Datenbanken I (T2INF2004)

Foliensatz 2: Datenbankentwurf I

Uli Seelbach, DHBW Mannheim, 2023

Foliensatz freundlicherweise zur Verfügung gestellt von Mirko Schick

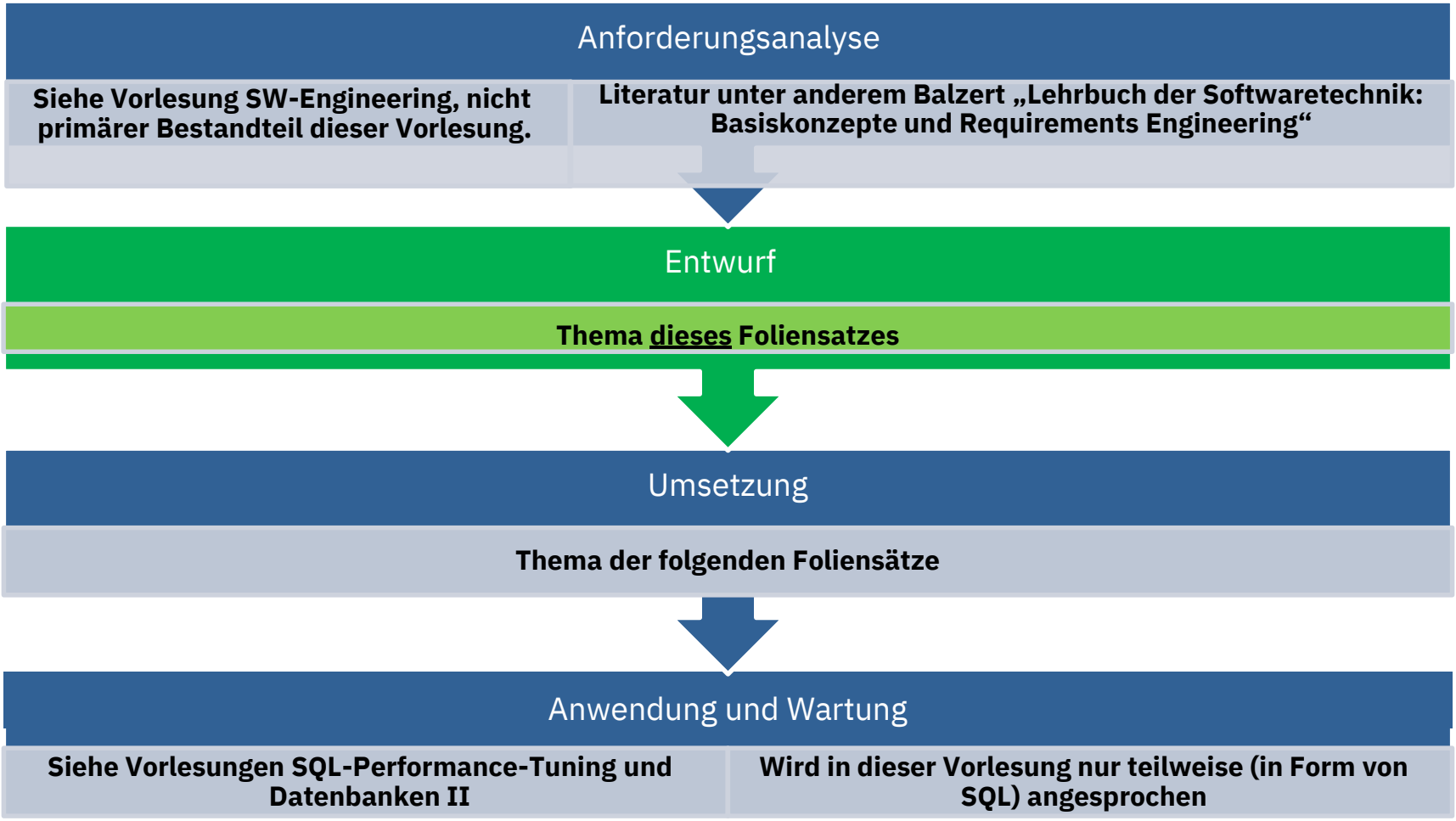
Datenbankentwurf

Wozu?

- Fördert strukturiertes Vorgehen, unter anderem:
  - Hinterfragen von Anforderungen
  - Besseres Verständnis der Datenbank
  - Dokumentation für später
  - Ggf. Automatisierung
- Bewertung
  - Speicherplatz
  - Effizienz
  - Erweiterbarkeit
- Kosten!!!
  - Ein guter Entwurf ist bei konzernkritischen Anwendungen unabdingbar
  - Spätere Änderungen sind um ein vielfaches teurer als einmalig viel Arbeit während der Entwurfsphase (➔ Vorlesung SW-Engineering)

Datenbankentwurf

Wozu?



Anforderungsanalyse

Merkmale

- Erster Schritt in der SWE:
  - Gehört eigentlich nicht zum Entwurf (auch wenn manche Autoren das meinen!)
  - Kommunikation zwischen Entwicklern, Business Analysten und Kunden
  - Zeitlich sehr aufwändig, aber trotzdem der erste Schritt, um Kunden ein Gefühl für die Software zu geben und den ersten Baustein für Akzeptanz zu legen
- Wichtig: Bewährtes beibehalten
  - Firmeninterne Strukturen
  - Vorgehen bei der Befragung, Ergebnisdokumente etc.
  - Dokumentationen persistent hinterlegen
- Ergebnis ist ein Pflichtenheft oder eine Fachdokumentation

Anforderungsanalyse

Grobes Vorgehen bei der Anfo-Analyse und den Folgeschritten

- Vollständig „verstehen, was der Kunde will“:
  Funktionale Anforderungen (welche Daten sollen warum und in welcher Menge gespeichert werden)
- **Konzept** erstellen, welches die DB-Struktur grundsätzlich beschreibt, ggf. in Teilschemata, die dann zusammengefasst werden
- **Nichtfunktionale Anforderungen** werden spätestens jetzt wichtig (wie schnell soll das System sein, wie viele Nutzer sollen gleichzeitig darauf zugreifen können)
- Danach bestimmen, **welches DBS** für die Anforderung geeignet ist (denn erst hier ist klar, welche Funktionen das DBS überhaupt unterstützen muss)
- Daraus ein **logisches Datenmodell** entwickeln (ginge theoretisch in den meisten Fällen vor Auswahl des DBS)
- Ein datenbankspezifisches **physisches Datenmodell** erstellen

Entwurfsprozess

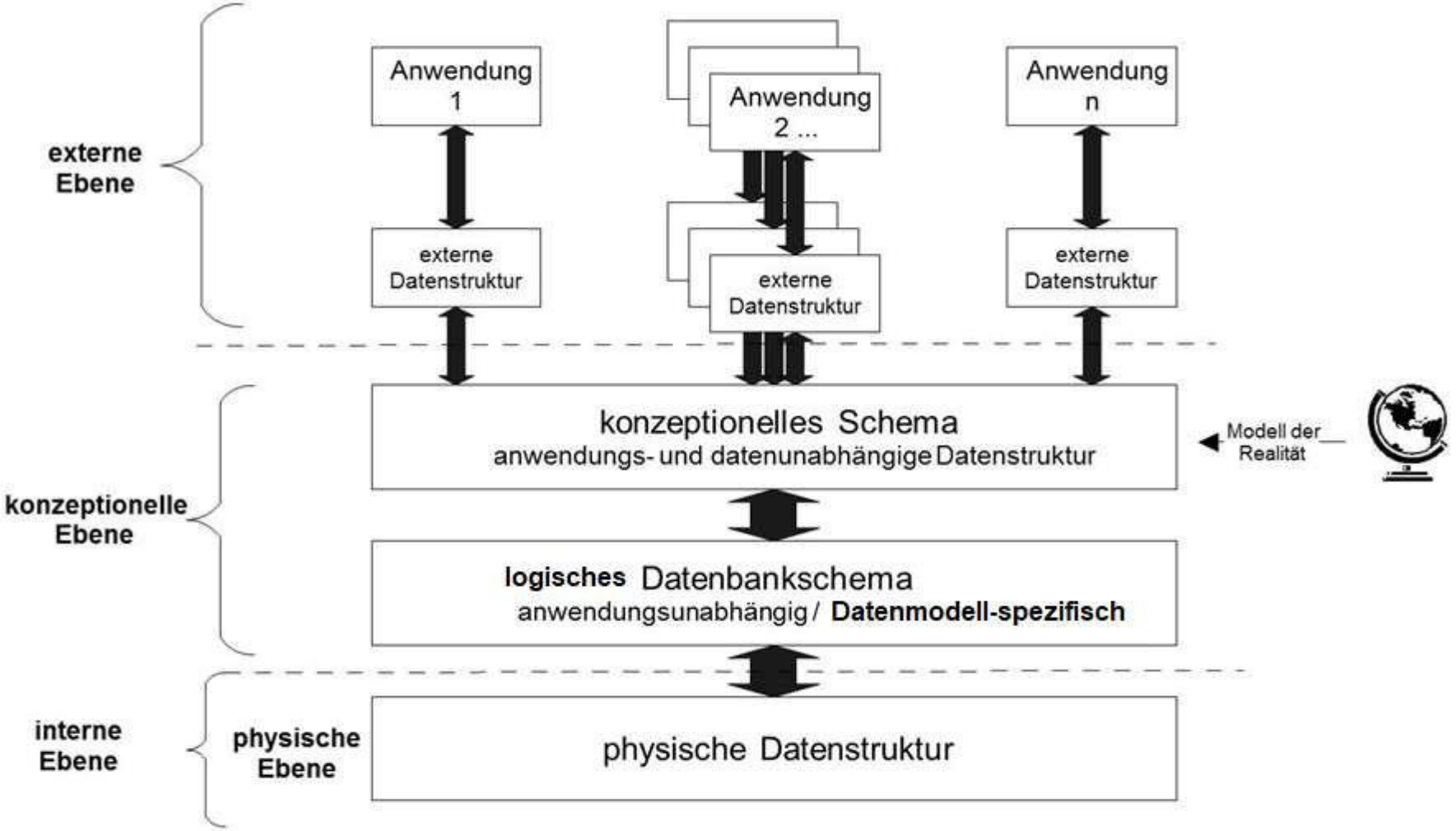
Grobes Vorgehen

Phase	Inhalt und Struktur von Daten		Datenbank-Anwendung
Phase 1 Anforderungen erfassen und analysieren	Datenanforderungen		Datenverarbeitungs-anforderungen
Phase 2 konzeptueller Entwurf	Konzeptuelles Schemadesign (DBMS-unabhängig)		Transaktions- und Anwendungsdesign (DBMS-unabhängig)
Phase 3* Auswahl eines DBMS	Basierend auf Basis der Datenanforderungen und der nichtfunktionalen Anforderungen	↔	Gedanken machen über
Phase 4 Datenmodell-Abbildung (logischer Entwurf)	Logisches Schemadesign und Umsetzung der Views (DBMS-abhängig)		Nutzerzahlen, Zugriffshäufigkeiten, Lastspitzen, Caching, Umsetzung von nichtfunktionalen Anforderungen (Performance, Ausfallsicherheit, ...)
Phase 5 physischer Entwurf	internes Schemadesign (DBMS-abhängig)	↔	
Phase 6 Implementierung, Wartung, Tuning	SQL-Statements Monitoring Tests		Implementierung Tests

*) Achtung: Phase 3 und 4 können auch vertauscht werden, wenn im logischen Entwurf auf DBMS-spezifische Charakteristika verzichtet wird! Dazu gehören beispielsweise besondere Datentypen. Vor der Erstellung des logischen Entwurfs muss das Datenmodell feststehen, aber nicht zwingend das DBMS.

Datenbankentwurf

Aufteilung in Ebenen: Wiederholung

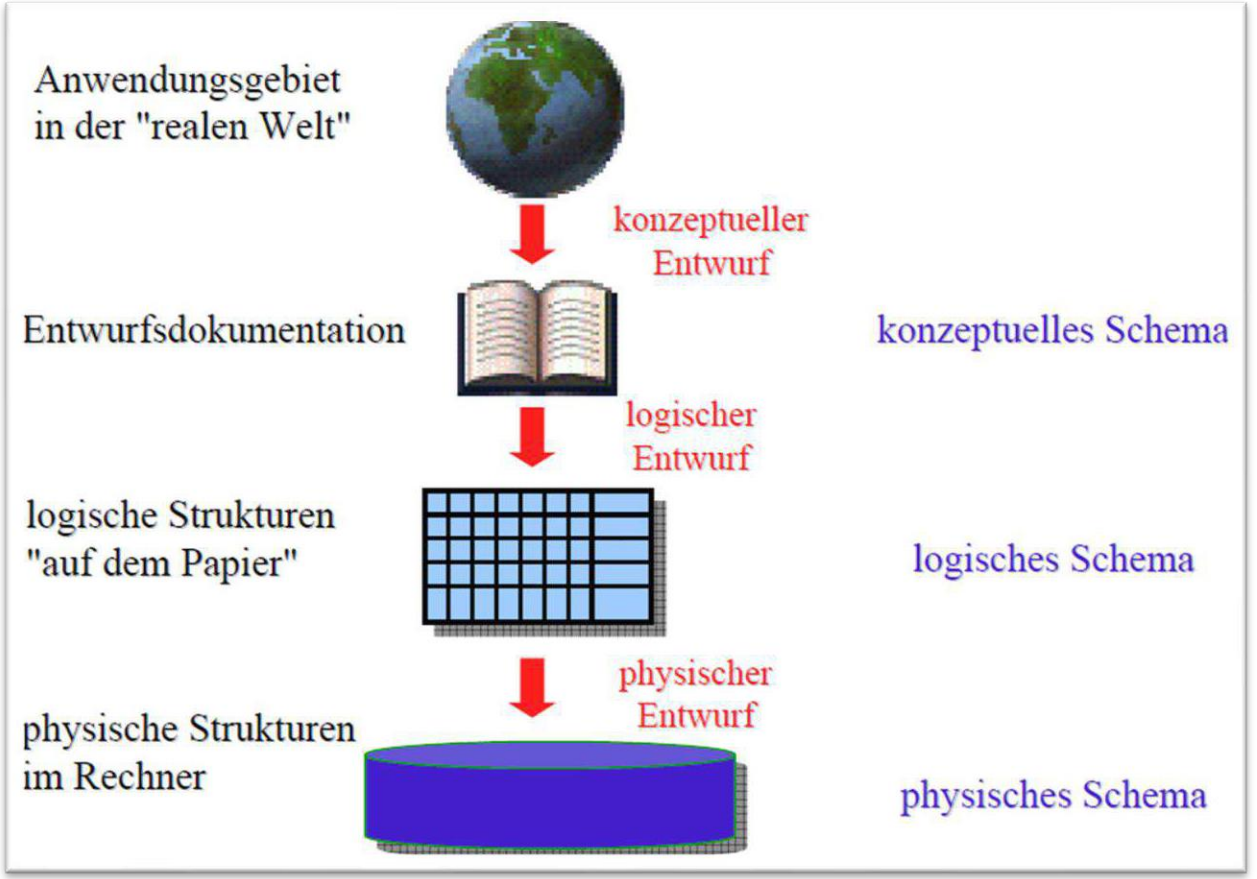


Quelle: http://wikis.gm.fh-koeln.de/wiki_db/uploads/Datenbanken/ANSI-3-Ebenenmodell/ansi_3_ebenen_modell.jpg

Datenbankentwurf

Aufteilung in Ebenen: Motivation

- Nur der Detaillierungsgrad wird gezeigt, der noch überblickt werden kann und auch wirklich benötigt wird
- Änderungen an der Basis müssen das Ergebnis nicht beeinflussen
- Stichwort: Datenunabhängigkeit



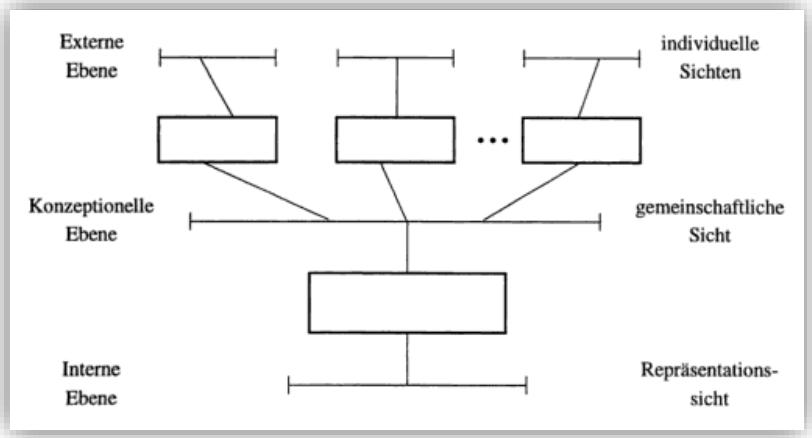
Quelle: Foliensatz 2006 Prof. Dr. Rainer Manthey

Datenbankentwurf

Aufteilung in Ebenen: Datenunabhängigkeit

Formen der Datenunabhängigkeit sind

- **Logische Datenunabhängigkeit**
  - Die individuelle nutzerspezifische Sicht eines Anwenders ist abgeschirmt von Änderungen unterhalb der externen Ebene
  - Durch Sichten und „unsichtbare“ Geschäftslogik kann dies bis zu einem gewissen Grad erreicht werden, jedoch bei größeren Änderungen **nicht immer möglich**
- **Physische Datenunabhängigkeit**
  - Änderung an den physischen Strukturen der internen Ebene ändert nichts am logischen Schema auf konzeptueller Ebene
  - **Kann fast immer realisiert werden**



Datenbankentwurf

Konzeptueller Datenbankentwurf

- Wichtig ist die Modellierung der im Dialog mit dem Kunden / Fachanwender besprochenen Sachverhalte
  - Datentypen, Algorithmen und nichtfunktionale Anforderungen spielen hier erst einmal keine nennenswerte Rolle (das wird oft vergessen)
- Das Entity-Relationship (ER)-Modell ist ein oft verwendetes Modell, welches intuitiv zusammen mit jedem nicht IT-versierten Anwender genutzt werden kann
- Bei IT-versierten Fachanwendern wird das konzeptuelle Schema meist großzügig vernachlässigt und auf Basis des Fachkonzeptes direkt mit dem logischen Schema begonnen
  - Vertretbares Vorgehen, welches jedoch schnell zu Missverständnissen und späteren Änderungen führen kann

Konzeptueller Datenbankentwurf

ER-Modell

- 1976 Peter Chen
- Grafische Notation zur Datenmodellierung, welche sich mit ihrer Semantik an Problemen der realen Welt orientiert
- Parallelen zur Objektorientierung unschwer erkennbar
- Zahlreiche Weiterentwicklungen und Werkzeuge (teilweise nur von historischer Bedetung)
  - Im Großen, zum Beispiel:
    - Structured-Entity-Relationship-Modell / Spezialisierungen und Generalisierungen; UML als Alternative
  - Im Kleinen, zum Beispiel
    - Notationsformen zur Darstellung der Kardinalitäten
- Überschaubare Anzahl an Regeln und Konzepten, die auf den folgenden Folien vorgestellt werden sollen

