

Sommersemester 2026

# Datenmanagement & -analyse

**Prof. Dr. Christoph M. Flath**

*Lehrstuhl für Wirtschaftsinformatik und Business Analytics*

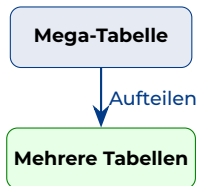
*Julius-Maximilians-Universität Würzburg*

## ► 1 Rückblick & Motivation

- 2 Entitäten & Attribute
- 3 Beziehungen & Kardinalitäten
- 4 Erweiterte Konzepte
- 5 Notationen im Vergleich
- 6 Modellierungspraxis
- 7 Zusammenfassung

## Probleme der Mega-Tabelle:

- **Redundanz** – gleiche Daten mehrfach
- **Änderungsanomalie** – Inkonsistenzen
- **Einfügeanomalie** – abhängige Daten
- **Löschanomalie** – Datenverlust



## Lösung:

Daten auf mehrere Tabellen aufteilen!

# Wie wissen wir, welche Tabellen wir brauchen?

⇒ **Entity-Relationship-Modellierung**

## Entity-Relationship-Modell (ER-Modell):

- Grafische Notation für Datenstrukturen
- Entwickelt 1976 von Peter Chen
- Beschreibt die “Welt” in:
  - **Entitäten** (Dinge)
  - **Beziehungen** (Verbindungen)
  - **Attributen** (Eigenschaften)
- Unabhängig von der Datenbank



*Einfaches ER-Diagramm*

## Vorteile:

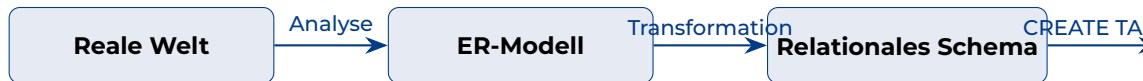
- ✓ Visuelle Kommunikation
- ✓ Gemeinsame Sprache mit Fachexperten
- ✓ Fehler früh erkennen
- ✓ Dokumentation der Datenstruktur
- ✓ Grundlage für SQL-Schema

## Einsatzgebiete:

- Neue Datenbankprojekte
- Systemanalyse
- Requirements Engineering
- Reverse Engineering bestehender DBs
- Klausuren und Prüfungen

## Praxisrelevanz

ER-Modelle sind Standard in der Softwareentwicklung – jeder IT-Professional sollte sie lesen und erstellen können.



- **Heute:** Reale Welt → ER-Modell
- **Nächste Vorlesung:** ER-Modell → Relationales Schema

- 1 Rückblick & Motivation
- ▶ **2 Entitäten & Attribute**
- 3 Beziehungen & Kardinalitäten
- 4 Erweiterte Konzepte
- 5 Notationen im Vergleich
- 6 Modellierungspraxis
- 7 Zusammenfassung



## Definition:

Eine **Entität** ist ein unterscheidbares “Ding” der realen Welt.

## Beispiele:

- Ein konkreter Spieler (Thomas Müller)
- Ein konkreter Verein (Bayern München)
- Eine konkrete Bestellung (#12345)

## Entitätstyp:

Menge gleichartiger Entitäten  
(z.B. “alle Spieler”)

**Spieler**

*Entitätstyp*

**Verein**

*Entitätstyp*

## Schritt-für-Schritt:

- 1 **Substantive markieren** in der Anforderungsbeschreibung
- 2 **Filtern:** Welche sind eigenständige “Dinge”?
- 3 **Prüfen:** Haben sie eigene Eigenschaften?
- 4 **Abgrenzen:** Sind sie von anderen unterscheidbar?

## Beispiel-Anforderung:

*“Kunden bestellen Produkte. Jeder Kunde hat einen Namen und eine Adresse. Produkte haben einen Preis und gehören zu einer Kategorie.”*

⇒ **Entitäten:** Kunde, Produkt, Kategorie

⇒ **Keine Entitäten:** Name, Adresse, Preis (das sind Attribute!)

