



FRIEDRICH-SCHILLER-  
UNIVERSITÄT  
JENA

# Datenbanksysteme I

---

Prof. Dr. Viktor Leis

Professur für Datenbanken und Informationssysteme

# Datenbankentwurf

---

- Besonders wichtig für Datenbanksysteme ist die Modellierung der Daten, die in der Datenbank gespeichert werden sollen
- Dies wird im konzeptuellen Entwurf behandelt

- Entity-Relationship-Modell (ER-Modell) wird häufig für konzeptuellen Entwurf eingesetzt
- Ein ER-Schema ist eine graphische Repräsentation der konzeptuellen Modellierung der Daten
- Identifiziert *Entitäten* (Entities) und *Beziehungen* (Relationships) zwischen Entitäten einer Anwendungsdomäne

- Ein ER-Schema
  - enthält alle Informationseinheiten, die im zu implementierenden DBS enthalten sein sollen
  - dient auch der Kommunikation mit den späteren Benutzern (um Vollständigkeit und Korrektheit sicherzustellen)
  - kann (recht) einfach in ein logisches Schema transformiert werden

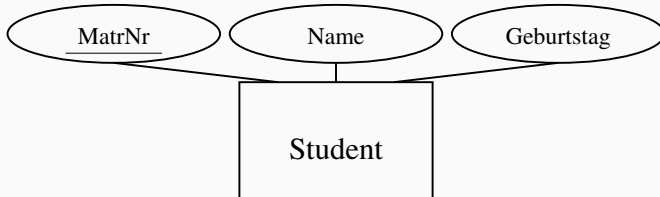
- Entitätentyp
- Attribut
- Schlüssel
- Relationship (Beziehungstyp)
- Rolle

- Entität: ein Objekt aus der realen Miniwelt, das sich von anderen Objekten unterscheidet (z.B. Dinge, Personen, Konzepte)
- Entitätentypen werden durch Rechtecke repräsentiert:



# Attribute

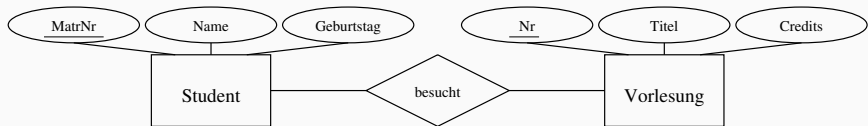
- Eine Entität wird durch Attribute beschrieben
- Zu jedem Attribut gehört ein Wertebereich (z.B. Integer, String) der die möglichen Werte dieses Attributs beschreibt
- Ein Attribut das eine Entität eindeutig beschreibt wird *Schlüssel* genannt
- Attribute werden durch eine Ellipse repräsentiert:





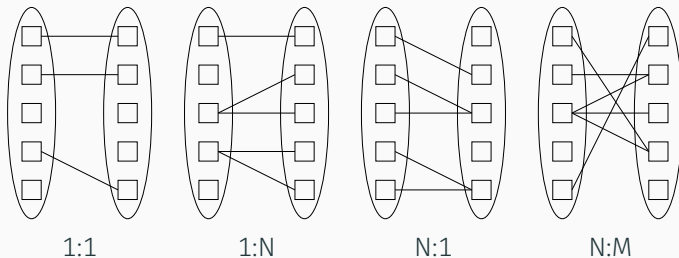
# Beziehungen (Relationships)

- Relationship: verbindet zwei oder mehr Entitätstypen miteinander
- Wird durch eine Raute repräsentiert:

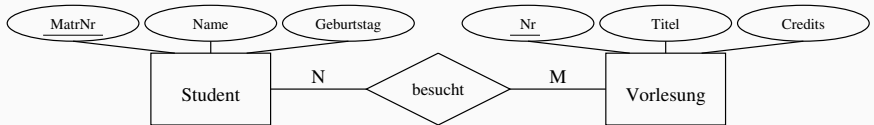


# Funktionalitäten

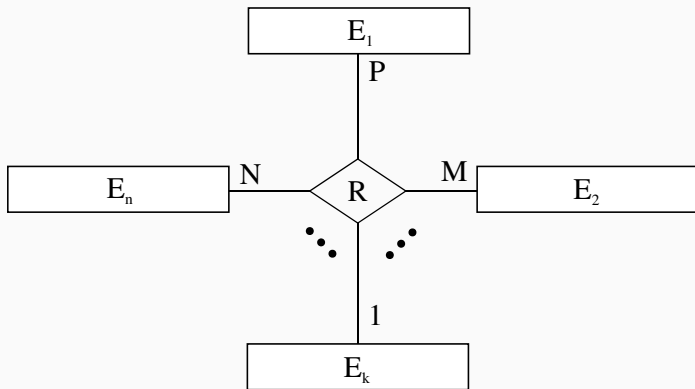
- Weitergehende Beschreibung von Beziehungen durch Funktionalitäten
- Funktionalitäten drücken aus mit wievielen anderen Entitäten eine Entität eine Beziehung eingeht



- `besucht` ist eine N:M Beziehung:



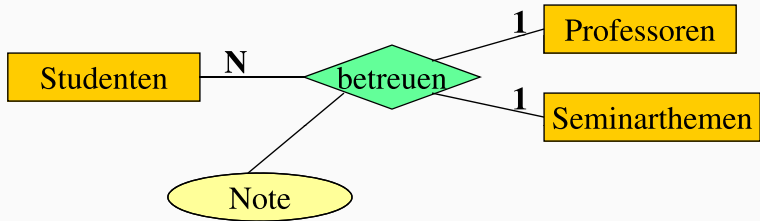
## Funktionalitäten(3)



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

## Funktionalitäten(4)

Beispielbeziehung: *betreuen*



$\textit{betreuen} : \textit{Professoren} \times \textit{Studenten} \rightarrow \textit{Seminarthemen}$

$\textit{betreuen} : \textit{Seminarthemen} \times \textit{Studenten} \rightarrow \textit{Professoren}$

# Funktionalitäten(5)

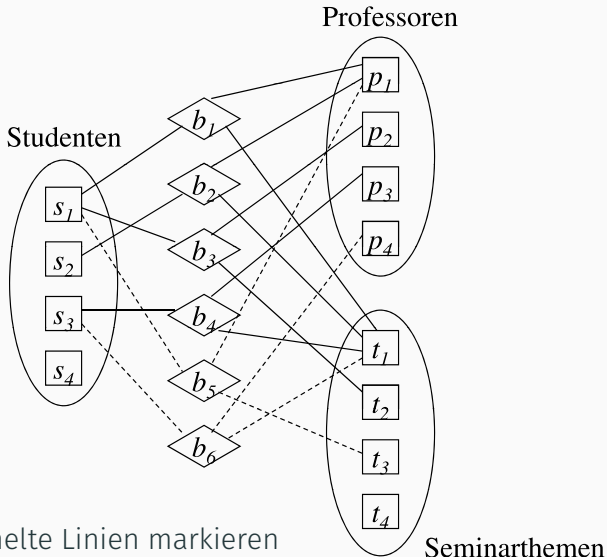
Dadurch erzwungene Konsistenzbedingungen:

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.

Es sind aber folgende Datenbankzustände nach wie vor möglich:

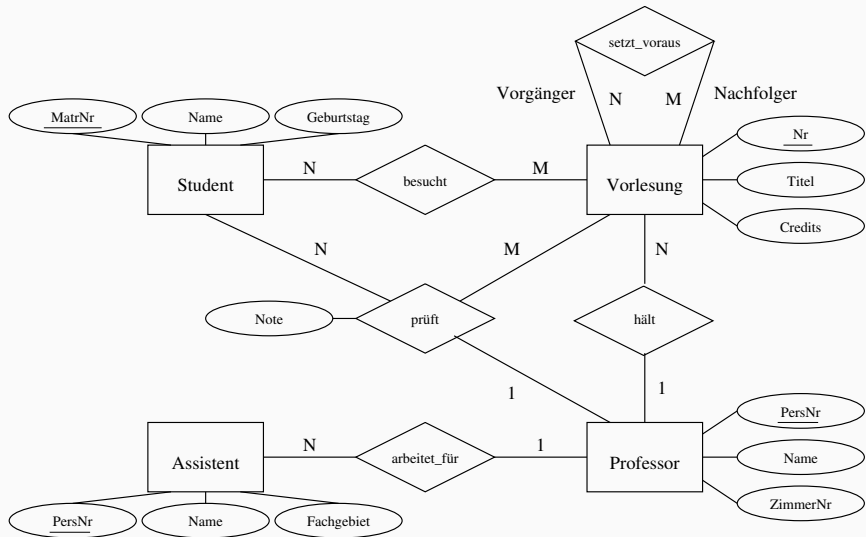
- Professoren können dasselbe Seminarthema "wiederverwenden" – also dasselbe Thema auch mehreren Studenten erteilen.
- Ein Thema kann von mehreren Professoren vergeben werden – aber an unterschiedliche Studenten.

