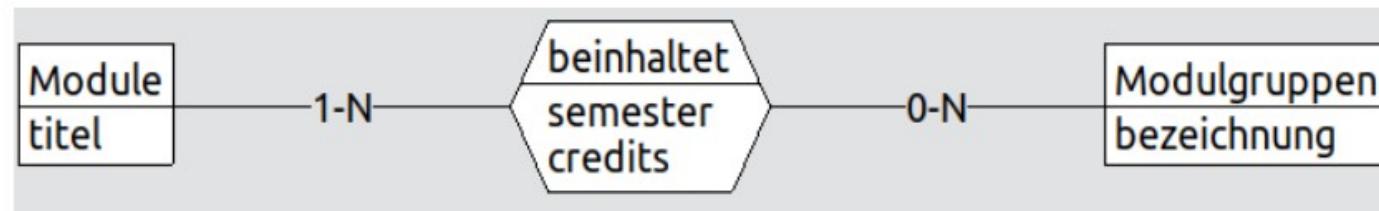
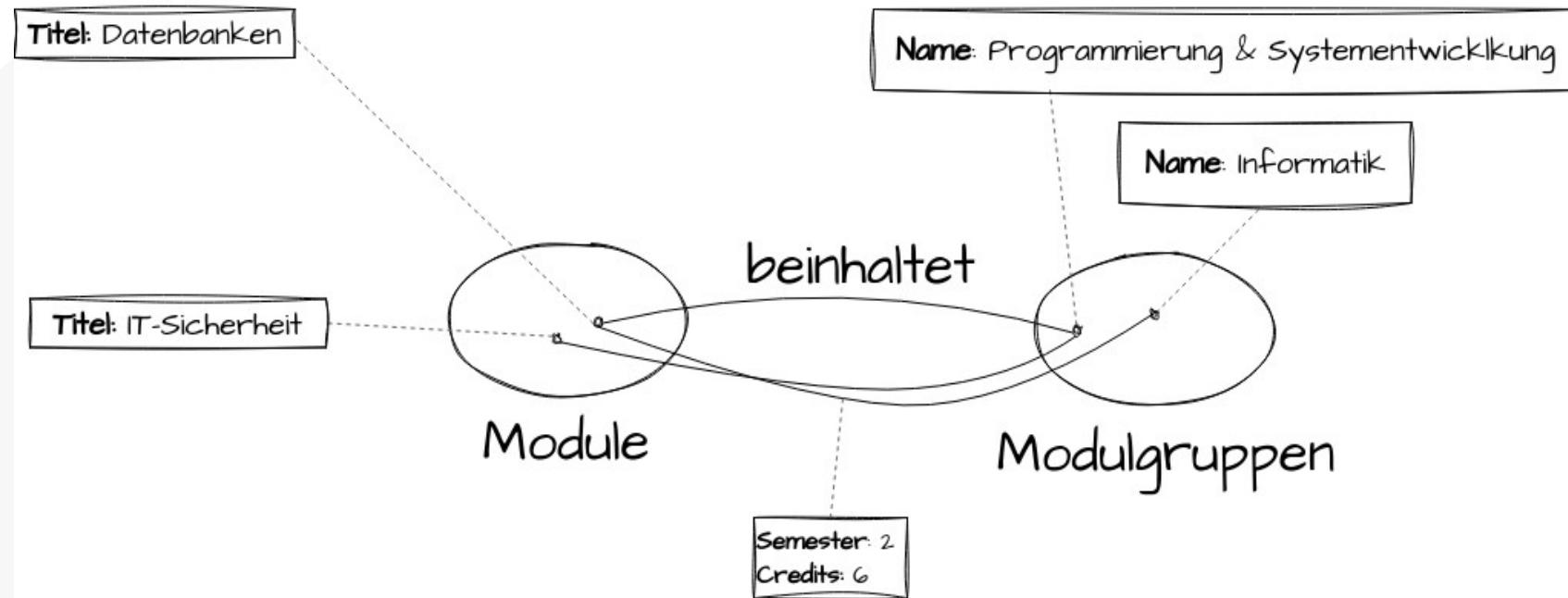
## N:M Beziehung