## Eher Entität, wenn:

- Eigene Eigenschaften vorhanden
- Mehrere Beziehungen zu anderen
- Eigenständig speicherbar
- Mehrfach referenziert

## Beispiel:

“Adresse” mit Straße, PLZ, Ort, Land  
⇒ Entität (wenn wiederverwendet)

## Eher Attribut, wenn:

- Nur ein Wert pro Entität
- Keine eigenen Beziehungen
- Einfacher Datentyp
- Gehört fest zur Entität

## Beispiel:

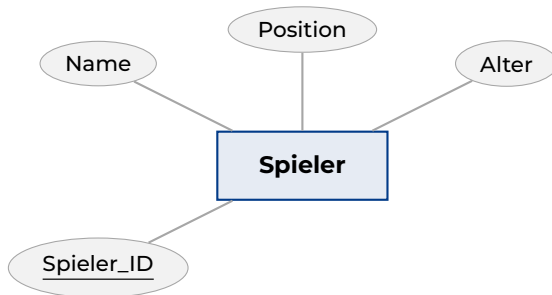
“Geburtsdatum” eines Kunden  
⇒ Attribut

## Faustregel

Im Zweifelsfall: Eher zu Entität tendieren. Refactoring zu Attribut ist einfacher als umgekehrt.

## Definition:

**Attribute** beschreiben Eigenschaften von Entitäten oder Beziehungen.



Schlüsselattribut

## Einfache Attribute:

- Atomarer Wert (nicht teilbar)
- Beispiel: Alter, Punktzahl

Alter

## Zusammengesetzte Attribute:

- Besteht aus Teilattributen
- Beispiel: Adresse (Straße, PLZ, Ort)



## Mehrwertige Attribute:

- Mehrere Werte pro Entität
- Beispiel: Telefonnummern

Telefon

## Abgeleitete Attribute:

- Berechenbar aus anderen
- Beispiel: Alter (aus Geburtsdatum)

Alter

## Schlüsselattribut:

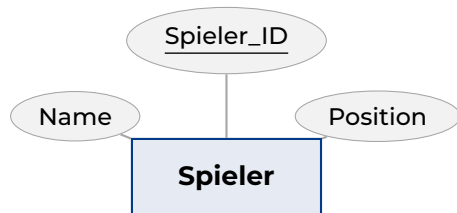
- Identifiziert eine Entität eindeutig
- Wird unterstrichen dargestellt
- Entspricht dem Primärschlüssel

## Beispiele:

- Spieler\_ID
- Verein\_ID
- Matrikelnummer

## Zusammengesetzter Schlüssel:

Mehrere Attribute zusammen bilden den Schlüssel



## Starke Entität:

- Existiert unabhängig
- Hat eigenen Primärschlüssel
- Beispiel: Verein

**Verein**

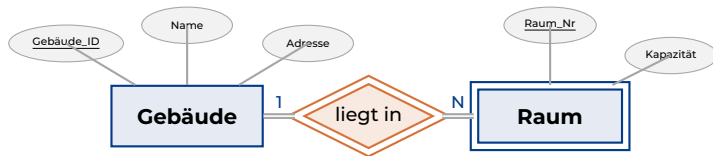
## Schwache Entität:

- Existiert nur mit "Owner"
- Braucht Fremdschlüssel zur Identifikation
- Beispiel: Raum (in einem Gebäude)

**Raum**

## Praxis

Schwache Entitäten sind selten. Im Zweifelsfall: starke Entität mit eigenem Schlüssel.



## Eigenschaften schwacher Entitäten:

- **Partieller Schlüssel:** Raum\_Nr (nur innerhalb eines Gebäudes eindeutig)
- **Identifizierende Beziehung:** "liegt in" (doppelte Raute)
- **Vollständiger Schlüssel:** Gebäude\_ID + Raum\_Nr

**Weitere Beispiele:** Kontozeile (in Konto), Bestellposition (in Bestellung), Stockwerk (in Gebäude)



- 1 Rückblick & Motivation
- 2 Entitäten & Attribute
- ▶ **3 Beziehungen & Kardinalitäten**
- 4 Erweiterte Konzepte
- 5 Notationen im Vergleich
- 6 Modellierungspraxis
- 7 Zusammenfassung

## Definition:

Eine **Beziehung** (Relationship) beschreibt eine Verbindung zwischen Entitäten.

