

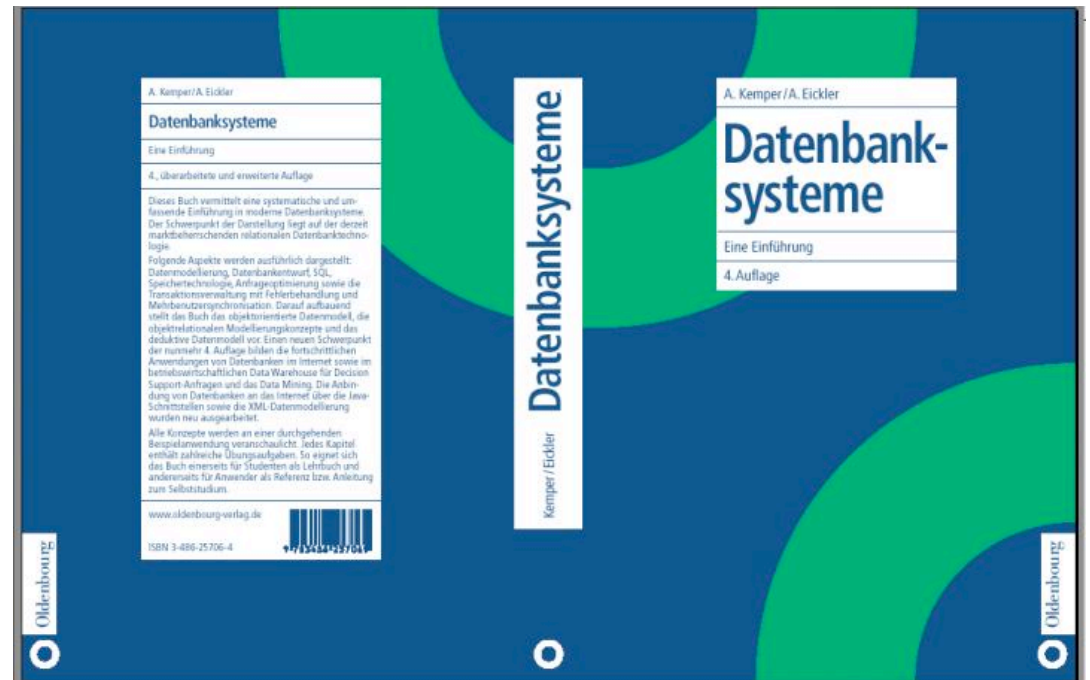
Einführung in Datenbanken

Ralf Möller, TU Hamburg Harburg

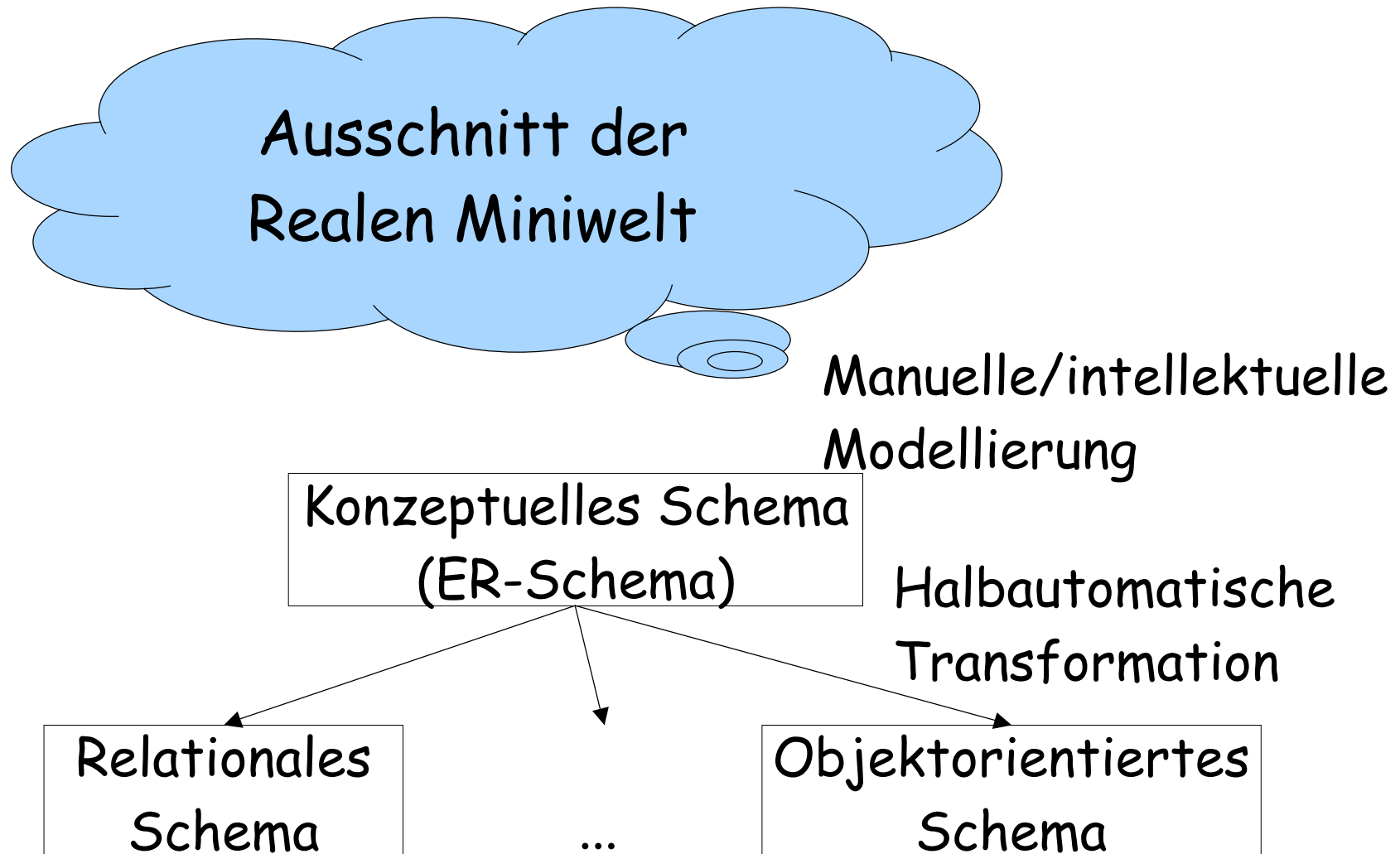
- Voraussetzungen:
 - Einführung in Entity-Relationship-Modellierung
- Lernziele heute:
 - Vertiefung der ER-Modellierung
- Vorbereitung für
 - Relationales Datenmodell
 - Objektorientiertes Datenmodell

