

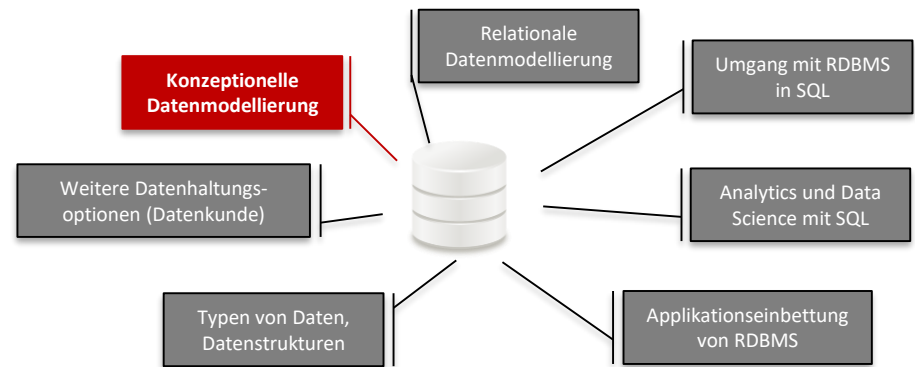
**Prof. Dr. Peer Küppers**

**DSCB140**

**Datenbanken und Datenkunde I**

VL2 – Konzeptionelle Modellierung

## Konzeptionelle Datenmodellierung



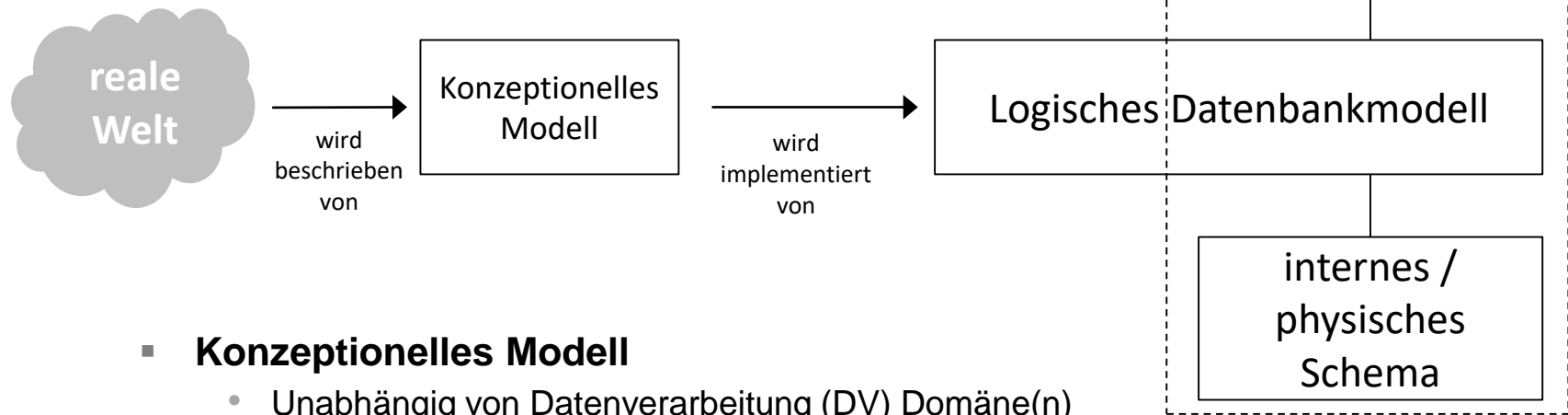
- Das grundlegende Vorgehen in der Datenmodellierung von Abläufen der „realen Welt“ bis hin zu Datenstrukturen in einer Datenbank kennenlernen.
- Das Entity-Relationship-Model (ERM) als eine verbreitete Modellierungssprache zur konzeptionellen Datenmodellierung kennenlernen und anwenden können.
- Die Anwendung fortgeschrittener Modellierungskonzepte in ERM erlernen.