ER-Modell

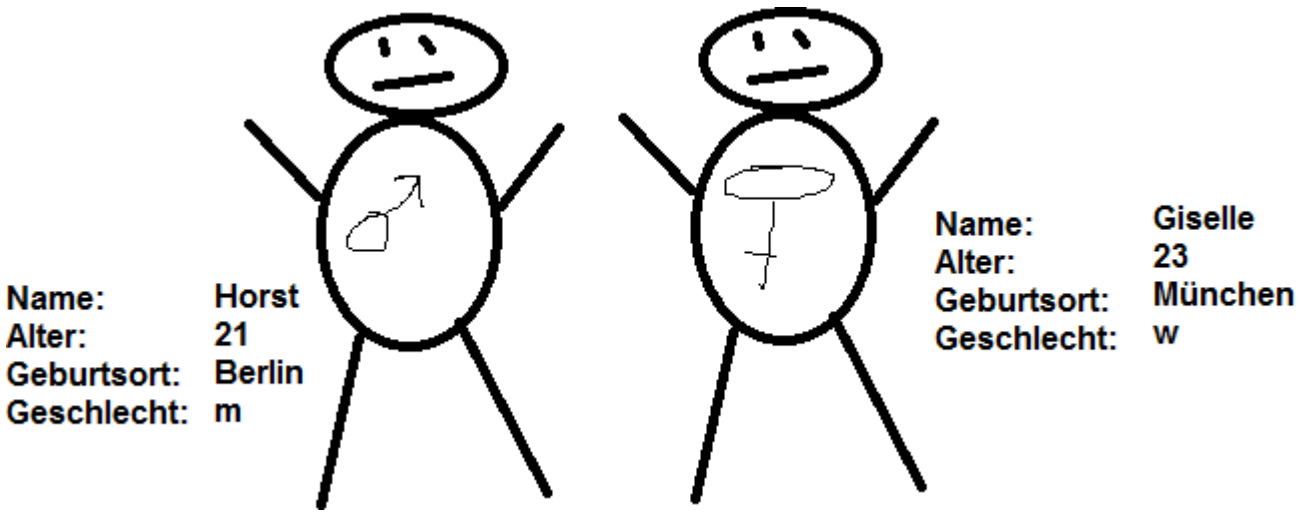
Begriffe

- Übersicht:
  - **Entität** (Entitätstyp)
    - schwacher Entitätstyp
  - **Beziehung** (Beziehungstyp)
    - Komplexität
    - Kardinalität
    - Rollen
  - Eigenschaft (Attribut)
    - Für Entitäten
    - Für Beziehungen
  - Primärschlüssel
  - Aggregation (Komposition)
  - Generalisierung (Spezialisierung)

ER-Modell

Entity (Entität)

- Gegenstände (Entities) sind wohlunterscheidbare physisch oder gedanklich existierende Konzepte der zu modellierenden Welt
- Ein Gegenstand wird durch seine Eigenschaften definiert und ist von allen anderen Gegenständen unterscheidbar
- Im Sinne der OOP: ein Objekt
- Das folgende sind 2 Entitäten (man beachte: sie haben gleiche Eigenschaften, nur andere Ausprägungen selbiger)



ER-Modell

Attribute (Eigenschaft) sowie Attributtyp (Attribut)

- Eigenschaften wie „Horst“ oder „Giselle“ sind mit einem Wert belegte Attribute (hier der „Name“)
- Jede Entität besteht aus Eigenschaften
- Attribute werden in der gängigen Praxis grafisch als Oval dargestellt
- Dies kann noch um Wertebereiche erweitert werden, was im Rahmen des konzeptuellen Designs i.d.R. nicht gemacht wird (implizite Annahmen)
- Primärschlüsselattribute werden unterstrichen (bedeutet: durch die Ausprägung dieses Attributs / dieser Attribute ist eine Entität eindeutig in der Menge aller Entitäten („Population“) bestimmbar

ER-Modell

Primärschlüssel?

- **Superschlüssel:** Menge von Attributen, deren Eigenschaften eine Entität eindeutig identifiziert
- **Schlüsselkandidat:** nicht mehr reduzierbare Menge von Attributen, deren Eigenschaften eine Entität eindeutig identifiziert

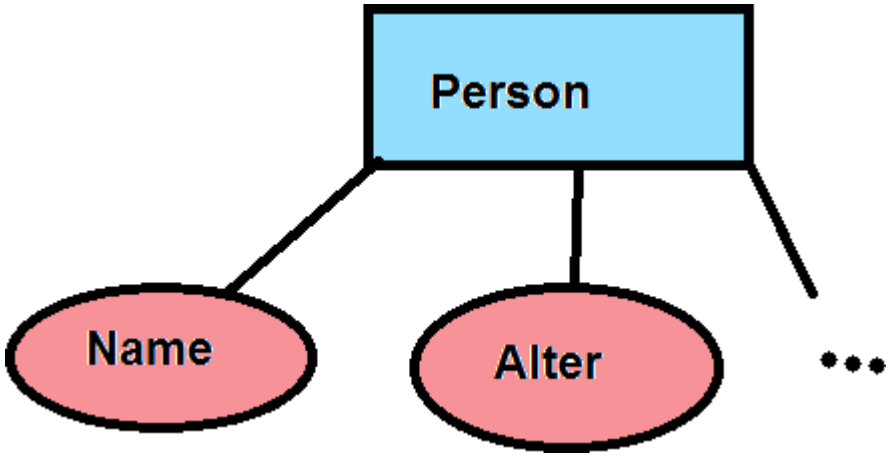
Telefonbuch		
Name	**Straße**	**Telefon#**
Mickey Mouse	Main Street	4711
Minnie Mouse	Broadway	94725
Donald Duck	Broadway	95672
…	…	…

- **Primärschlüssel:**
  - Einer der Schlüsselkandidaten wird als Primärschlüssel ausgewählt
  - Wird im E/R-Diagramm unterstrichen dargestellt
  - Hat eine besondere Bedeutung bei der Referenzierung von Entitäten
  - Fachliche ↔ technische Primärschlüssel

ER-Modell

Entity type (Entitätstyp / Entity-Typ)

- Gleichartige Entitäten werden zu Entity-Typen zusammengefasst
  - Mindestens die Attribute, welche eine Entität charakterisieren, müssen gleich sein
  - Eine Entität bildet dann die Instanz eines Entitätstypen (alle Instanzen sind die *Population* des Entitätstypen)
- In der OOP wäre das also die Klasse
- Darstellung: Rechteck mit Name des Typen, verbunden mit den Eigenschaftsnamen, die eine Ausprägung haben kann



Einzahl oder Mehrzahl:

Was macht hier mehr Sinn?

ER-Modell

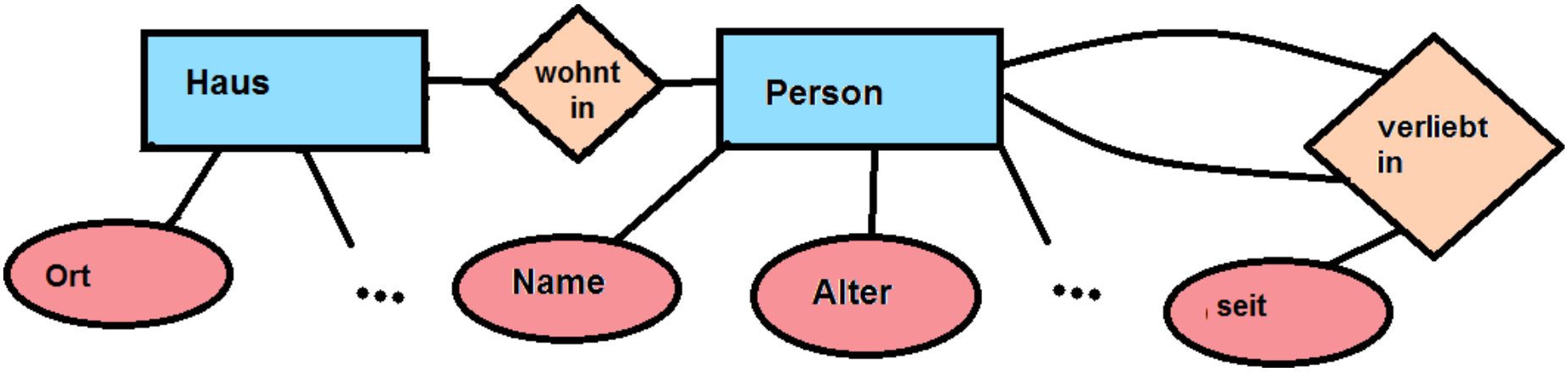
Relationship (Beziehung)

- elementare Beziehungen zwischen zwei oder mehr Entitäten
- ggf. mit eigenen Attributwerten
- Folgendes Beispiel: stellt eine Beziehung zwischen 2 Entitäten mit einem Attributwert dar:

ER-Modell

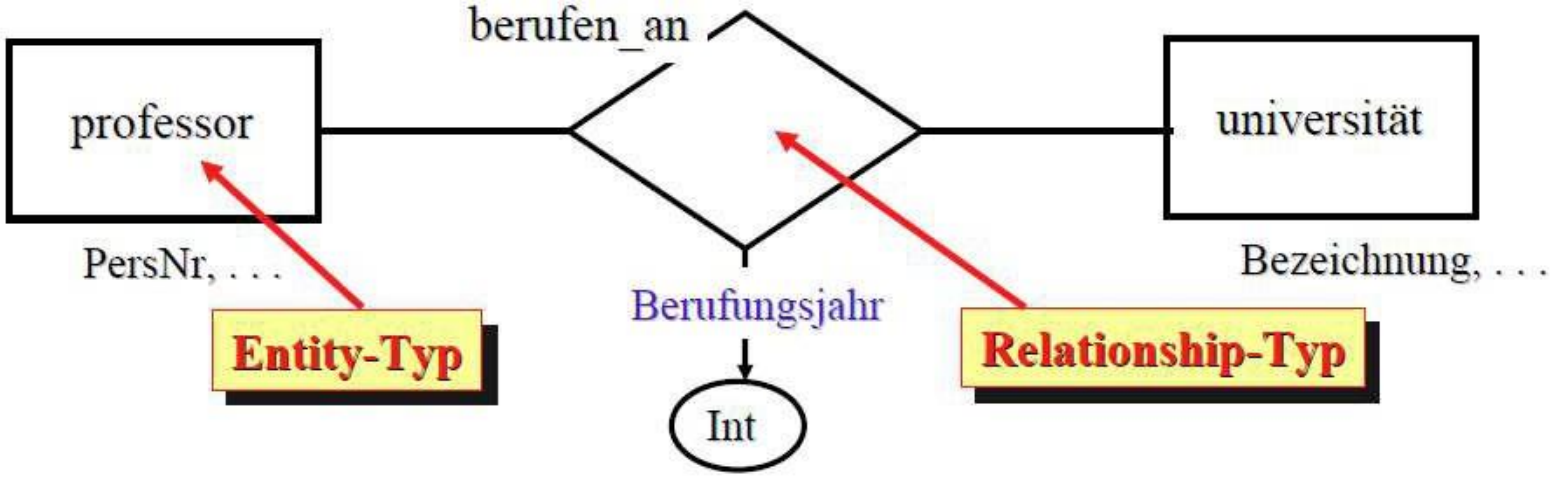
Relationship type (Relationship-Typ, Beziehungstyp)

- Gleichartige Relationships werden zu Beziehungstypen zusammengefasst
- Voraussetzung: gleiche Struktur der Attribute und gleiche Entitätstypen
- Grafisch durch Raute dargestellt, Attribute analog zum Entitätstypen

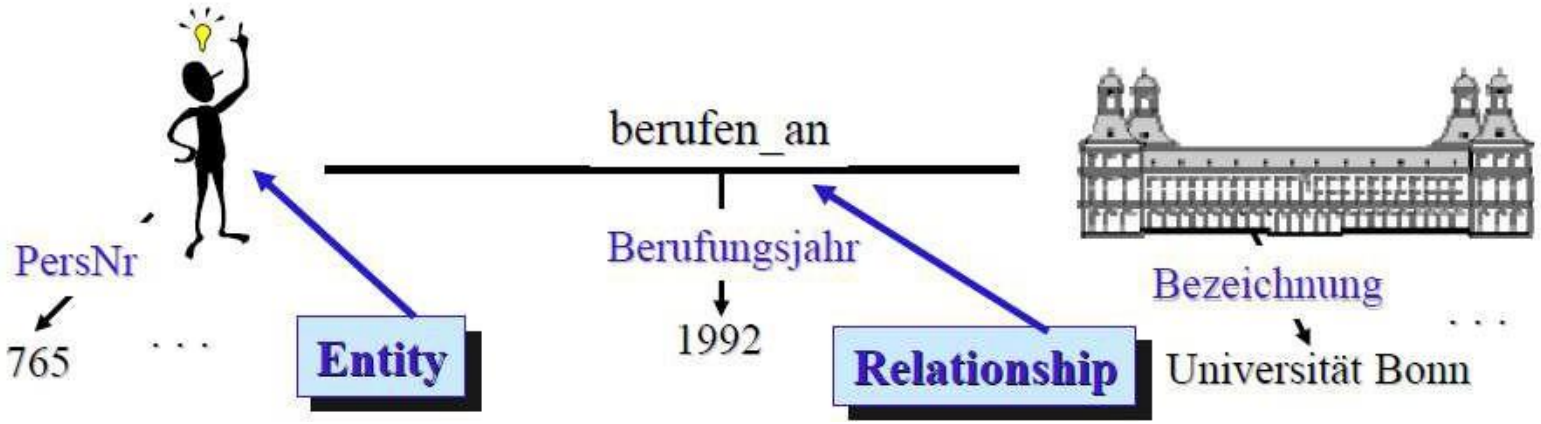


ER-Modell

Beispiel



Die Unterscheidung zwischen Typen und Instanzen fällt vielen (gerade auch) Fachleuten erstaunlicherweise schwer: Gewöhnen Sie sich gleich Präzision dabei an!

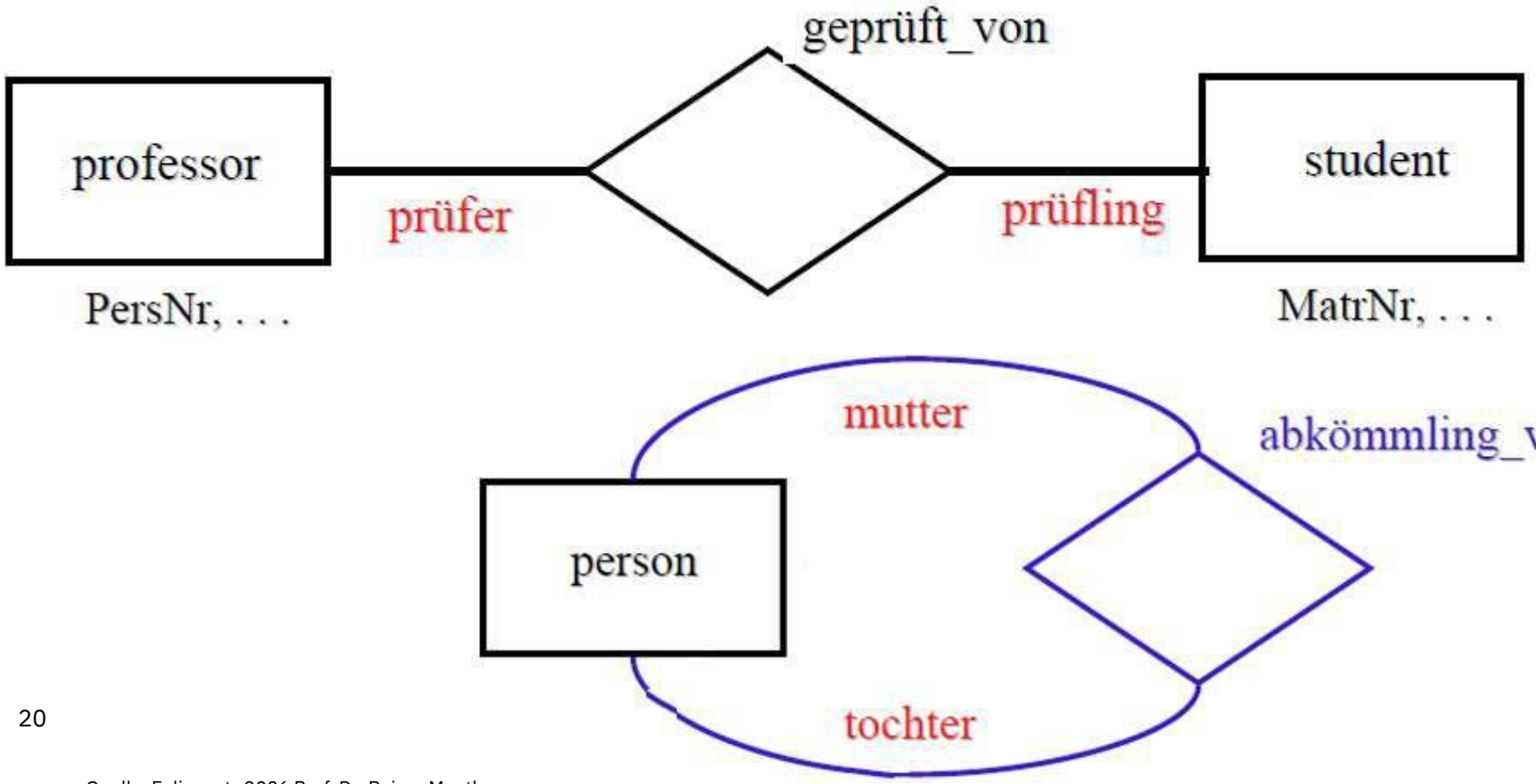


Quelle: Foliensatz 2006 Prof. Dr. Rainer Manthey

ER-Modell

Rollen

- Bezeichnungen für Beziehungen zwischen Entitäten, welche die Rolle einer Entität auf ihrer Seite der Beziehung festlegen

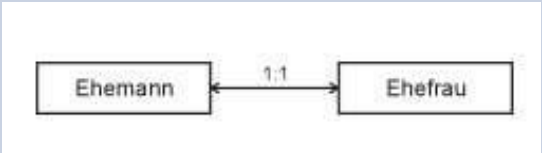
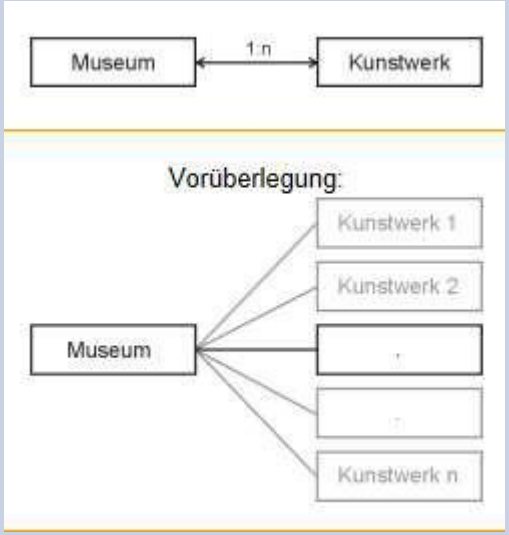
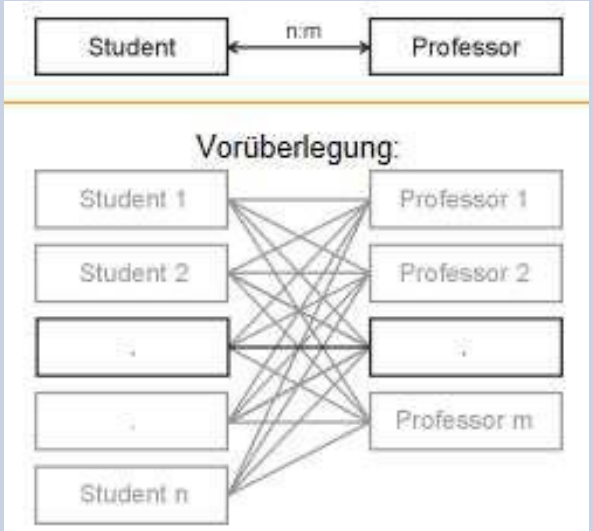


Quelle: Foliensatz 2006 Prof. Dr. Rainer Manthey

Wiederholung: Datenmodelle

Kardinalität.