## Beziehungstyp:

Menge gleichartiger Beziehungen  
(z.B. "Spieler spielt für Verein")

## Darstellung:

Raute (Diamond) zwischen Entitäten



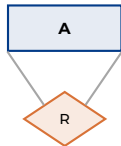
*"Müller spielt für Bayern"*

## Binär (Grad 2): Zwei Entitätstypen



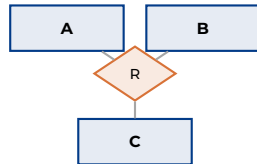
Spieler – Verein

## Unär (Grad 1): Ein Entitätstyp



Mitarbeiter – Vorgesetzter

## Ternär (Grad 3): Drei Entitätstypen

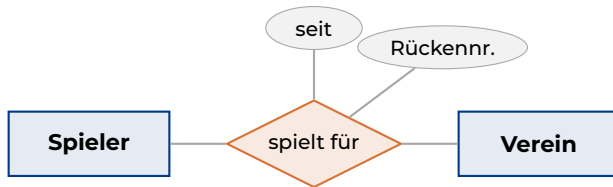


Projekt – Mitarbeiter – Rolle

## Praxis

Binäre Beziehungen sind am häufigsten. Ternäre lassen sich oft in binäre zerlegen.

Auch Beziehungen können Attribute haben:



## Typische Beziehungsattribute:

- Zeitangaben (seit, bis, Datum)
- Mengen (Anzahl, Stückzahl)
- Rollen (Position, Funktion)

Wie viele Entitäten auf der einen Seite  
können mit wie vielen auf der anderen verbunden  
sein?

- **1:1** – Eins zu Eins
- **1:N** – Eins zu Viele
- **M:N** – Viele zu Viele

**Jede Entität auf einer Seite ist mit höchstens einer auf der anderen verbunden.**



## Beispiele:

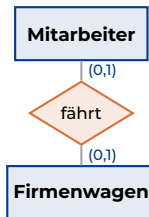
- Land – Hauptstadt
- Person – Personalausweis
- Mitarbeiter – Parkplatz (falls jeder max. einen hat)

## Hinweis

1:1-Beziehungen sind selten. Oft könnte man die Entitäten zusammenlegen.

## Gründe für separate Entitäten trotz 1:1:

- 1 **Optionalität:**  
Nicht jeder Mitarbeiter hat Firmenwagen
- 2 **Unterschiedliche Lebenszyklen:**  
Personalausweis wird erneuert
- 3 **Viele Attribute:**  
Trennung für Übersichtlichkeit



*Nicht jeder hat einen,  
nicht jeder Wagen ist zugeteilt*

Eine Entität auf einer Seite kann mit vielen auf der anderen verbunden sein.



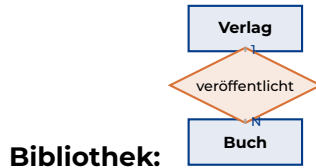
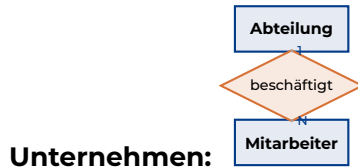
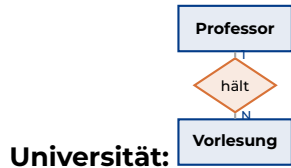
## Beispiele:

- Verein – Spieler (ein Spieler, ein Verein)
- Abteilung – Mitarbeiter
- Kunde – Bestellungen
- Mutter – Kinder

## Häufigste Kardinalität!

Die meisten Beziehungen in der Praxis sind 1:N.





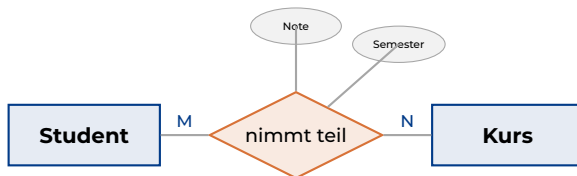
**Entitäten auf beiden Seiten können mit vielen auf der anderen verbunden sein.**



## Beispiele:

- Student – Kurs (viele Studenten, viele Kurse)
- Autor – Buch (Co-Autoren)
- Schauspieler – Film
- Produkt – Zutat (Rezepte)

## M:N-Beziehungen haben oft eigene Attribute:



## Weitere Beispiele mit Attributen:

- Mitarbeiter – Projekt: *Rolle, Stunden, Zeitraum*
- Schauspieler – Film: *Rolle, Gage*
- Lieferant – Produkt: *Preis, Lieferzeit*

## Wichtig

M:N-Beziehungen werden bei der Transformation zu eigenen Tabellen!

