
Einführung in Datenbanksysteme

Prof. Dr. Ralf Möller

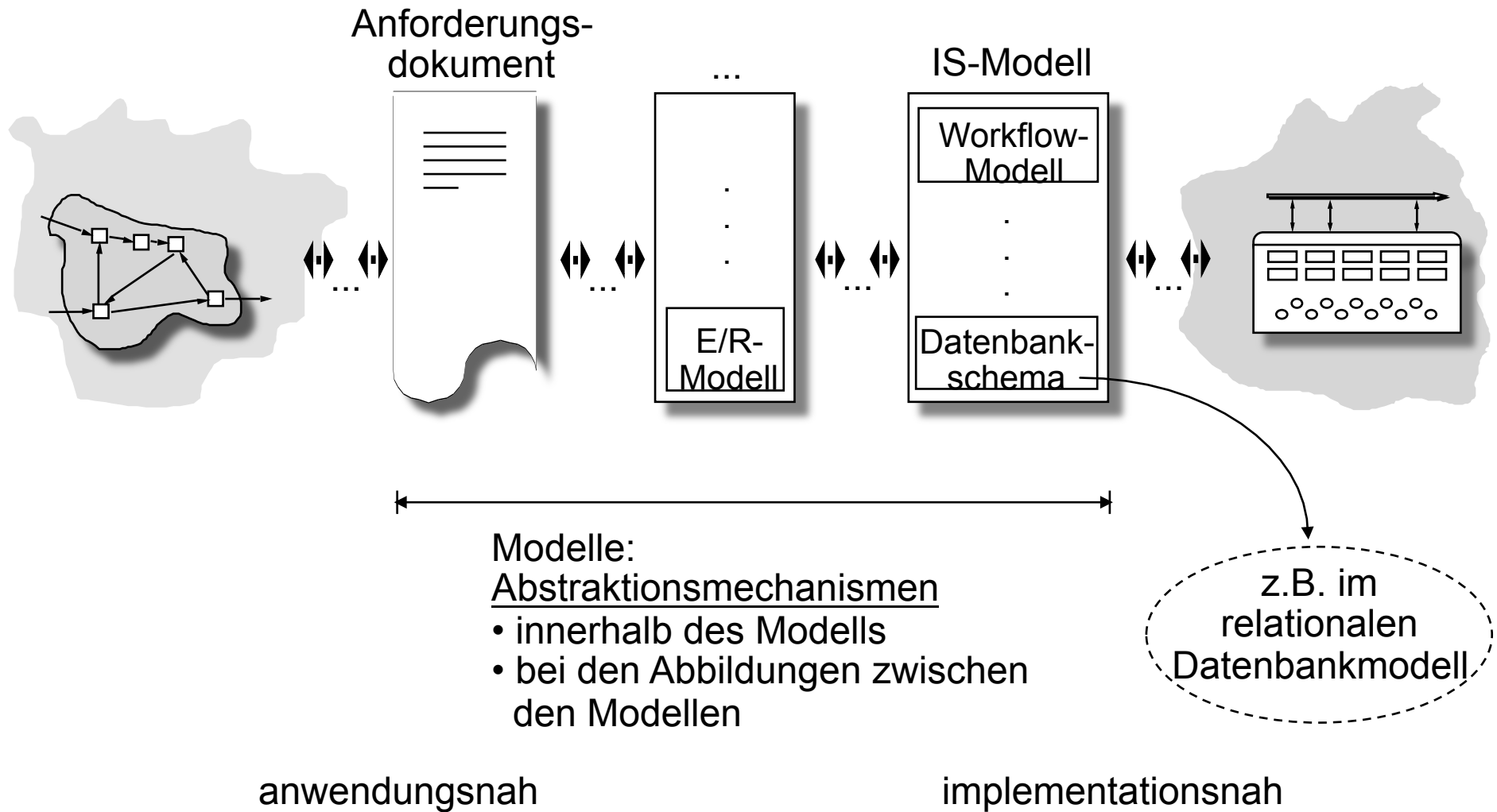
Technische Universität Hamburg-Harburg
Institut für Softwaresysteme (STS)

Mon., 09:45 - 11:15, TUHH ES40 N0007

Übung

Karsten Martiny
Dienstags 13:15-14:00, ES42 Raum 0526

Modelle und Abstraktion



Kapitel 1: Einführung

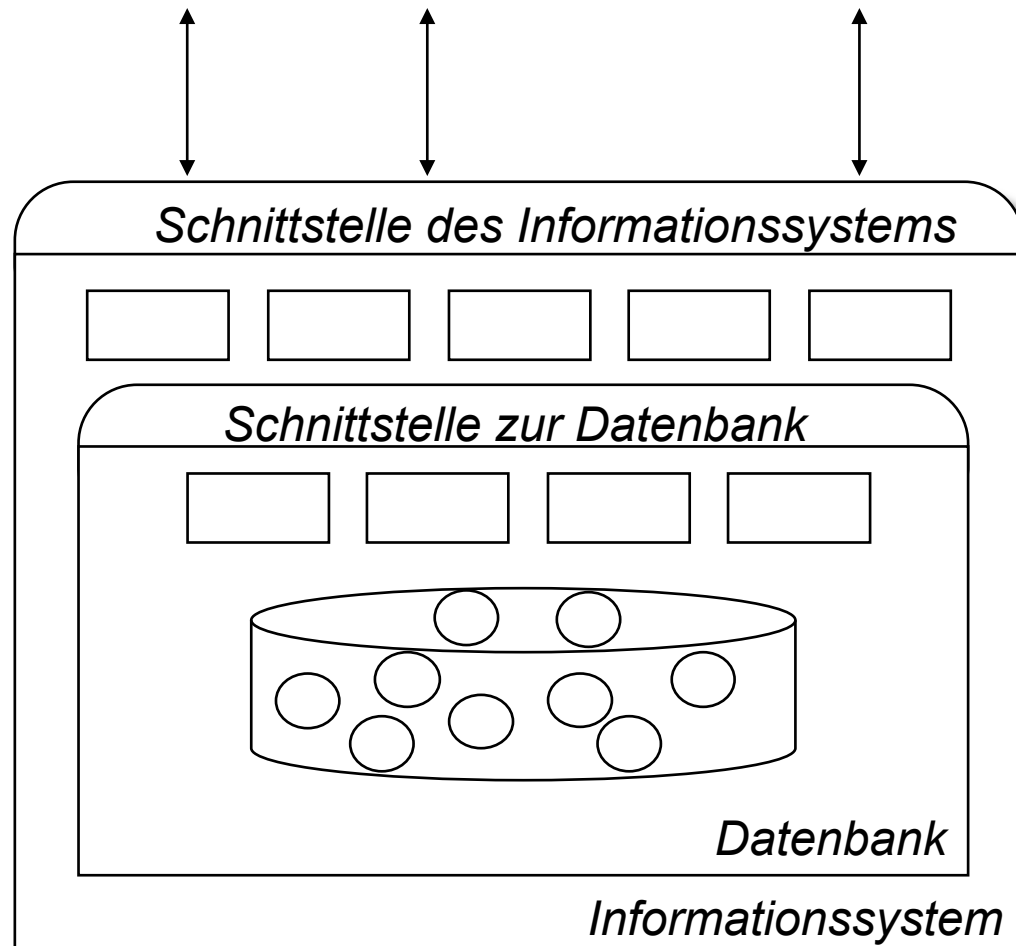
Kennzeichen von Daten in Datenbanken

- lange Lebensdauer (Jahre, Jahrzehnte)
- reguläre Strukturen
- große Datenobjekte, große Datenmengen
- stetig anwachsende integrierte Bestände (Giga-, Terabyte an Informationen)

Decimal		
Value		SI
1000	k	kilo
1000 ²	M	mega
1000 ³	G	giga
1000 ⁴	T	tera
1000 ⁵	P	peta
1000 ⁶	E	exa
1000 ⁷	Z	zetta
1000 ⁸	Y	yotta

Datenbanksysteme

Realisierung eines Informationssystems mit einer Datenbank:

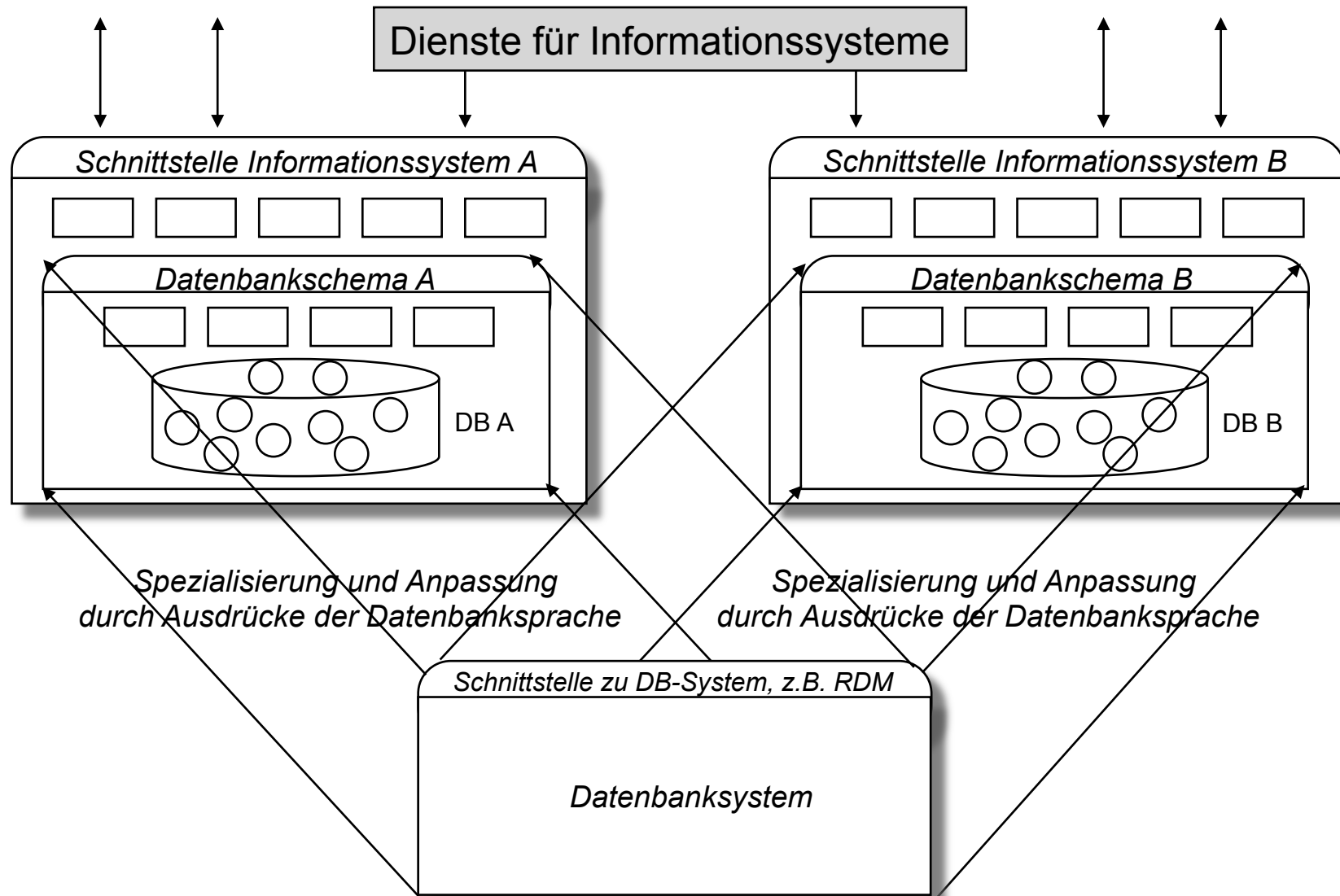


Algorithmen zur Informationsdarstellung, -verarbeitung und zur Integritätssicherung