Literatur und Danksagung

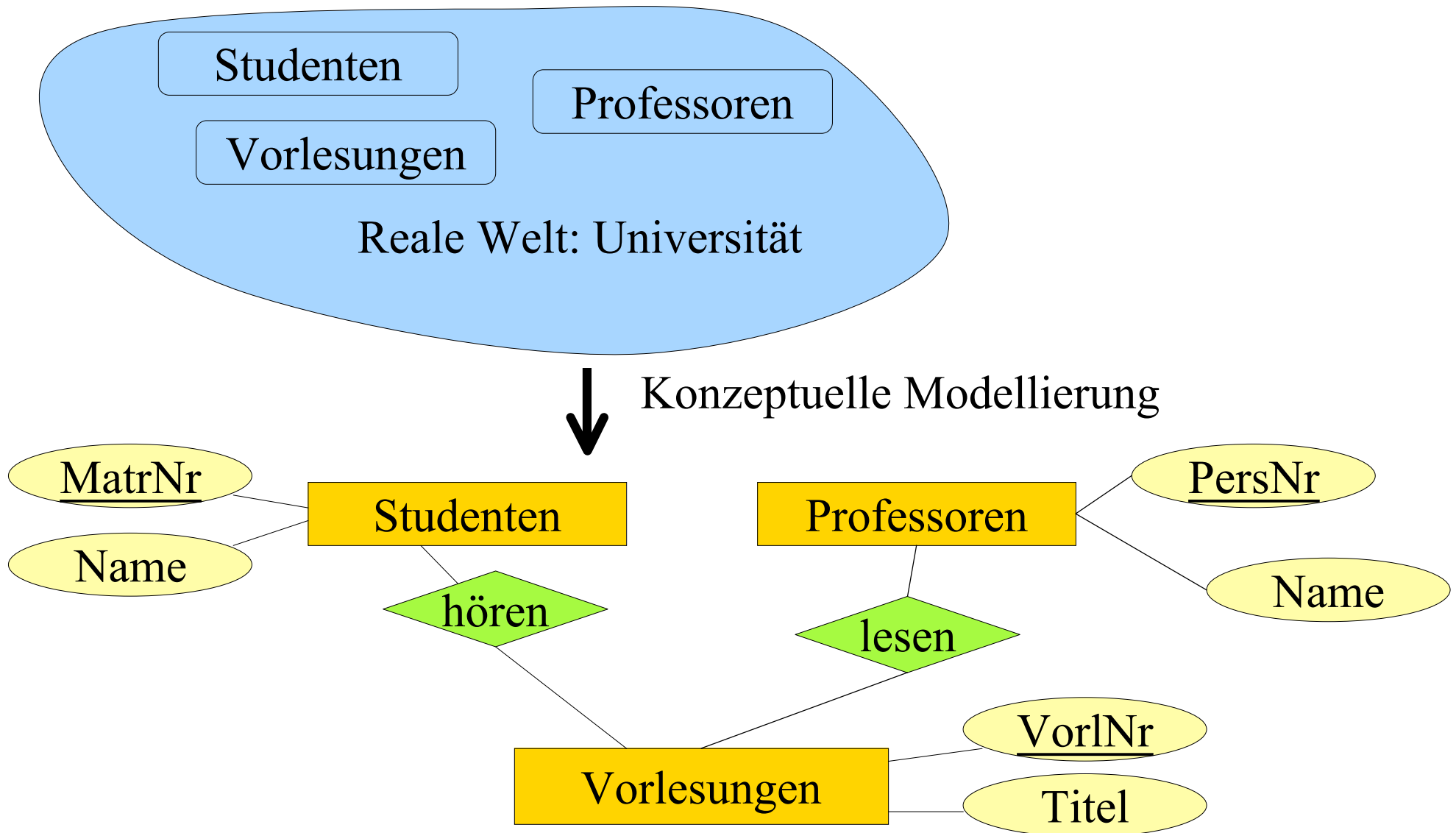
- A. Kemper, A. Eickler,
Datenbanksysteme: Eine Einführung
- Diese Vorlesung
basiert auf
Präsentations-
material zu
diesem Buch