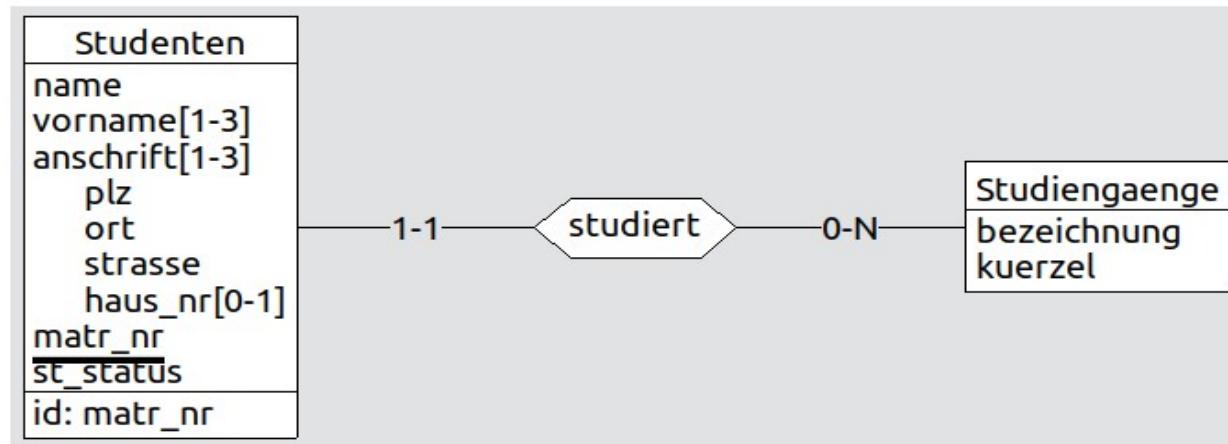
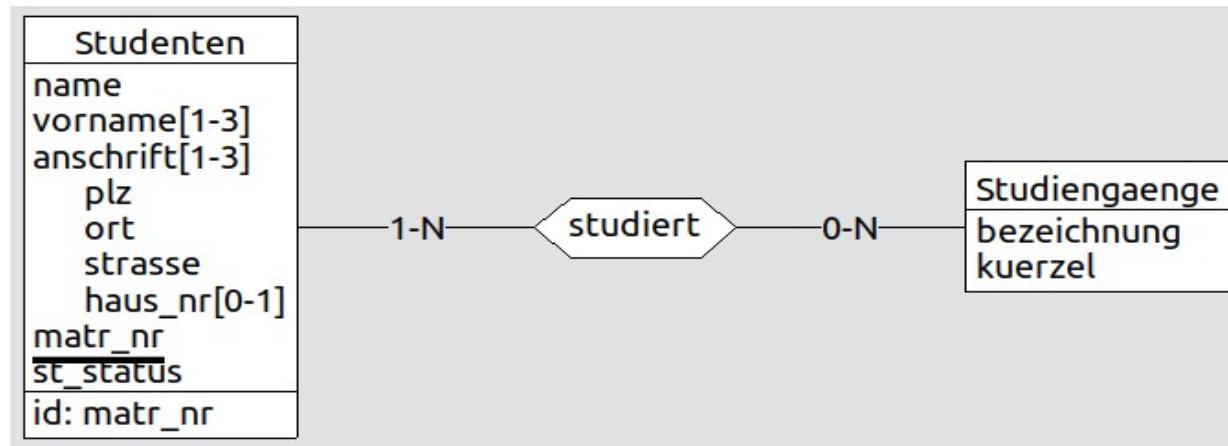
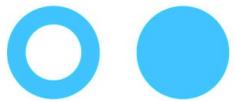


- Wie vielen Modulgruppen muss ein Modul zugeordnet werden? -> 1
- Wie vielen Modulgruppen kann ein Modul zugeordnet werden? -> N
- Wie viele Module muss eine Modulgruppe beinhalten? -> 0
- Wie viele Module kann eine Modulgruppe beinhalten? -> N

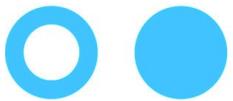


## N:M Beziehung





... Worin unterscheiden sich diese beiden Varianten?

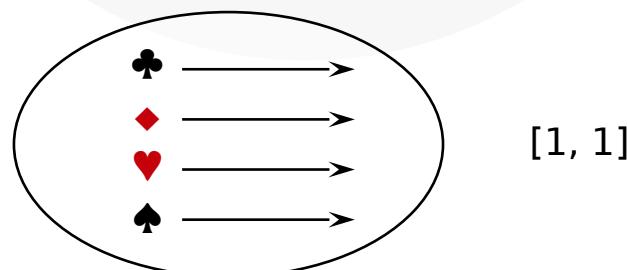


## - Komplexitätsgrad einer Beziehung -

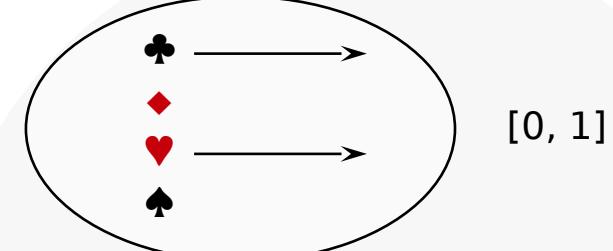
### Kardinalität

- „min-max“-Notation von Jean-Raymond Abrial (1974)
- gibt an, mit wie vielen Entities einer anderen Art ein betrachtetes Entity im Rahmen eines betrachteten Beziehungs-Typs verbunden sein kann
- signifikanter Unterschied zur CHEN-Notation
- Komplexitätsgrad: [min, max]