## Genauere Angabe der Kardinalitäten:



## Bedeutung:

- (11,30) bei Verein: Ein Verein hat 11 bis 30 Spieler
- (1,1) bei Spieler: Ein Spieler gehört zu genau 1 Verein

## Spezialfälle:

- (0,1) – optional, höchstens einer
- (1,1) – genau einer (Pflicht)
- (0,N) oder (0,\*) – optional, beliebig viele
- (1,N) oder (1,\*) – mindestens einer, beliebig viele

## Bestimmen Sie die Kardinalität:

### ① Buch – Verlag

- Ein Buch erscheint bei einem Verlag
- Ein Verlag veröffentlicht viele Bücher
- $\Rightarrow$

## Bestimmen Sie die Kardinalität:

### ① Buch – Verlag

- Ein Buch erscheint bei einem Verlag
- Ein Verlag veröffentlicht viele Bücher
- $\Rightarrow$  1:N (Verlag:Buch)

## Bestimmen Sie die Kardinalität:

### ① Buch – Verlag

- Ein Buch erscheint bei einem Verlag
- Ein Verlag veröffentlicht viele Bücher
- $\Rightarrow$  1:N (Verlag:Buch)

### ② Patient – Arzt

- Ein Patient kann mehrere Ärzte haben
- Ein Arzt behandelt mehrere Patienten
- $\Rightarrow$

## Bestimmen Sie die Kardinalität:

### ① Buch – Verlag

- Ein Buch erscheint bei einem Verlag
- Ein Verlag veröffentlicht viele Bücher
- $\Rightarrow$  1:N (Verlag:Buch)

### ② Patient – Arzt

- Ein Patient kann mehrere Ärzte haben
- Ein Arzt behandelt mehrere Patienten
- $\Rightarrow$  M:N



## Bestimmen Sie die Kardinalität:

### ① Buch – Verlag

- Ein Buch erscheint bei einem Verlag
- Ein Verlag veröffentlicht viele Bücher
- $\Rightarrow$  1:N (Verlag:Buch)

### ② Patient – Arzt

- Ein Patient kann mehrere Ärzte haben
- Ein Arzt behandelt mehrere Patienten
- $\Rightarrow$  M:N

### ③ Auto – Kennzeichen (in Deutschland)

- Ein Auto hat genau ein Kennzeichen
- Ein Kennzeichen gehört zu genau einem Auto
- $\Rightarrow$

## Bestimmen Sie die Kardinalität:

### ① Buch – Verlag

- Ein Buch erscheint bei einem Verlag
- Ein Verlag veröffentlicht viele Bücher
- $\Rightarrow$  1:N (Verlag:Buch)

### ② Patient – Arzt

- Ein Patient kann mehrere Ärzte haben
- Ein Arzt behandelt mehrere Patienten
- $\Rightarrow$  M:N

### ③ Auto – Kennzeichen (in Deutschland)

- Ein Auto hat genau ein Kennzeichen
- Ein Kennzeichen gehört zu genau einem Auto
- $\Rightarrow$  1:1

## Bestimmen Sie die Kardinalität:

### ④ Ehepartner – Ehepartner (aktuell, monogam)

- Eine Person hat höchstens einen Ehepartner
- $\Rightarrow$

## Bestimmen Sie die Kardinalität:

- ④ Ehepartner – Ehepartner (aktuell, monogam)
- Eine Person hat höchstens einen Ehepartner
  - $\Rightarrow$  **1:1 unär (rekursiv)**

## Bestimmen Sie die Kardinalität:

### 4 Ehepartner – Ehepartner (aktuell, monogam)

- Eine Person hat höchstens einen Ehepartner
- $\Rightarrow$  **1:1 unär (rekursiv)**

### 5 Rezept – Zutat

- Ein Rezept enthält mehrere Zutaten
- Eine Zutat kommt in mehreren Rezepten vor
- $\Rightarrow$

## Bestimmen Sie die Kardinalität:

### 4 Ehepartner – Ehepartner (aktuell, monogam)

- Eine Person hat höchstens einen Ehepartner
- $\Rightarrow$  **1:1 unär (rekursiv)**

### 5 Rezept – Zutat

- Ein Rezept enthält mehrere Zutaten
- Eine Zutat kommt in mehreren Rezepten vor
- $\Rightarrow$  **M:N** (mit Attribut "Menge"!)

## Bestimmen Sie die Kardinalität:

### ④ Ehepartner – Ehepartner (aktuell, monogam)

- Eine Person hat höchstens einen Ehepartner
- $\Rightarrow$  **1:1 unär (rekursiv)**

### ⑤ Rezept – Zutat

- Ein Rezept enthält mehrere Zutaten
- Eine Zutat kommt in mehreren Rezepten vor
- $\Rightarrow$  **M:N** (mit Attribut "Menge"!)