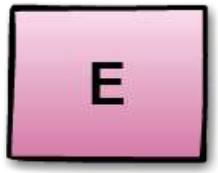
Datenmodellierung (einfache Chen-Notation)

1:1	1:n	n:m
		

Quelle:http://de.wikipedia.org/wiki/Kardinalit%C3%A4t_(Datenbankmodellierung)

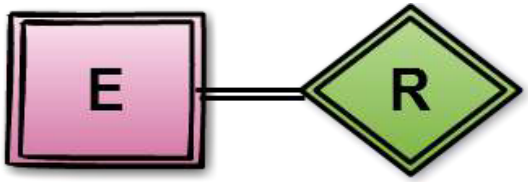
ER-Modell

Überblick der Symbole

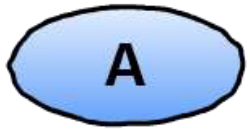


Entity-Typ

Relationship-Typ



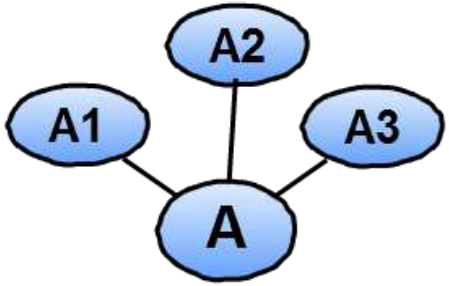
Schwacher Entity-Typ mit identifizierendem Relationship-Typ



Attribut

Schlüsselattribut

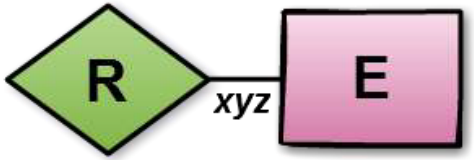
mehrwertiges Attribut



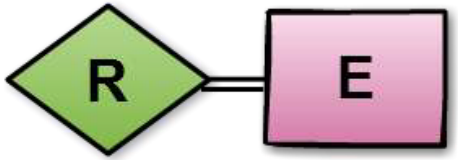
zusammengesetztes Attribut

abgeleitetes Attribut

Relationship-Typ mit Kardinalitätsangaben



Relationship-Typ mit Rollenangaben



Totale Teilnahme von E an R

Aufrufbar unter: https://cacoo.com/diagrams/QguwZ4T3RzAJV0S6
Bearbeitbar unter: https://cacoo.com/diagrams/kveko1arbUpefH6t (bitte nicht „kaputtmachen“!)

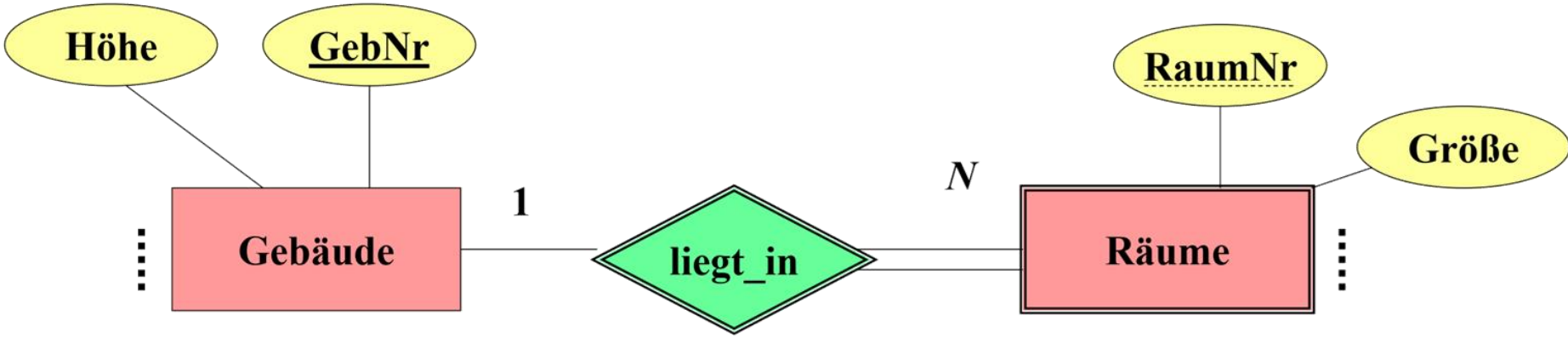
ER-Modell

Überblick der Symbole – da waren doch neue Sachen dabei…

- **Mehrwertiges Attribut**
  - Bei einem Buch zum Beispiel „Autoren“, bei einem Kunden zum Beispiel „Telefonnummern“
- **Zusammengesetztes Attribut**
  - Adresse: Straße, Hausnummer, Etage, …
- **Abgeleitete Attribute**
  - Attribut: Geburtsdatum
  - Abgeleitet: Alter
- **Schwache Entity-Typen**
  - Haben keinen eigenen vollständigen Satz an Schlüsselkandidaten und können nur über ihre Beziehung zu einer anderen Entity-Menge identifiziert werden
  - Teilschlüssel oft gestrichelt unterstrichen
  - Beispiel nächste Folien

ER-Modell

Schwache Entity-Typen



- Beziehung zwischen „starkem“ und schwachem Typ ist fast immer 1:N (in seltenen Fällen 1:1)
- Schlüssel ist: GebNr und RaumNr

?

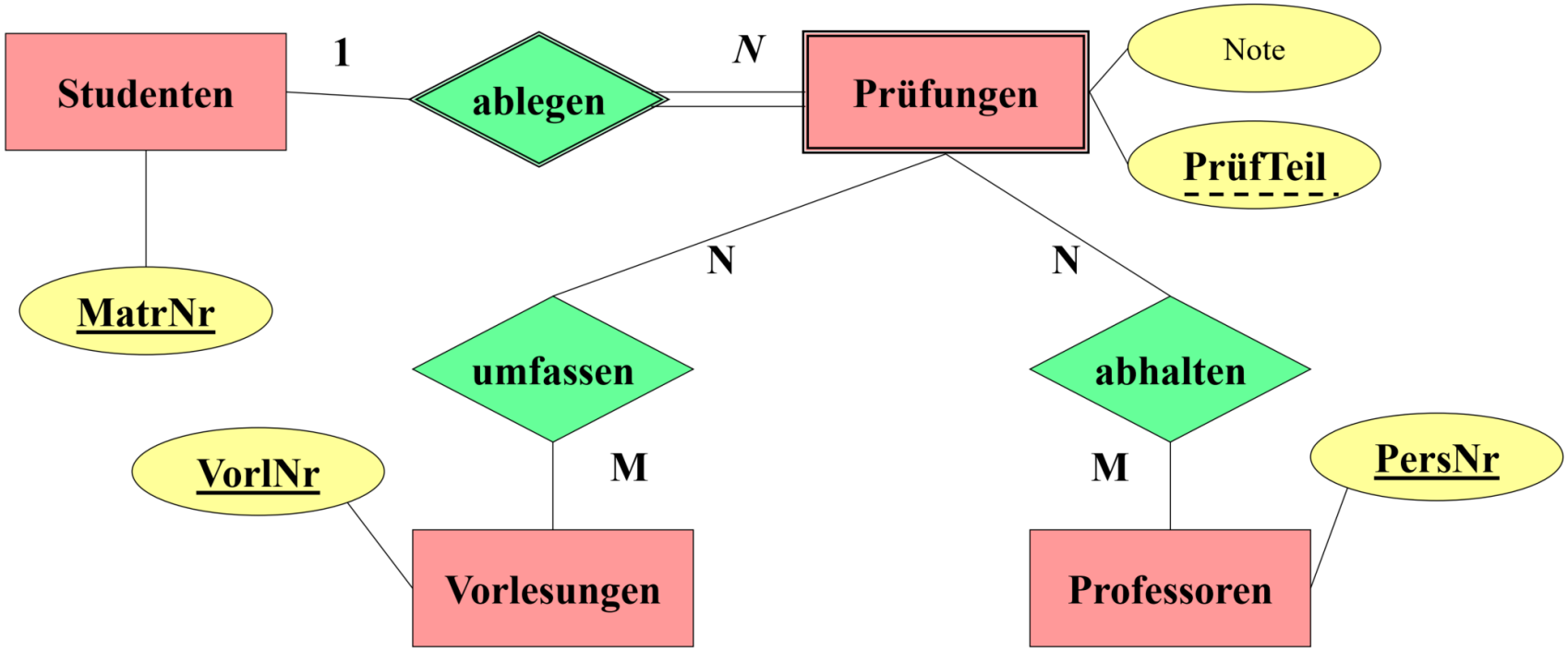
Und warum keine N:M-Beziehung?

!

Existenzabhängigkeit!

ER-Modell

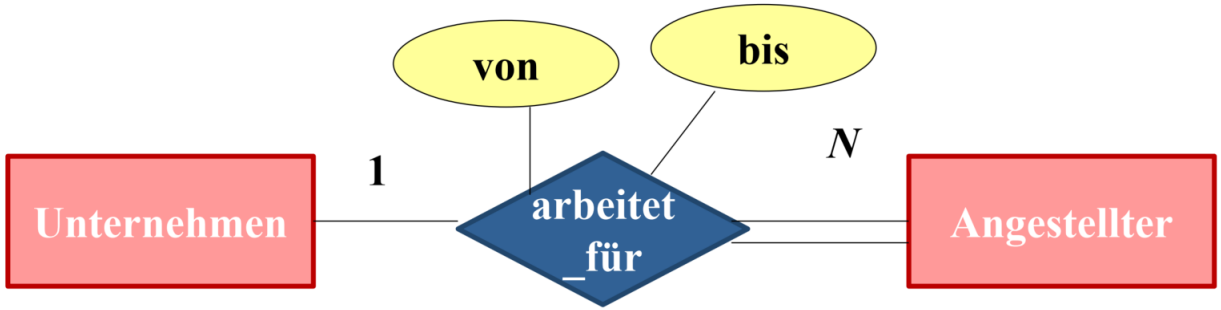
Schwache Entity-Typen



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

ER-Modell

Totale Teilnahme



- Annahme: ein Konzern mit Tochterunternehmen speichert seine Angestellten in einer Datenbank ab
- Jeder Angestellte muss in Beziehung mit einem Unternehmen stehen

ER-Modell

Übung

?

Erstellen Sie ein ER-Diagramm mit folgenden Anforderungen:

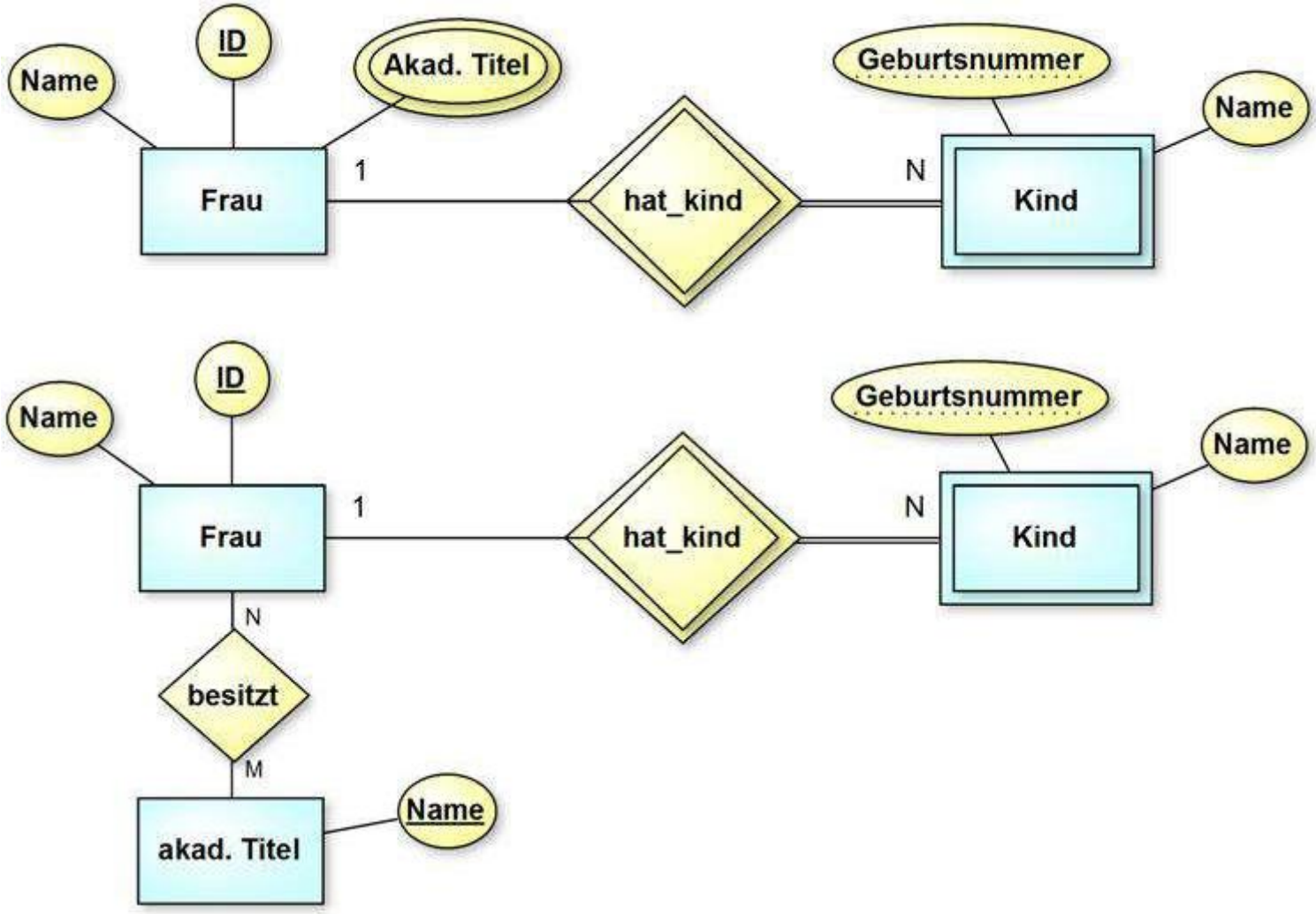
Es sollen weibliche Erwachsene mit folgenden Attributen verwaltet werden: ID, Name und eine beliebige Anzahl akademischer Titel

Darüber hinaus sollen Kinder der Frauen mit den Attributen Name und Geburtsreihenfolge je Frau (1. Kind, 2. Kind, ...) gesondert verwaltet werden.

Bearbeitungszeit: 15 Minuten

ER-Modell

2 Lösungsbeispiele

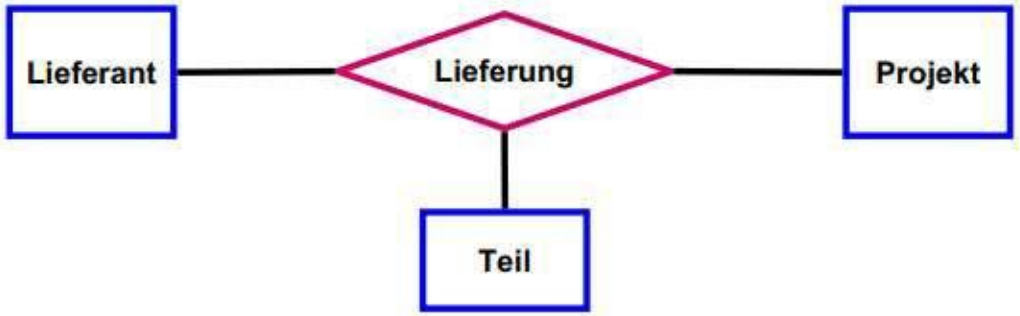


Erstellt mit pacestar uml diagrammer

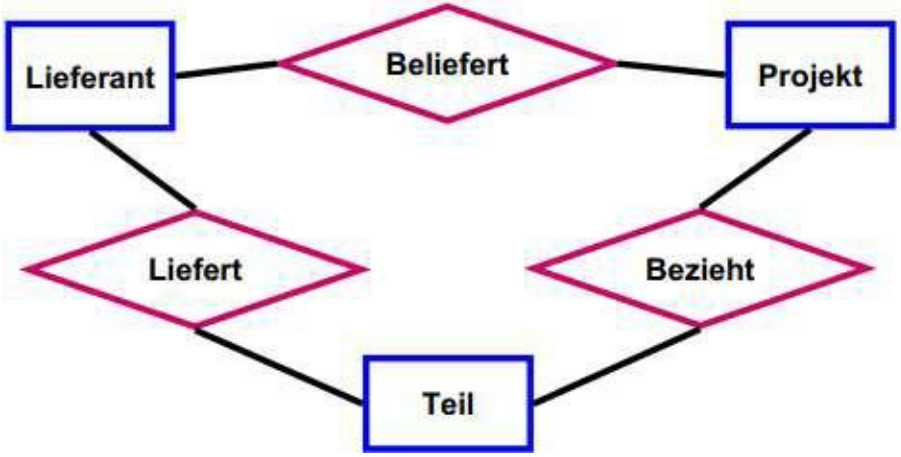
Beispiel

Ternäre Beziehungen

Beispiel einer 3-stelligen Relationship-Menge



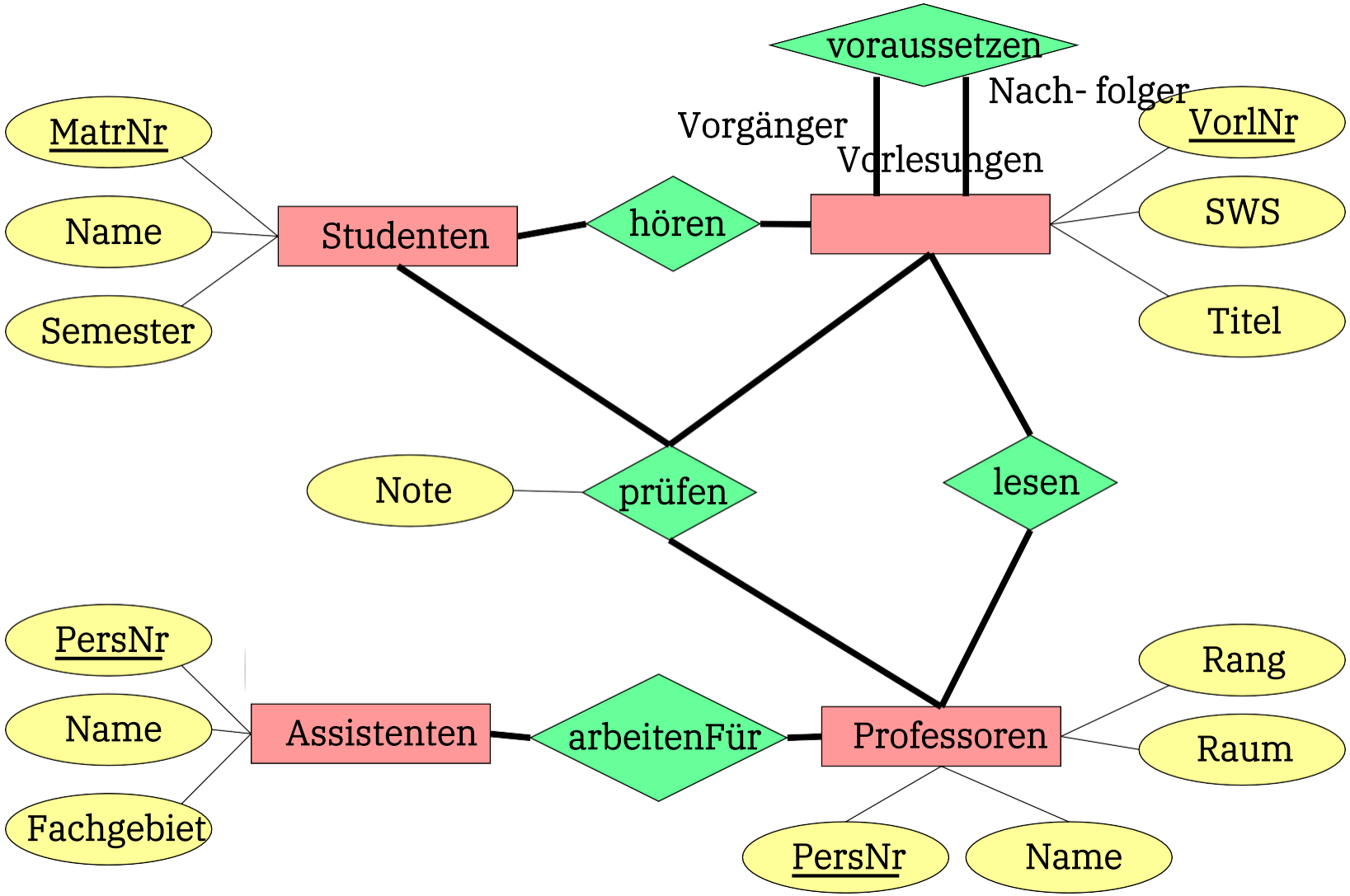
nicht gleichwertig mit drei 2-stelligen (binären) Relationship-Mengen!