# Agenda

- **Grundlagen der Datenmodellierung und Datenbanktypen**
- ☐ **Konzeptionelle Modellierung mit ERM – Grundlagen**
- ☐ **Konzeptionelle Modellierung mit eERM – fortgeschrittene Konzepte**
- ☐ **Fallstudie**
- ☐ **Zusammenfassung & Ausblick**

# Modellierungskonzepte

## ■ Mehrstufiger Abstraktionsansatz



### ■ Konzeptionelles Modell

- Unabhängig von Datenverarbeitung (DV) Domäne(n) der realen Welt beschreiben (bspw. Lagerverwaltung)
- Bildet die Grundlage für das Datenbanksystem

### ■ Dem konzeptionellen Modell wird ein **logisches Modell** zur Seite gestellt

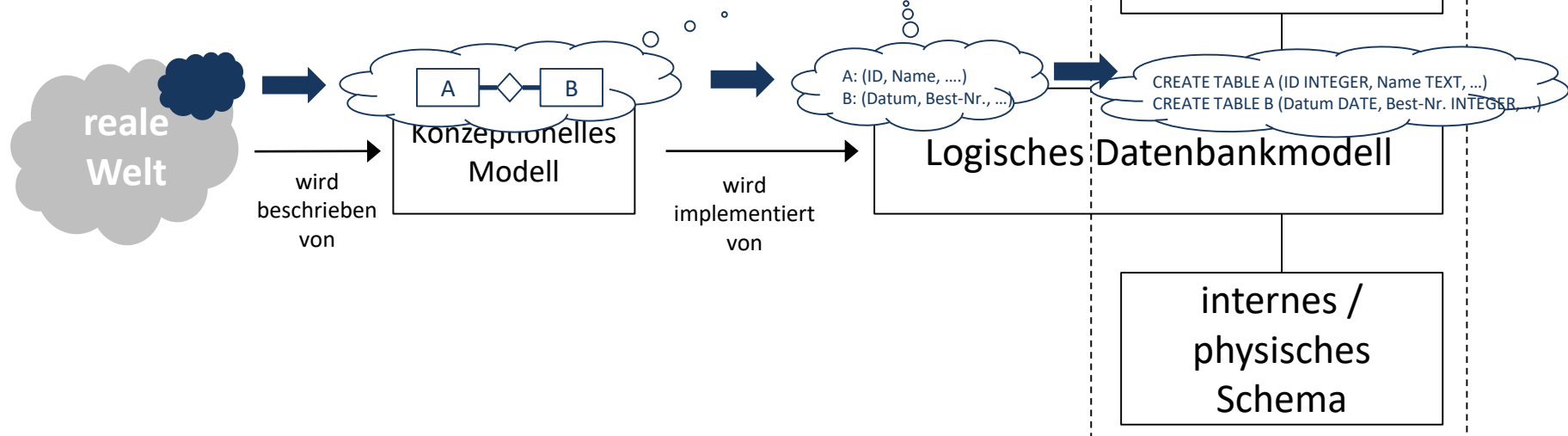
- Abbildung der Gesamtheit der Daten unabhängig von der internen physischen Speicherung, aber dennoch unter Berücksichtigung von Implementierungsaspekten
  - Bspw. kann ein konzeptionelles Modell sowohl als relationales Modell oder durch Verwendung von NoSQL-Ansätzen logisch abgebildet werden.
  - Das logische Datenbankmodell ist nicht gebunden an ein Datenbanksystem (bspw. kann ein relationales Modell auf verschiedenen DBMS wie MySQL und PostgreSQL identisch aufgebaut sein)

# Modellierungskonzepte

Relationales Modell überführt  
in **konkretes** Datenbankschema  
(Tabellen, Spalten, Datentypen,  
Beziehungen, ...)