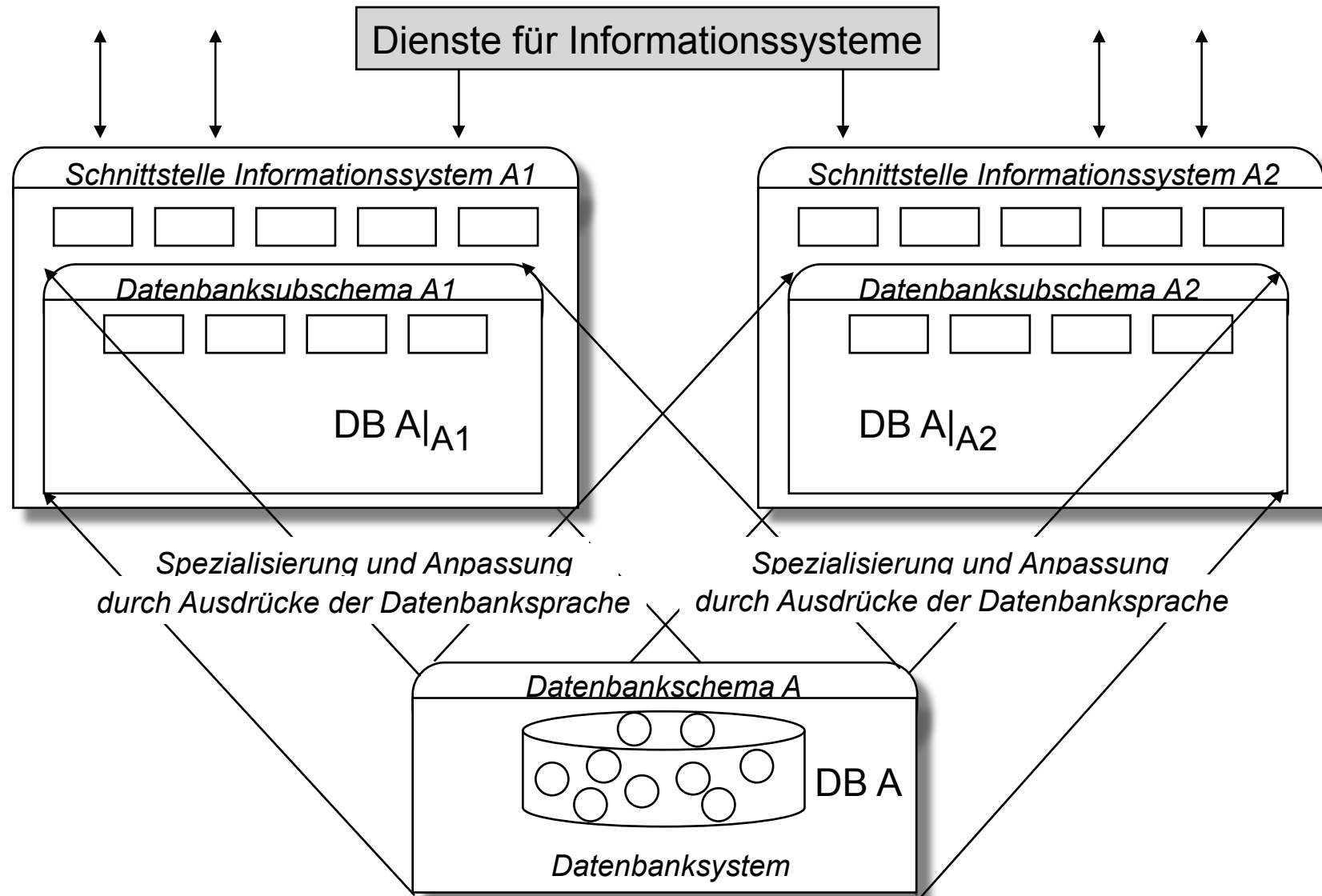
Dienste des Datenbanksystems zur Datenspeicherung, -anfrage und Integritätssicherung (Datenbankschema)

Datenbankzustand

Generisches Datenbankmodell und -system



Sichten: DB-Subschemata und Subdatenbanken



Datenabstraktionskonzepte z. Informationsstrukturierung

In fast allen Datenbankmodellen findet man Konstrukte für die folgenden Abstraktionskonzepte:

- Klassifikation und Instantiierung
- Generalisierung und Spezialisierung
- Assoziation
- Aggregation
- Identifikation und Schlüssel

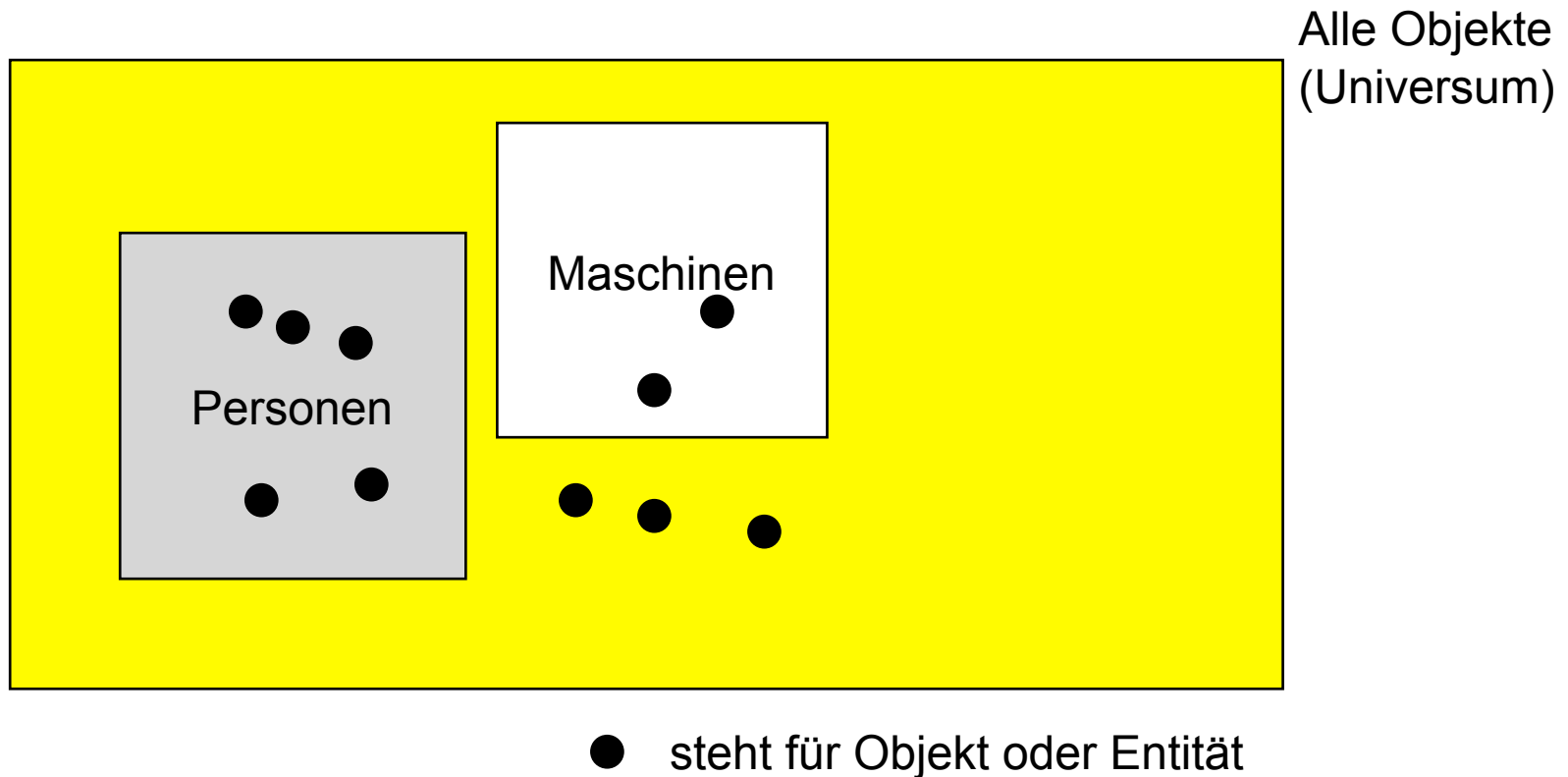
In späteren Kapiteln werden diese Konstrukte beschrieben. Nachfolgend werden die Abstraktionskonzepte anhand einer populären graphischen Notation erklärt.

Entity-Relationship-Diagramme wurden von P.P.S. Chen vorgeschlagen (vgl. P.P.S. Chen. "The Entity Relationship Model - Toward a Unified View of Data. In: ACM Transactions on Database Systems, Vol. 1, No. 1, März 1976, S. 9 ff.) und mehrfach erweitert (⇒ extended E/R diagram, EE/R Modell).

NB: ER-Notation nicht einheitlich.

Klassifikation und Instantiierung (1)

- ❑ Objekte mit ähnlichen Eigenschaften können zu Klassen (Mengen) zusammengefaßt werden.
- ❑ Jedes Objekt ist "Instanz" einer oder mehrerer Klassen.
- ❑ Extension (Menge aller Instanzen einer Klasse)



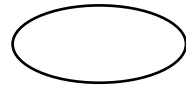
Grundlegende Elemente von ER-Diagrammen

❑ Objekttyp



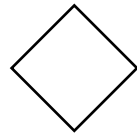
(auch Entitätstyp oder Klasse genannt,
Menge von Objekten)

❑ Werttyp



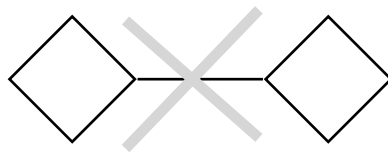
(für Basisdatentypen, Menge von „Werten“ bzw. Literale)

❑ Beziehungstyp

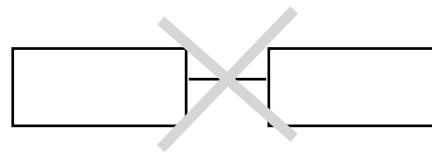


(Menge von Tupeln von Objekten)

❑ Die Elemente von ER-Diagrammen bilden einen bipartiten Graphen:



und

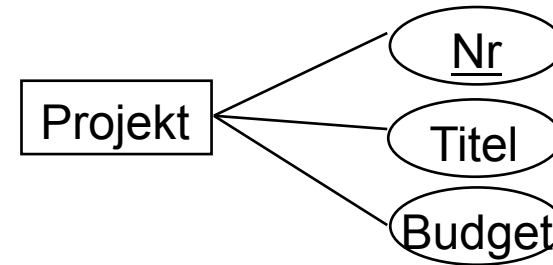


❑ Verbindungen zwischen Symbolen der gleichen Typen sind nicht erlaubt.

Objekte/Entitäten und Attribute

Beispiele:

- ❑ Ein Projekt wird beschrieben durch
 - eine Nummer
 - einen Titel
 - das Budget