Quelle: frei nach dem Foliensatz von Prof. Rahm

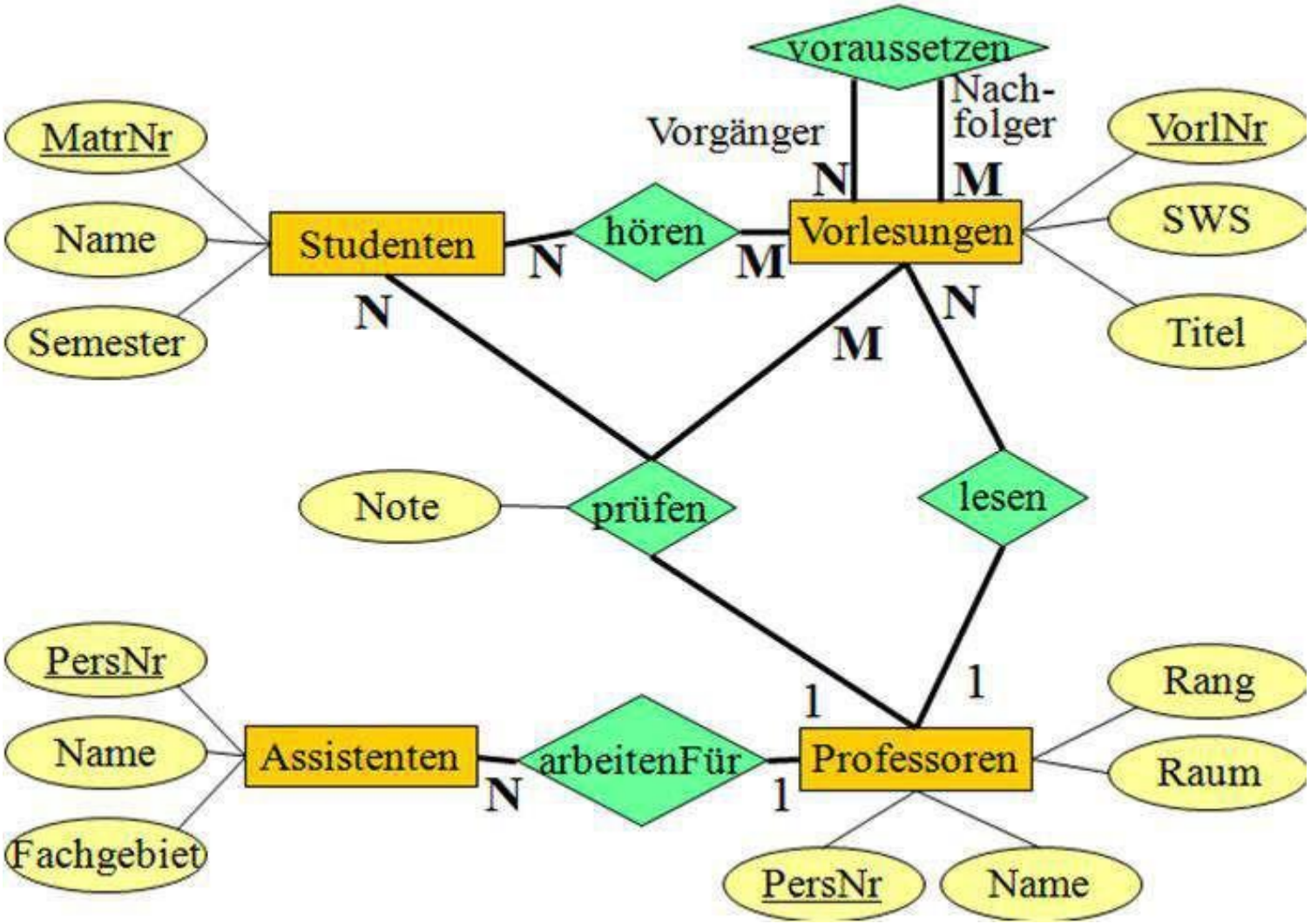
Beispiel

Uni-Datenbank ohne Kardinalitäten



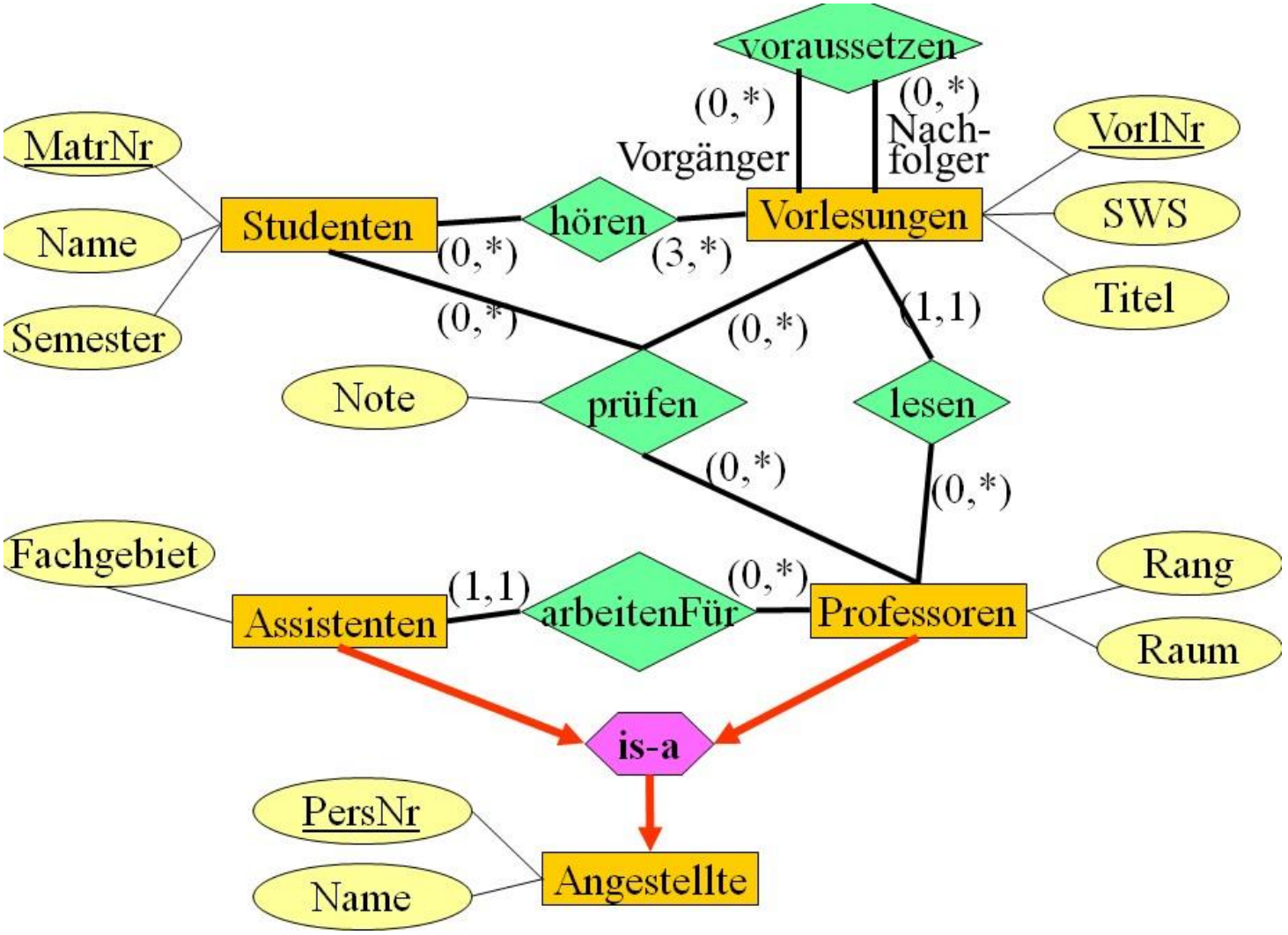
Kardinalitäten

in einfacher Chen-Notation



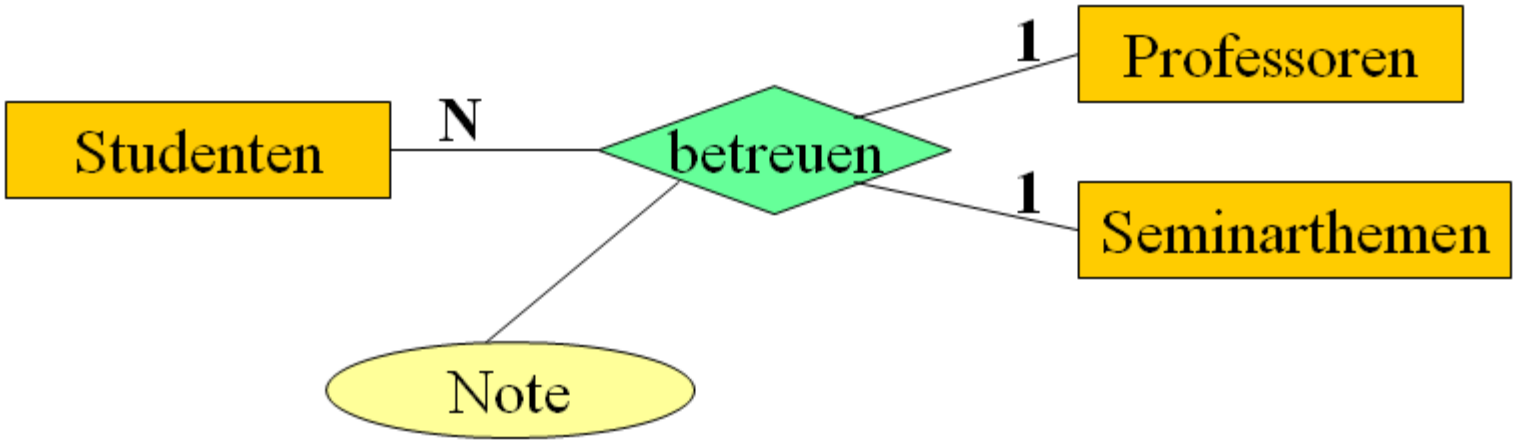
Kardinalitäten

in (min,max)-Notation



Kardinalitäten

in einfacher Chen-Notation, „Funktionalitäts“-Angabe

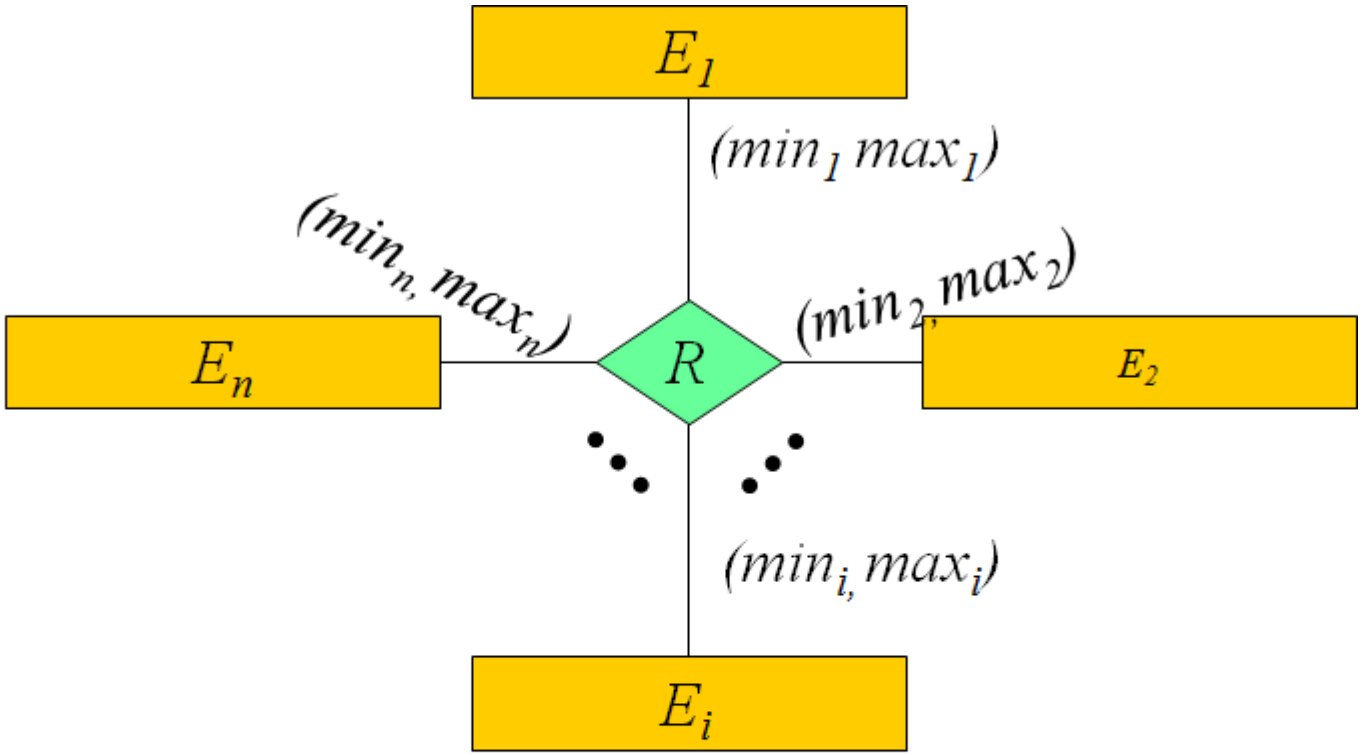


- Konsistenzbestimmungen
  - Studenten dürfen bei demselben Prof nur ein Seminarthema "ableisten" (damit ein breites Spektrum abgedeckt wird).
  - Studenten dürfen dasselbe Thema nur 1x bearbeiten, also nicht bei anderen Prof ein schon einmal erteiltes Thema nochmals bearbeiten.

- Weiterhin 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.

Kardinalitäten

in (min,max)-Notation



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

Für jedes $e_i \in E_i$ gibt es in R

- Mindestens min_i Tupel der Art $(\ldots, e_i, \ldots)$ und
- Höchstens max_i viele Tupel der Art $(\ldots, e_i, \ldots) \in R$

Kardinalitäten

in (min,max)-Notation… hä?!

- Eine Person darf bei den zusammengetragenen Beziehungen mindestens einmal, höchstens aber dreimal zu finden sein

ENTITÄTEN

Person	
Name	**Alter**
Horst	20
Peter	21
Maria	22
Maraike	23

BEZIEHUNGEN

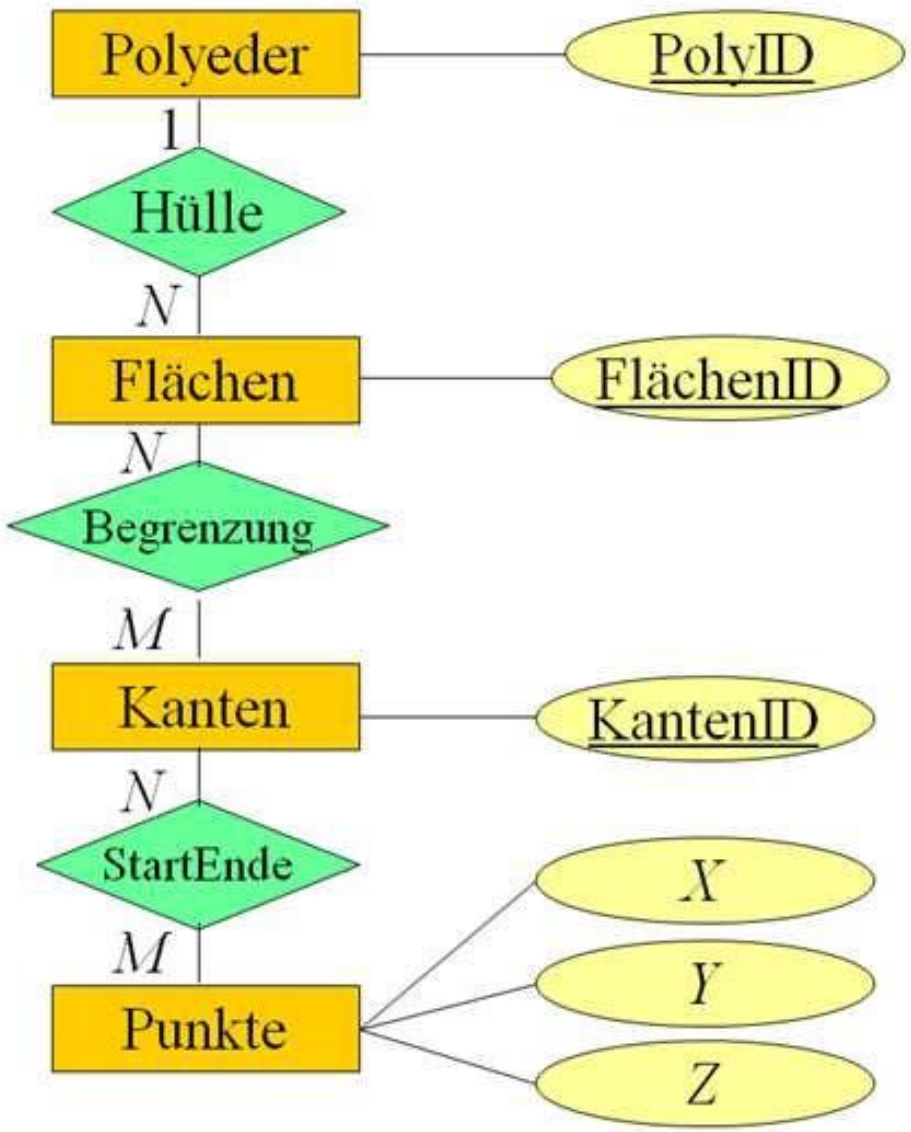
arbeitet_bei	
Name	**Firmenname**
Horst	Bank X
Peter	Kaufhaus Y
Maria	Hotel Z
Maraike	Bank X
Maraike	Kaufhaus Y
Maraike	Hotel Z