# Funktionalitäten(6)



Gestrichelte Linien markieren illegale Ausprägungen.

# Vollständiges Schema

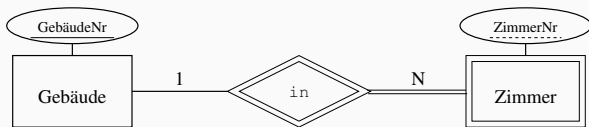




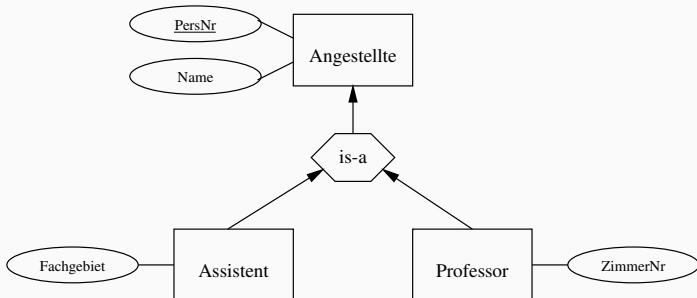
- Schwache Entitäten
- Generalisierungen
- Aggregationen

# Schwache Entitäten

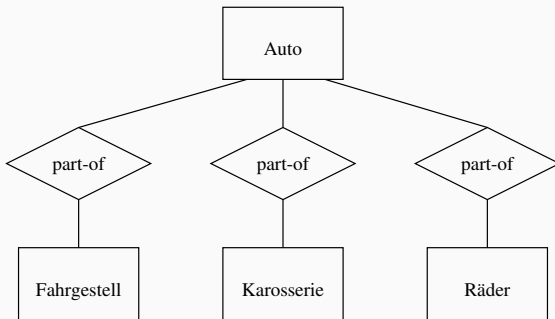
- Schwache Entität: kann nicht als eigenständiges Gebilde existieren
- Braucht eine "starke" Entität, um eindeutig identifiziert werden zu können
- Eine schwache Entität und zugehörige starke Entität gehen eine 1:N (oder seltener 1:1) Beziehung ein



- Wie in objekt-orientierten Programmiersprachen, können Attribute von einem Obertyp an einen Untertyp vererbt werden



- Wird bei der Modellierung von ist-Teil-von Beziehungen benutzt

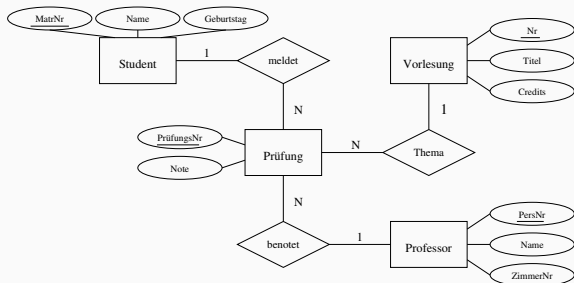


- Entität vs. Attribut
- Entität vs. Beziehung
- Binäre vs. ternäre Beziehungen
- Connection Trap

- Ein eigener Entitätstyp ist flexibler, ein Attribut einfacher
- Beispiel: wir wollen Adressinformationen von Kunden speichern
  - Falls es mehrere Adressen pro Kunde geben kann, modelliere Adresse als eigene Entität (da Attribute nicht mengenwertig sein dürfen)
  - Falls eine Adresse in einzelne Bestandteile zerlegt werden soll (Straße, Hausnummer, PLZ, Stadt, usw.), dann zerlege Adresse in einzelne Attribute (diese können dann eventuell in eigene Entität ausgegliedert werden)