### ⑥ Mitarbeiter – Vorgesetzter

- Ein Mitarbeiter hat einen Vorgesetzten
- Ein Vorgesetzter hat mehrere Mitarbeiter
- $\Rightarrow$

## Bestimmen Sie die Kardinalität:

### 4 Ehepartner – Ehepartner (aktuell, monogam)

- Eine Person hat höchstens einen Ehepartner
- $\Rightarrow$  **1:1 unär (rekursiv)**

### 5 Rezept – Zutat

- Ein Rezept enthält mehrere Zutaten
- Eine Zutat kommt in mehreren Rezepten vor
- $\Rightarrow$  **M:N** (mit Attribut "Menge"!)

### 6 Mitarbeiter – Vorgesetzter

- Ein Mitarbeiter hat einen Vorgesetzten
- Ein Vorgesetzter hat mehrere Mitarbeiter
- $\Rightarrow$  **1:N unär (rekursiv)**

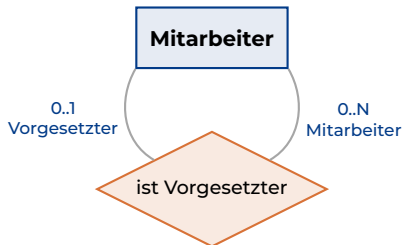


# Pause

15 Minuten

- 1 Rückblick & Motivation
- 2 Entitäten & Attribute
- 3 Beziehungen & Kardinalitäten
- ▶ **4 Erweiterte Konzepte**
- 5 Notationen im Vergleich
- 6 Modellierungspraxis
- 7 Zusammenfassung

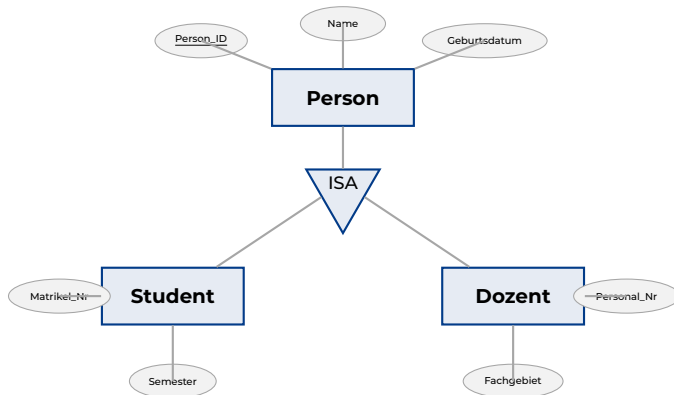
**Eine Entität steht in Beziehung zu sich selbst:**



**Weitere Beispiele rekursiver Beziehungen:**

- Person – “ist verheiratet mit” – Person (1:1)
- Produkt – “ist Ersatzteil für” – Produkt (M:N)
- Kategorie – “ist Unterkategorie von” – Kategorie (1:N)

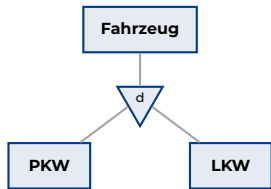
## Vererbungsbeziehung zwischen Entitätstypen:



**Bedeutung:** Student *ist eine* Person, Dozent *ist eine* Person

## Disjoint (disjunkt):

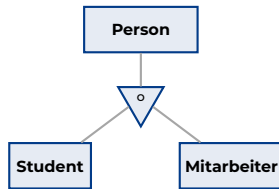
- Entität gehört zu **höchstens einem** Subtyp
- Markierung: “d” im Dreieck



Ein Fahrzeug ist PKW **oder** LKW

## Overlapping (überlappend):

- Entität kann zu **mehreren** Subtypen gehören
- Markierung: “o” im Dreieck

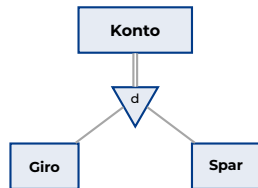


HiWi ist Student **und** Mitarbeiter

# ISA: Total vs. Partial

## Total (vollständig):

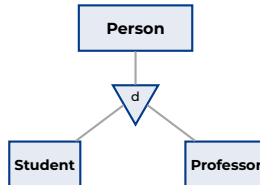
- Jede Entität gehört zu **mindestens einem** Subtyp
- Doppelte Linie zum ISA