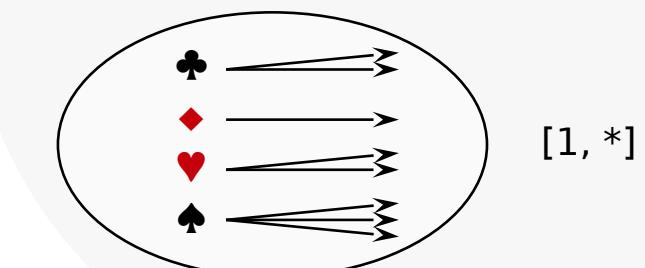
$$\boxed{\text{Symbole}} = \{\clubsuit, \diamondsuit, \heartsuit, \spadesuit\}$$



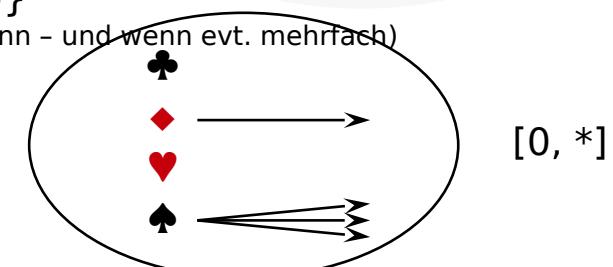
- bedingt: min = 0, max = 1  
(kann - und wenn höchstens einmal)

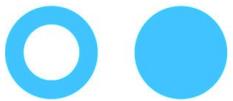


- mehrfach: min = 1, max = \*, \* ∈ N+ / {0}  
(muss - und evtl. mehrfach)



- bed. mehrfach: min = 0, max = \*, \* ∈ N+ / {0}  
(kann - und wenn evtl. mehrfach)





## - Binäre 1:1-Beziehung („min-max“-Notation nach Abrial) -

### 1:1-Beziehung

- einem Entity vom Typ E1 ist höchstens ein Entity vom Typ E2 zugeordnet
- einem Entity vom Typ E2 ist höchstens ein Entity vom Typ E1 zugeordnet

### Semantik

- Entities der Typen E1 und E2 können in einem Beziehungs-Typ R maximal einmal teilnehmen

### Bezeichnung

- $\text{max1} = \text{max2} = 1$
- $R(E1 [ \text{min1}, \text{max1}], E2 [ \text{min2}, \text{max2}])$

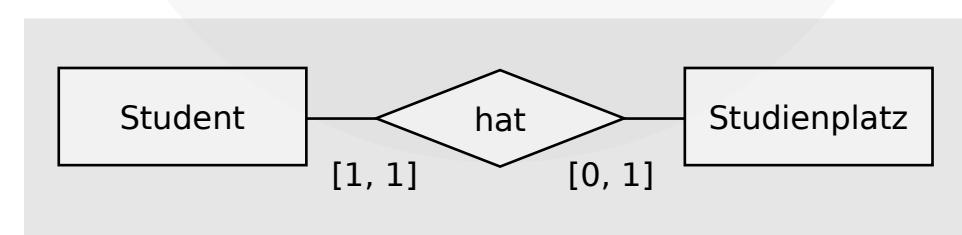
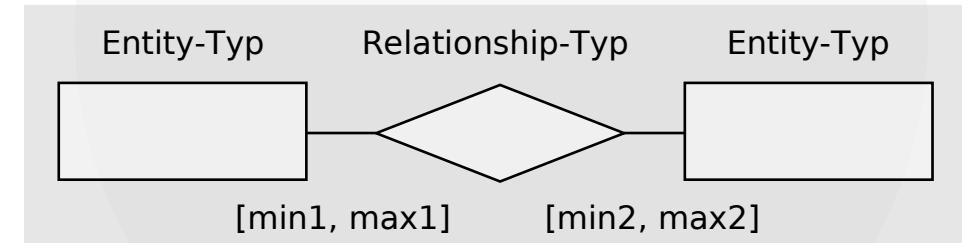
$0 \leq \text{min1 bzw. min2} \leq 1$

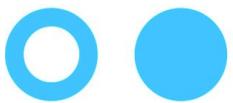
$0 = \text{bedingt (kann)}$   
 $\text{oder}$   
 $1 = \text{einfach (muss)}$

### Beispiele

- $R(E1[\text{min1}, 1], E2[\text{min2}, 1])$   
 $R(E1[1,1], E2[0,1])$   
 $R(E1[1,1], E2[1,1])$   
 $R(E1[0,1], E2[0,1])$   
 $R(E1[0,1], E2[1,1])$
- $1 = \text{einfach}$   
 $(\text{muss}) \text{ oder}$   
 $0 = \text{bedingt}$   
 $(\text{kann})$

### Graphische Darstellung





## - Binäre 1:N-Beziehung („min-max“-Notation nach Abrial) -

### 1:N-Beziehung

- einem Entity vom Typ E1 können N Entities vom Typ E2 zugeordnet sein
- einem Entity vom Typ E2 ist höchstens ein Entity vom Typ E1 zugeordnet

### Semantik

- Entities der Typen E1 können in einem Beziehungs-Typ R maximal N-mal teilnehmen, Entitäten der Typen E2 maximal nur einmal