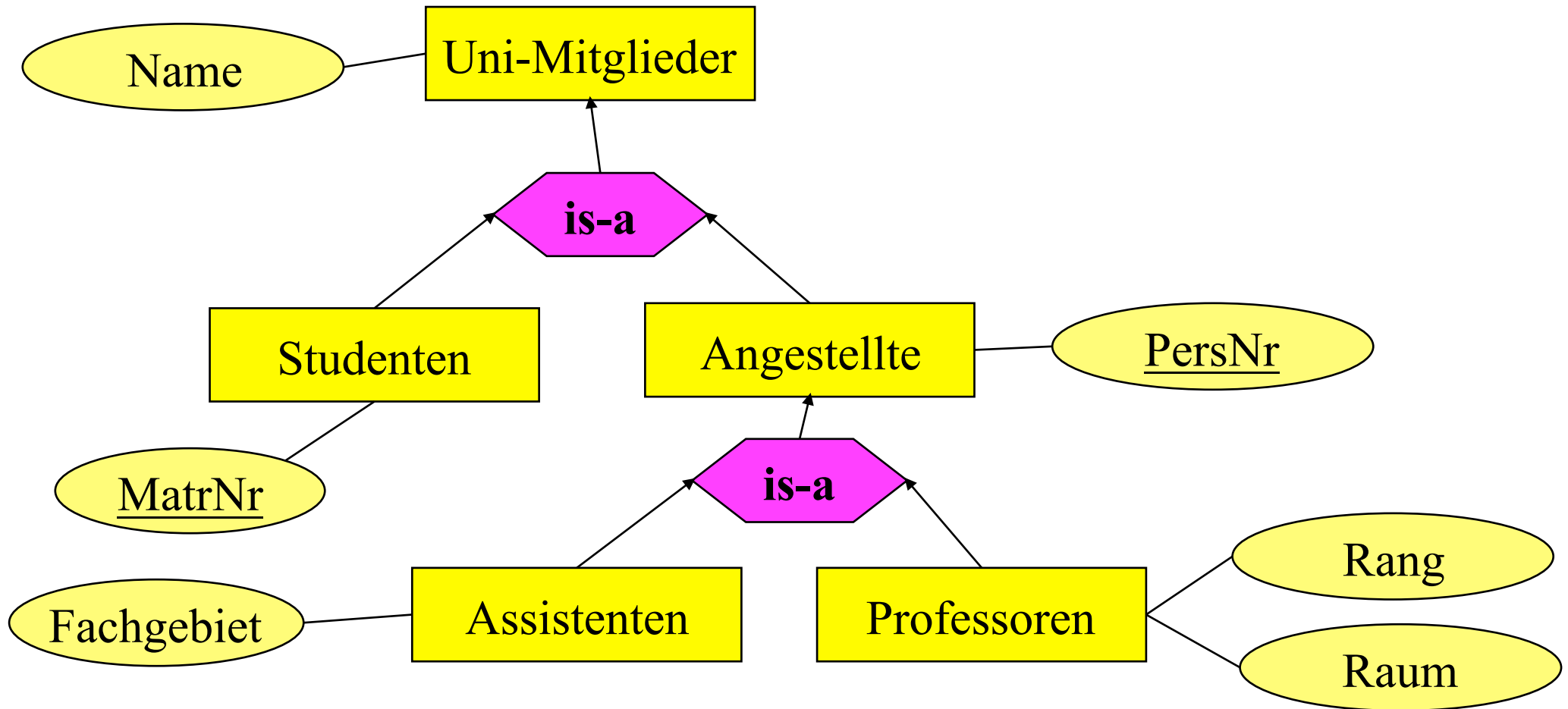


- ❑ Mathematische Bedeutung von Projekt: Menge von Tupeln von Werten
- ❑ Ein Tupel kann als Aggregat von Basiswerten aufgefaßt werden.

Generalisierung und Spezialisierung (1)

- ❑ Spezialisierung bezeichnet die Verfeinerung einer Klasse (mehr Information/ Anforderungen bzgl. des jeweiligen Individuen).
- ❑ Generalisierung ist die Vergrößerung einer Klasse (weniger Information/ Anforderungen bzgl. der jeweiligen Individuen).
- ❑ Spezielle Klassen (Subklasse) und allgemeine Klassen (Superklasse) bilden eine Subklassenhierarchie (⇒ Subtypisierung, Typhierarchie).
- ❑ Instanzen einer Klasse sind auch Instanzen der Superklasse.
- ❑ Spezialisierungen können disjunkt oder überlappend sein.
- ❑ Subklassen erben die Eigenschaften der Superklasse und fügen evtl. neue hinzu (⇒ Vererbung von Beschreibungen für Tupelkomponenten).

Generalisierung und Spezialisierung (2)

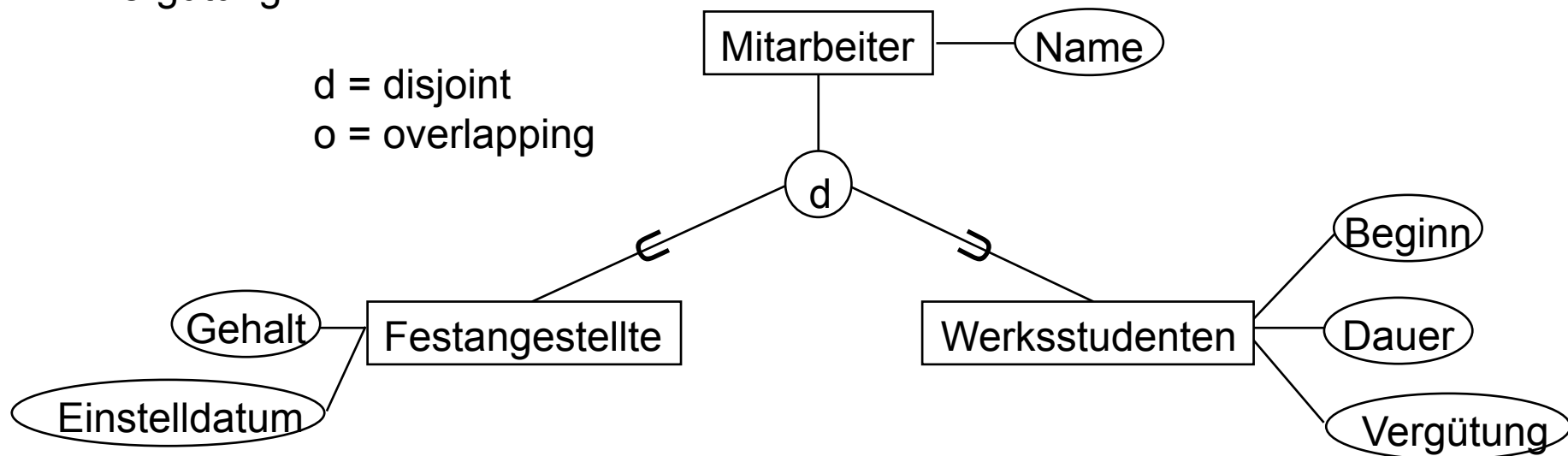


Generalisierung und **Spezialisierung** (3)

Erweitere Entity-Relationship-Diagramme

Beispiel:

- ❑ Festangestellte und Werksstudenten sind Mitarbeiter. Festangestellte sind keine Werkstudenten
- ❑ Festangestellte haben die zusätzliche Eigenschaften Gehalt und Einstelldatum.
- ❑ Werksstudenten haben die zusätzliche Eigenschaften Beginn, Dauer und Vergütung.

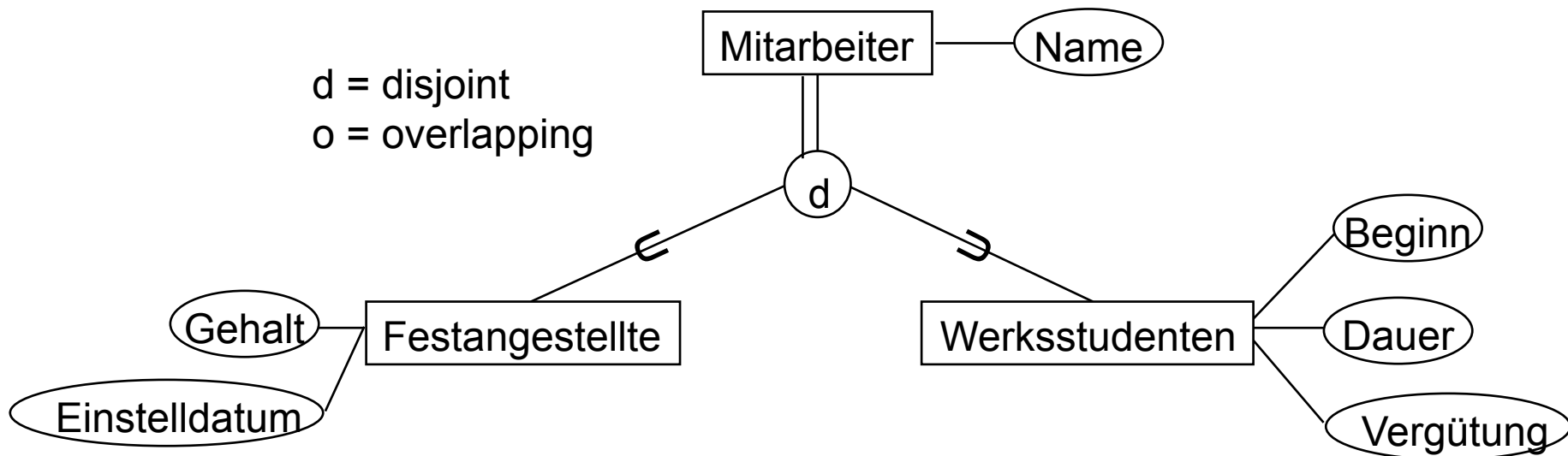


Generalisierung und Spezialisierung (4)

Erweitere Entity-Relationship-Diagramme

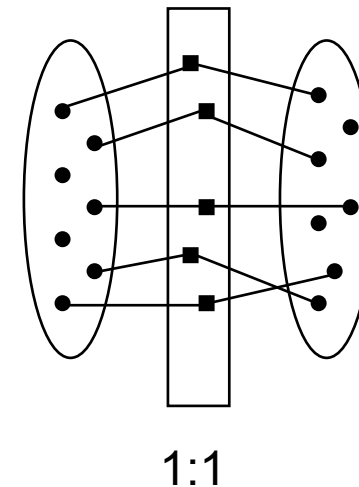
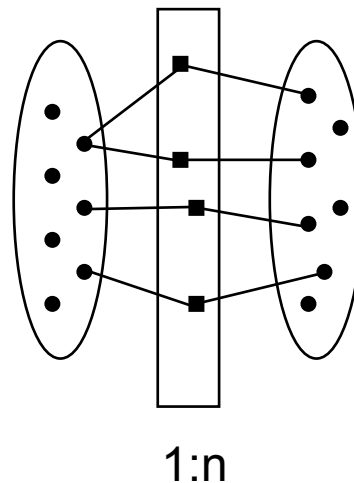
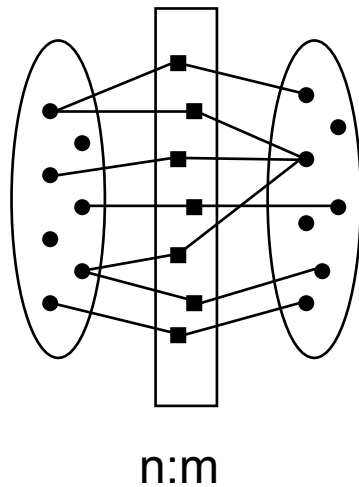
Beispiel:

- ❑ Festangestellte und Werksstudenten sind Mitarbeiter. Festangestellte sind keine Werkstudenten
- ❑ Die Menge der Mitarbeiter ist gleich der Vereinigung der Mengen Festangestellte und Werkstudenten, d.h. ein Mitarbeiter ist entweder festangestellt oder Werkstudent



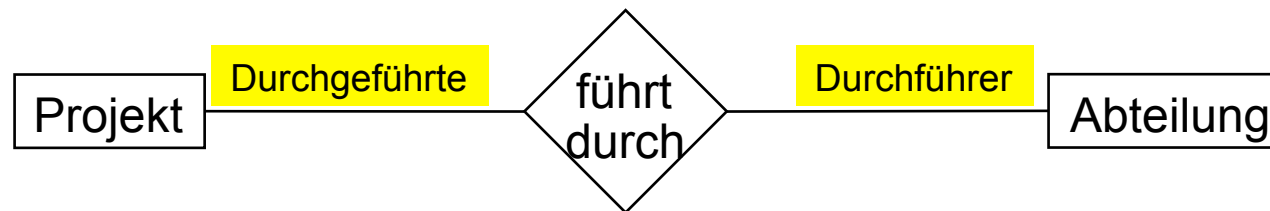
Assoziation (1)

- ❑ Objekte können miteinander in Beziehung gesetzt (assoziiert) werden:
 - Binäre (ternäre, ...) Beziehungen assoziieren zwei (drei, ...) Klassen oder Objekte.
 - Allgemein: n-äre Beziehungen zwischen n Klassen oder Objekten, wobei n der Grad der Beziehung ist.
 - Beziehungen können dynamisch geknüpft, ausgewertet, gelöst werden.
- ❑ Kardinalitätsbeschränkungen legen die genaue Zahl oder ein Intervall für die Anzahl der in Beziehung stehenden Instanzen fest.