ENTITÄTEN

Firma	
Firmenname	**Ort**
Bank X	Berlin
Kaufhaus Y	Hamburg
Hotel Z	Berlin

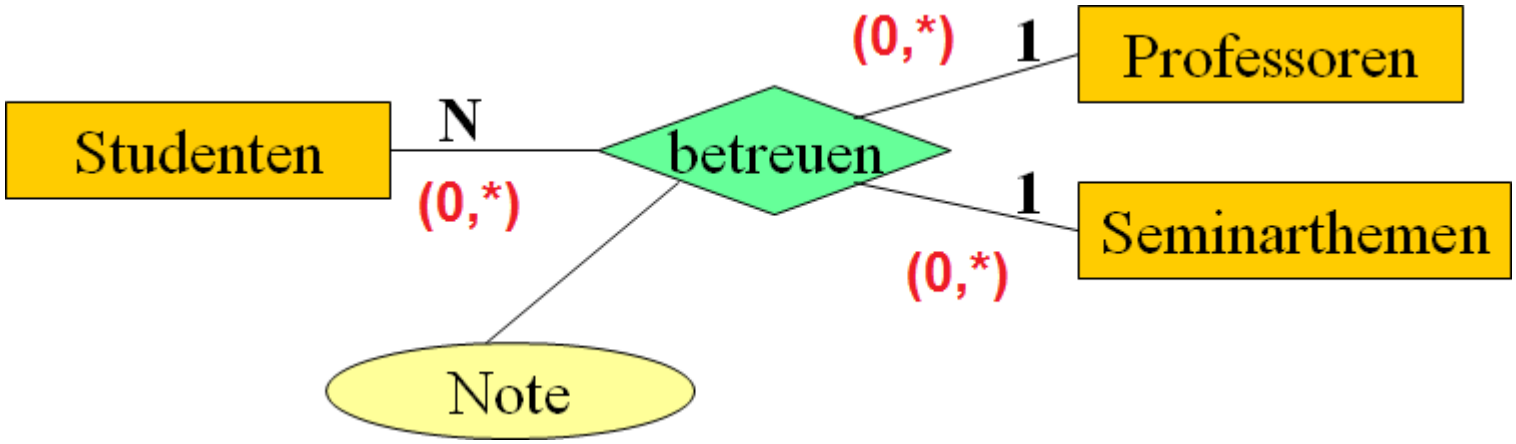
Kardinalitäten

Beispiel Begrenzungsflächendarstellung



Kardinalitäten

Unterschied Funktionalitätsangabe ↔ min/max-Notation

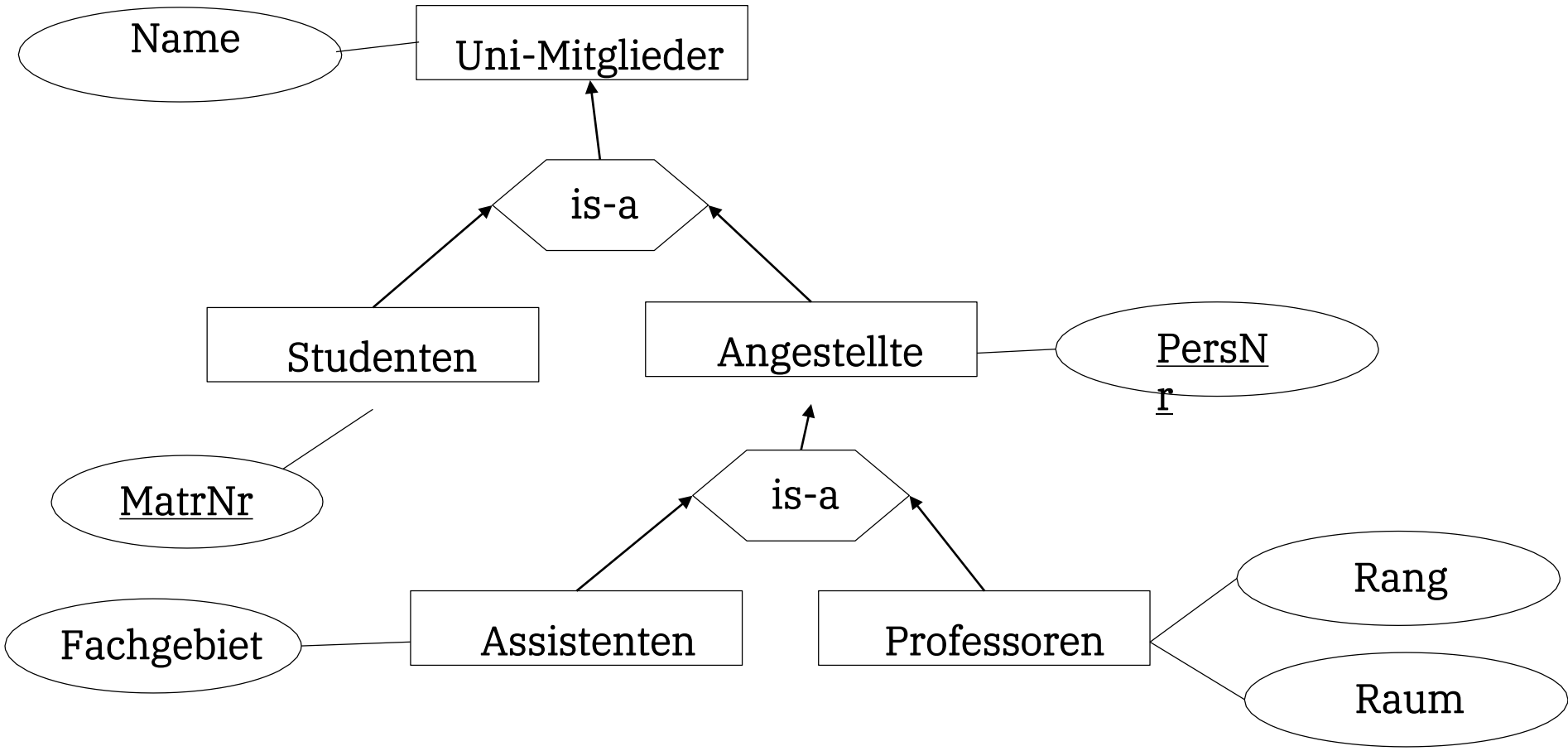


- Konsistenzbestimmungen der Funktionalitätsangabe in Chen-Notation werden mit der min,max-Notation ausgehebelt.

  ➔ Hier könnte nun ein Student das gleiche Seminarthema bei unterschiedlichen Professoren ableisten, um sich nur 1x die Arbeit zu machen, aber mehrmals dafür bewertet zu werden

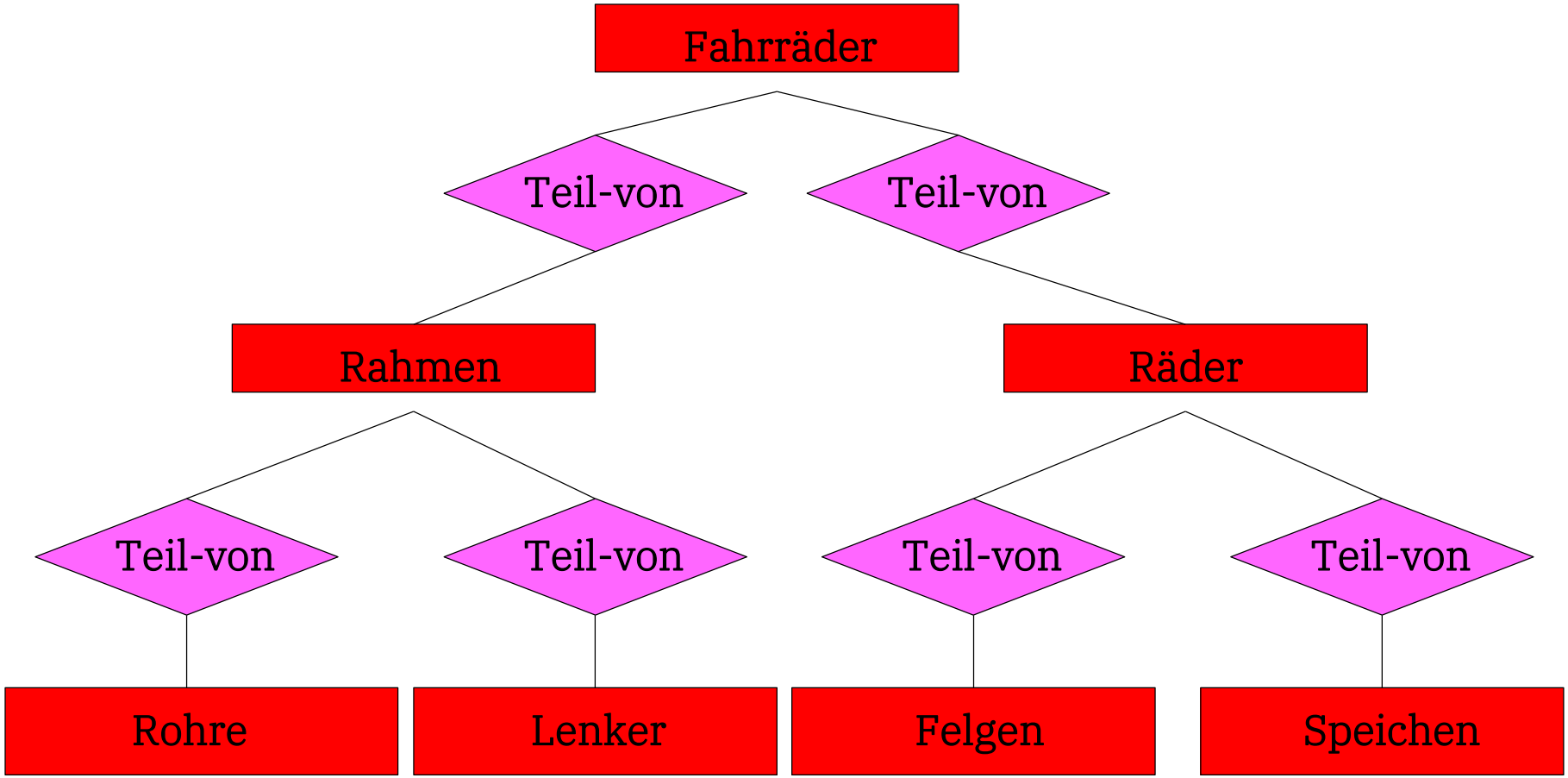
Generalisierung

Beispiel



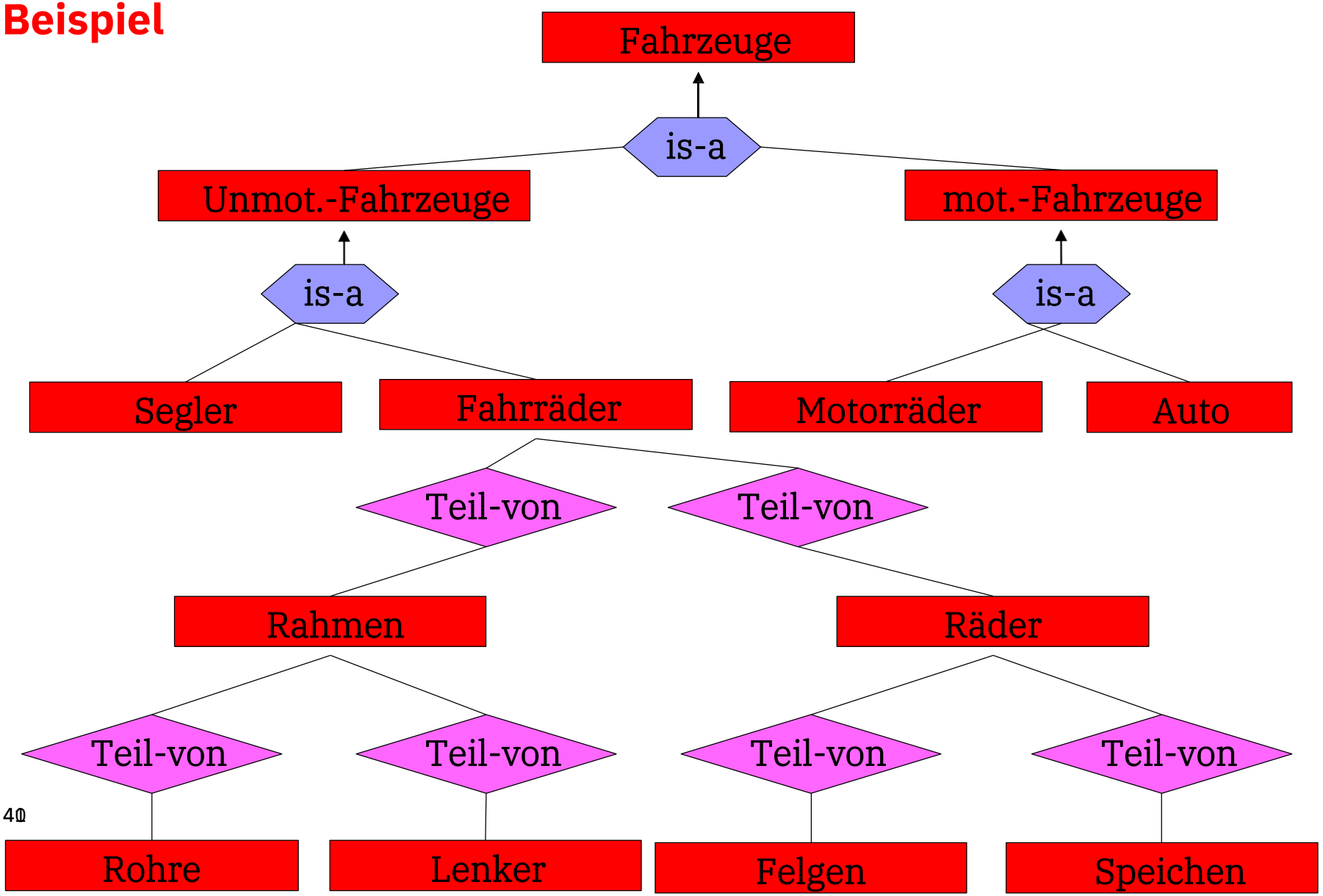
Aggregation

Beispiel (auch „part of“)



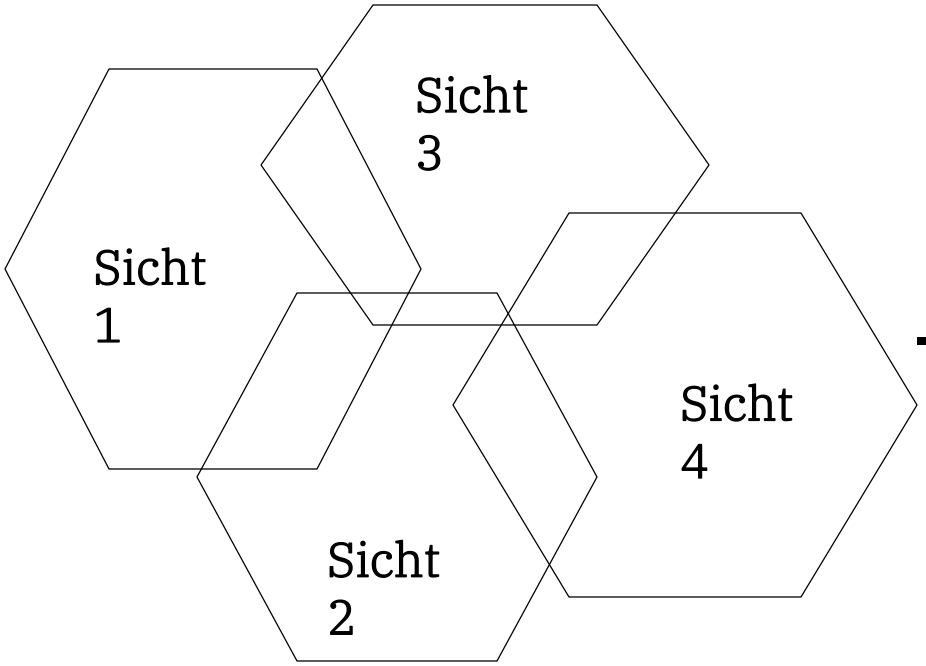
Aggregation und Generalisierung

Beispiel



Konsolisierung von Teilschemata

Allgemein



Konsoli-
dierung

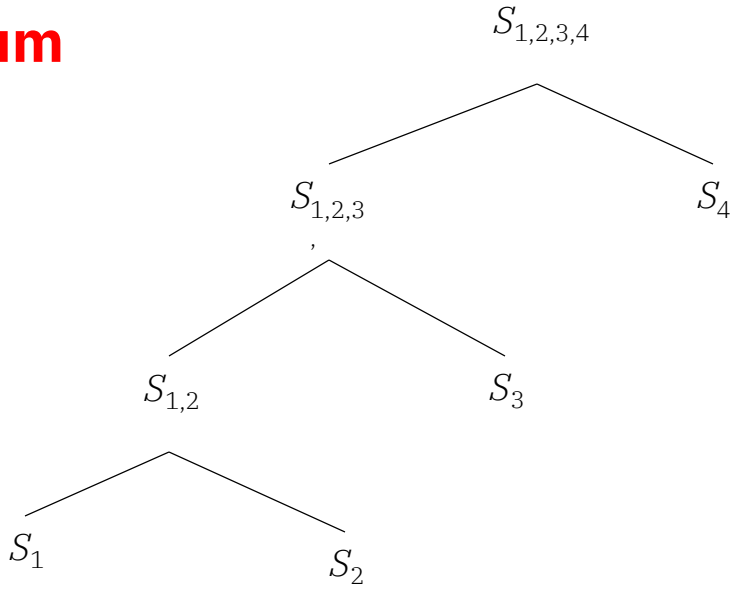
Globales Schema

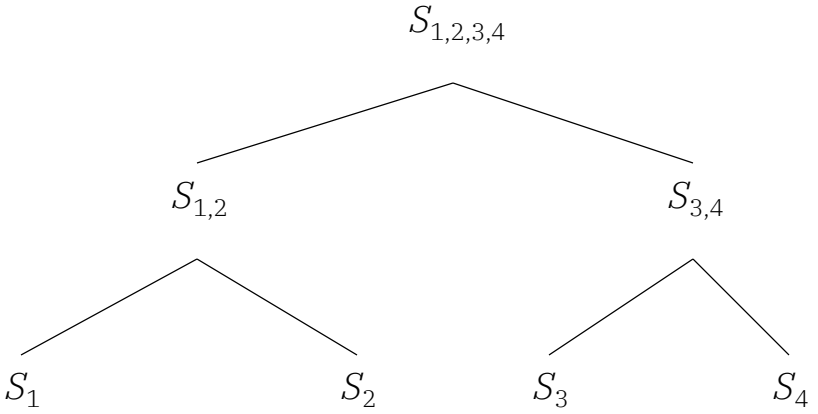
- Redundanzfrei
- Widerspruchsfrei
- Synonyme bereinigt
- Homonyme bereinigt