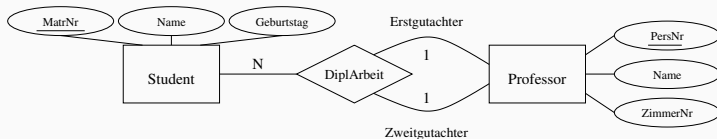
# Entität vs. Beziehung

- Entität ist flexibler, Beziehung einfacher
- Beziehungen haben keine eigene Identität: die selben Entitäten können nicht mehr als einmal eine Beziehung eingehen
- Beispiel: mehr als eine Note für die gleiche Vorlesung vom gleichen Professor:



# Binäre vs. ternäre Beziehung

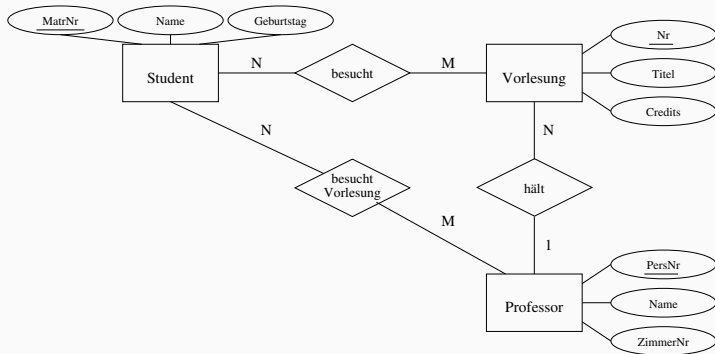
- Echte ternäre Beziehungen benötigen alle drei teilnehmenden Entitäten, um Beziehung zu beschreiben
- Beziehung **prüft** im vollständigen Schema ist echt ternär
- Die folgende Beziehung ist es nicht:





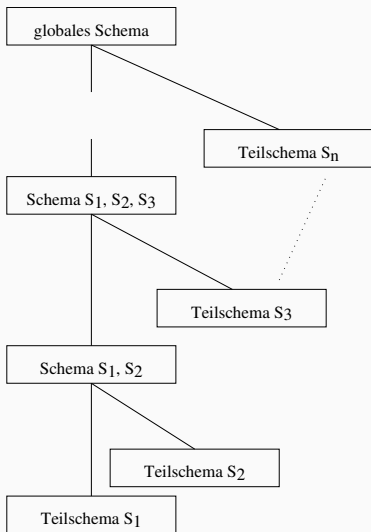
# Connection Trap

- Eine Falle in die man leicht fällt ist die connection trap: redundante, zyklische Beziehungen
- **besucht Vorlesung** ist redundant, sie wird bereits durch **besucht** und **hält** ausgedrückt



- Oft werden Anwendungen von verschiedenen Personen modelliert, die auch verschiedene Blickpunkte auf die Anwendung haben
- Es entstehen Teilschemata, die in ein Gesamtschema zusammengeführt werden müssen

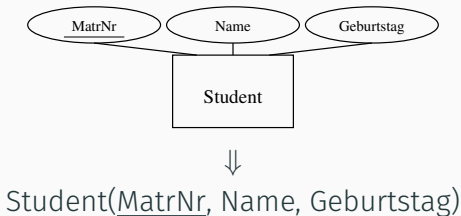
## Schemakonsolidierung(2)



- Für ein relationales DBMS muss ein ER-Schema in ein relationales Schema übersetzt werden
- Dies geschieht in drei Schritten:
  - Übersetzung der Entitäten
  - Übersetzung der Beziehungen
  - Vereinfachung des Schemas

# Übersetzung von Entitäten

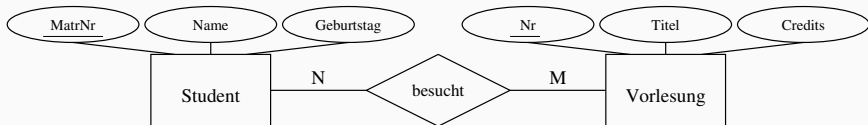
- Wandle jede Entität in eine Relation um
- Jedes Attribut der Entität wird zu einem Attribut der Relation (Schlüssel der Entität wird Primärschlüssel der Relation)