Assoziation (2)

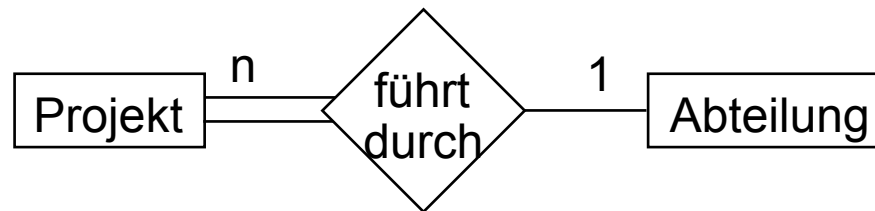
- ❑ **Totale Partizipation:** Jede Instanz einer Klasse muß mit einer Instanz der zweiten Klasse in Beziehung stehen (====).
- ❑ **Partielle Partizipation:** Eine Instanz einer Klasse kann in Beziehung zu einer Instanz der zweiten Klasse stehen (-----) s. Beispiel auf der nächsten Folie).
- ❑ Rollennamen identifizieren die Menge der Instanzen, die mit einer anderen Instanz in Beziehung stehen.
- ❑ Rollen können als abgeleitete Attribute verstanden werden, die die Menge der Instanzen als Attributwerte besitzen.



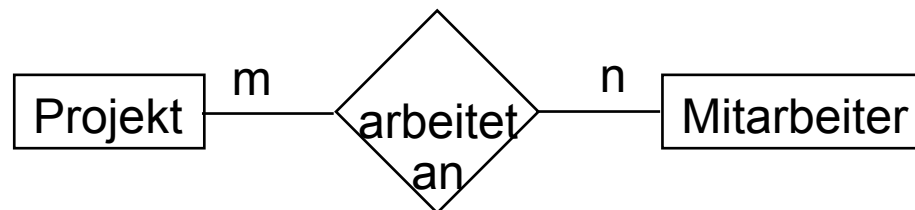
Assoziation (3)

Beispiele:

- ❑ Projekte werden von Abteilungen durchgeführt. Jedes Projekt muß einer Abteilung zugeordnet sein. Eine Abteilung kann mehrere Projekte ausführen.



- ❑ An Projekten arbeiten Mitarbeiter. Ein Mitarbeiter kann an mehreren Projekten arbeiten. Jedes Projekt wird von beliebig vielen Mitarbeitern bearbeitet.



- ❑ Bemerkung: In der Literatur findet man auch andere Beschriftungsregeln.

Beispiel 1:

Gegeben: "Anforderungsdefinition"

An einer Universität werden verschiedene Vorlesungen angeboten, die Teil mehrerer Studienfächer sind.

Diese Vorlesungen werden von genau einem Dozenten gehalten.

Jeder Dozent ist Mitglied genau eines Fachbereiches.

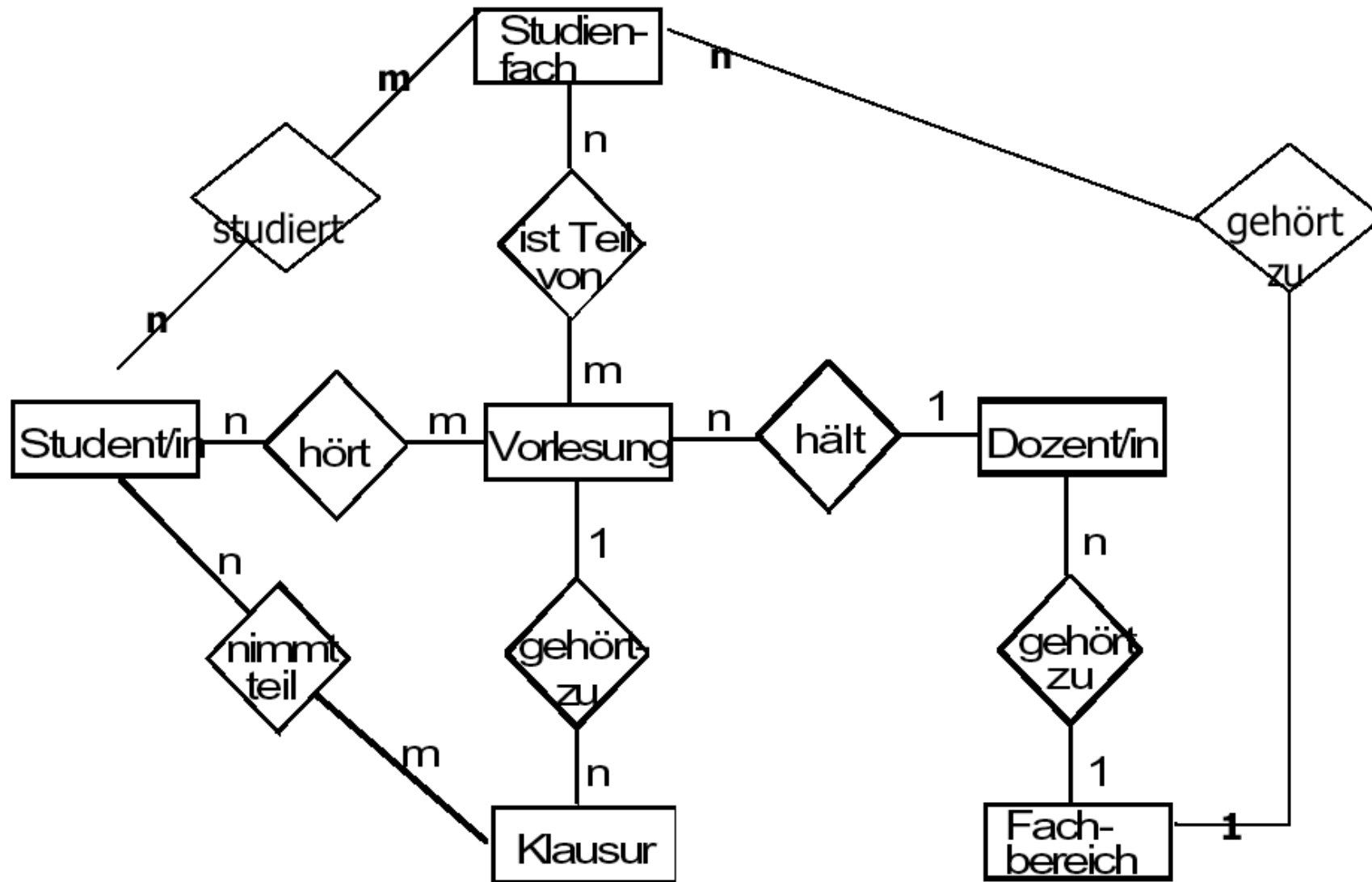
Ein Fachbereich hat mehrere Studienfächer.

Die Vorlesungen werden von Studenten gehört, die jeweils ein oder mehrere Studienfächer belegt haben.

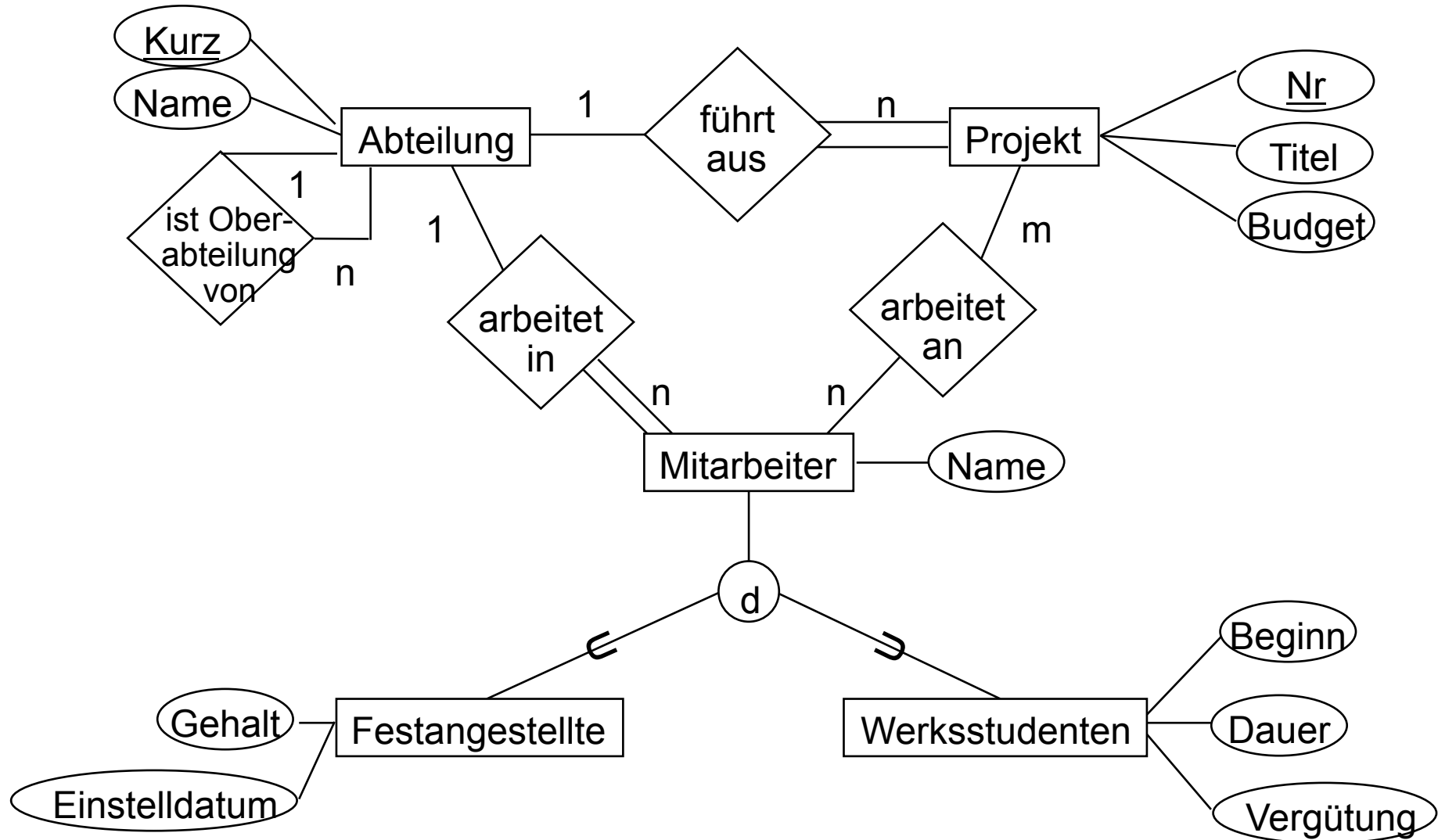
Zu jeder Vorlesung werden mehrere Klausuren angeboten, die von den Studenten geschrieben werden.

Gesucht: ER-Diagramm

Mögliche Lösung (ohne Totalitätsangaben)