Jedes Konto **muss** Giro oder Spar sein

## Partial (partiell):

- Entität muss **nicht** zu einem Subtyp gehören
- Einfache Linie zum ISA



Person kann auch "nur Person" sein

## Kombination

Die vier Varianten kombinieren: **disjoint/total**, **disjoint/partial**, **overlapping/total**, **overlapping/partial**

- 1 Rückblick & Motivation
- 2 Entitäten & Attribute
- 3 Beziehungen & Kardinalitäten
- 4 Erweiterte Konzepte
- **5 Notationen im Vergleich**
- 6 Modellierungspraxis
- 7 Zusammenfassung

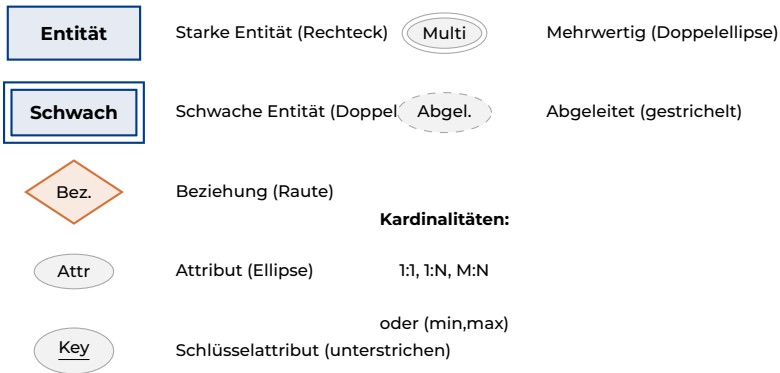
## Es gibt verschiedene ER-Notationen:

Notation	Erfinder/Quelle	Einsatz
Chen-Notation	Peter Chen (1976)	Lehre, Theorie
Crow's Foot (Min,Max)	verschiedene verschiedene	UML, Praxis-Tools Präzise Constraints
UML-Klassendiagramm	OMG	Software Engineering





### Wichtig

Alle Notationen drücken dasselbe aus – nur die Symbole unterscheiden sich!





**In Praxis-Tools (ERwin, Visio, MySQL Workbench) weit verbreitet:**

Symbol	Bedeutung	Chen
	Genau eins (mandatory)	(1,1)
	Null oder eins (optional)	(0,1)
	Eins oder mehr (mandatory)	(1,N)
	Null oder mehr (optional)	(0,N)

## Dieselbe Beziehung in beiden Notationen:

### Chen-Notation:



Ein Verein hat viele Spieler,  
ein Spieler gehört zu einem Verein.

### Crow's Foot Notation:

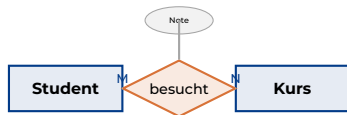


Dieselbe Semantik,  
kompaktere Darstellung.

## Empfehlung

In der Klausur: Chen-Notation verwenden (expliziter, weniger fehleranfällig).  
In der Praxis: Notation des verwendeten Tools nutzen.

## Chen-Notation:



## Crow's Foot (mit Assoziationstabelle):



## Wichtiger Unterschied:

- Chen: M:N als Beziehung mit Attributen
- Crow's Foot: M:N wird zu Assoziations-Entität aufgelöst

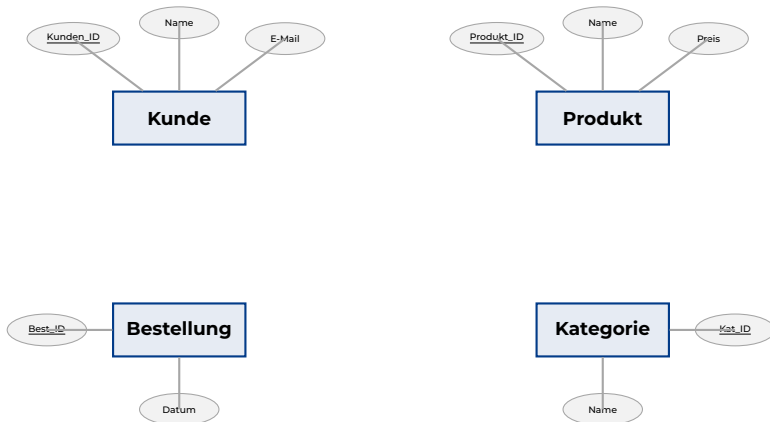
- 1 Rückblick & Motivation
- 2 Entitäten & Attribute
- 3 Beziehungen & Kardinalitäten
- 4 Erweiterte Konzepte
- 5 Notationen im Vergleich
- ▶ **6 Modellierungspraxis**
- 7 Zusammenfassung