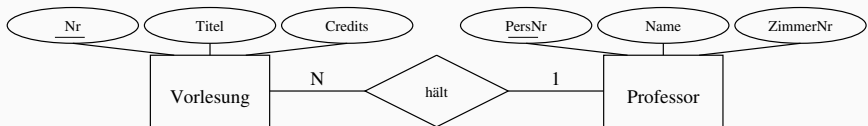
# Übersetzung von Beziehungen

- Wandle jede Beziehung in eine Relation um
- Attribute dieser Relation sind die Schlüssel der teilnehmenden Relationen (+ Attribute der Beziehung selbst)
- Der Schlüssel einer (binären) Beziehung hängt von der Art der Beziehung ab
  - N:M - Schlüssel setzt sich aus Kombination beider Entitätsschlüssel zusammen
  - 1:N, N:1 - Schlüssel der N-Seite wird Schlüssel
  - 1:1 - einer der beiden Entitätsschlüssel kann gewählt werden

# Beispiele



besucht(MatrNr, Nr)



hält(Nr, PersNr)

- Nach dem zweiten Schritt können Relationen mit dem gleichen Schlüssel existieren
- Diese Relationen werden in einer einzigen Relation zusammengefaßt
- Dies passiert bei 1:N, N:1 und 1:1 Beziehungen, da die Beziehungsrelation den gleichen Schlüssel wie eine der Entitätsrelationen hat



- **Vorlesung** und **hält** haben den gleichen Schlüssel:

Vorlesung(Nr, Titel, Credits)

hält(Nr, PersNr)



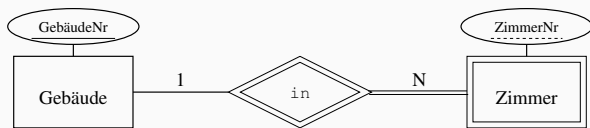
Vorlesung(Nr, Titel, Credits, PersNr)

- **PersNr** wird *Fremdschlüssel* von **Vorlesung** genannt, da er den Schlüssel in **Professor** referenziert (und auch die gleichen Werte enthält)
- Häufig Umbenennung, um Referenzen klarzumachen

Vorlesung(Nr, Titel, Credits, ProfPersNr)

# Schwache Entitäten

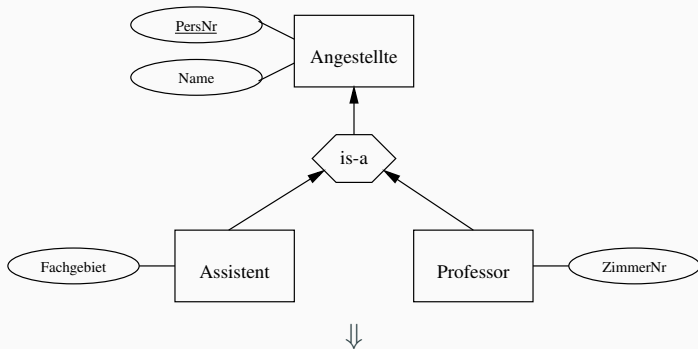
- Schwache Entitäten werden im Prinzip wie 1:N Beziehungen umgesetzt
- Einziger Unterschied: die schwache Entität hat einen zusammengesetzten Schlüssel



Gebäude(GebäudeNr, ...) Zimmer(GebäudeNr, ZimmerNr, ...)

# Generalisierungen

- Generalisierungen können auf drei verschiedene Weisen umgesetzt werden (es gibt keine direkte äquivalente Übersetzung ins relationale Modell)



(1) Nur Obertyp, (2) nur Untertypen, (3) alle Typen

## Generalisierungen(2)

- Nur Obertyp:  
Angestellte(PersNr, Name, Fachgebiet, ZimmerNr)  
Problem: NULL-Werte
- Nur Untertypen:  
Assistent(PersNr, Name, Fachgebiet)  
Professor(PersNr, Name, ZimmerNr)  
Problem: Angestellte die weder Assistenten noch Professoren sind
- Alle Typen:  
Angestellte(PersNr, Name)  
Assistent(PersNr, Fachgebiet)  
Professor(PersNr, ZimmerNr)  
Problem: Information ist verteilt auf mehrere Relationen

- ER Modell ist ein weit verbreitete Art der konzeptuellen Modellierung
- Die Konzepte sind nicht allzu schwer zu begreifen, d.h. der Sachverhalt kann auch mit Informatiklaien (die den Fachbereich kennen) diskutiert werden
- Entwurf ist auch immer etwas subjektives, es gibt viele verschiedene Wege einen Sachverhalt zu modellieren (sich für den passendsten zu entscheiden ist eine Sache der Erfahrung)
- Das konzeptuelle Modell wird schließlich ins relationale Modell überführt