Konsolisierung von Teilschemata

Möglicher Konsolidierungsbaum

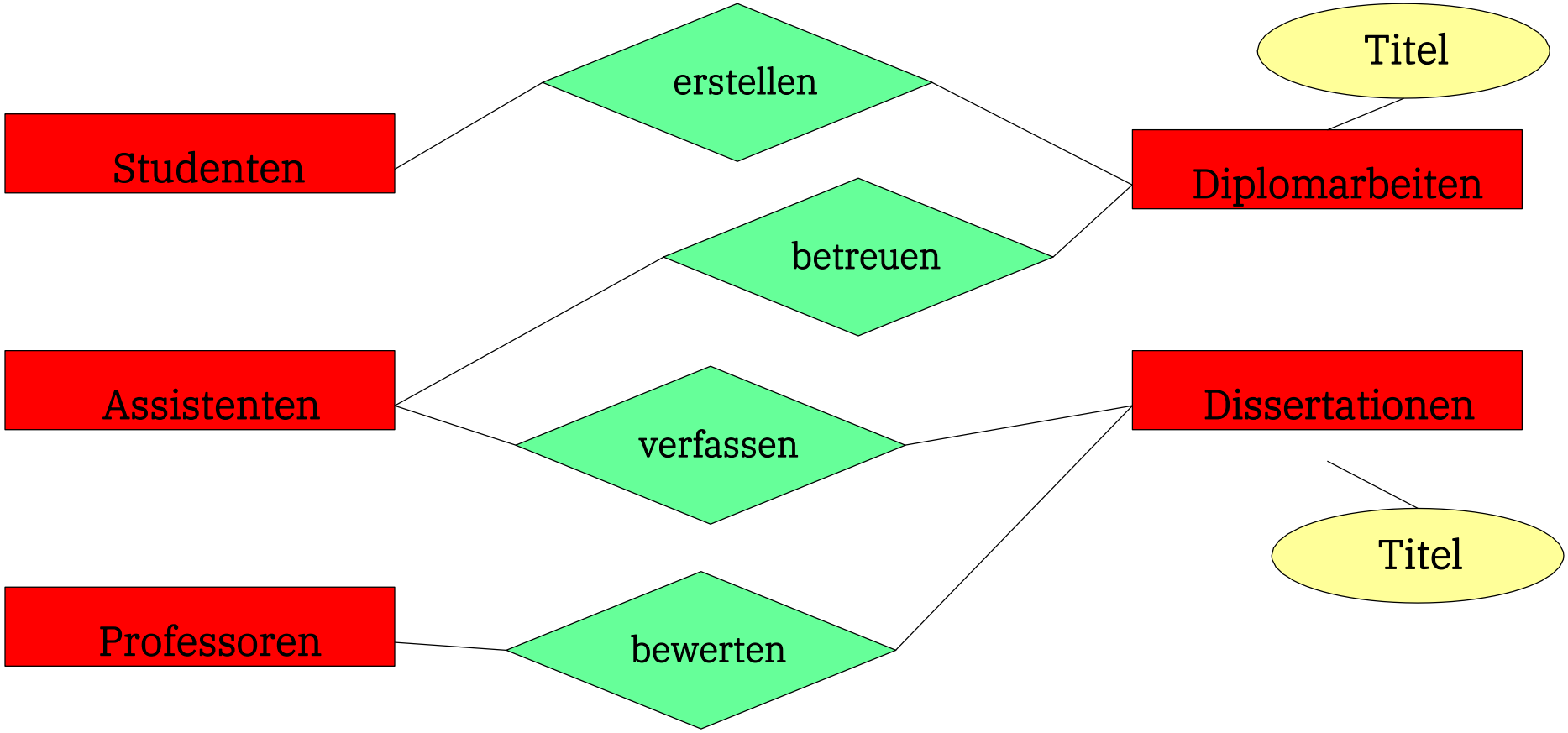
- Mögliche Konsolidierungsbäume zur Herleitung des globalen Schemas $S_{1,2,3,4}$ aus 4 Teilschemata S_1, S_2, S_3, und S_4
- Oben ein maximal hoher Konsolidierungsbaum
  - links-tief“ (left-deep)
- Unten ein minimal hoher Konsolidierungsbaum
  - Balanciert
- Beide Vorgehensweisen haben Vor- und Nachteile





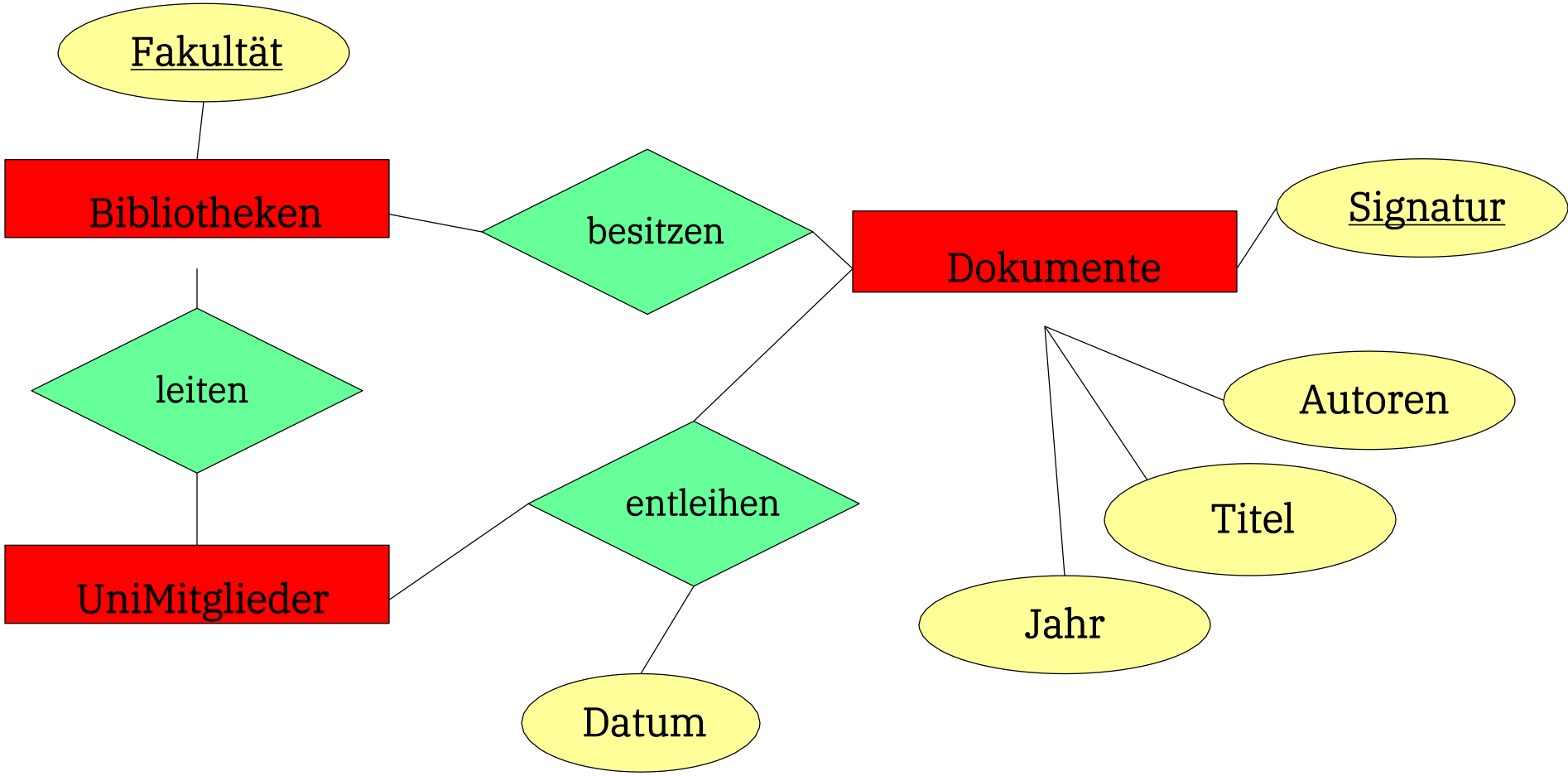
Beispiel: 3 Sichten auf eine Uni-Datenbank

Sicht 1: Erstellung von Dokumenten als Prüfungsleistung



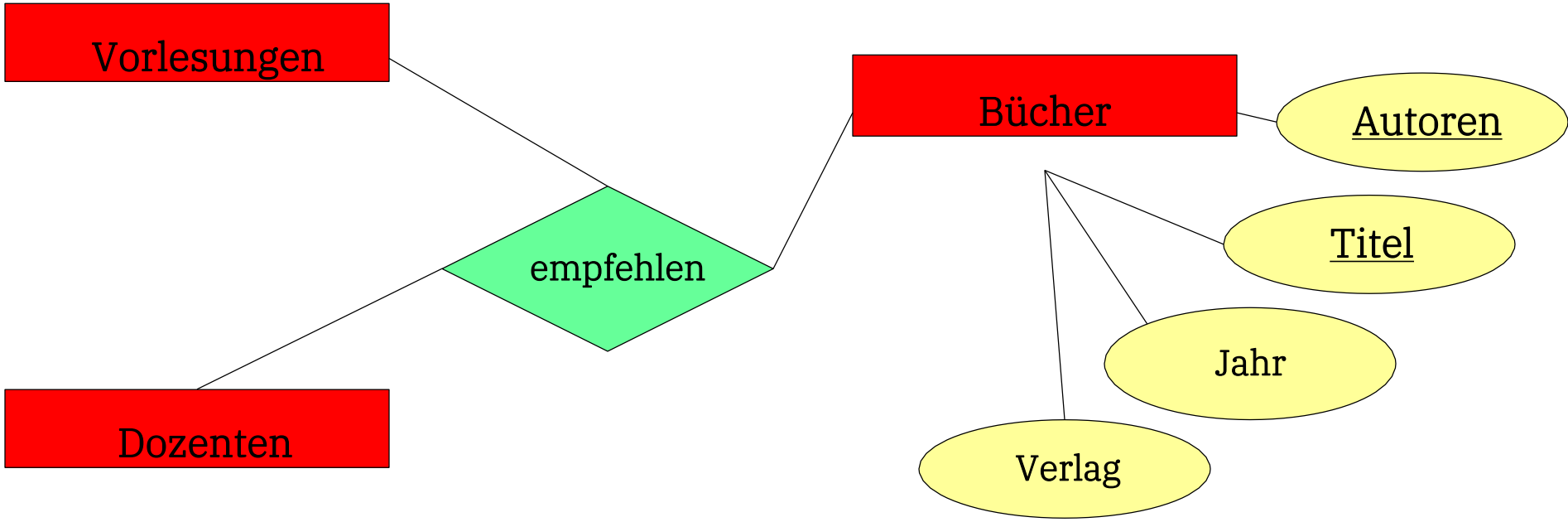
Beispiel: 3 Sichten auf eine Uni-Datenbank

Sicht 2: Bibliotheksverwaltung



Beispiel: 3 Sichten auf eine Uni-Datenbank

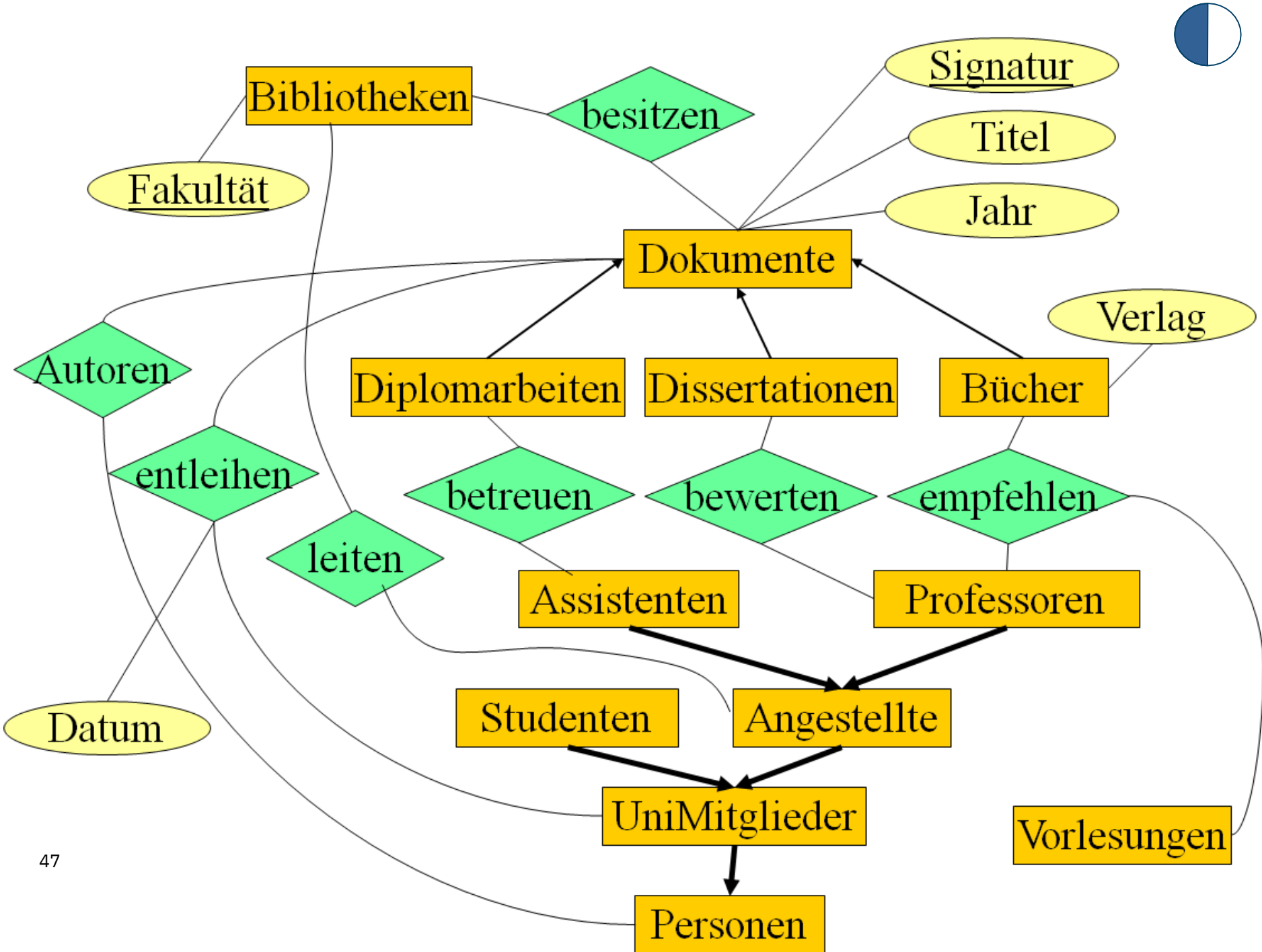
Sicht 3: Buchempfehlungen für Vorlesungen



Beispiel: 3 Sichten auf eine Uni-Datenbank

Beobachtungen

- Begriffe **Dozenten** und **Professoren** sind synonym verwendet
- 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.
- Die in Sicht 1 festgelegten Beziehungen erstellen und verfassen modellieren denselben Sachverhalt wie das Attribut Autoren von Büchern in Sicht 3.
- Wir können davon ausgehen, dass alle an der Universität erstellten Diplomarbeiten und Dissertationen in Bibliotheken verwaltet werden.
  - Alle in einer Bibliothek verwalteten Dokumente werden durch Signatur identifiziert.


Bibliotheken
besitzen
Signatur
Titel
Jahr
Fakultät
Dokumente
Verlag
Autoren
Diplomarbeiten
Dissertationen
Bücher
entleihen
betreuen
bewerten
empfehlen
leiten
Assistenten
Professoren
Datum
Studenten
Angestellte
UniMitglieder
Vorlesungen
Personen

Zusammenfassung zum ERM

Vor- und Nachteile?

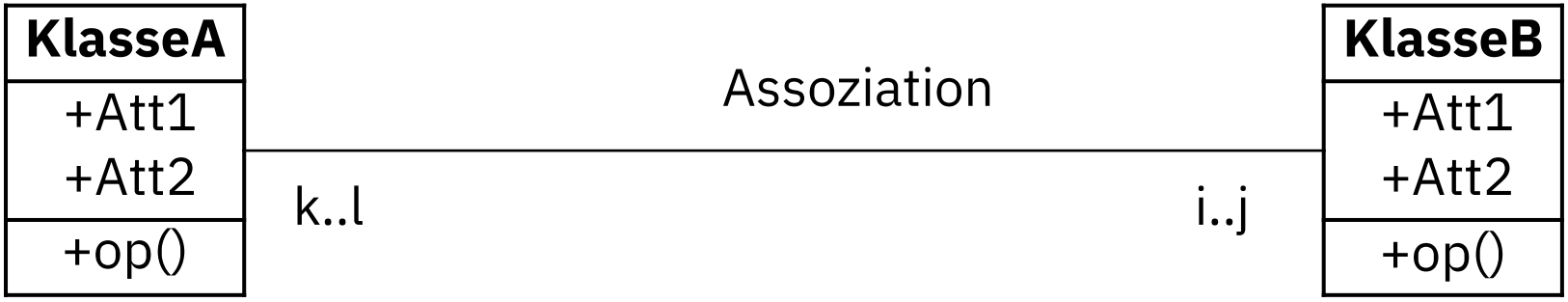
- Vorteile
  - Unabhängig von technischer Infrastruktur recht genaue Modellierung möglich
  - Darstellung ist nicht zu technisch und kann auch von Kunden nach Erklärung intuitiv verstanden werden, da „Umgangssprache“ verwendet werden kann
  - Man muss sich keine Gedanken um Datentypen und Datenintegrität machen
- Nachteile
  - Es gibt wenige Programme, mit denen ER-Diagramme qualitativ hochwertig und einfach erstellt werden können
  - Erstellung der Diagramme sehr aufwendig
  - Code-Generierung aus ER-Diagrammen so gut wie unmöglich ohne weitere Meta-Informationen
  - Verschiedene Notationsvarianten, historisch bedingt
- Vor- und Nachteil
  - Lässt manche Fragen ungeklärt und bietet somit Interpretationsspielräume (mehrwertige Attribute, ...)

UML – eine Alternative?
Überblick

- UML = Unified Modelling Language
- Standard für den OO-Softwareentwurf
- Zentrales Konstrukt ist die Klasse (class), mit der gleichartige Objekte hinsichtlich Struktur (Attribute) und Verhalten (Operationen/Methoden) modelliert werden
  - Erkennen Sie die Parallelen zum Entity-Typen?
- Assoziationen zwischen Klassen entsprechen den uns bekannten Beziehungstypen
  - Kardinalitäten werden hier in Form von Multiplizitäten dargestellt und orientieren sich an der Chen-Notation
- Darüber hinaus gibt es
  - Generalisierung
  - Aggregation
  - ...

UML im Datenbankentwurf

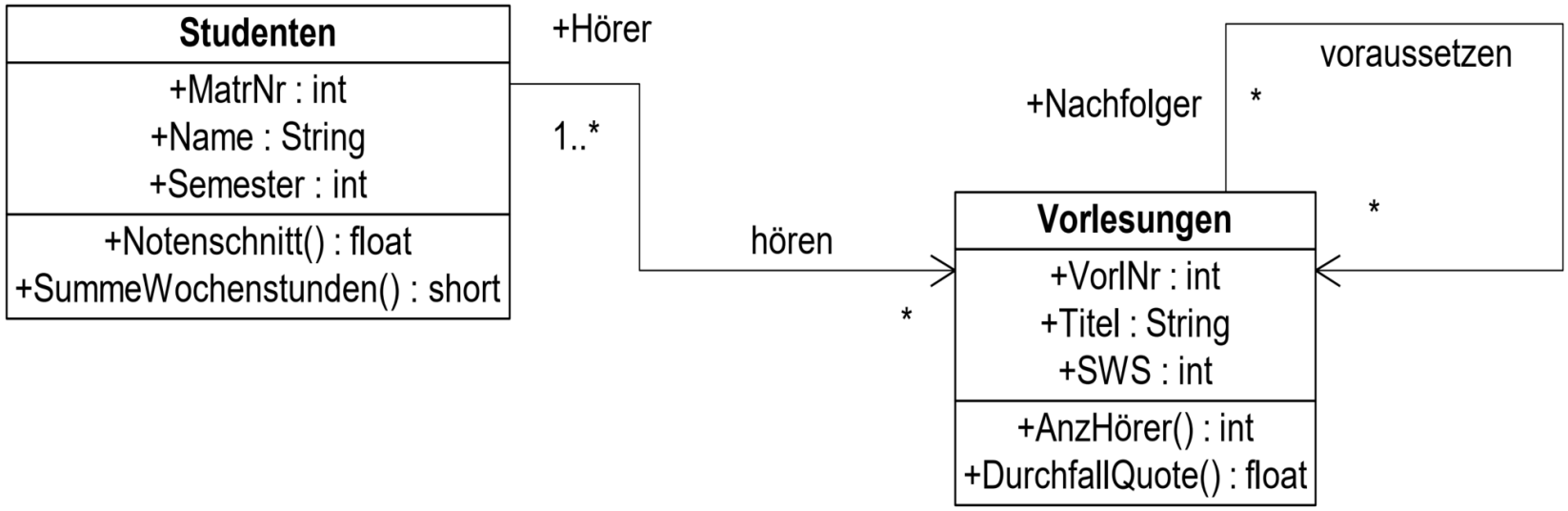
Multiplizität



- Jedes Element von KlasseA steht mit mindestens i Elementen der KlasseB in Beziehung und mit maximal j vielen KlasseB-Elementen
- Analoges gilt für das Intervall k..l
- Die Multiplizitätsangabe ist analog zur Funktionalitätsangabe im ER-Modell (Chen-Notation)
  - Beachten Sie: Auch wenn die Schreibweise ähnlich aussieht: die Multiplizität ist nicht vergleichbar mit der (min,max)-Notation!