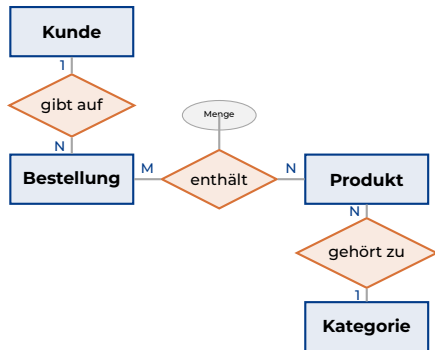
## Anforderungsbeschreibung:

*“Ein Online-Shop verkauft Produkte an Kunden. Jeder Kunde hat einen Namen und eine E-Mail-Adresse. Produkte haben einen Namen, einen Preis und gehören zu einer Kategorie. Kunden können Bestellungen aufgeben, die mehrere Produkte enthalten können.”*

## Schritt 1: Entitäten identifizieren

- Kunde
- Produkt
- Kategorie
- Bestellung

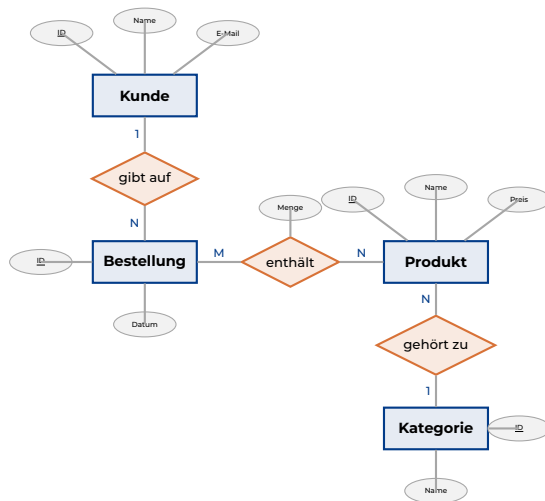


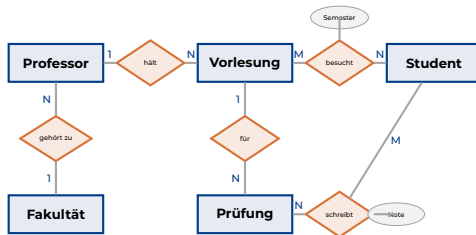


## Kardinalitäten:

- Kunde – Bestellung: 1:N (ein Kunde, viele Bestellungen)
- Bestellung – Produkt: M:N (mit Attribut “Menge”)
- Produkt – Kategorie: N:1 (viele Produkte, eine Kategorie)







**Beziehungen:** Professor hält Vorlesungen (1:N), Studenten besuchen Vorlesungen (M:N), Studenten schreiben Prüfungen zu Vorlesungen (M:N mit Attribut Note)

## Fehler vermeiden:

### **X Kardinalität verwechselt**

- “N bei Kunde” statt “N bei Bestellung”
- Tipp: Immer prüfen – “Ein X hat wie viele Y?”

### **X Attribut statt Entität**

- “Kategorienname” als Attribut von Produkt
- Tipp: Hat es eigene Attribute/Beziehungen?

### **X M:N nicht erkannt**

- “Student hat einen Kurs” (falsch!)
- Tipp: Beide Richtungen prüfen

## Noch mehr Fehler:

### **X Schlüssel vergessen**

- Jede Entität braucht Schlüsselattribut

### **X Beziehungsattribute vergessen**

- “Menge” bei Bestellung-Produkt
- “Note” bei Student-Prüfung

### **X Redundante Beziehungen**

- Direkter Pfad reicht meist
- Keine “Abkürzungen” einbauen

## Prozess:

- 1 Substantive → Entitäten
- 2 Verben → Beziehungen
- 3 Adjektive → Attribute
- 4 Kardinalitäten bestimmen
- 5 Schlüssel festlegen
- 6 Review mit Stakeholdern

## Qualitätskriterien:

- ✓ Vollständig (alle Anforderungen)
- ✓ Redundanzfrei
- ✓ Konsistent  
(Namenskonventionen)
- ✓ Lesbar (gutes Layout)
- ✓ Dokumentiert (Glossar)

## Goldene Regel

**Ein gutes ER-Modell ist für Fachexperten verständlich!**

Wenn der Kunde es nicht versteht, ist es zu kompliziert.