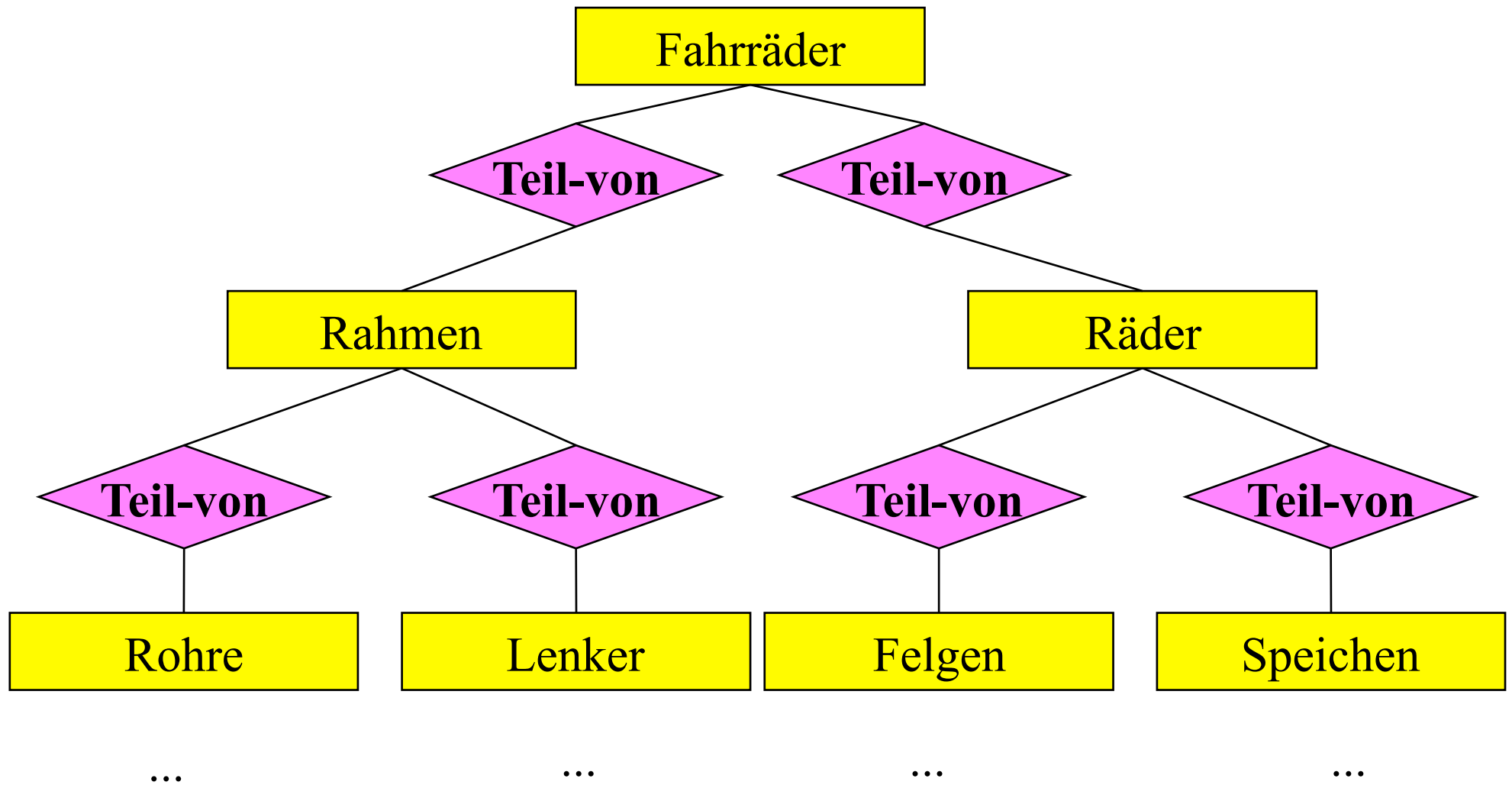
Beispiel 2: ER-Diagramm erläutern



Aggregation und Dekomposition (1)

- ❑ Objekte können zu übergeordneten Objekten aggregiert werden:
 - Beziehungen zwischen Komponenten und übergeordnetem Objekt
 - Übergeordnetes Objekt kann wiederum an Beziehungen teilnehmen.
- ❑ Im Vergleich zur "normalen" Assoziation wird die "Aggregation" in der Entity-Relationship-Modellierung nicht besonders unterstützt. Man verwende anwendungsspezifische Assoziationen

Aggregation



Identifikation und Schlüssel (1)

Zur **Identifikation** existieren zwei grundlegende Ansätze in Datenbankmodellen:

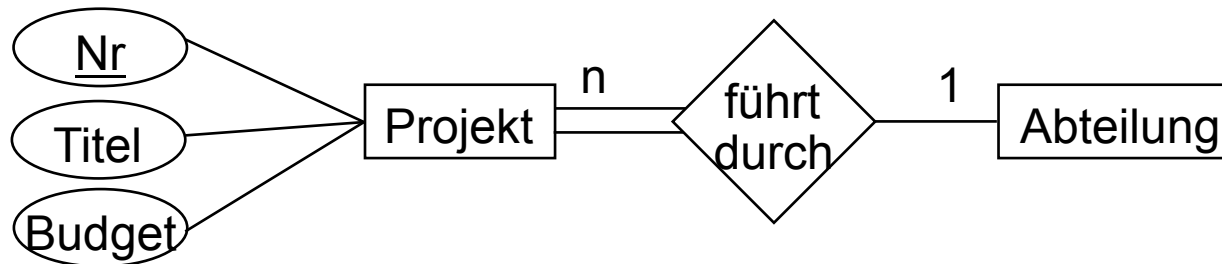
- ❑ **Referentielle Identifikation** bezeichnet direkte Verweise auf Objekte (»»» Zeiger in Programmiersprachen).
- ❑ **Assoziative Identifikation** verwendet die Werte von Attributen oder Attributkombinationen, um sich eindeutig auf Objekte zu beziehen (»»» Schlüssel: Ausweisnummer, Fahrgestellnummer, ...).
- ❑ In der Praxis benötigt man häufig beide Formen der Identifikation.

Schlüssel:

- ❑ Schlüssel sind Attribute oder Attributkombinationen mit innerhalb einer Klasse eindeutigen Werten und eignen sich deshalb zur Identifikation.
- ❑ Es kann mehrere Schlüsselkandidaten geben (Primärschlüssel, Sekundärschlüssel).
- ❑ Schlüssel stellen als Attributwerte Beziehungen zu anderen Objekten her (Fremdschlüssel).
- ❑ Durch Fremdschlüssel referenzierte Objekte müssen existieren (»»» referentielle Integrität).

Identifikation und Schlüssel (2)

Beispiel: Projekte können durch eine Nummer eindeutig identifiziert werden.



Dabei existieren zwei Möglichkeiten zur Identifikation von Projekten innerhalb der Assoziation "führt durch":

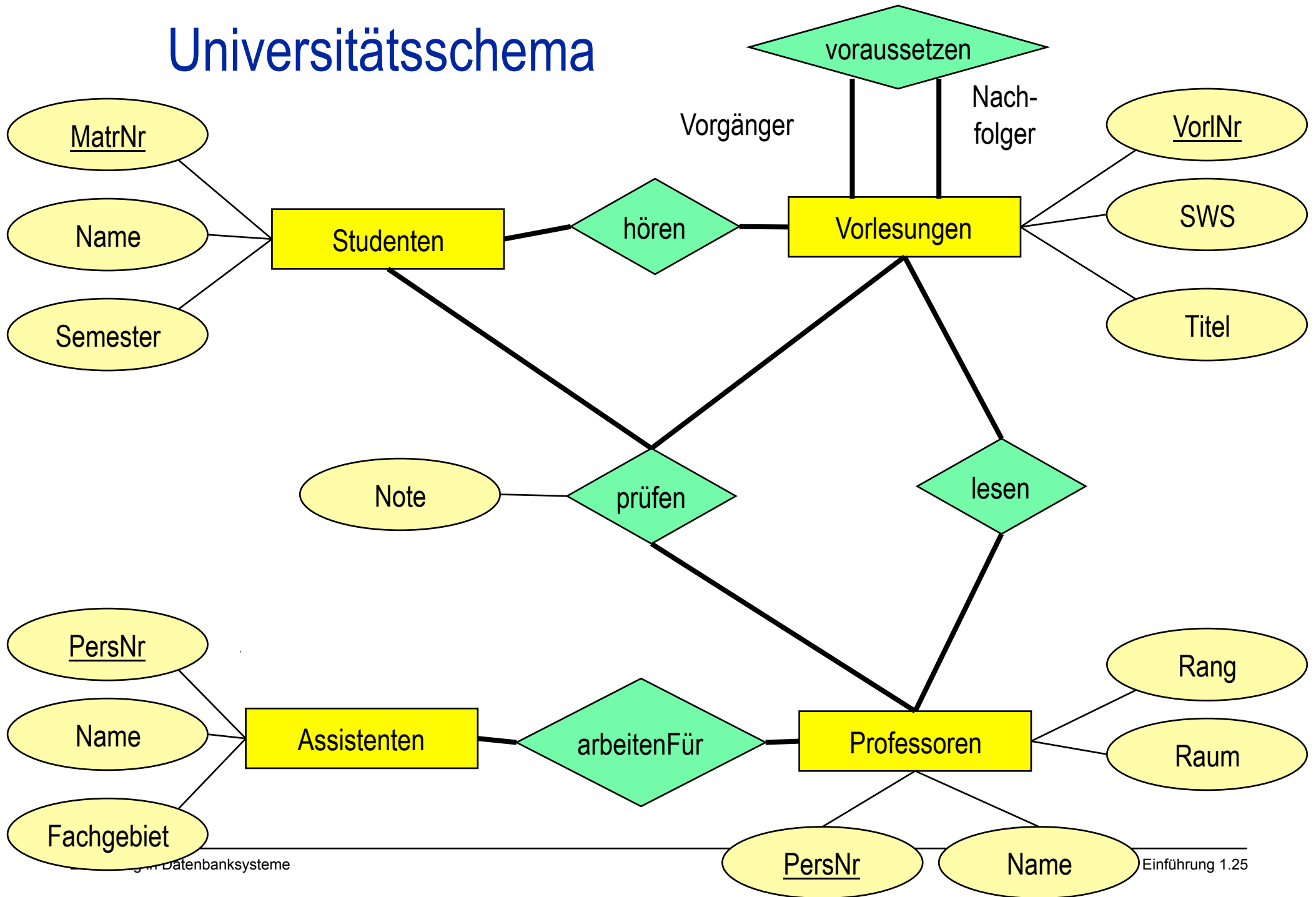
Referentielle Identifikation

	Projekt	Abteilung
←	•	...
←	•	...
←	•	...

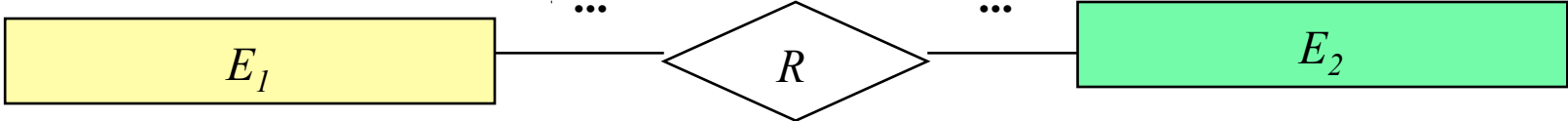
Assoziative Identifikation

Projekt	Abteilung
4711	...
4712	...
4713	...
...	...