UML im Datenbankentwurf

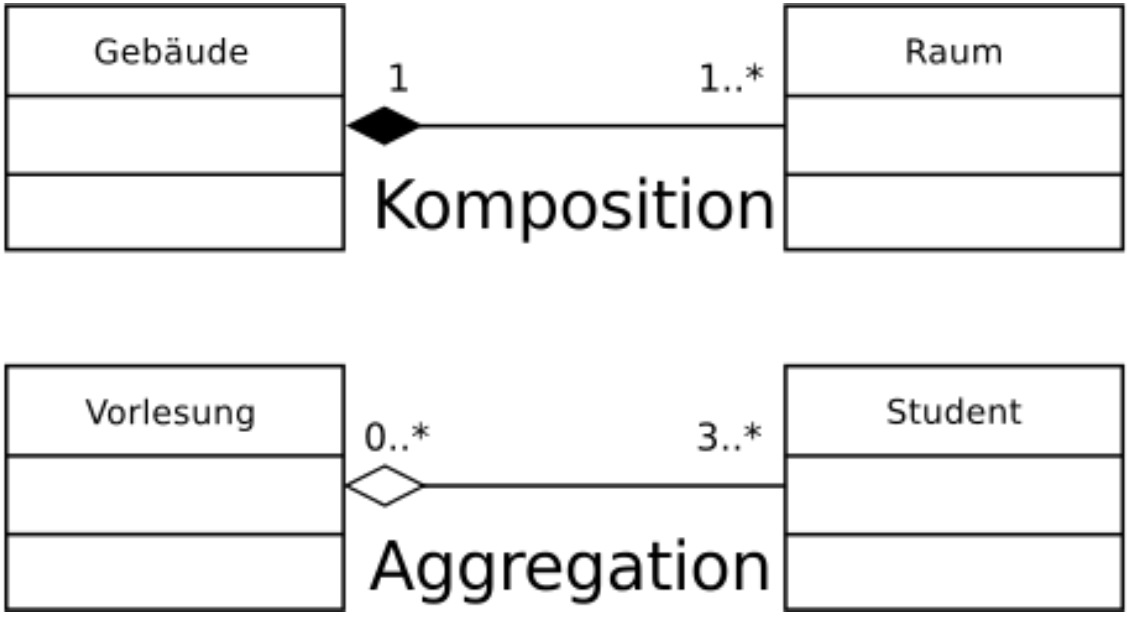
Klassen und Assoziationen



- Klassen mit schon bekannten Attributen und nun auch zusätzlich Methoden
- Angabe der Assoziationen mit Rollen, Sichtbarkeit der Rollen, Multiplizitäten, ggf. Eigenschaften, Navigationsrichtungen, Assoziationsklassen
- Datentypenangabe standardmäßig, jedoch im Rahmen eines konzeptuellen Entwurfs keine Pflicht

UML im Datenbankentwurf

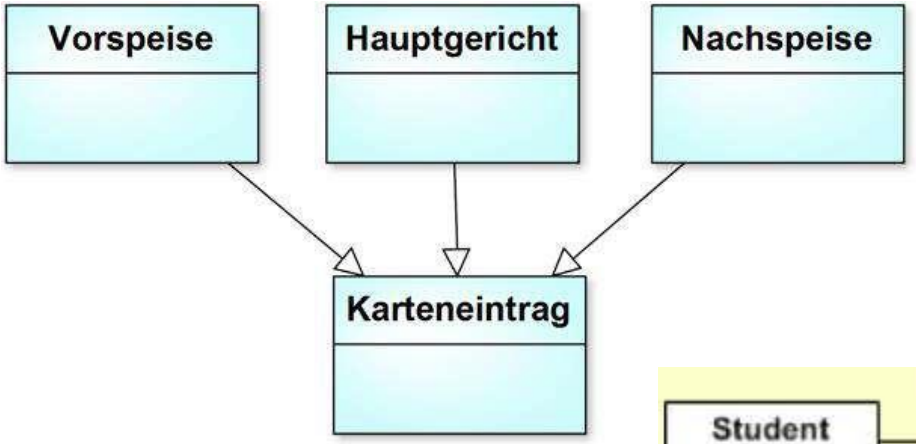
Klassen und Assoziationen: Aggregation und Komposition

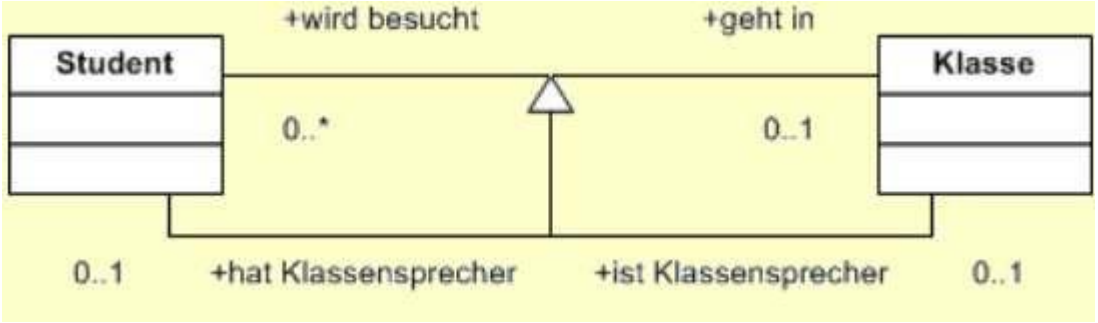


- Aggregation: Ist Teil von etwas anderem, kann aber selbständig existieren
- Komposition: Ist Teil von etwas und kann nicht selbständig existieren
  - Vgl. „Totale Teilnahme“ im ERM

UML im Datenbankentwurf

Klassen und Assoziationen: Generalisierung / Spezialisierung





- Vorspeise ist eine Spezialisierung von Karteneintrag
- Karteneintrag ist eine Generalisierung von Vorspeise
- Auch für Assoziationen möglich

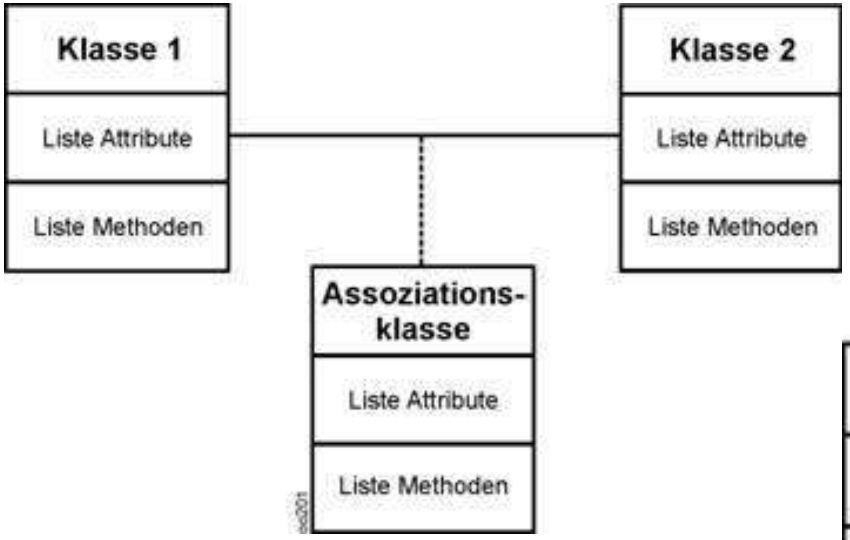
?

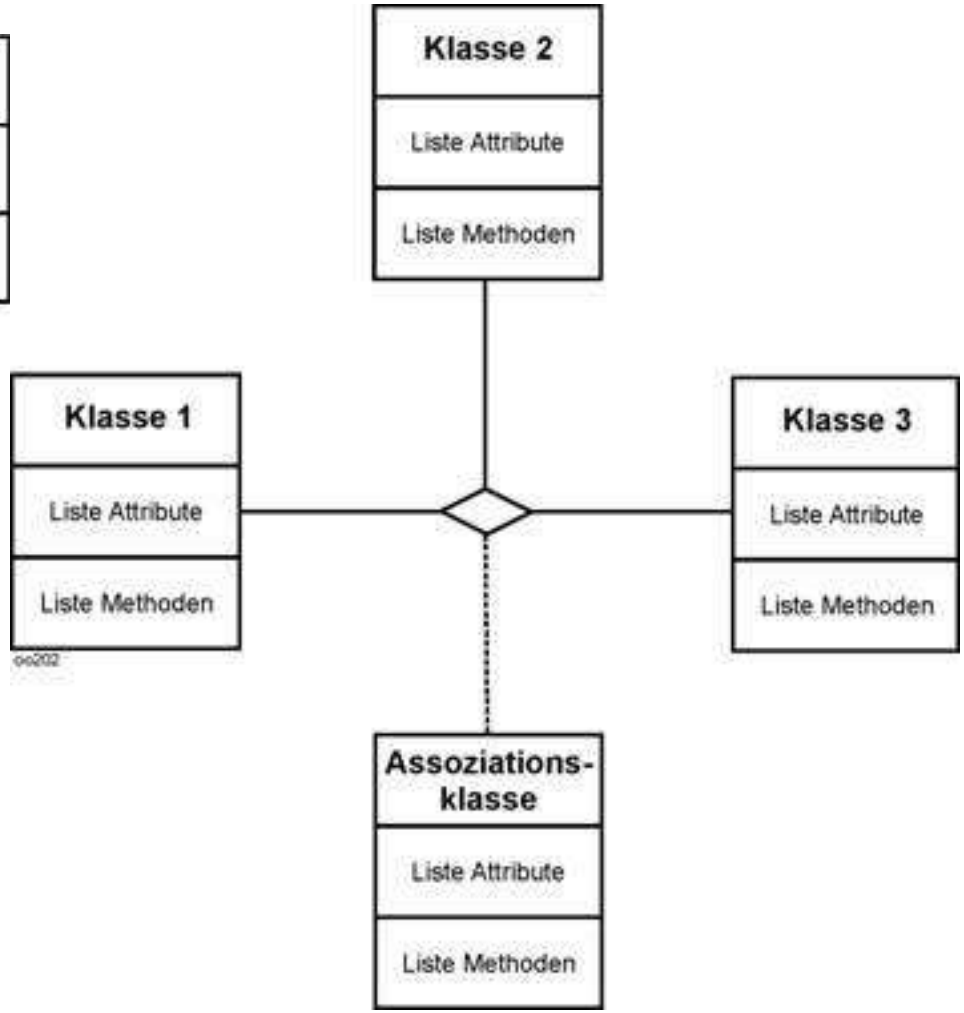
Ist das sinnvoll?

Quelle: http://home.f1.htw-berlin.de/scheibl/uml/index.htm?./Klassendiagramm/Symbol.htm

UML im Datenbankentwurf

Klassen und Assoziationen: Assoziationsklassen



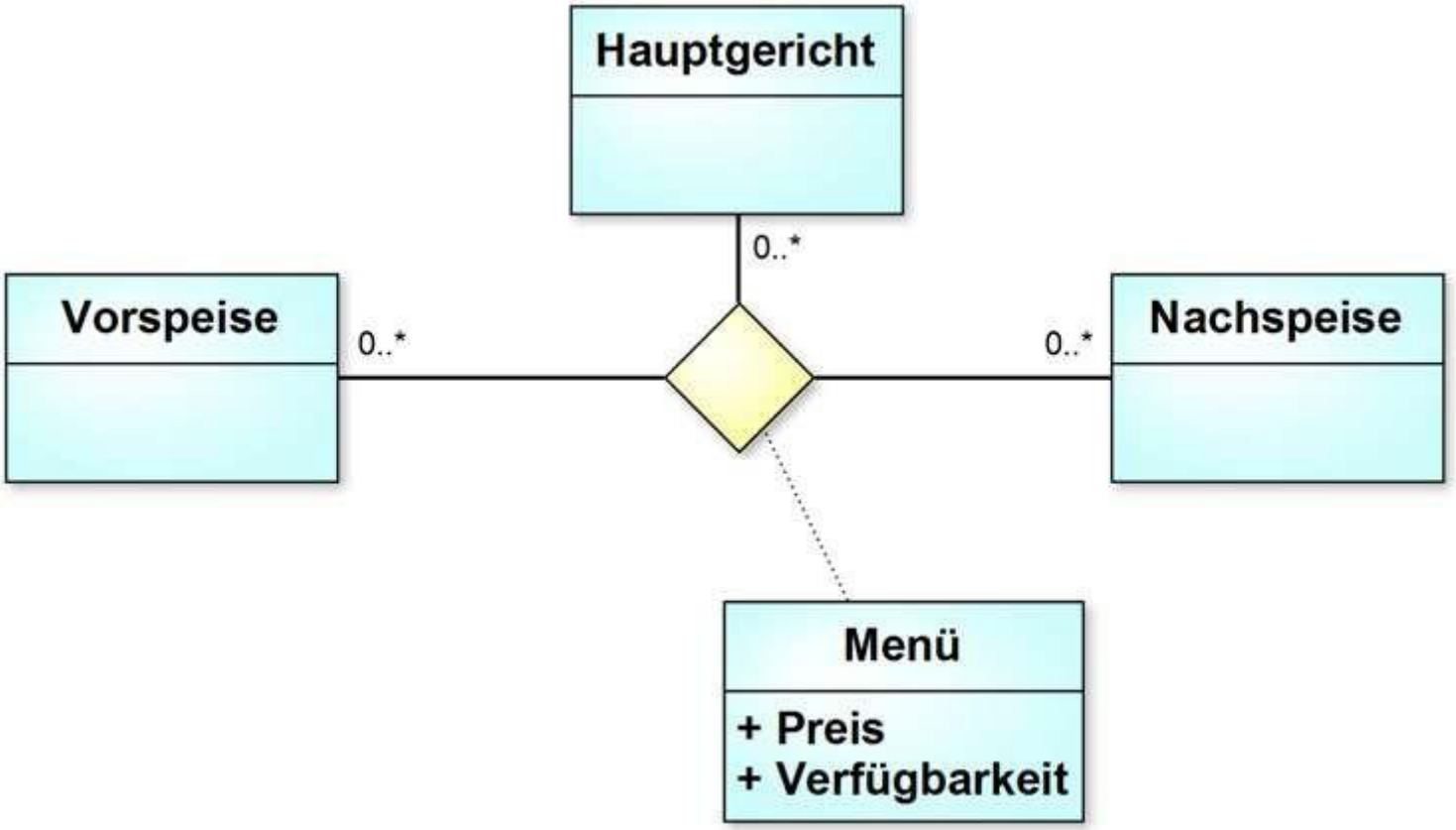


- Mehrwertige Assoziationen wie im ER-Modell möglich
- Assoziationen können auch Eigenschaften und Methoden via Assoziationsklassen erhalten

Quelle: http://www.staud.info/WebZumBuchUM/w7_f_4.htm

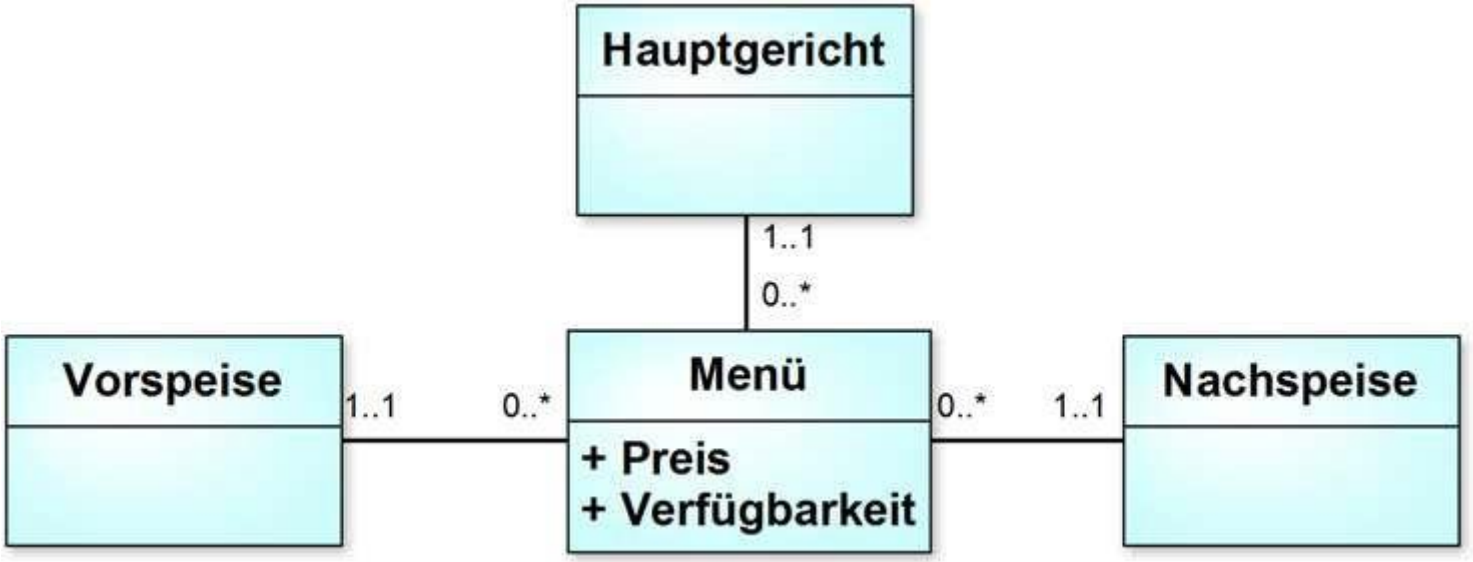
UML im Datenbankentwurf

Klassen und Assoziationen: Assoziationsklassen :: BEISPIEL



UML im Datenbankentwurf

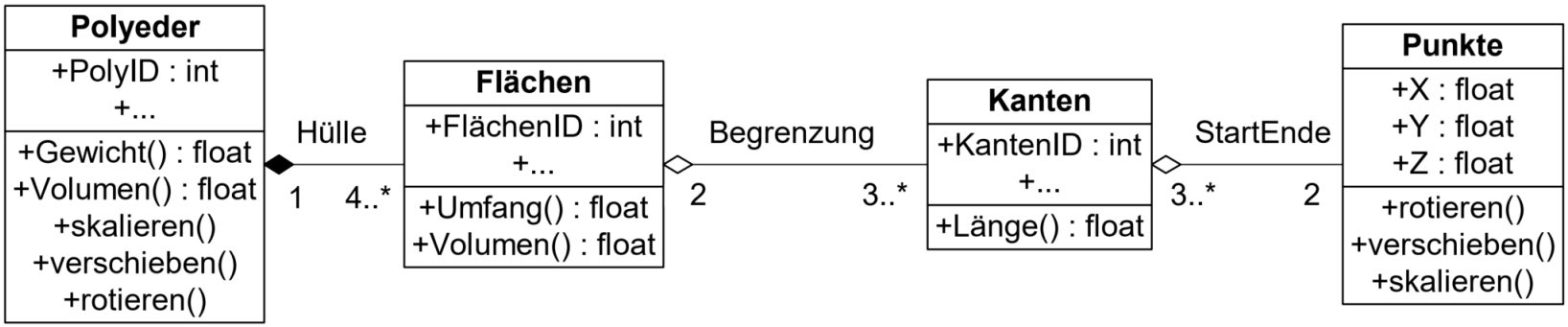
Klassen und Assoziationen: Assoziationsklassen :: BEISPIEL