# Hands-on

Eigenes ER-Diagramm erstellen

Papier oder draw.io

Aufgaben 6.1 – 6.4

## Erstellen Sie ein ER-Diagramm für eine Bibliothek:

*“Eine Bibliothek verleiht Bücher an Mitglieder. Jedes Buch hat einen Titel, ISBN und einen oder mehrere Autoren. Mitglieder haben einen Namen und eine Mitgliedsnummer. Ein Buch kann mehrfach vorhanden sein (Exemplare). Mitglieder können Bücher ausleihen.”*

### Hinweise:

- Identifizieren Sie die Entitäten
- Bestimmen Sie die Attribute (inkl. Schlüssel)
- Zeichnen Sie die Beziehungen mit Kardinalitäten
- Achtung: Buch vs. Exemplar!

## Erstellen Sie ein ER-Diagramm für eine Universität:

*“Professoren halten Vorlesungen. Studierende besuchen Vorlesungen und schreiben Prüfungen. Jede Prüfung gehört zu einer Vorlesung. Studierende können mehrere Prüfungen ablegen.”*

### Entitäten:

- Professor
- Vorlesung
- Student
- Prüfung (?)

**Diskussion:** Ist “Prüfung” eine Entität oder eine Beziehung?

## Erstellen Sie ein ER-Diagramm für ein Fitness-Studio:

*“Ein Fitness-Studio bietet Kurse an, die von Trainern geleitet werden. Mitglieder können sich für Kurse anmelden. Kurse finden in bestimmten Räumen statt. Trainer können mehrere Kurse leiten. Mitglieder haben verschiedene Mitgliedschaftstypen (Basic, Premium, VIP).”*

### Besondere Herausforderungen:

- Ist “Mitgliedschaftstyp” eine Entität oder Attribut?
- Welche Beziehungsattribute gibt es?
- Verwenden Sie (min,max)-Notation für Präzision



## Lieferplattform (aus Klausur SS2025):

*“Ein Unternehmen betreibt eine Plattform, über die Restaurants Mahlzeiten liefern. Restaurants haben Kennung, Name und Adresse. Gerichte gehören zu Restaurants und haben Preis, Kalorien und Allergene. Kunden haben Kennung, Kontaktdaten und Unverträglichkeiten. Bestellungen enthalten mehrere Positionen (Gericht + Menge).”*

### Aufgabe:

- Vollständiges ER-Diagramm
- (Min,Max)-Kardinalitäten
- Alle Attribute mit Schlüsseln

- 1 Rückblick & Motivation
- 2 Entitäten & Attribute
- 3 Beziehungen & Kardinalitäten
- 4 Erweiterte Konzepte
- 5 Notationen im Vergleich
- 6 Modellierungspraxis
- **7 Zusammenfassung**

## ER-Modellierung:

- Grafische Darstellung von Datenstrukturen
- Unabhängig von der Implementierung
- Kommunikationsmittel mit Fachexperten

## Kernelemente:

- **Entitäten** – “Dinge”
- **Attribute** – Eigenschaften
- **Beziehungen** – Verbindungen
- **Kardinalitäten** – Anzahl

## Modellierungsschritte:

- ① Entitäten identifizieren
- ② Attribute zuordnen
- ③ Schlüssel festlegen
- ④ Beziehungen definieren
- ⑤ Kardinalitäten bestimmen

## Erweiterte Konzepte:

- Schwache Entitäten
- ISA-Hierarchien
- Rekursive Beziehungen

## Vollständigkeit:

- ☐ Alle Entitäten aus Anforderung?
- ☐ Alle Attribute zugeordnet?
- ☐ Alle Beziehungen modelliert?
- ☐ Schlüsselattribute markiert?

## Korrektheit:

- ☐ Kardinalitäten korrekt?
- ☐ Keine redundanten Beziehungen?
- ☐ Attribute am richtigen Ort?

## Qualität:

- ☐ Einheitliche Namensgebung?
- ☐ Übersichtliches Layout?
- ☐ Keine kreuzenden Linien?

## Validierung:

- ☐ Beispieldaten durchspielen
- ☐ Fachexperten befragen
- ☐ Grenzfälle prüfen

## Vorlesung 7: Relationales Modell & Transformation

- ER-Modell → Relationales Schema
- Transformationsregeln
- CREATE TABLE mit Schlüsseln
- Fremdschlüssel-Constraints



# Fragen?

[christoph.flath@uni-wuerzburg.de](mailto:christoph.flath@uni-wuerzburg.de)