Datenmodellierung

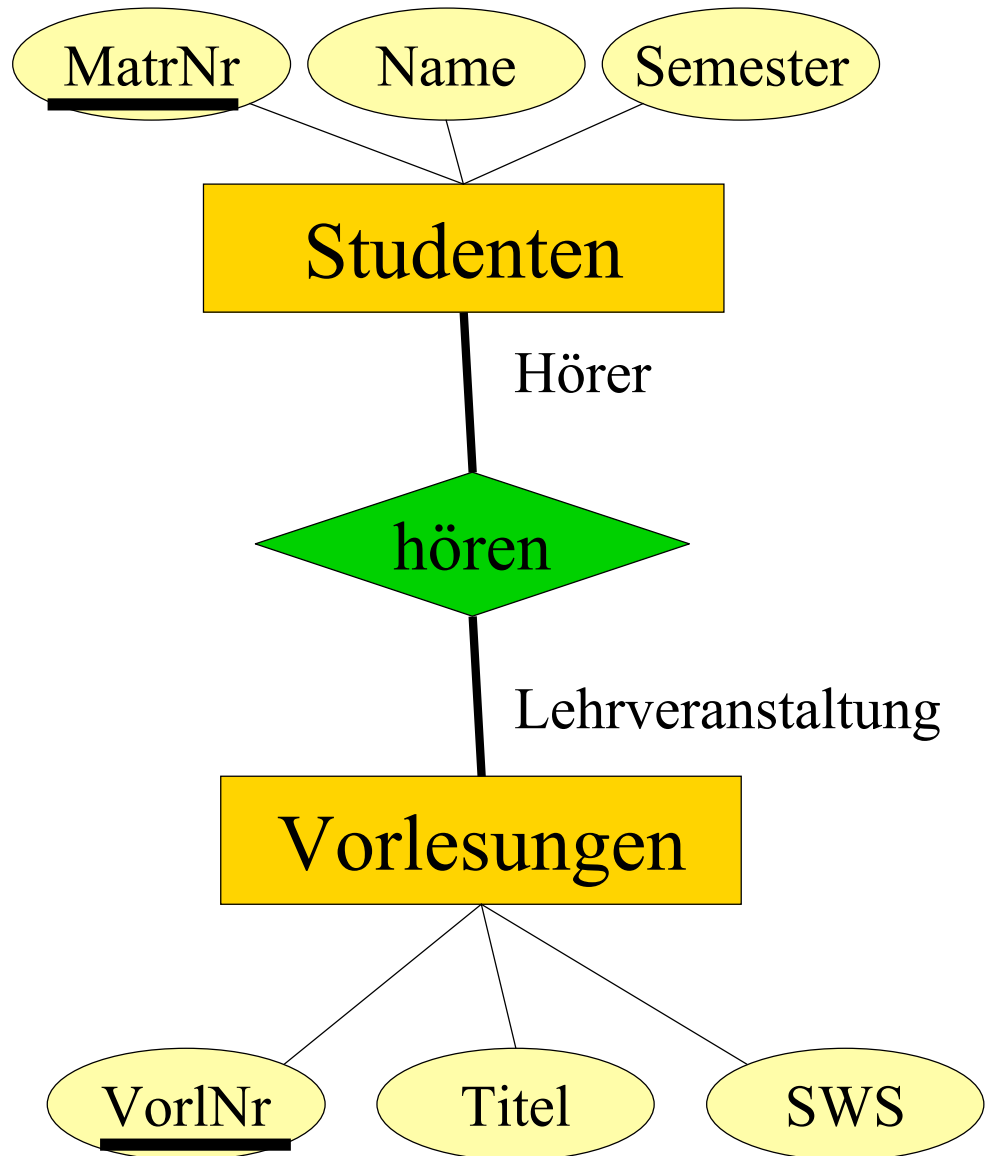


Modellierung einer kleinen Beispielanwendung

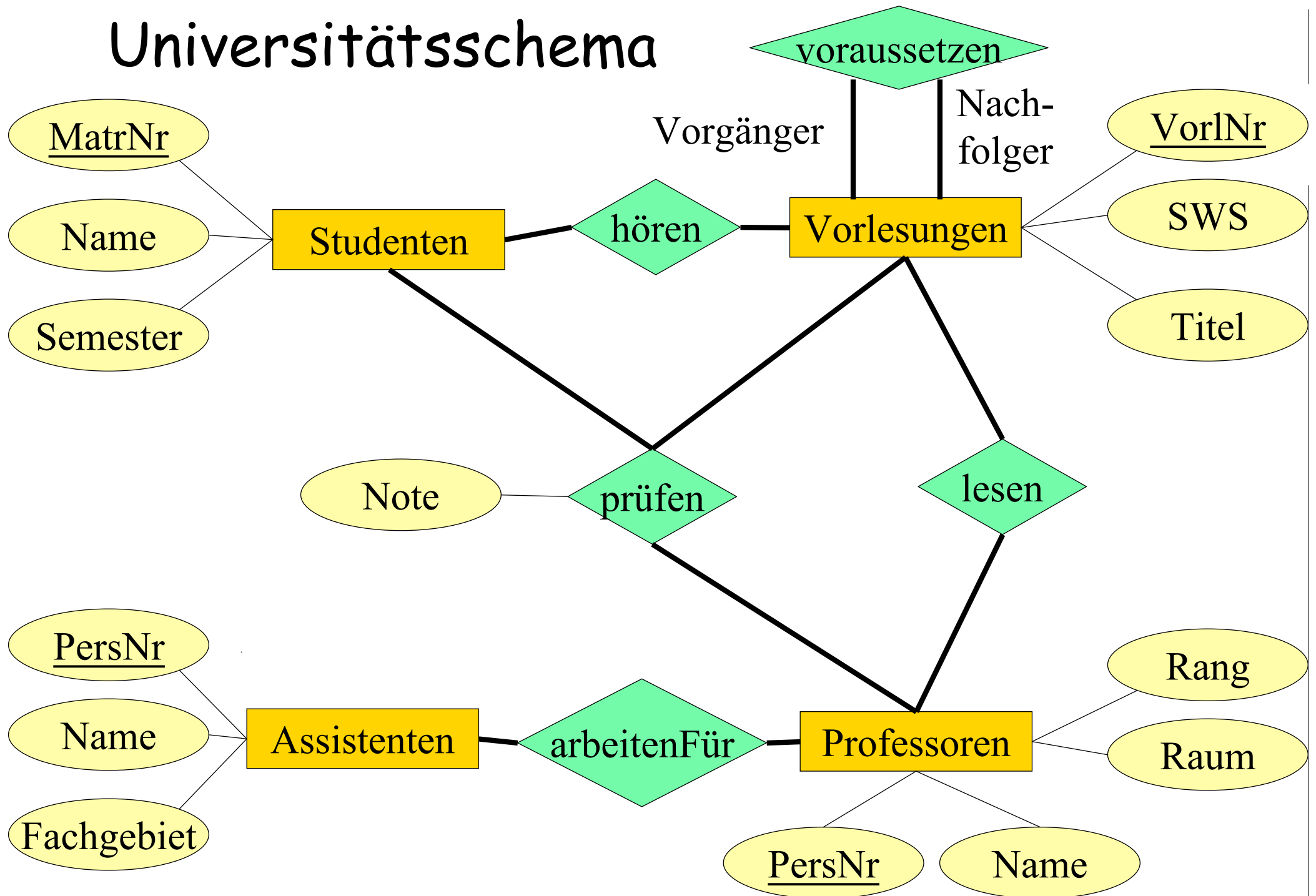


Entity/Relationship-Modellierung

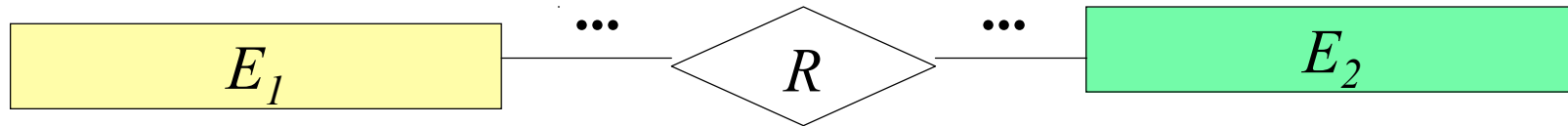
- Entity (Gegenstandstyp)
- Relationship (Beziehungstyp)
- Attribut (Eigenschaft)
- Schlüssel (Identifikation)
- Rolle



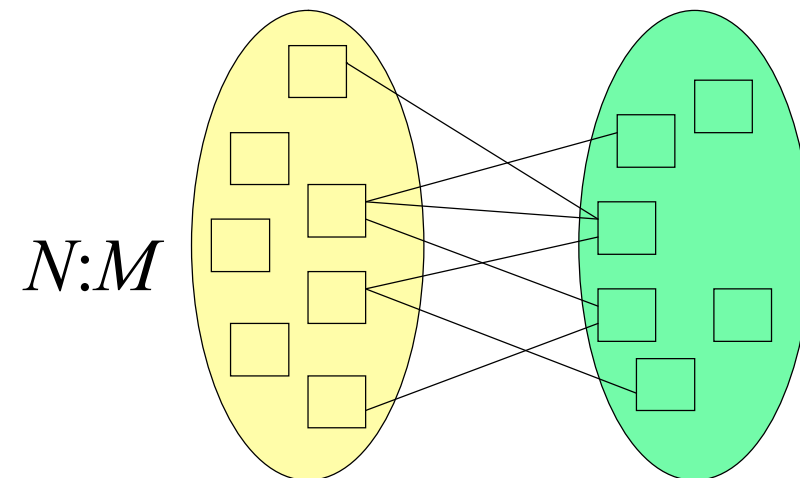
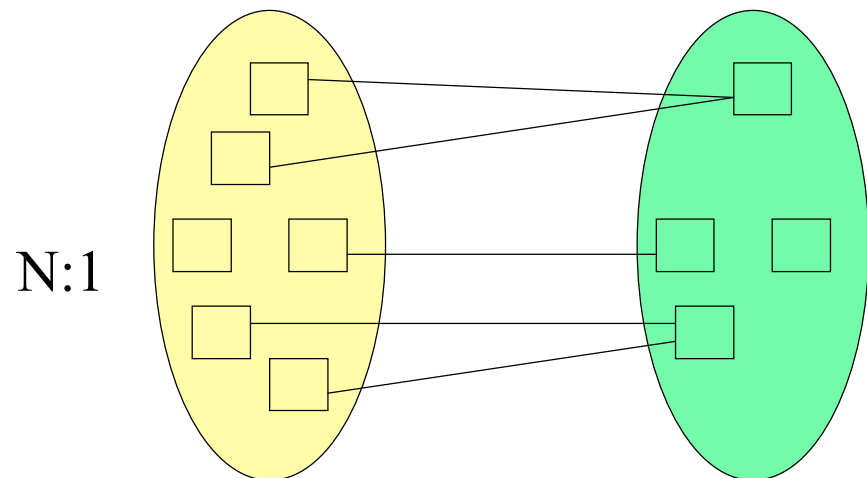
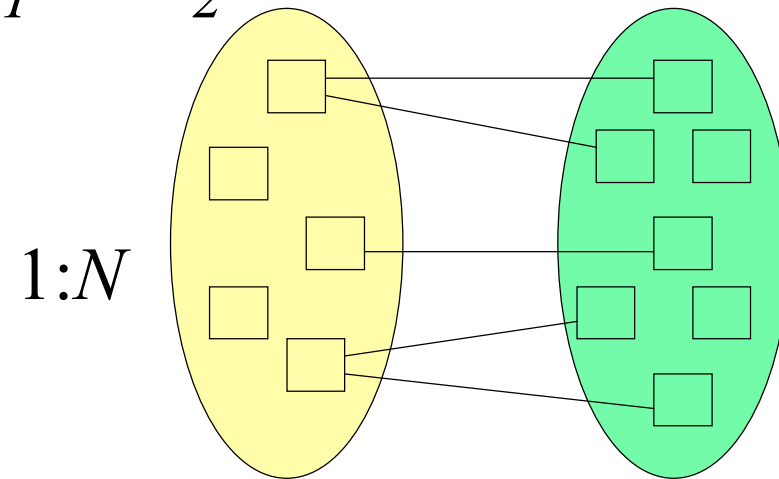
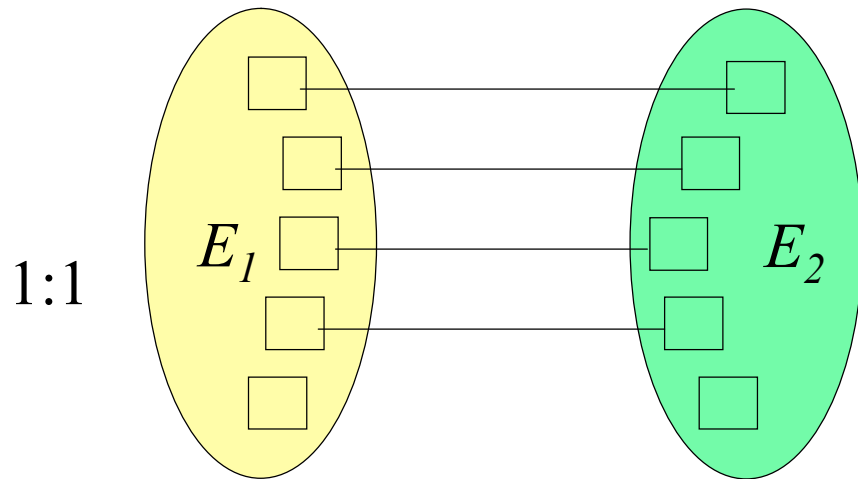
Universitätsschema