### Bezeichnung

- $\text{max1} = *$ ,  $\text{max2} = 1$
- $R(E1[\text{min1}, *], E2[\text{min2}, 1])$

$0 \leq \text{min1} \leq *$   
bedingt mehrfach oder mehrfach

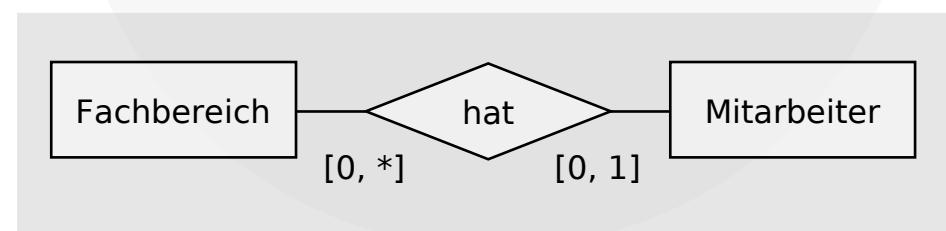
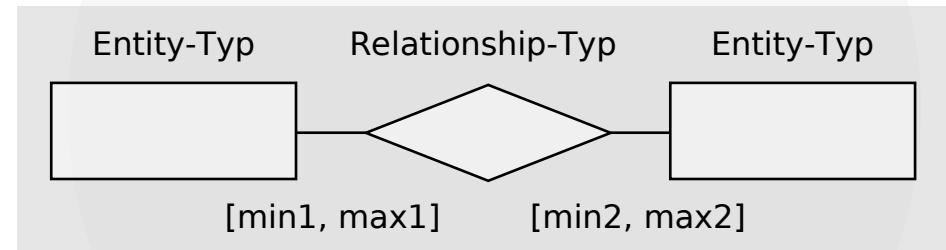
$0 \leq \text{min2} \leq 1$   
bedingt oder einfach

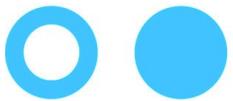
Gilt auch vertauscht:  
 $R(E1[\text{min2}, 1], E2[\text{min1}, *])$

### Beispiele

- $R(E1[0, *], E2[0, 1])$
- $R(E1[0, 1], E2[1, *])$

### Graphische Darstellung





## - Binäre N:M-Beziehung („min-max“-Notation nach Abrial) -

### N:M-Beziehung

- ♦ einem Entity vom Typ E1 können Entities vom Typ E2 zugeordnet sein
- ♦ einem Entity vom Typ E2 können Entities vom Typ E1 zugeordnet sein

### Semantik

- ♦ Entities der Typen E1 und E2 können in einem Beziehungs-Typ R maximal N- bzw. M-mal teilnehmen

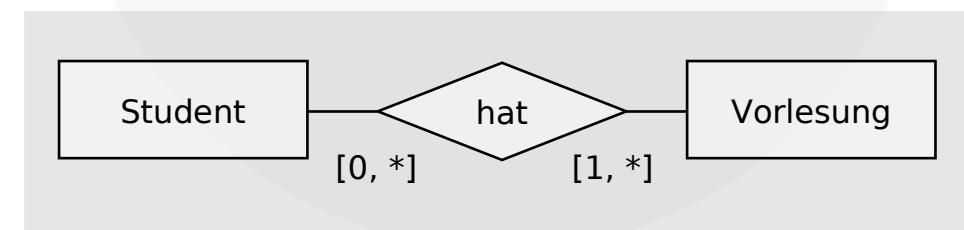
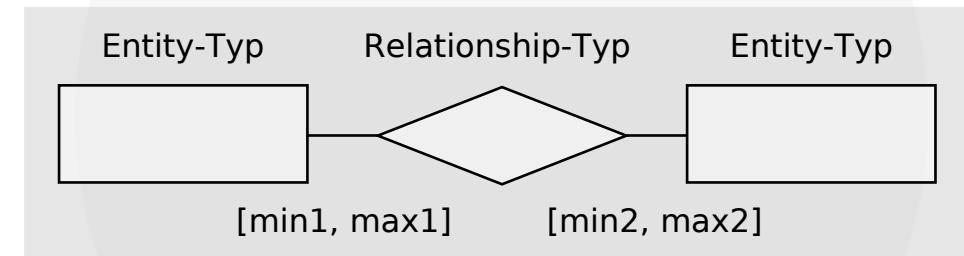
### Bezeichnung

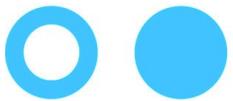
- ♦  $\max_1 = \max_2 = *$
- ♦  $R(E1[\text{mehrfach o. bed. mehrf.}, *], E2[\text{mehrfach o. bed. mehrf.}, *])$

### Beispiele

- ♦  $R(E1[0,*], E2[0,*] )$
  - ♦  $R(E1[1,*], E2[0,*] )$
- ...