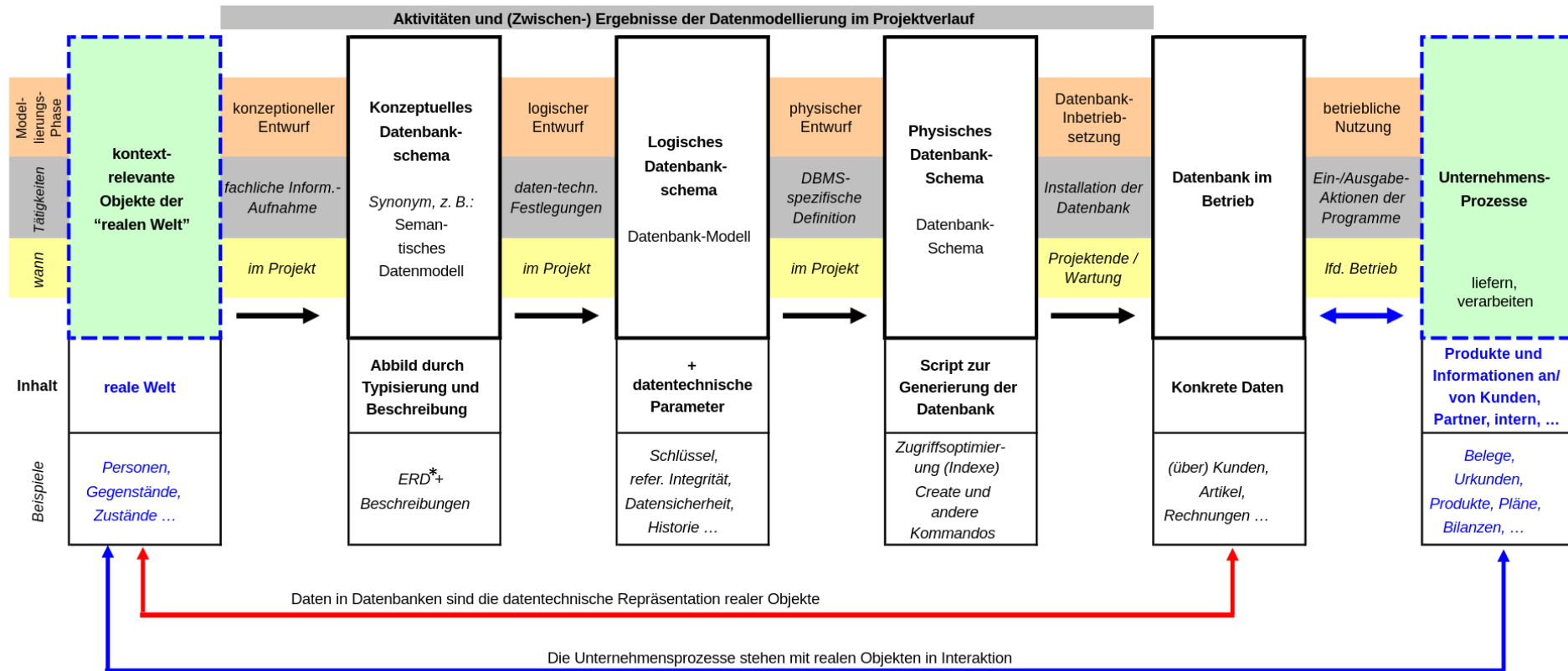
## Beispiele

Klassendiagramm in der UML  
Entity-Relationship-Model (**ERM**)



# Modellierungskonzepte

## Datenmodellieren: Entwicklung von der fachlichen, implementierungsunabhängigen Konzeption bis zur Datenbank



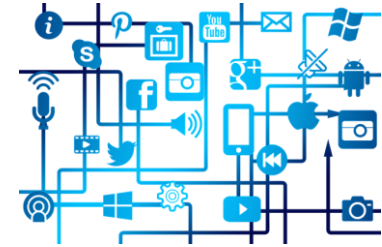
\*Hinweis: ERD = Entity-Relationship-Diagramm = Entity-Relationship-Model = ERM

Quelle: <https://de.wikipedia.org/wiki/Datenmodellierung>

# Arten von DBMS

- Netzwerk, hierarchische und objektorientierte DBMS
  - Nicht durchgesetzt, evtl. Legacy-Systeme (=Altbestand)
- **Relationale DBMS (auch RDBMS)**
  - Marktbeherrschend, z. B. als Grundlage der wichtigen ERP-Systeme (Enterprise Resource Planning → eine Klasse umfassender Unternehmens-Software)
- **NoSQL-Datenbanken**
  - insbesondere bei Anwendungen mit sehr großen Datenmengen, um Skalierung und zu erreichen (z.B. beim Verarbeiten von Sensordaten)
- **Data Warehouse-Systeme** / multidimensionale Datenbanken
  - Strukturell nah an RDBMS
  - z. B. als Basis für Business Intelligence Anwendungen
- **In-Memory-Datenbanken** für hohe Zugriffsgeschwindigkeiten
  - Oftmals auch strukturell nah an RDBMS oder NoSQL-Ansätzen
- **Graphdatenbanken**
  - Abbildung von Graphstrukturen, bspw. soziale Netzwerke

## Weiterführendes Material



- Video zu Modellierungsschritten

<https://www.youtube.com/watch?v=nOmSmTGy39o>



### 13.07 Übersicht über die Modellierungsschritte: von der

Jens Dittrich • 16.000 Aufrufe • vor 4 Jahren

Komplette Liste der Videos und zusätzliches Material auf <http://datenbankenlernen.de> Informatik, Uni Saarland:



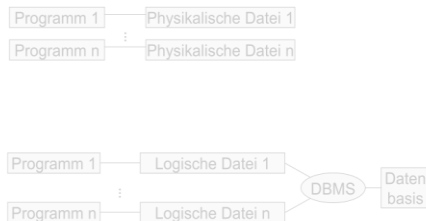
# Agenda

- ☐ Grundlagen der Datenmodellierung und Datenbanktypen
- ☒ Konzeptionelle Modellierung mit ERM – Grundlagen
- ☐ Konzeptionelle Modellierung mit eERM – fortgeschrittene Konzepte
- ☐ Fallstudie
- ☐ Zusammenfassung & Ausblick

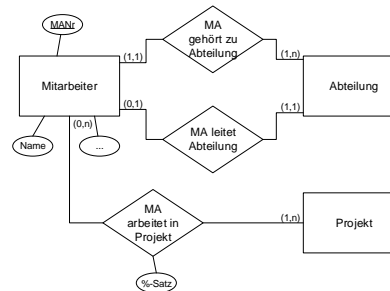
# Themenüberblick



## A Datenbankgrundlagen



## B Konzeptionelle Modellierung



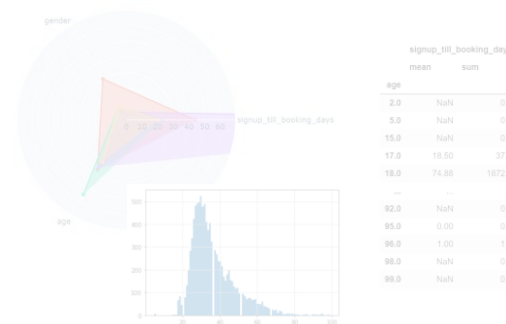
## C Logische Modellierung & Normalisierung

MID	Name	Adresse	Gehalt
1234	Müller	Hofstraße 7	23.000 €
1235	Meyer	Hauptstraße 13b	45.678 €
1236	Schmidt	Kurze Straße 1	55.555 €
1237	Heinze	Neue Straße 34	44.333 €
1238	Mustermann	Musterstraße 8	80.000 €
1239	...	...	...

## D Structured Query Language (SQL)

- filter your columns  
**SELECT** col1, col2, col3, ... **FROM** table1
- filter the rows  
**WHERE** col4 = 1 **AND** col5 = 2
- aggregate the data  
**GROUP** by ...
- limit aggregated data  
**HAVING** count(\*) > 1
- order of the results  
**ORDER BY** col2