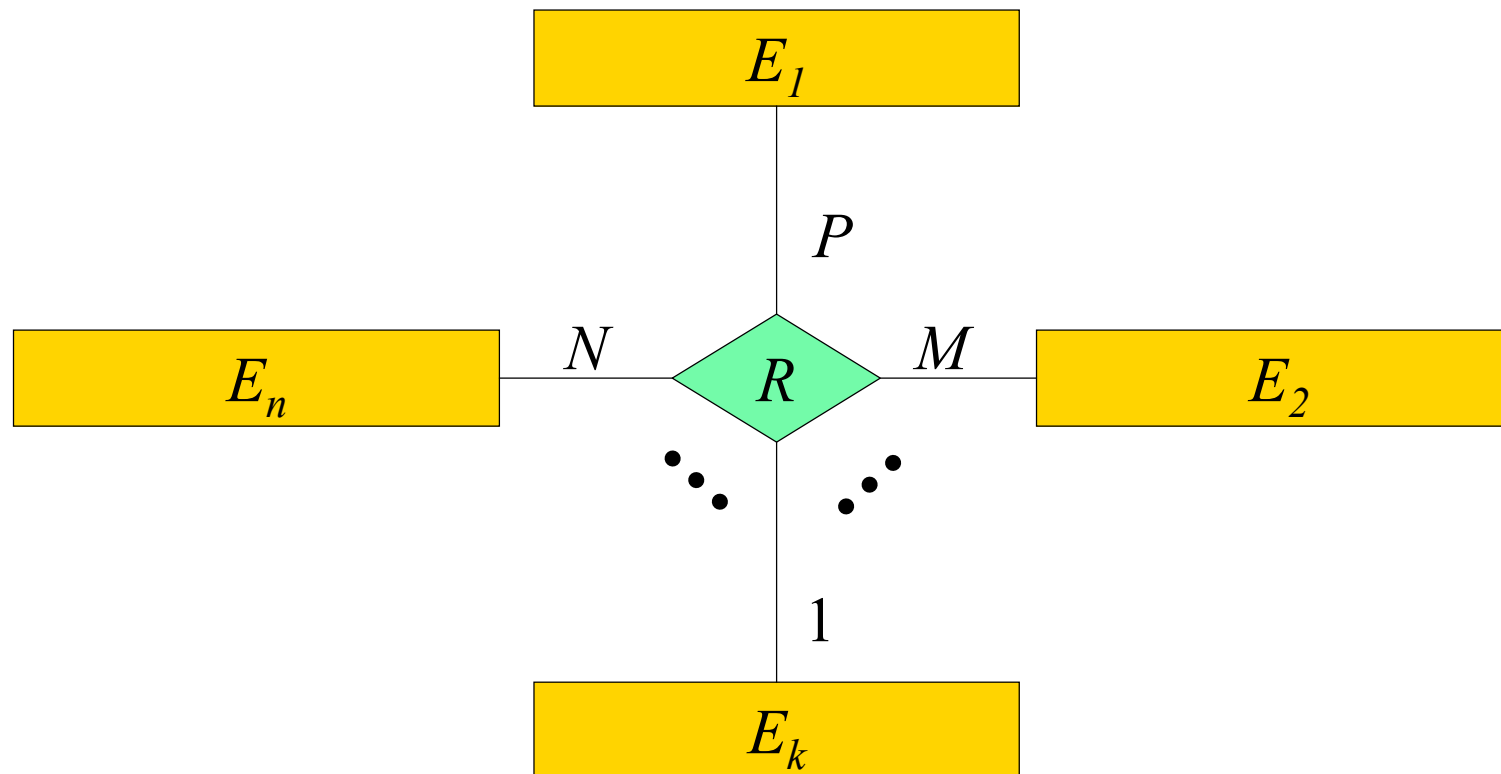
Funktionalitäten



$$R \sqsubseteq E_1 \times E_2$$

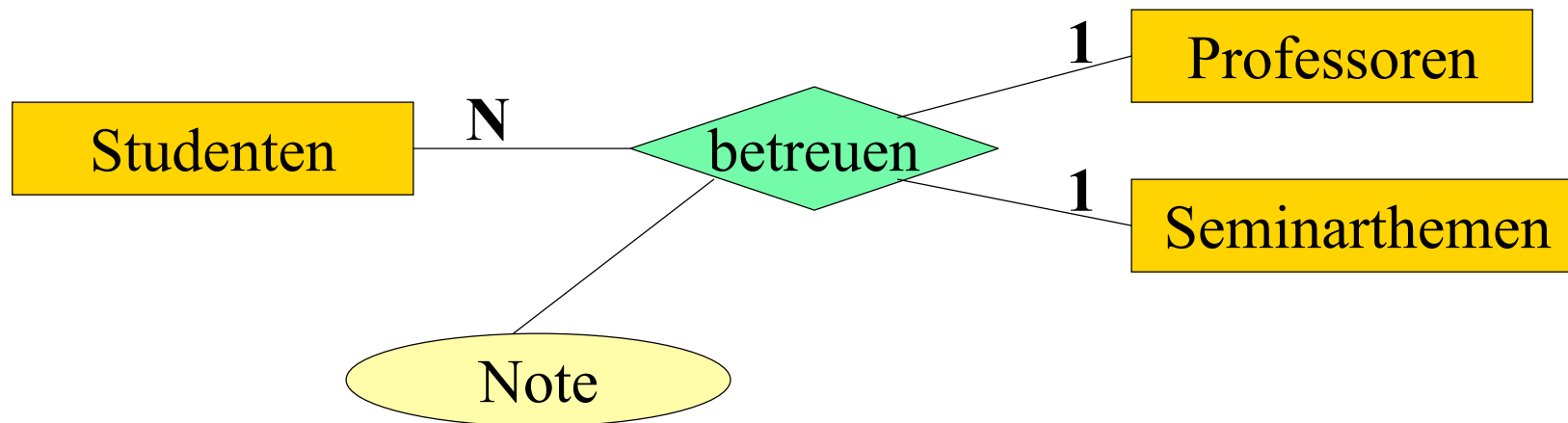


Funktionalitäten bei *n*-stelligen Beziehungen



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

Beispiel-Beziehung: *betreuen*



betreuen : Professoren x Studenten \square Seminarthemen