### Graphische Darstellung





## - Binäre Muss-Beziehung („min-max“-Notation nach Abrial) -

### Muss-Beziehung (totale Beziehung)

- ♦ Jedem Entity vom Typ E1 muss mindestens ein Entity vom Typ E2 zugeordnet sein

### Semantik

- ♦ Entities der Typen E1 müssen in einem Beziehungs-Typ R mindestens einmal teilnehmen

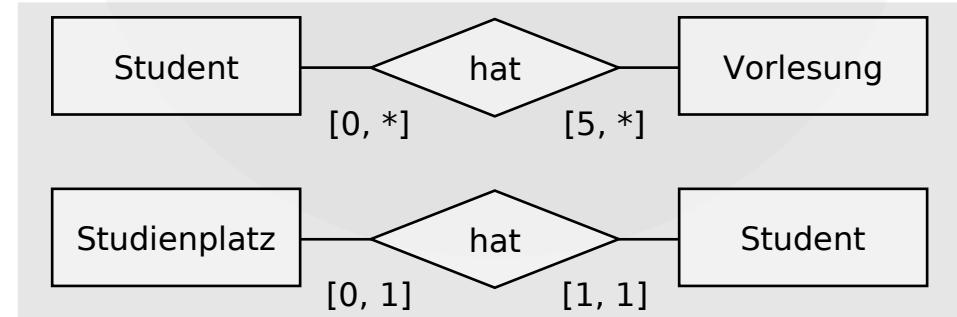
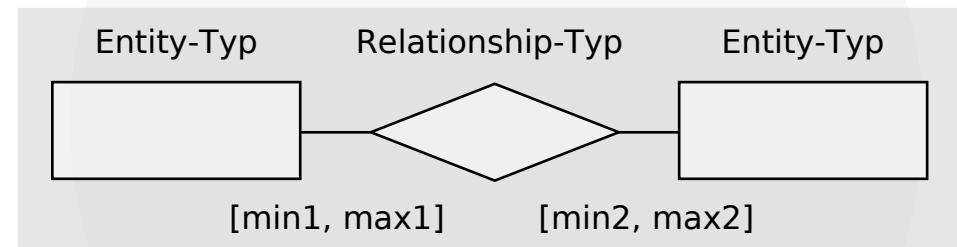
### Bezeichnung

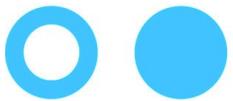
- ♦  $\text{min1} \geq 1$  und / oder  $\text{min2} \geq 1$
- ♦  $R(E1 [ \text{einfach o. mehrfach} */1 ], E2 [ \text{einfach o. bedingt o. mehrfach o. bed. mehrf.} */1 ] )$
- ♦  $R(E2 [ \text{einfach o. bedingt o. mehrfach o. } */1 ], E1 [ \text{einfach o. mehrfach } */1 ] )$

### Beispiele

- ♦  $R(E1[5,*], E2[0,*])$
- ♦  $R(E2[0,1], E1[1,1])$

### Graphische Darstellung

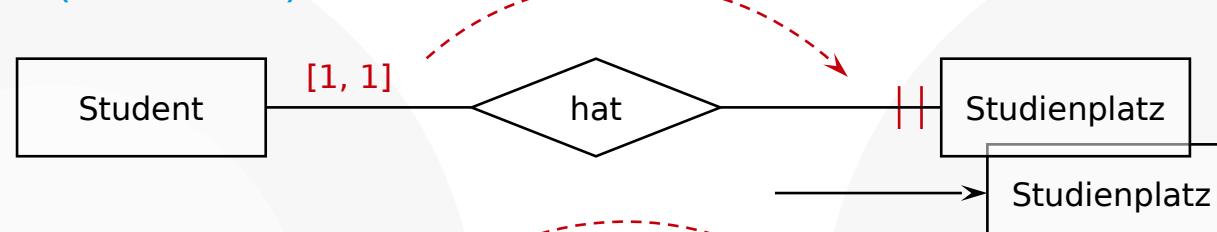




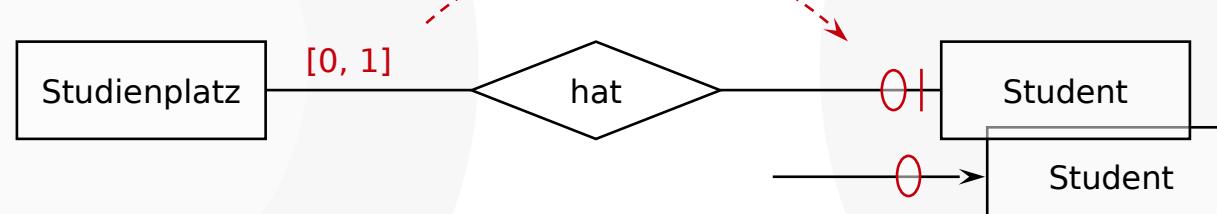
## - Viele unterschiedliche Notationen -

Min-Max-Notation  
(funktional)