## E Analytics mit SQL

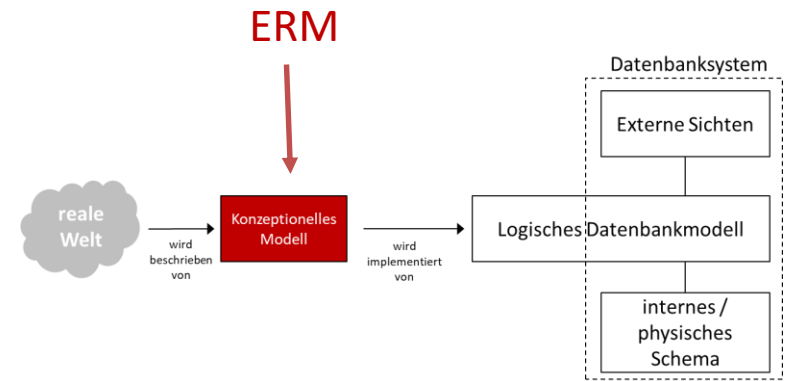


## F Weitere Formen und Grundlagen der Datenhaltung



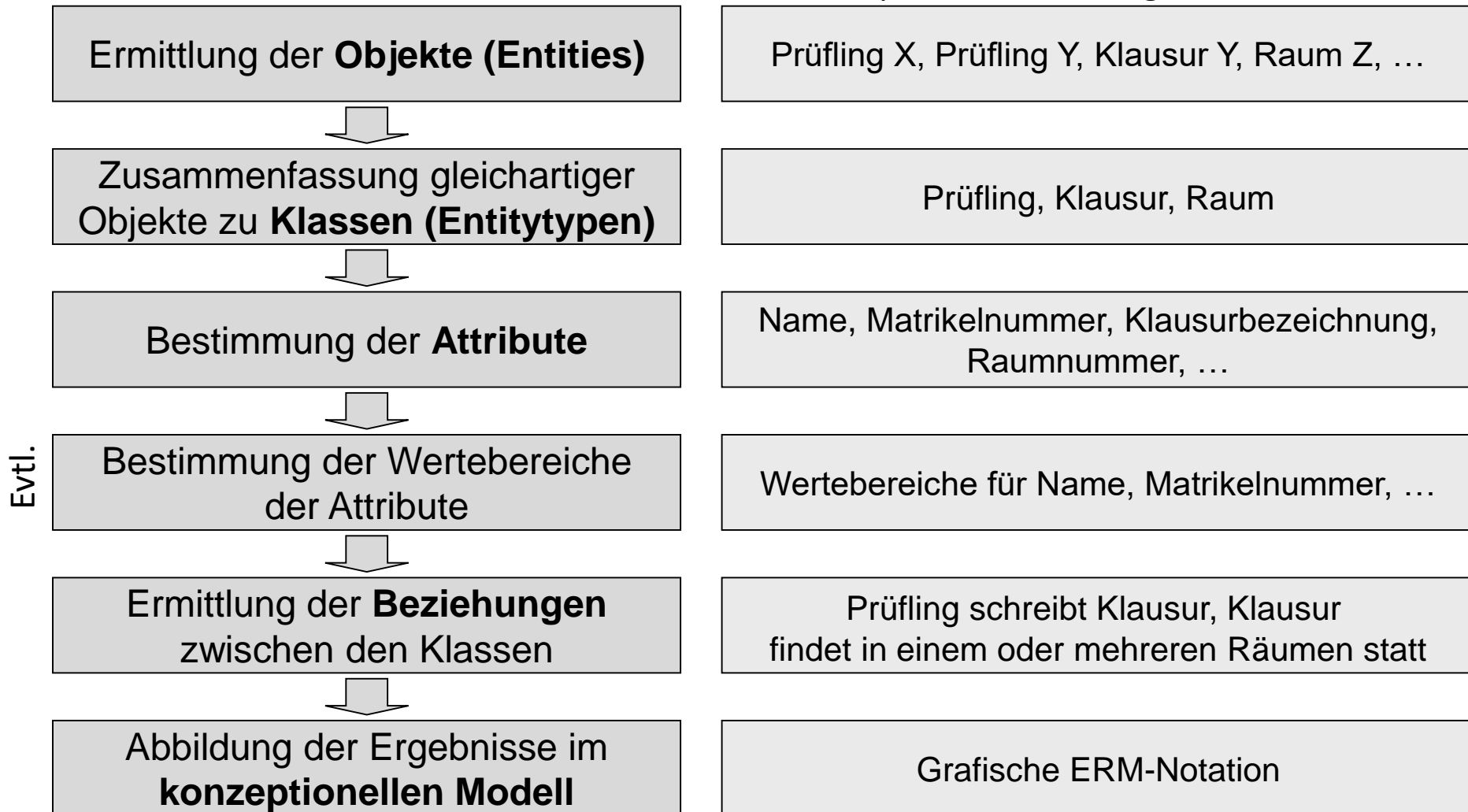
# Entity Relationship Model (ERM)

- **Grafische Notation** zur formellen Darstellung von konzeptionellen Datenmodellen.
- Unabhängig von einem bestimmten Datenbankmanagementsystem und der logischen Modellierung.
- Hält die im Betrachtungsbereich (**Domäne**) relevanten Objekte (**Entities**) und Beziehungen (**Relationships**) zwischen diesen formal fest.
  - Die konkreten Objekte werden abstrakter dargestellt durch Zusammenfassung zu Entitytypen. Auch die konkreten Beziehungen werden abstrahiert (Relationshiptypen). Dazu später mehr.
- Objekte und Beziehungen können durch Eigenschaften (Attribute) näher beschrieben werden.
- Beziehungen zwischen Objekten können genau spezifiziert werden.