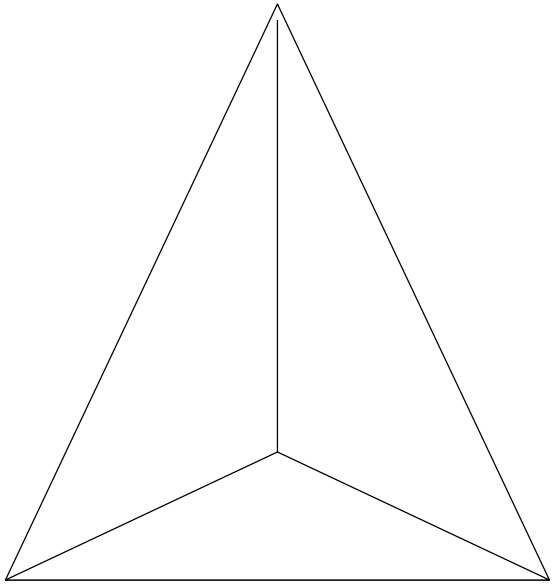


Beim Übergang von Analyse- zu Entwurfsphase: Auflösen der Assoziationsklasse und Einführen einer eigenständigen Klasse

UML im Datenbankentwurf: Multiplizitäten

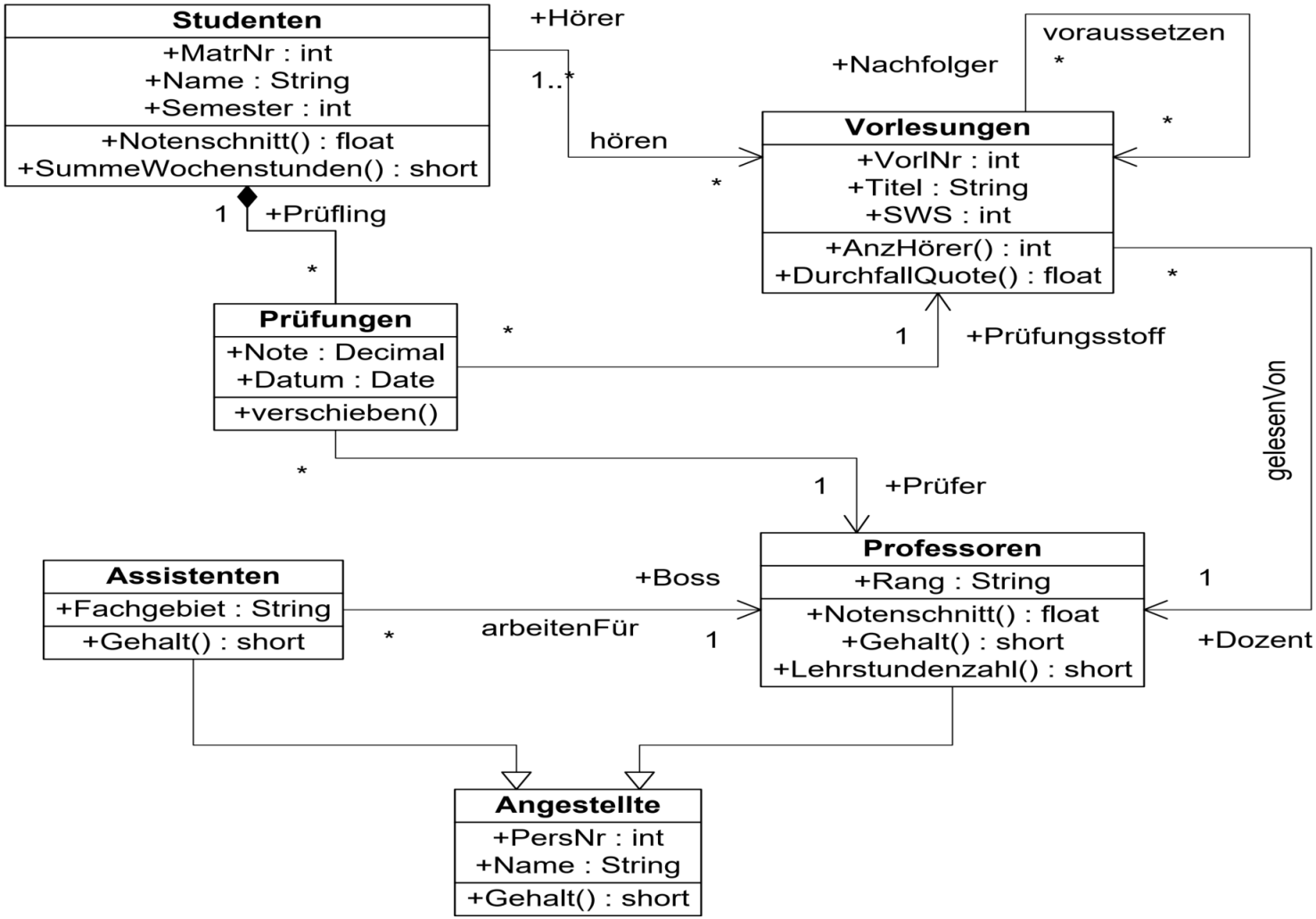
Beispiel Begrenzungsflächendarstellung





UML im Datenbankentwurf

Beispiel Uni-Datenbank



Zusammenfassung zu UML im DB-Entwurf

Vor- und Nachteile?

- Vorteile
  - Wird von sehr vielen Modellierungsprogrammen unterstützt (UML ist nicht auf DB-Entwurf beschränkt)
  - Darstellung von Generalisierung und Aggregation sind hier „fest“ definiert
  - Auch Methoden sind schon vermerkbar, obwohl es sich dabei nicht um Daten handelt
  - Code-Generierung (DDL in SQL) ist aufgrund des definierten Meta-Modells mit sehr vielen Angabemöglichkeiten einfacher und wird auch häufiger angeboten
- Nachteile
  - UML ist sehr komplex, da sehr umfangreich
  - Nicht einfach per Hand zu zeichnen (z. B. Whiteboard)
  - Es kann schnell vergessen werden, dass es sich ggf. um einen konzeptionellen Entwurf handelt, der Infrastrukturentscheidungen etc. noch nicht berücksichtigen soll

Übungen

ER-Modellierung

!

Beispiel:
Professor – Doktorand– Dissertation

?

Wenn Sie an eine Universität denken. Fällt Ihnen eine sinnvolle ternäre 1:1:1-Beziehung ein?

Falls nicht: finden Sie Beispiele außerhalb des universitären Bereichs.

Bearbeitungszeit:
2 Minuten

Übungen

ER-Modellierung

!

Existenzabhängigkeit und Definition der Primärschlüsselattribute!

?

Warum gibt es überhaupt die Möglichkeit, schwache Entity-Typen, totale Teilnahmen und part-of-Beziehungen parallel einzusetzen? Sind diese nicht redundant?

Übungen

ER-Modellierung

!

Der schwache Entitytyp S benötigt auch die PK-Attribute des identifizierenden Entitytypen I. Außerdem muss eine totale Teilnahme definiert werden bzw. in der (min,max)-Notation (1,1) auf der Seite des ehemalig schwachen Entitytypen angegeben werden.

ABER: Informationsverlust trotzdem vorhanden, siehe Zeichnung Tafel

?

Wie lässt sich ein schwacher Entity-Typ als starker Entity-Typ darstellen? Ist das dann inhaltlich dasselbe?

Bearbeitungszeit: 5 Minuten

Übungen

DB-Entwurf mit UML, Aufgabe

?

Sie verwalten ein Auskunftssystem für Züge (darunter S-Bahn, IC, ICE). Aus dem System sollen Startbahnhöfe, Zielbahnhöfe, Zwischenhalte, Gleise und jeweilige Abfahrtszeiten hervorgehen.

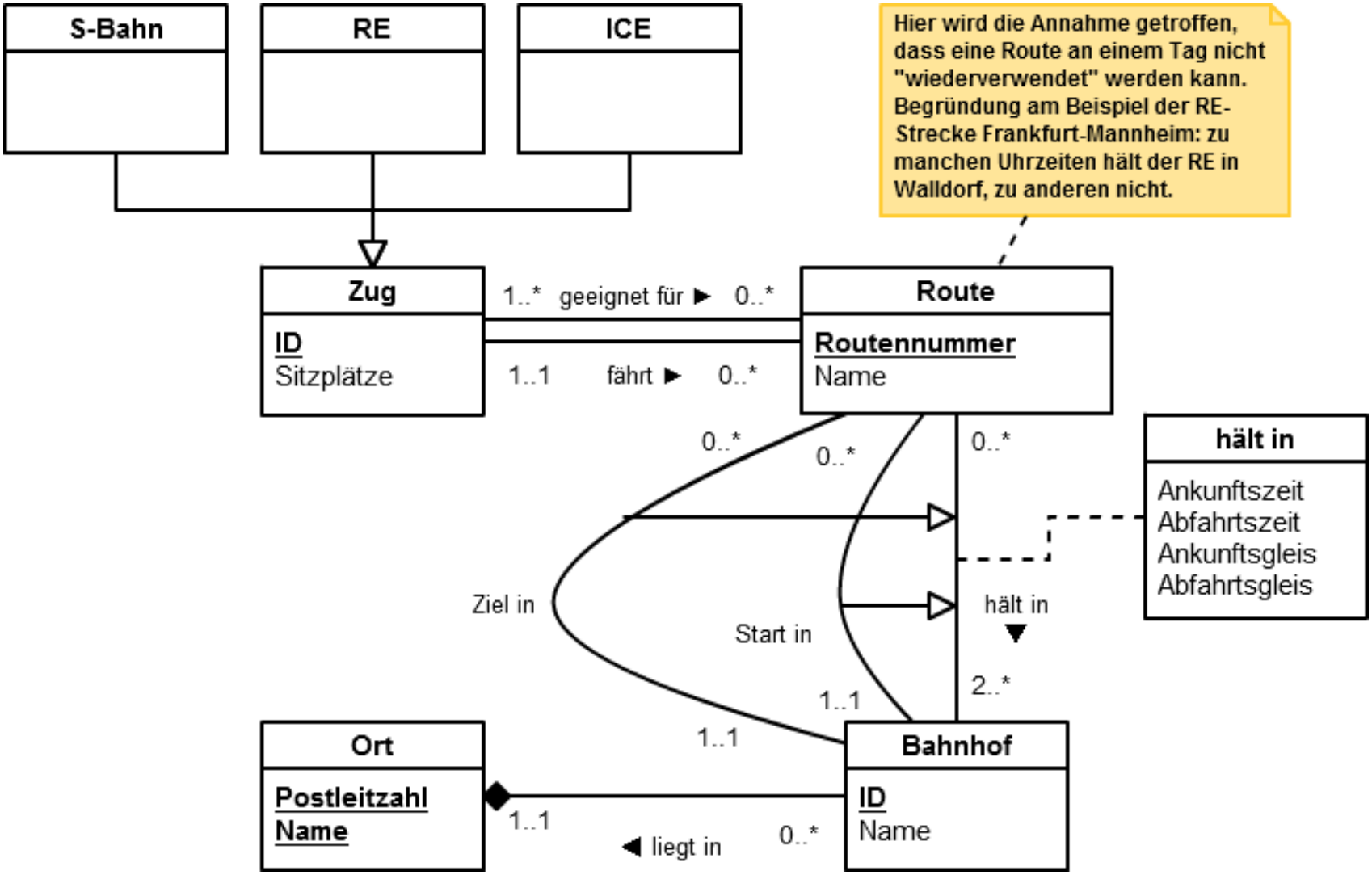
Erstellen Sie ein konzeptuelles DB-Schema in UML.

Falls Sie es nicht per Hand zeichnen wollen: Verwenden Sie dazu eine geeignete UML-Software. (Sollten Sie keine kennen / installiert haben, suchen Sie bitte nicht lange und verwenden eine online-Software wie cacoo.com oder draw.io)

Bearbeitungszeit: 20 Minuten

Übungen

DB-Entwurf mit UML, Lösungsvorschlag



Datenbankentwurf

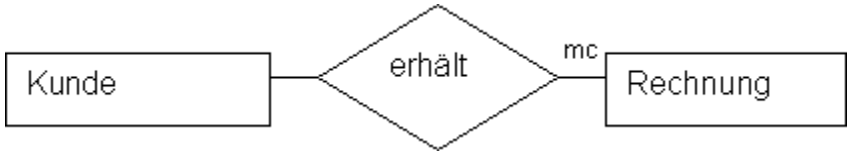
Modifizierte Chen-Notation

- Sie kennen nun die Chen-Notation mit den Kardinalitäten 1:1, 1:N und N:M
  - Das sagt aber nichts darüber aus, inwiefern eine der Seiten verpflichtender Teilnehmer ist
  - „Totale Teilnahme“ ist eine gewisse Alternative
- Abhilfe schafft hier die „modifizierte Chen-Notation“
  - Für die Datenstrukturierung interessieren wie bei der normalen Chen-Notation nicht die genauen Zahlen, sondern nur die Typen, welche ein wenig erweitert werden:
    - genau eine Entität (1)
    - keine oder eine Entität (c - von engl. can)
    - mehrere Entitäten (m- von engl. multiple)
    - keine, eine oder mehrere Entitäten (mc - von engl. multiple can)

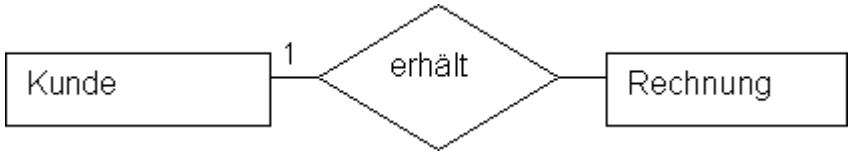
Datenbankentwurf

Modifizierte Chen-Notation

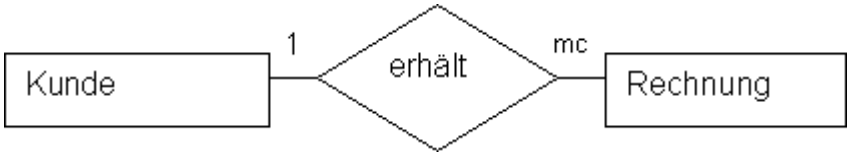
- Beispiel: Versandhaus-Datenbank mit Kunden und Rechnungen
  - Ein Kunde erhält keine (wenn er nichts bestellt hat) oder mehrere (wenn er öfters bestellt hat) Rechnung



  - liest man die Beziehung "erhält" in umgekehrter Richtung als "geht an", so liegt die Kardinalität 1 vor, d. h. eine Rechnung geht an genau einen.

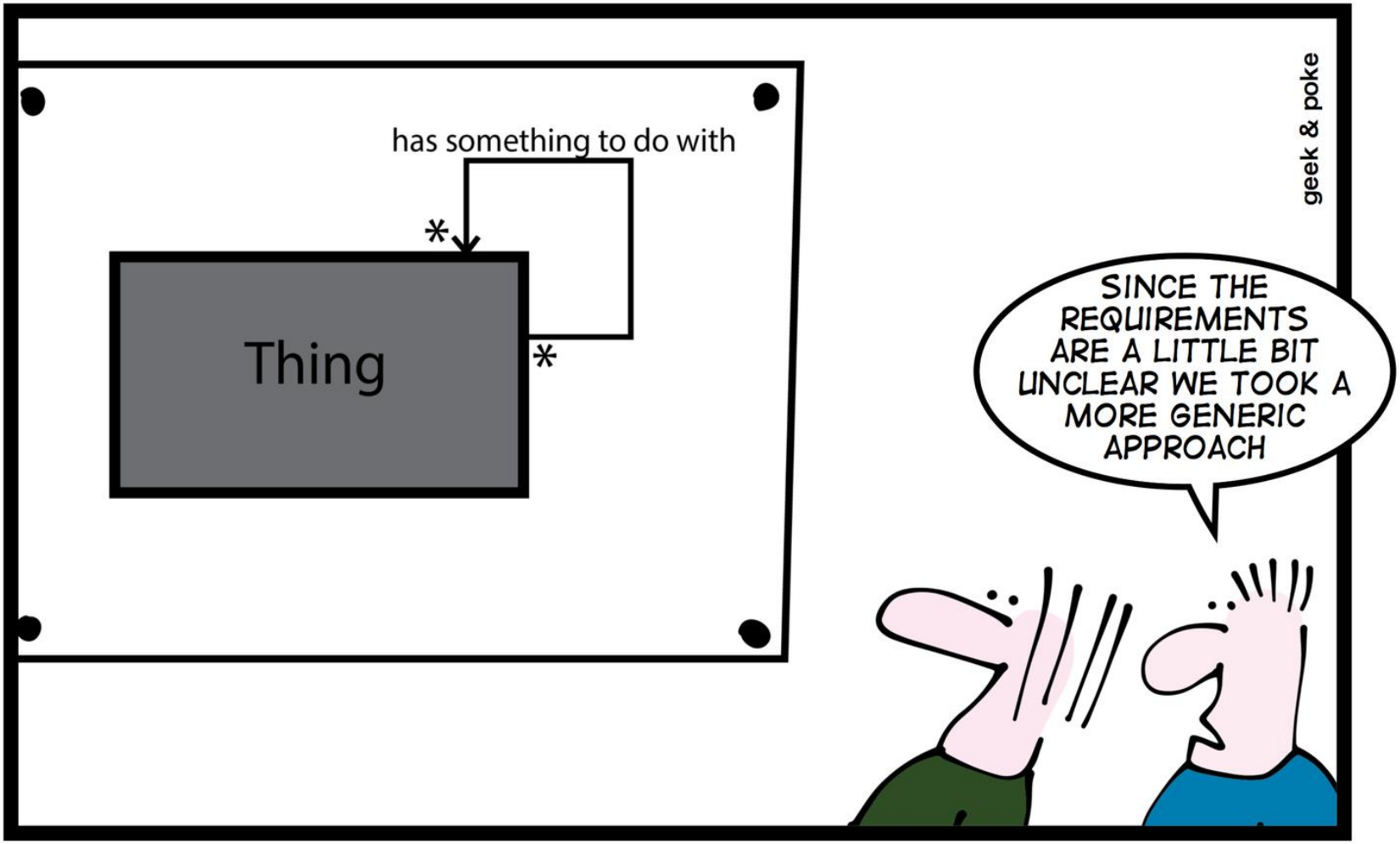


  - Insgesamt ergibt sich also:



Quelle: http://www.tinohempel.de/info/info/datenbank/kardinalitaet_mc.htm

How to create a stable data model



Hausaufgaben

bis zur nächsten Vorlesung

@

Machen Sie sich mit der modifizierten Chen-Notation vertraut

Bsp.:
http://www.tinohempel.de/info/info/datenbank/kardinalitaet_mc.htm

Oder im Dokument „Datenbanken I - HA Kardinalität - modifizierte Chen-Notation.pdf“

Hausaufgaben

bis zur nächsten Vorlesung

@

Schauen Sie sich verschiedene Modellierungstools für ER-Diagramme und UML an und denken Sie über Vorteile, Nachteile und Bedienbarkeit nach.

Vorschläge für kostenlose Tools sind hier z. B.:

Cacoo: https://cacoo.com

http://draw.io

Dia:

https://wiki.gnome.org/Apps/Dia #

Microsoft Visio (via MSDNAA)

Visual Paradigm for UML 11.0 Community Edition

(für Interessierte: DB-MAIN: http://www.db-main.eu)

Machen Sie sich dem Tool vertraut, mit dem Sie „am besten“ ER-Diagramme erstellen können / am schnellsten Ergebnisse erzielen.

(freiwillig, aber ratsam!) Erweitern Sie das Zugauskunftssystem um eine Personalplanung mit Lokführern, Fahrkartenkontrolleuren, Kellnern und deren Arbeits- und Urlaubszeiten.

Falls Kommentierung zur Lösung gewünscht wird, diese bitte an Mail-Adresse des Dozenten senden.