Universitätschema

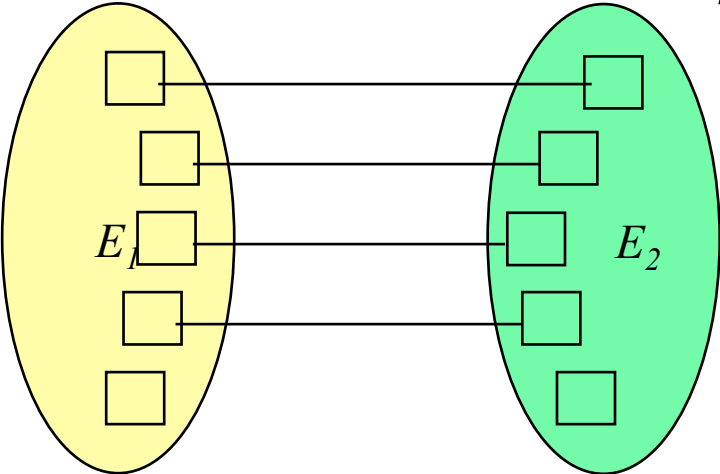


Funktionalitäten

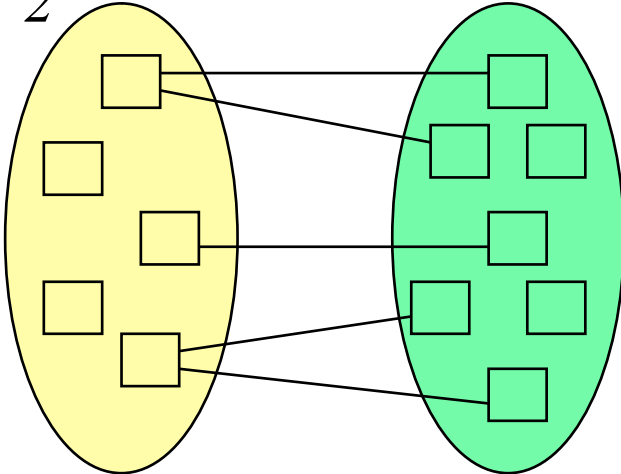


$$R \subseteq E_1 \times E_2$$

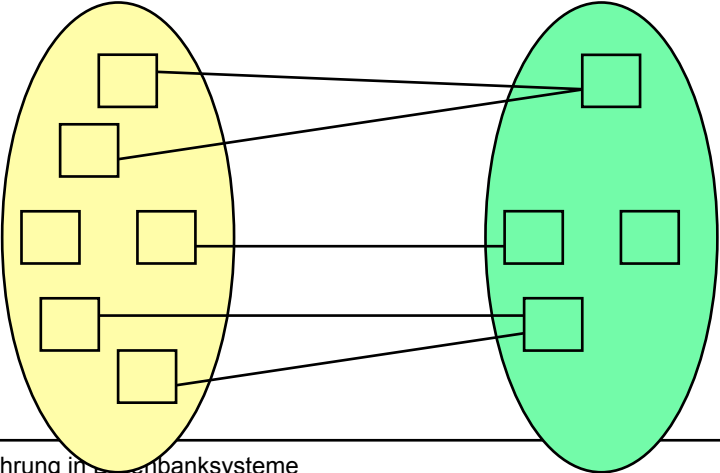
1:1



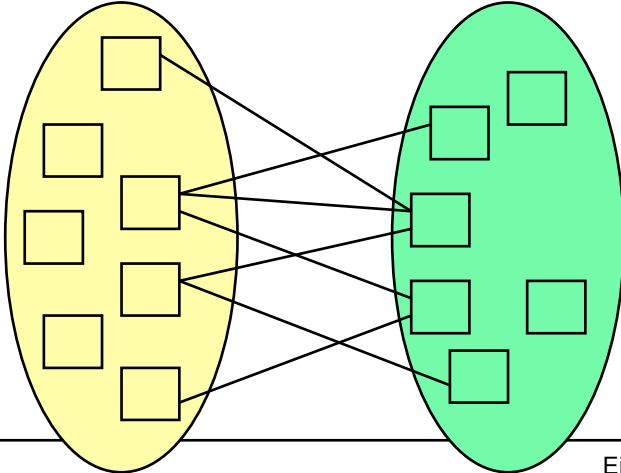
1:N



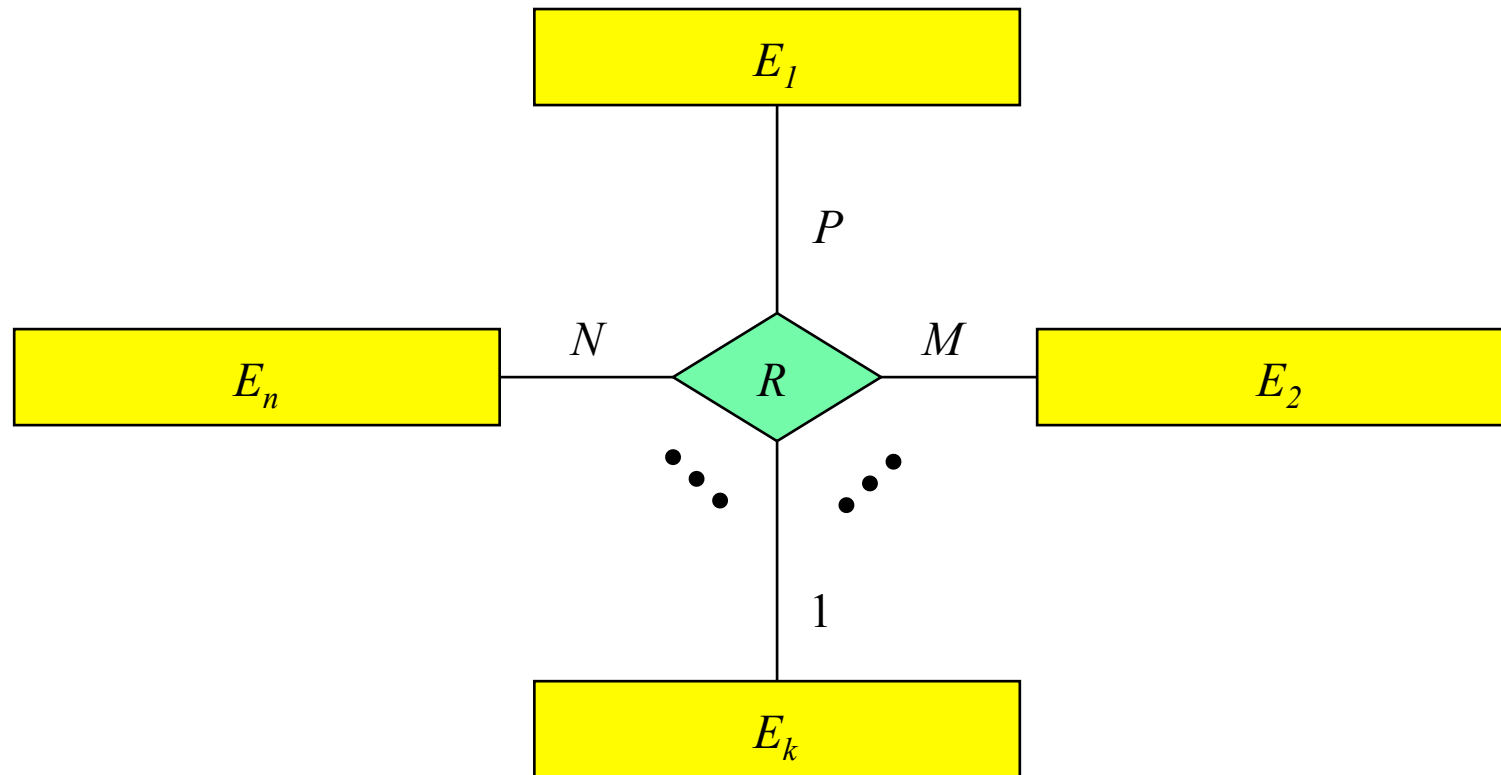
N:1



N:M

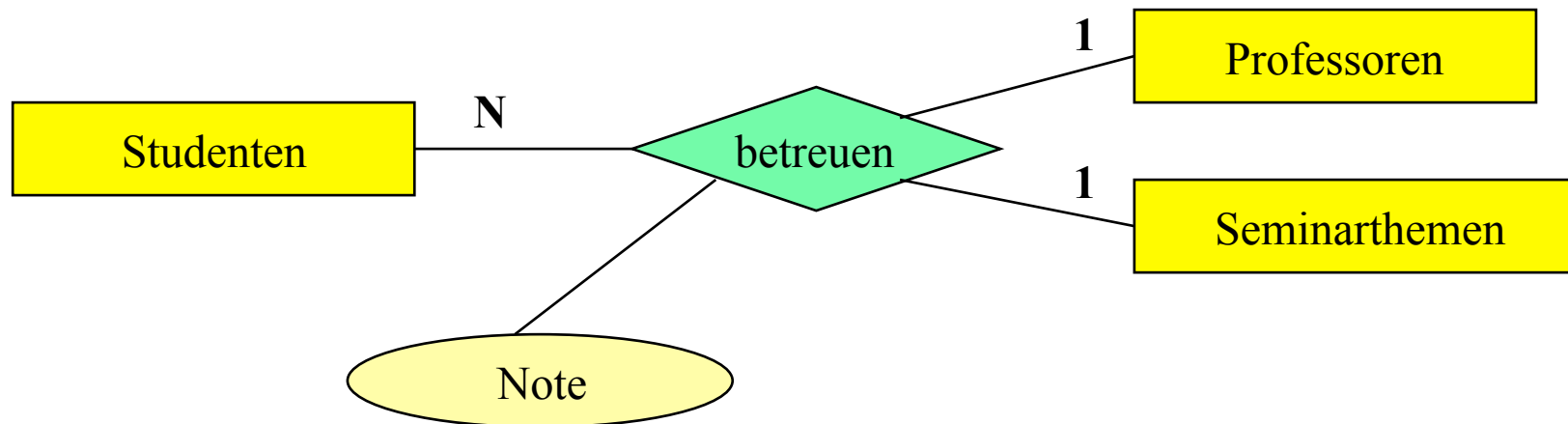


Funktionalitäten bei n -stelligen Beziehungen



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

Beispiel-Beziehung: *betreuen*



betreuen : Professoren x Studenten \rightarrow Seminarthemen

betreuen : Seminarthemen x Studenten \rightarrow Professoren

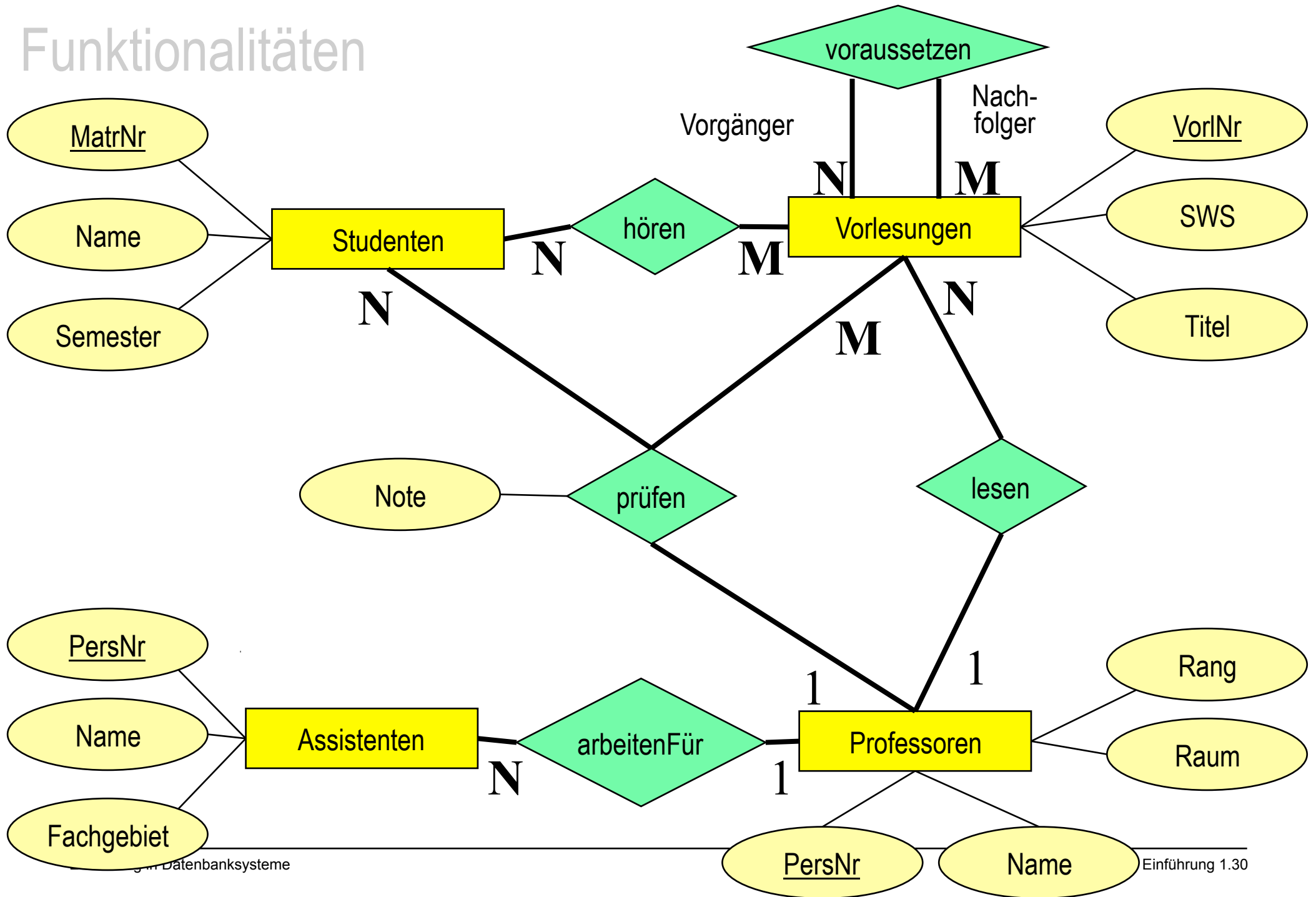
Dadurch erzwungene Konsistenzbedingungen

1. Studenten dürfen bei demselben Professor bzw. derselben Professorin nur ein Seminarthema "ableisten" (damit ein breites Spektrum abgedeckt wird).
2. Studenten dürfen dasselbe Seminarthema nur einmal bearbeiten – sie dürfen also nicht bei anderen Professoren ein schon einmal erteiltes Seminarthema nochmals bearbeiten.