# ERM: Schrittweises Vorgehen

## Beispiel: Modellierung von Klausuren



# ERM: Entitytypen und Attribute

## ■ Entitytypen repräsentieren zum Beispiel

- Real existierende Objekte: Person, Auto, Buch
- Ereignisse: Artikelbestellung, Gespräch führen
- Rollen: Mitarbeiter, Abteilungsleiter, ...
- Organisationen,
- Transaktionen: Bestelleingang, Warenausgang, Wareneingang, Rechnungsstellung (vgl. Ereignis)



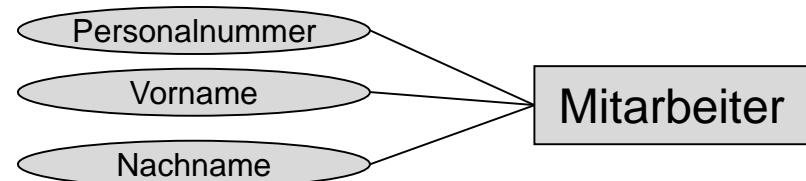
## ■ Attribut

- Weist einem Entity- oder Relationshiptypen Eigenschaften zu.
- Mehrere Attribute können durch mehrere Ellipsen oder durch Komma getrennt in einer Ellipse dargestellt werden.



## ■ Verbindung

- Ungerichtete Kante, verbindet Attribute und Entitytypen

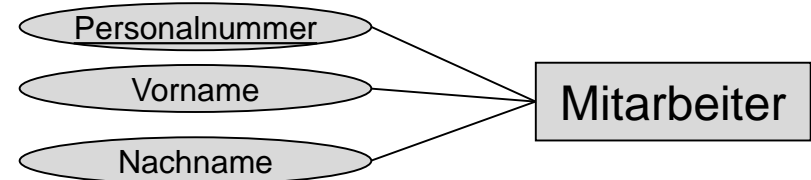


# ERM: Schlüsselattribute

<u>Pers.-Nr.</u>	Vorname	Nachname
1234	Peer	Küppers
5678	Hans	Meier

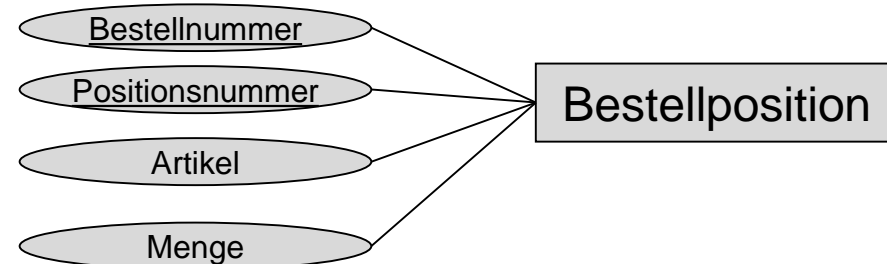
## ■ Schlüssel

- Welches **Attribut** bzw. welche **Kombination aus Attributen** kann ein Objekt (also eine Entität = einer Ausprägung eines Entitätstyps) eindeutig zu identifizieren?



## ■ Primärschlüssel

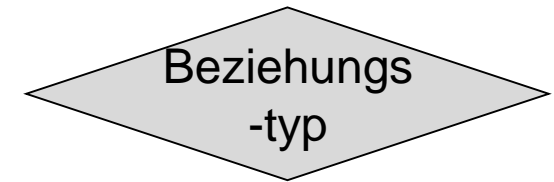
- Identifiziert ein Objekt **eindeutig**
- (Oft künstlicher) Schlüssel (ID), der auch aus mehreren Attributen zusammengesetzt sein kann
- Unterstrichen dargestellt



<u>Best.-Nr.</u>	<u>Pos.-Nr.</u>	Artikel	Menge
100	1	Apfel	10
100	2	Banane	15
200	1	Zitrone	20
200	2	Apfel	40

# ERM: Beziehungstypen und Kardinalitäten

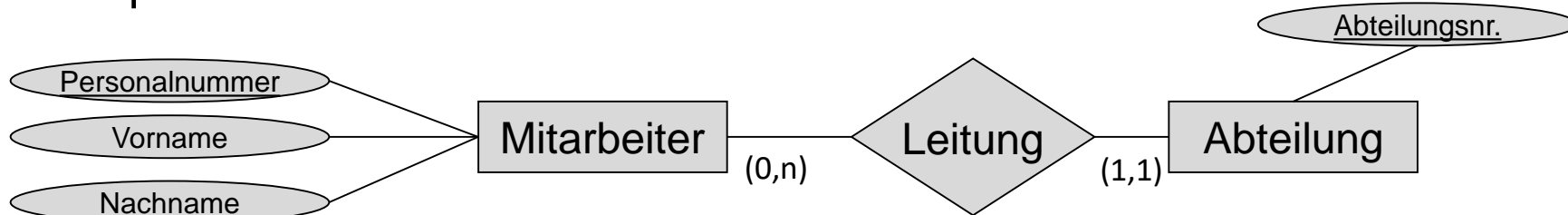
- Relationshiptyp (Beziehungstyp)
  - Logische Zuordnungen bzw. Beziehung zwischen Entitytypen, z.B. Kunde kauft Produkt.