betreuen : Seminarthemen x Studenten \square Professoren

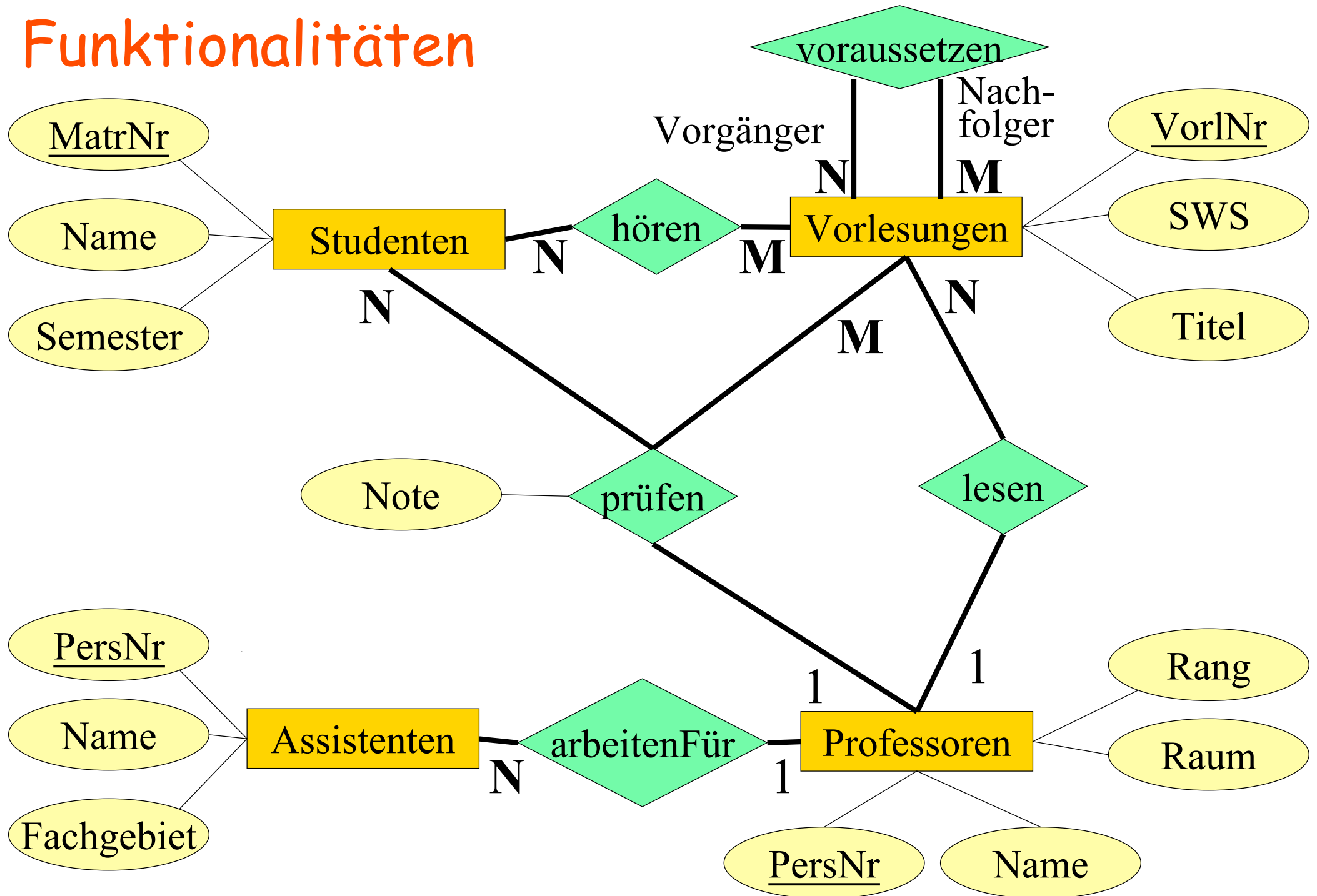
Dadurch erzwungene Konsistenzbedingungen

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.

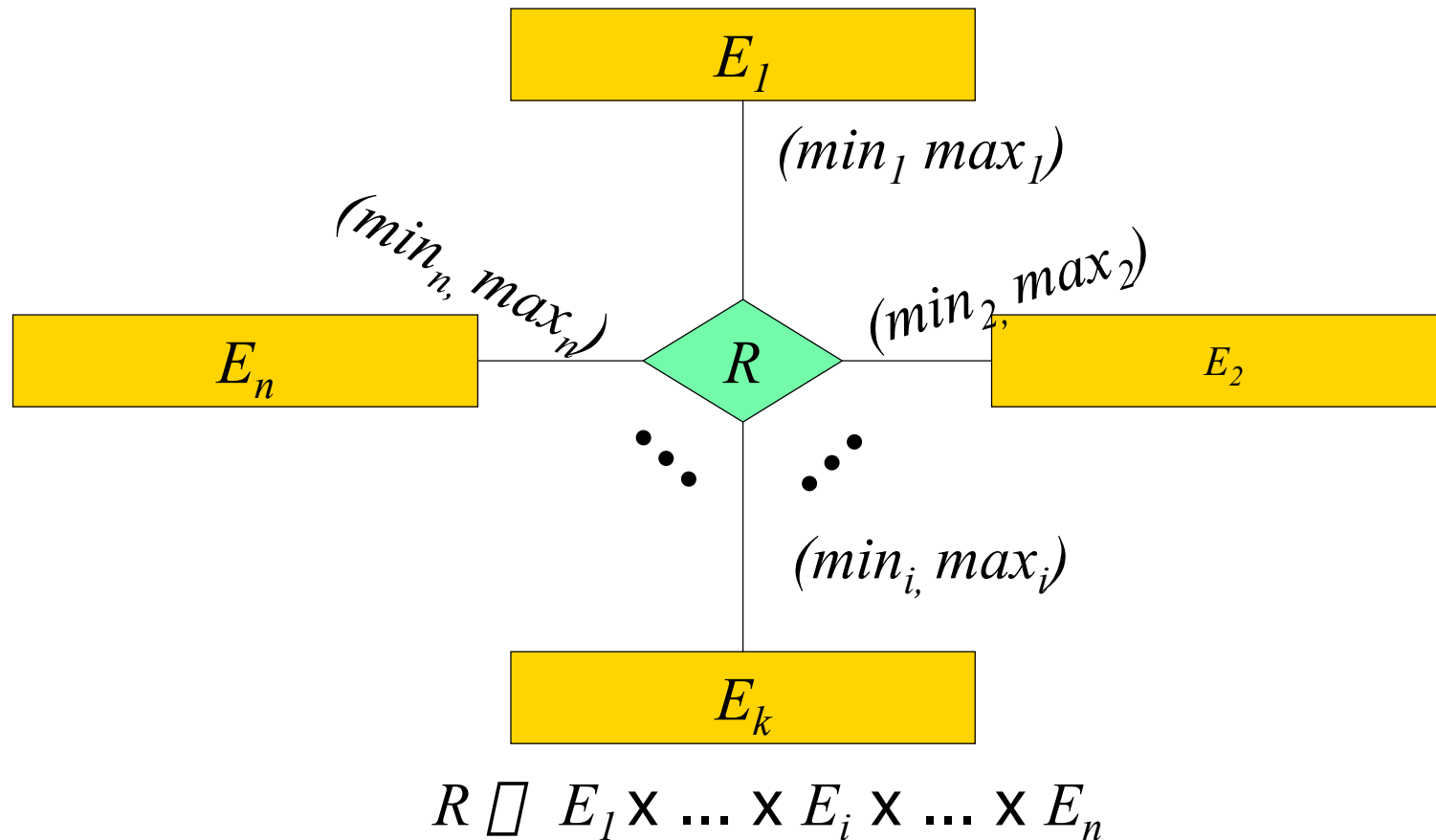
Folgende Datenbankzustände nach wie vor möglich:

- 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



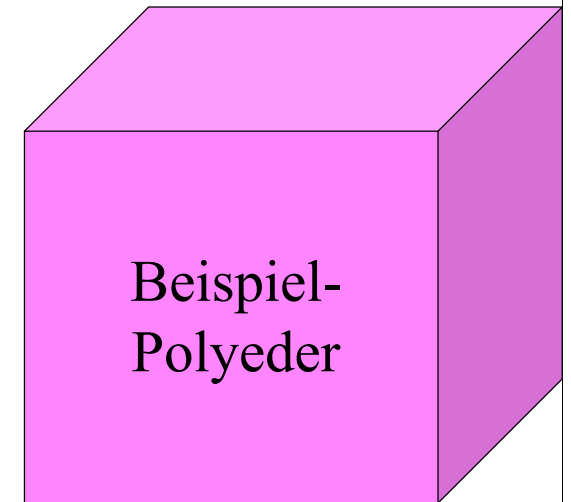
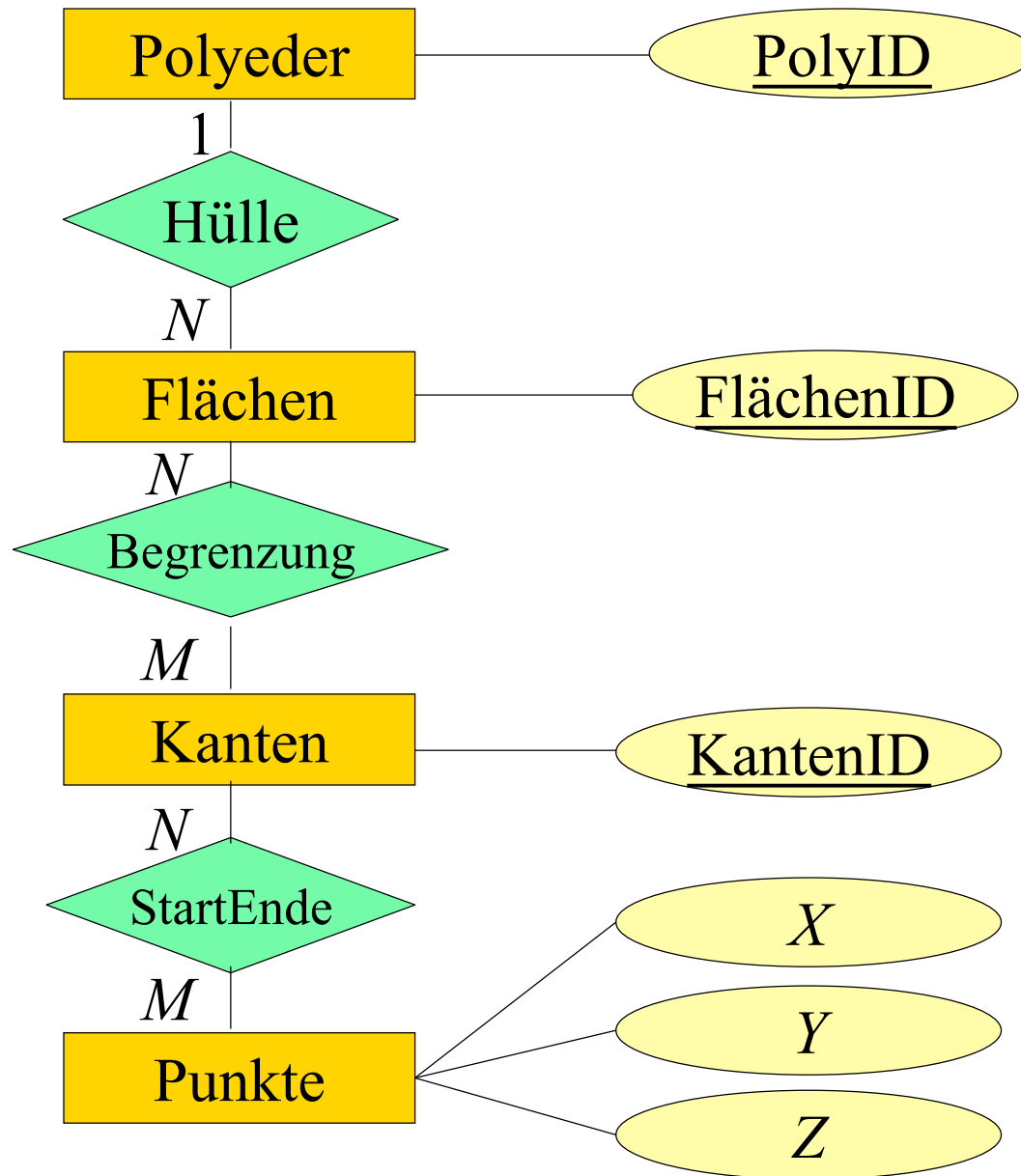
(Min, Max)-Notation



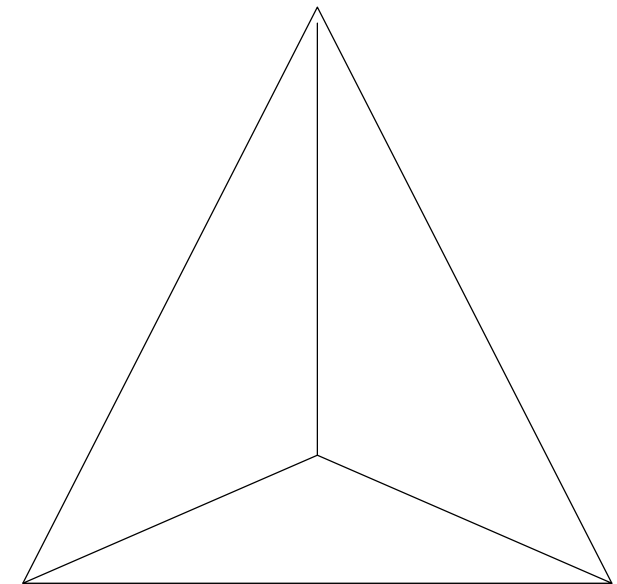
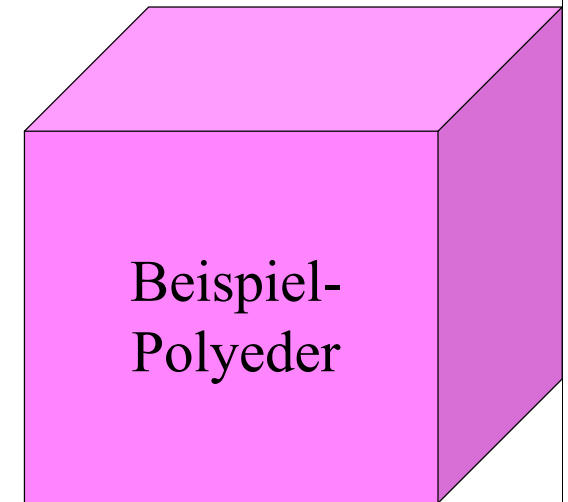
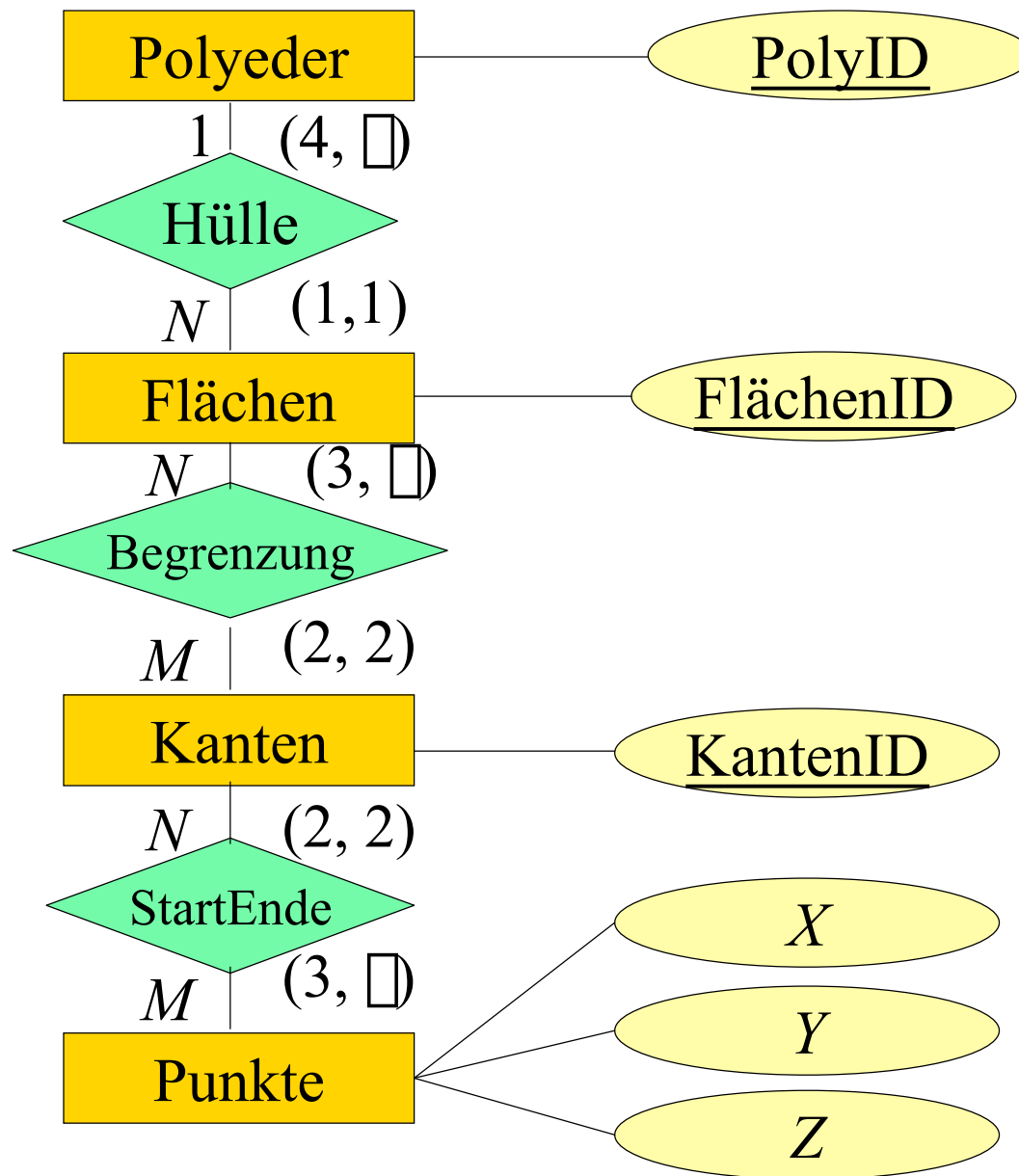
Für jedes $e_i \in E_i$ gibt es