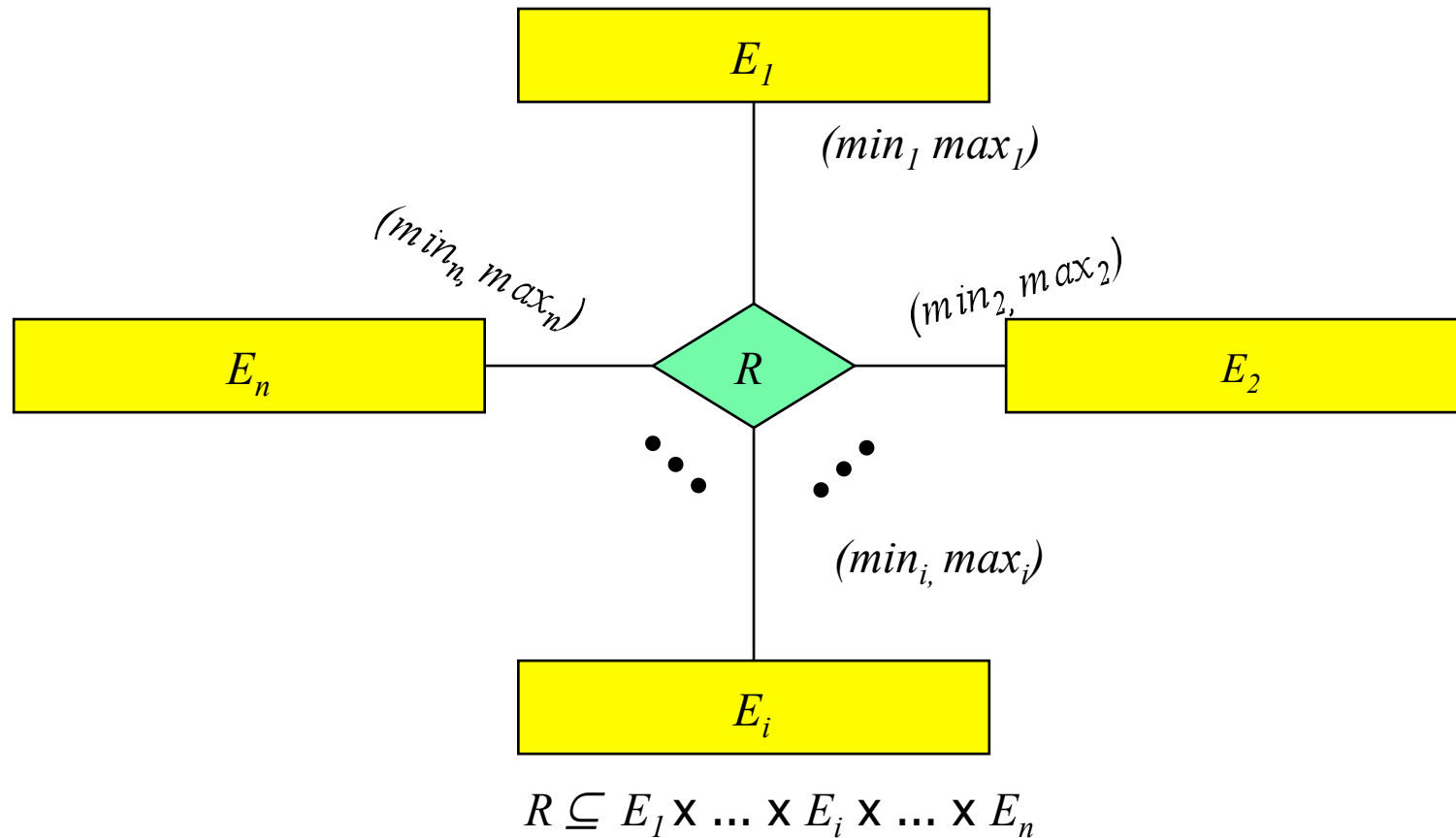
Folgende Datenbankzustände nach wie vor möglich:

- Professoren können dasselbe Seminarthema „wiederverwenden“ – also dasselbe Thema auch mehreren Studenten erteilen.
- Ein Thema kann von mehreren Professoren vergeben werden – aber an unterschiedliche Studenten.

Funktionalitäten



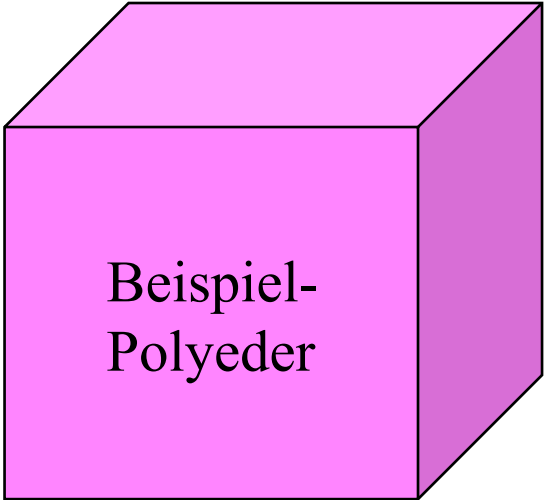
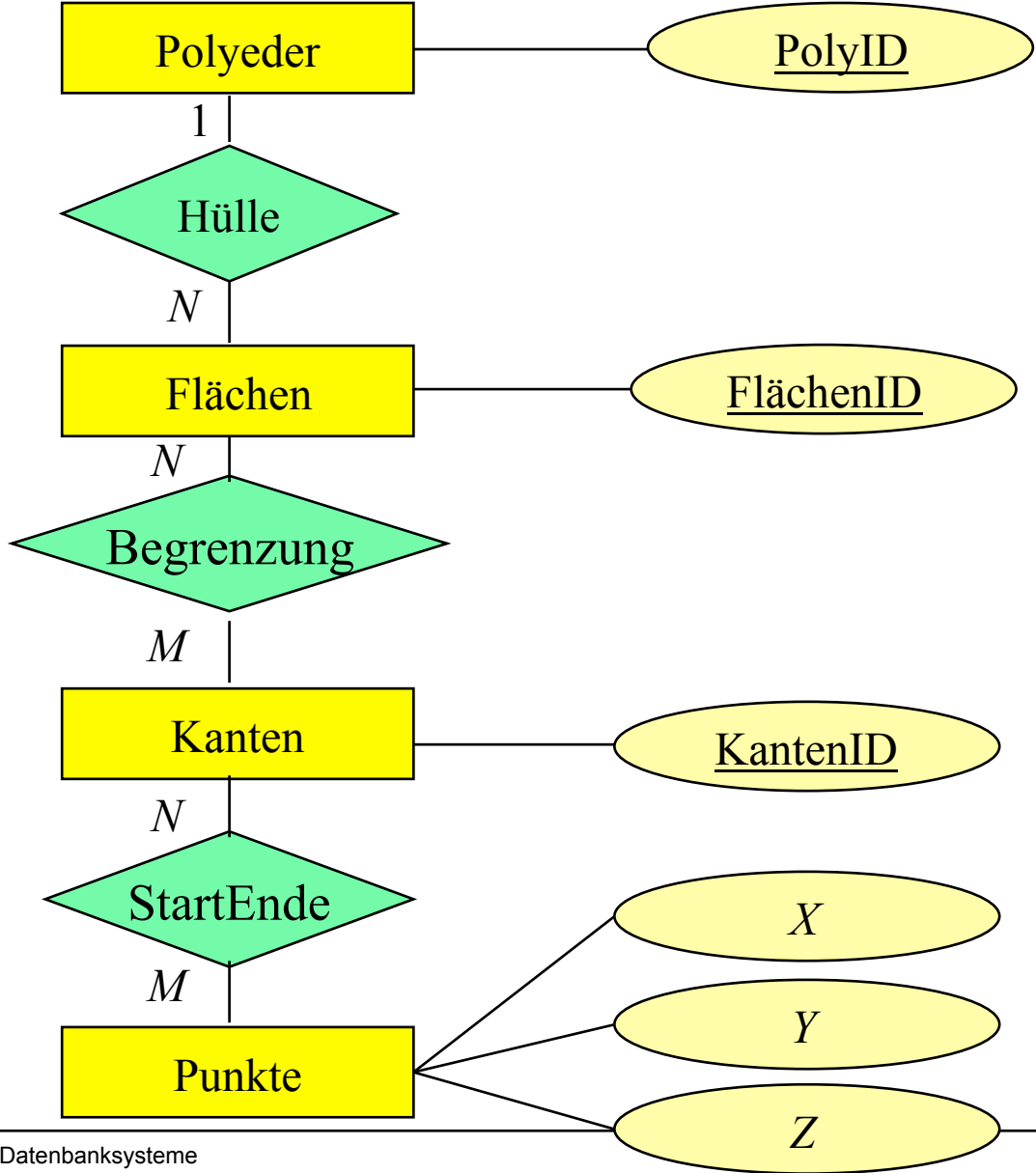
(Min, Max)-Notation



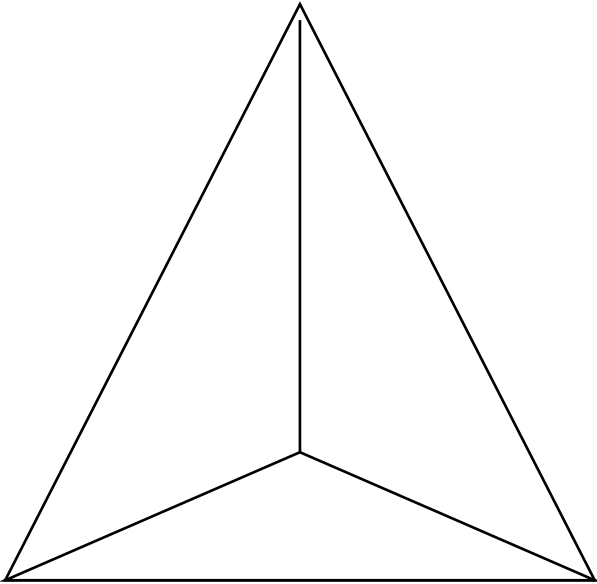
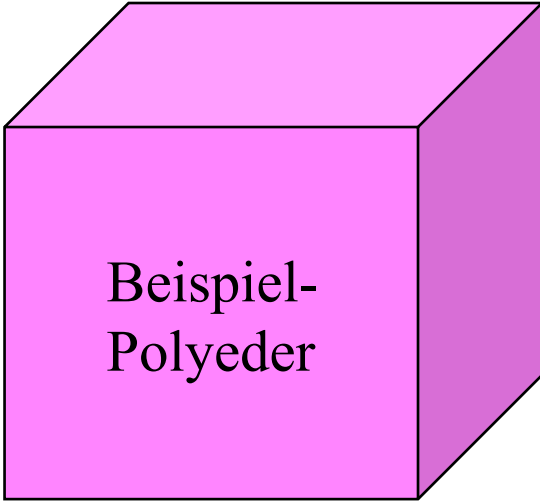
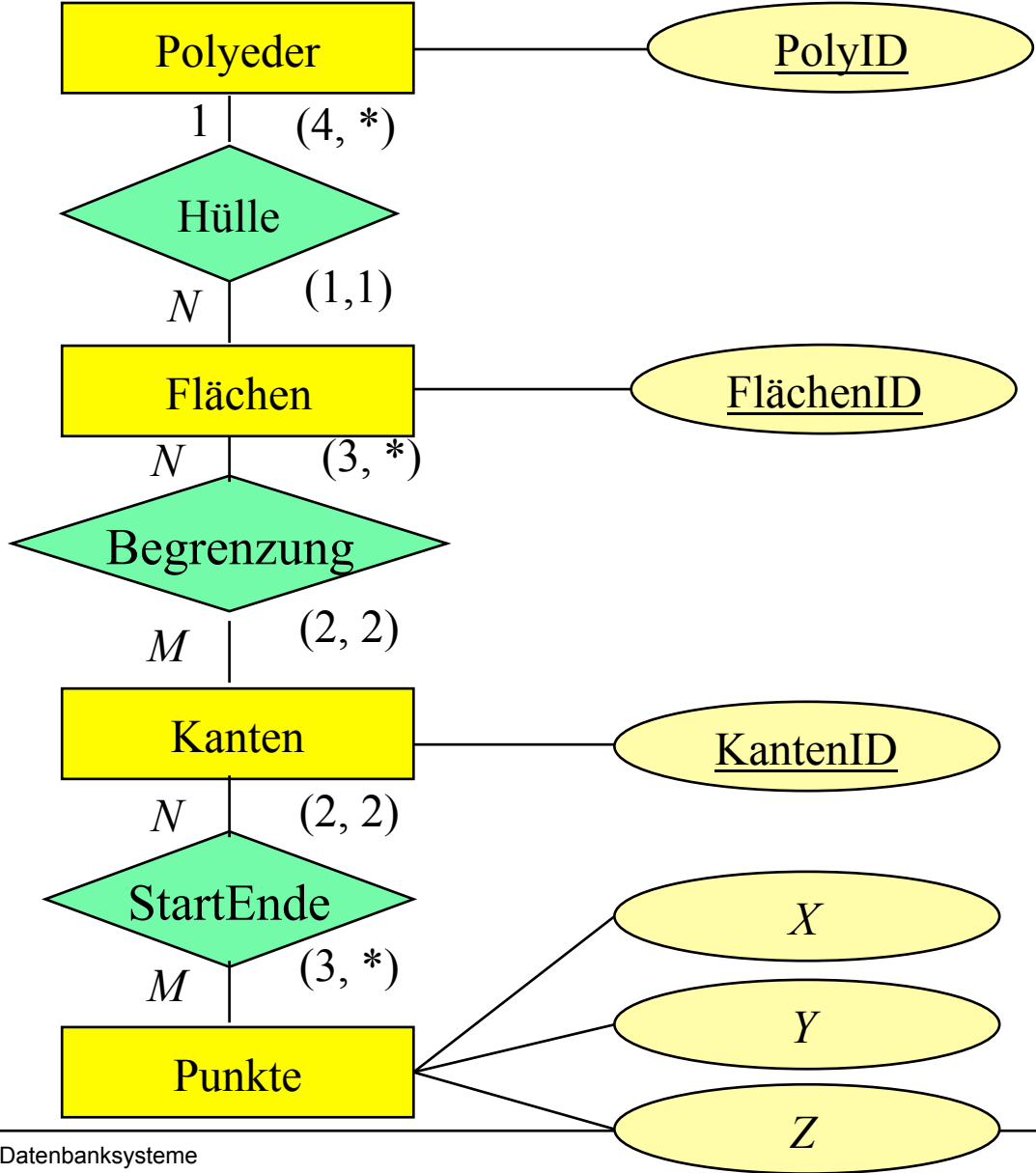
Für jedes $e_i \in E_i$ gibt es