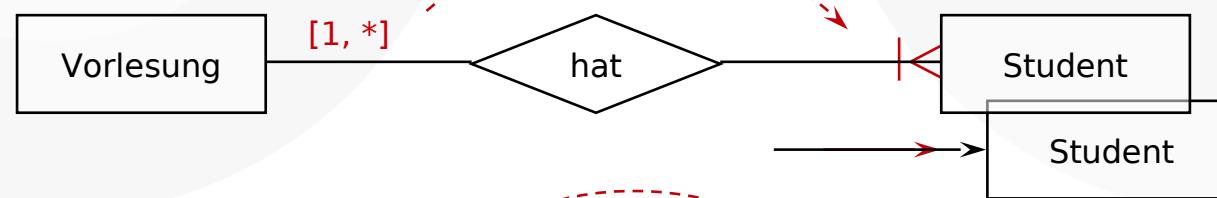
- ◆ einfach  
(..muss..)



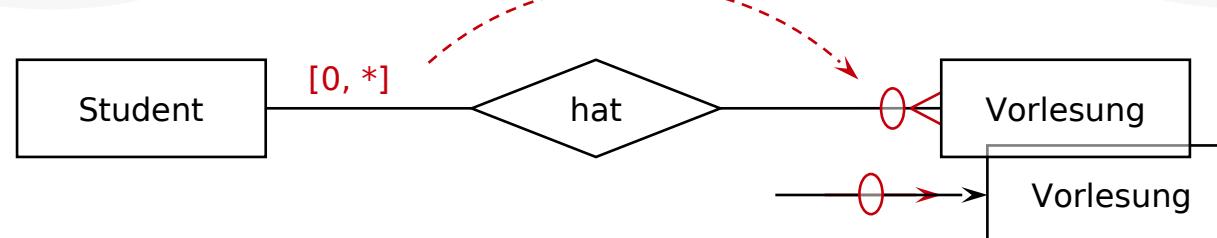
- ◆ bedingt  
(..kann..)



- ◆ mehrfach  
(..muss..)



- ◆ bedingt  
mehrfach  
(..kann..)



z.B. "Krähenfuß"-Notation, "Pfeil"-Notation

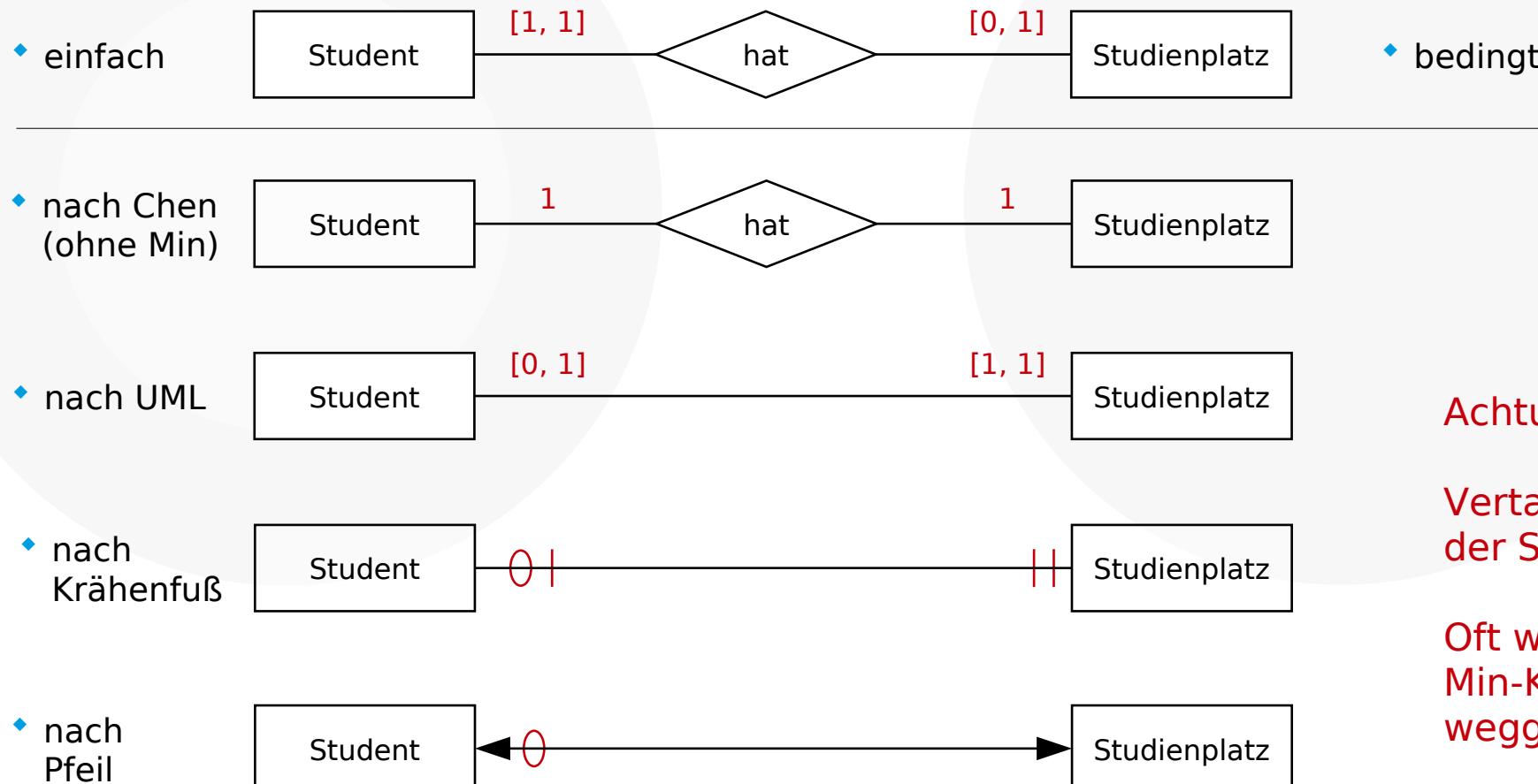
Achtung :  
Vertauschung  
der Seiten bei  
CHEN und  
auch bei UML !!!