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

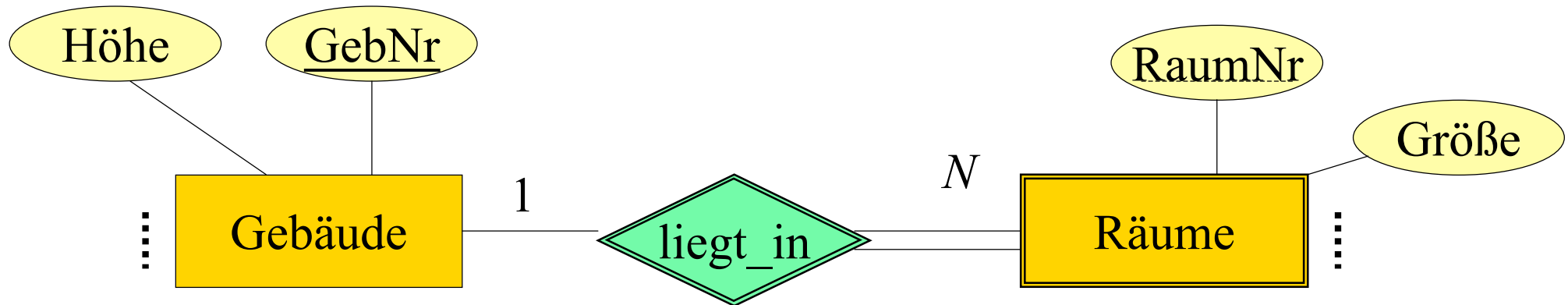
Komplex-strukturierte Entities



Komplex-strukturierte Entities

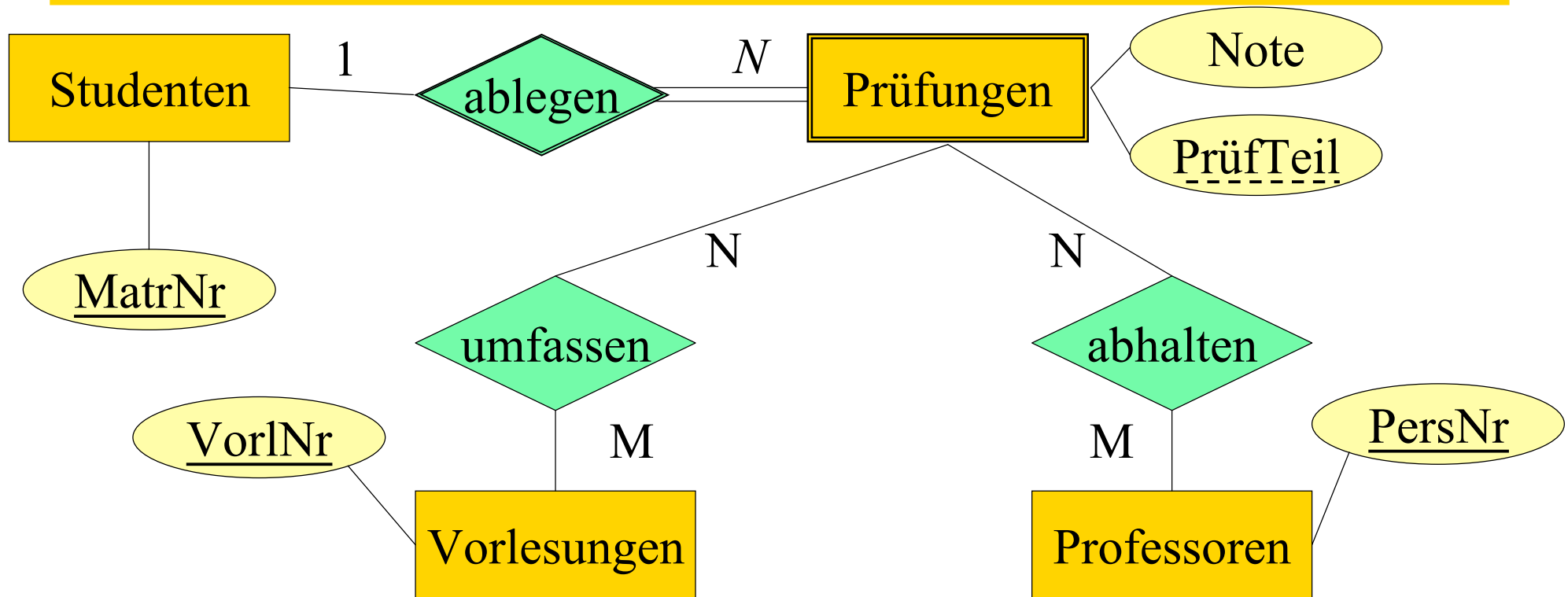


Schwache, existenzabhängige Entities



- 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

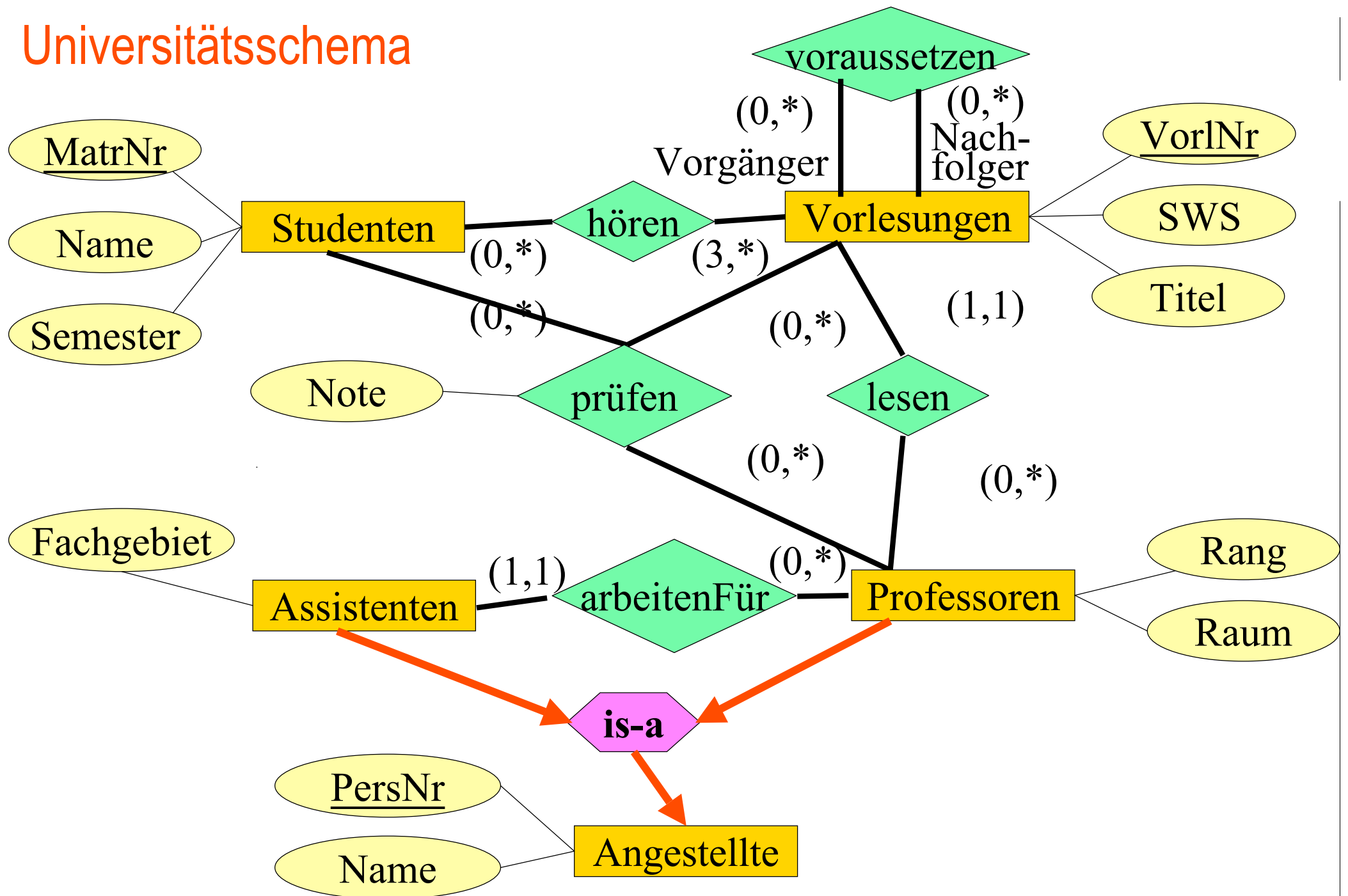
Prüfungen als schwacher Entitytyp



Mehrere Prüfer in einer Prüfung

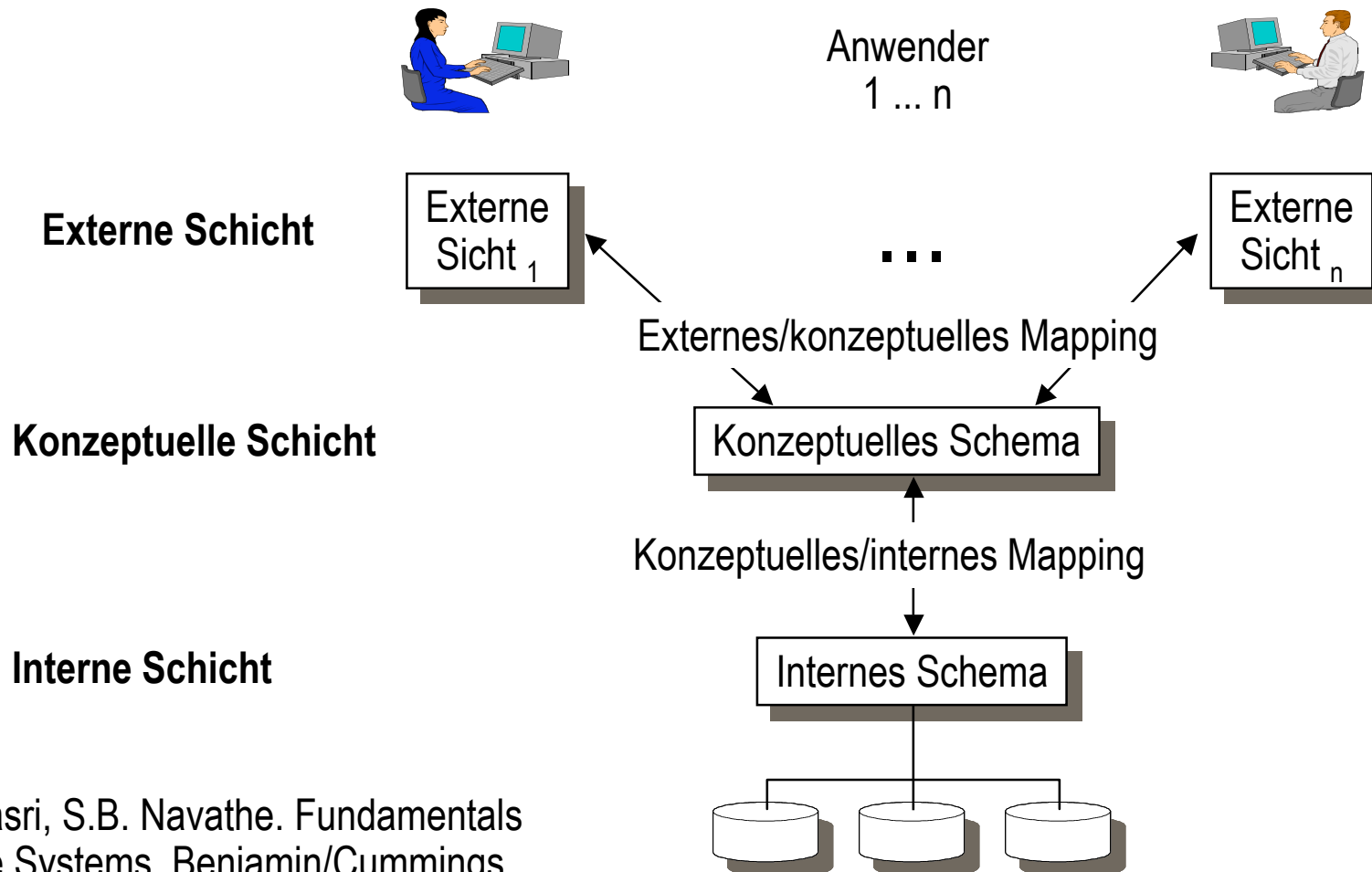
Mehrere Vorlesungen werden in einer Prüfung abgefragt

Universitätsschema



Datenunabhängigkeit durch Schemaarchitekturen (1)

■ Die Drei-Schichten-Architektur:



vgl. R. Elmasri, S.B. Navathe. Fundamentals of Database Systems. Benjamin/Cummings, Redwood City, CA. 1989.

Datenunabhängigkeit durch Schemaarchitekturen (2)

■ Externe Schicht:

- Jedes externe Schema (Benutzersicht: Benutzer = Anwendungsprogramm) beschreibt die Sicht eines oder mehrerer Benutzer auf die Daten.
- Für den Benutzer nicht relevante Daten werden vor ihm verborgen.
- Beispiel: Das Schema eines Projektinformationssystems verbirgt das Gehalt der Mitarbeiter.

■ Konzeptuelle Schicht:

- Das konzeptuelle Schema legt die Strukturen der konzeptuellen Sicht der gesamten Datenbank für die gesamte Benutzergemeinde fest (Vereinigung aller Anwendersichten zu einer gemeinschaftlichen Sicht).
- Berücksichtigt werden Entitäten, Datentypen, Beziehungen und Integritätsbedingungen, die physikalischen Speicherstrukturen werden verborgen.