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

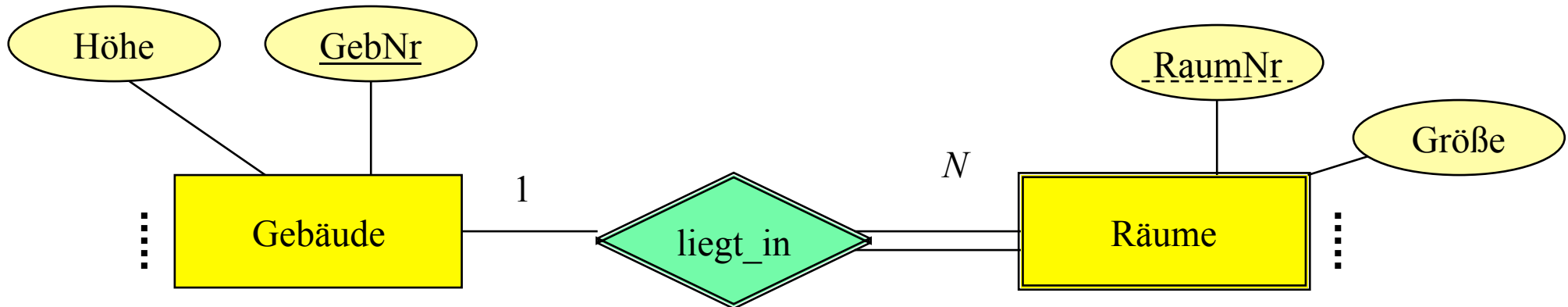
Komplex-strukturierte Entities



Komplex-strukturierte Entities



Schwache, existenzabhängige Entities



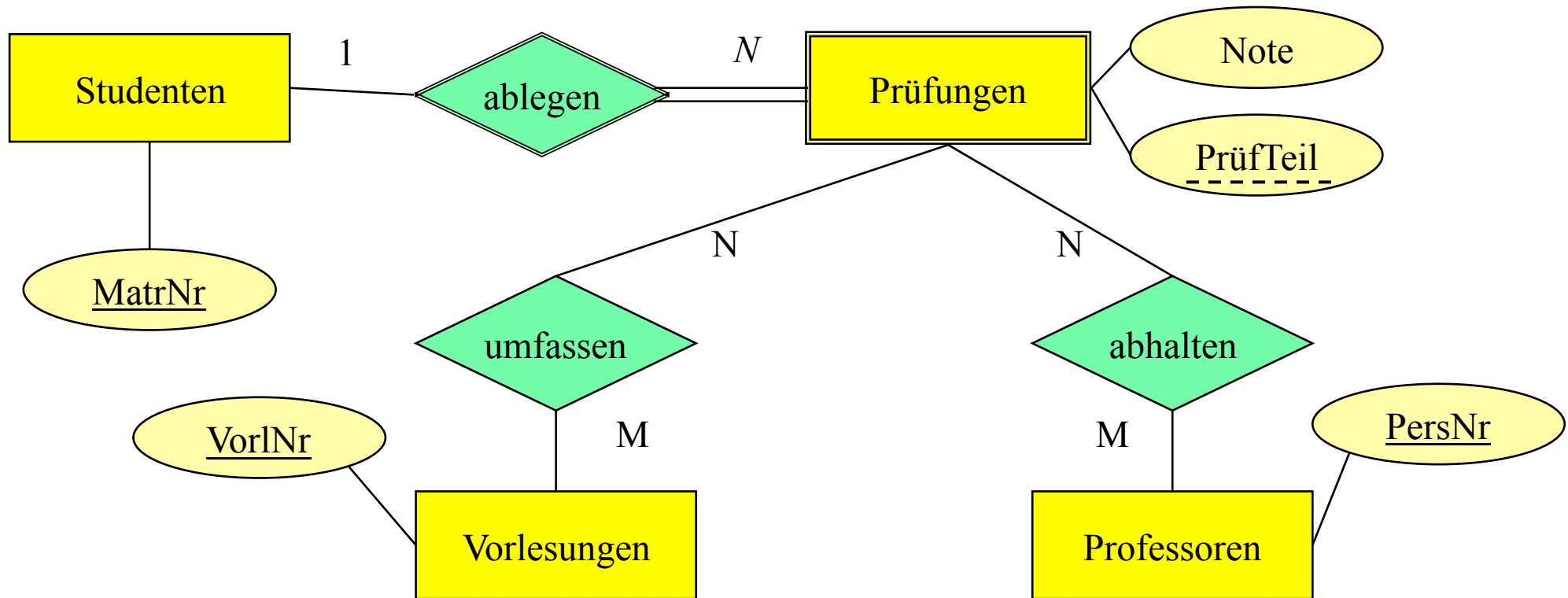
Beziehung zwischen "starken" und schwachem Typ ist immer 1:N (oder 1:1 in seltenen Fällen)

Warum kann das keine $N:M$ -Beziehung sein?

RaumNr ist nur innerhalb eines Gebäudes eindeutig

Schlüssel ist: GebNr **und** RaumNr

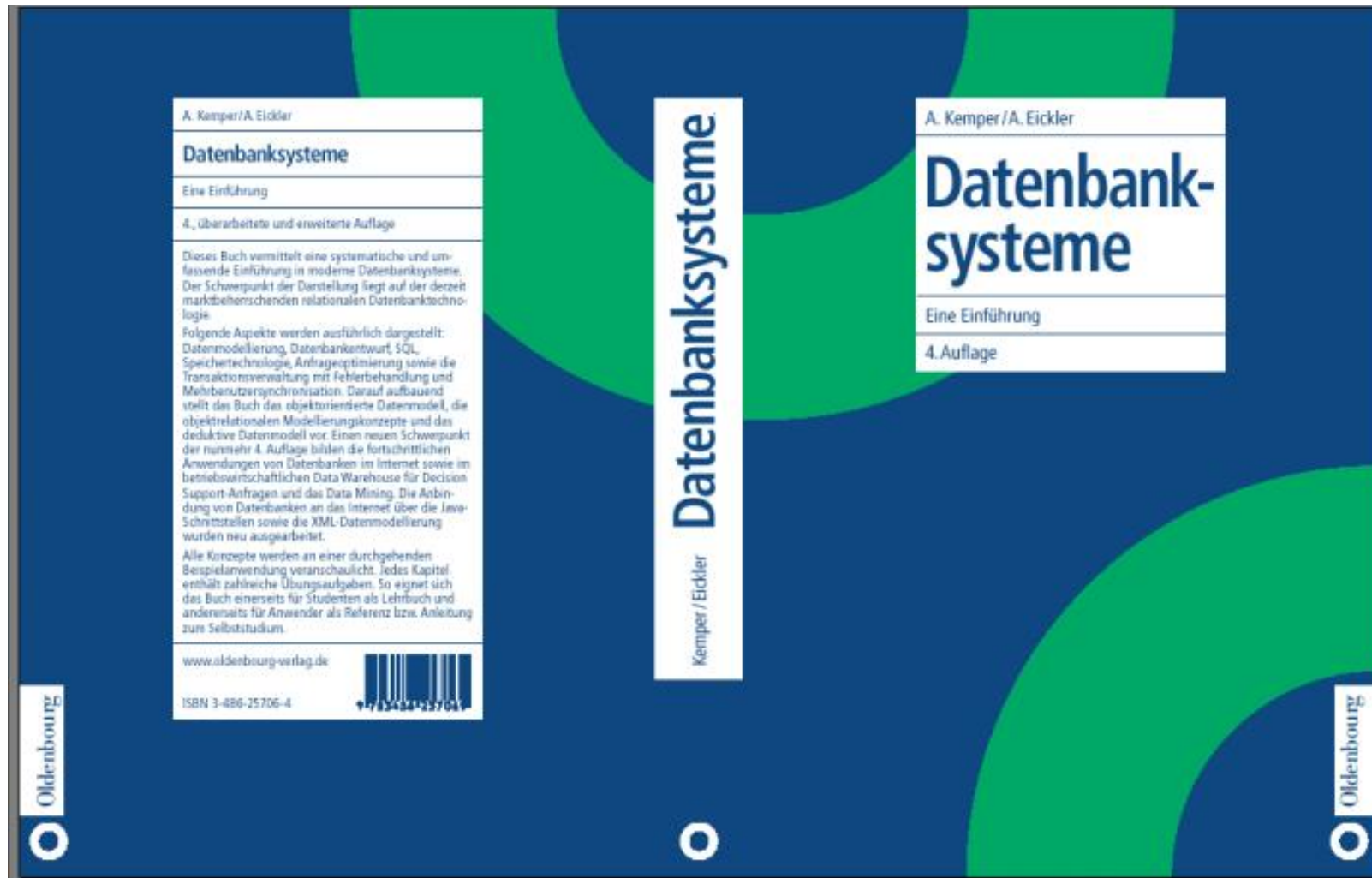
Prüfungen als schwacher Entitytyp



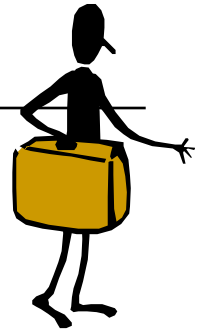
Mehrere Prüfer in einer Prüfung

Mehrere Vorlesungen werden in einer Prüfung abgefragt

Acknowledgments / Skript zur Vorlesung



Zusammenfassung, Kernpunkte



Grundlagen von Datenbanksystemen

- ❑ Grob-Architektur eines Datenbanksystems
- ❑ Entwurfsebene: Entity-Relationship-Modell