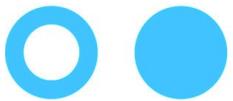
Oft wird die  
Min-Kardinalität  
weggelassen !!!

## - Viele unterschiedliche Notationsformen (I) -

### „min-max“-Notation nach Abrial

- ◆ Beispiel 1:1-Beziehungstyp



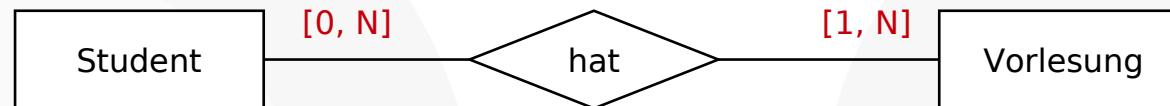


## - Viele unterschiedliche Notationsformen (II) -

### „min-max“-Notation nach Abrial

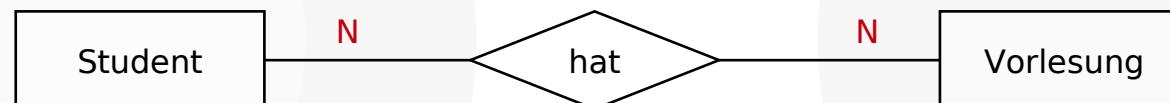
- ◆ Beispiel N:M-Beziehungstyp

- ◆ bedingt mehrfach

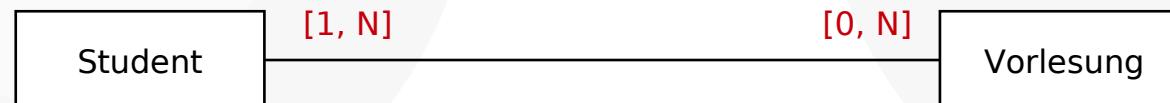


◆ mehrfach

- ◆ nach Chen (ohne Min)

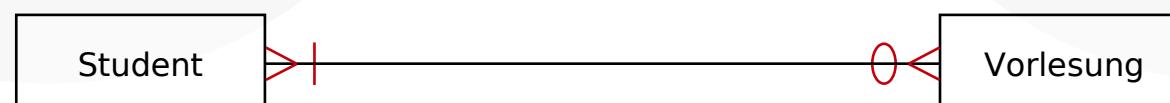


- ◆ nach UML



Achtung :

- ◆ nach Krähenfuß

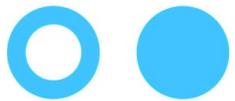


Vertauschung  
der Seiten

- ◆ nach Pfeil



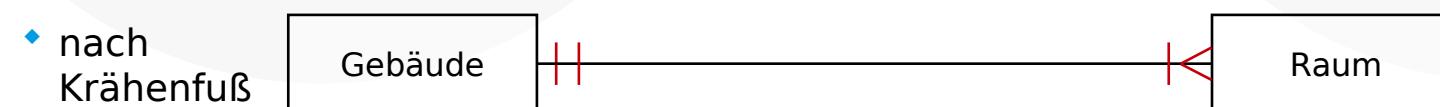
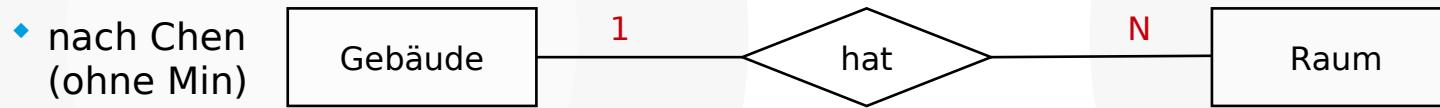
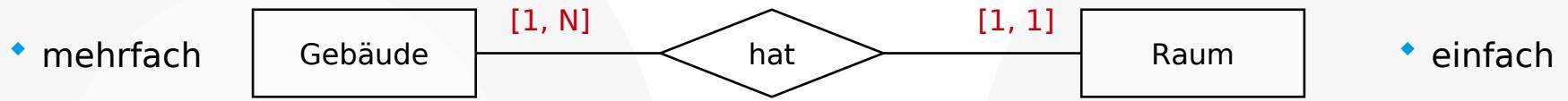
Oft wird die  
Min-Kardinalität  
weggelassen !!!