- Beispiel: Das Schema eines Firmeninformationssystems sammelt alle Informationen über die Mitarbeiter.

Datenunabhängigkeit durch Schemaarchitekturen (3)

■ Interne Schicht:

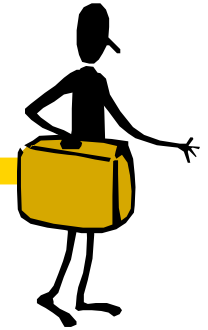
- Internes Schema beschreibt die physikalischen Speicherstrukturen der Datenbank.
- Unter Benutzung eines physikalischen Datenmodells werden Details der Datenspeicherung und Zugriffspfade beschrieben.
- **Beispiele:**
 - | Separate Speicherbereiche für Festangestellte und Werksstudenten
 - | B-Tree: Zugriff auf Projekte über die Projektnummer

Datenunabhängigkeit durch Schemaarchitekturen (4)

■ Datenunabhängigkeit:

- Schutz des Benutzers eines DBMS vor nachteiligen Auswirkungen im Zuge von Änderungen in der Systemumgebung (1996 ... 2040).
- Arten der Datenunabhängigkeit:
 - **Logische Datenunabhängigkeit:**
 - Das konzeptuelle Schema kann ohne Konsequenzen für das externe Schema geändert werden.
 - Beispiel: Erweiterung der Datenbank um eine neue Klasse oder Zusammenfassung mehrerer Klassen durch Generalisierung im konzeptuellen Schema.
 - **Physische Datenunabhängigkeit:**
 - Das interne Schema kann unabhängig vom konzeptuellen Schema geändert werden, ohne daß Funktionsänderungen in den Anwendungen auftreten.
 - Beispiel: Reorganisation der Daten oder Einrichtung neuer Zugriffspfade.

Zusammenfassung, Kernpunkte



- Grundlagen von Datenbanksystemen
 - Architektur eines Datenbanksystems
 - Entwurfsebene: Entity-Relationship-Modell

Was kommt beim nächsten Mal?



- Implementierungsebene:
Das relationale Datenmodell