## - Viele unterschiedliche Notationsformen (III) -

### „min-max“-Notation nach Abrial

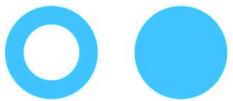
- ◆ Beispiel 1:N-Beziehungstyp



Achtung :

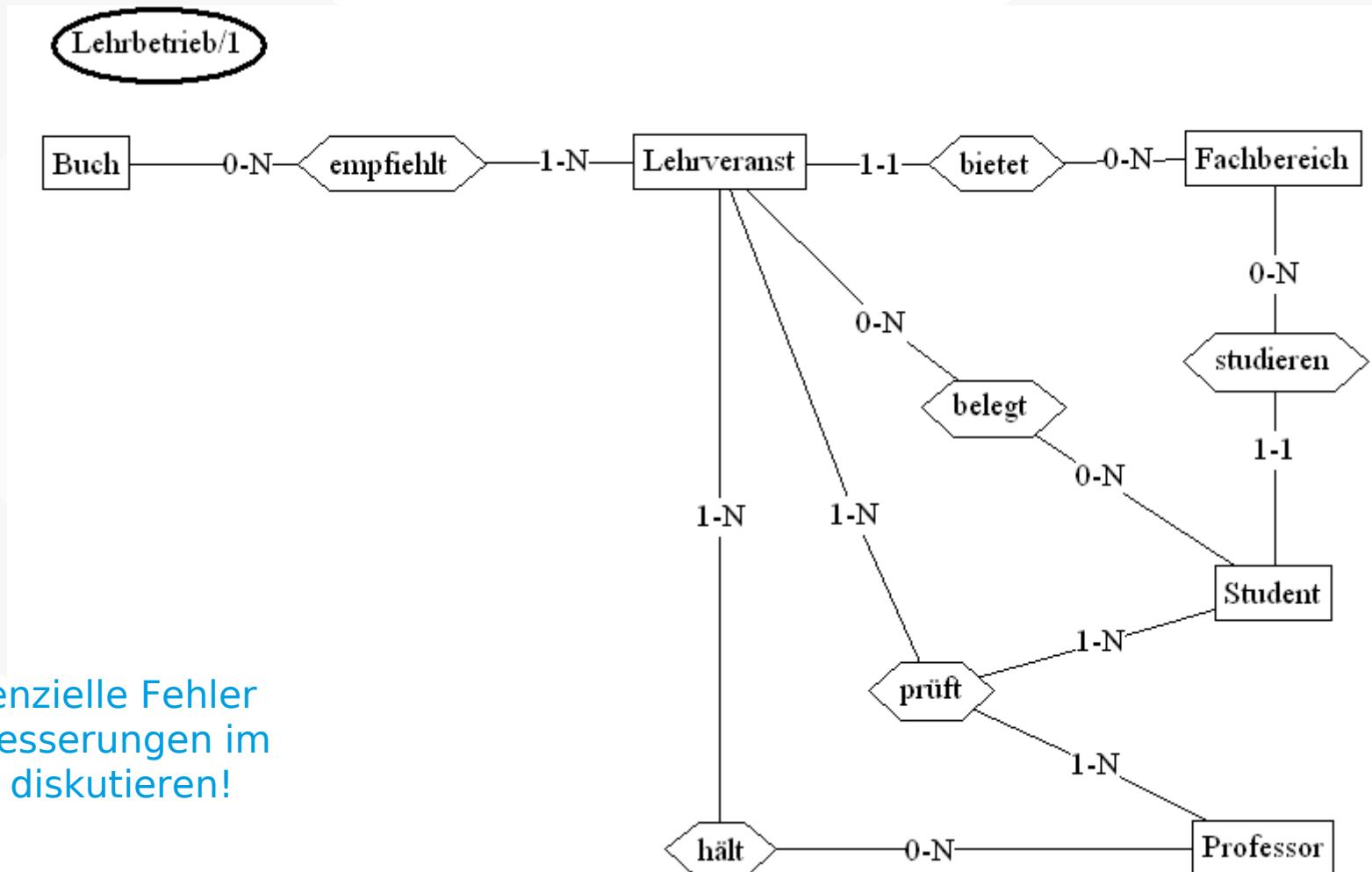
Vertauschung  
der Seiten

Oft wird die  
Min-Kardinalität  
weggelassen !!!



## - Beispiel in "min-max"-Notation nach Abrial -

Erstellt mit dem CASE-Werkzeug DB-MAIN



Bitte potenzielle Fehler  
bzw. Verbesserungen im  
Schema diskutieren!