- Kardinalität:
  - Gibt an, wie oft ein Entity eines Entitytypen A eine Beziehung mit einem Entity eines Entitytypen B eingehen kann (minimal / maximal).

(n,m)

- Beispiel:

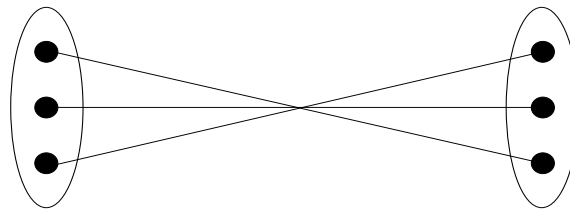


# Kardinalitäten in der min-max Notation (1/2)

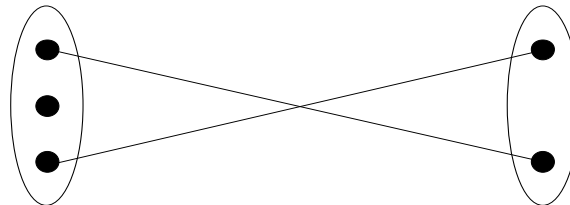
- Relationship-Typen werden durch die Angabe ergänzt, wie oft jedes Entity darin vertreten sein kann

1. Menge

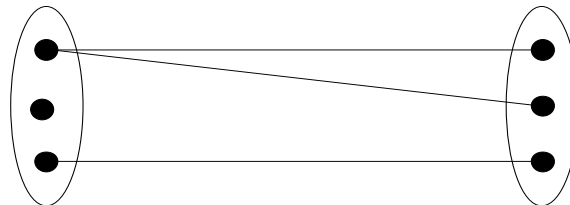
2. Menge



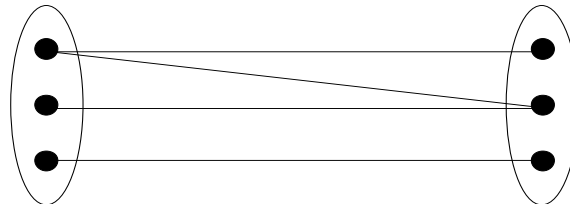
(1,1) : (1,1) - Beziehung



(0,1) : (1,1) - Beziehung



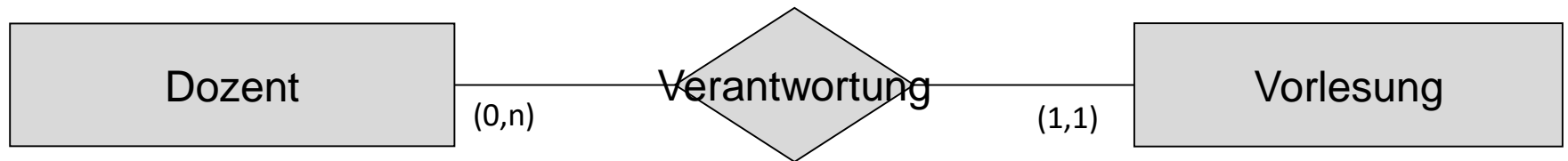
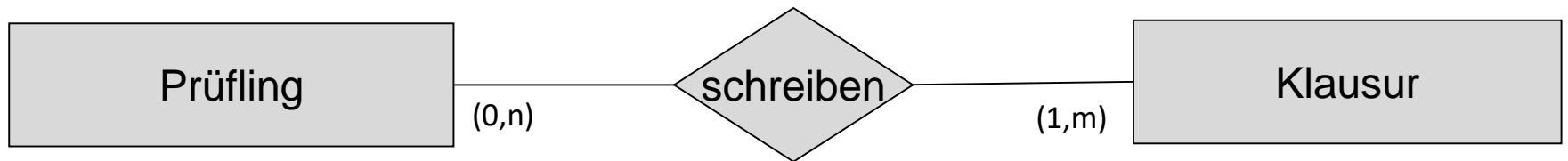
(0,n) : (1,1) - Beziehung



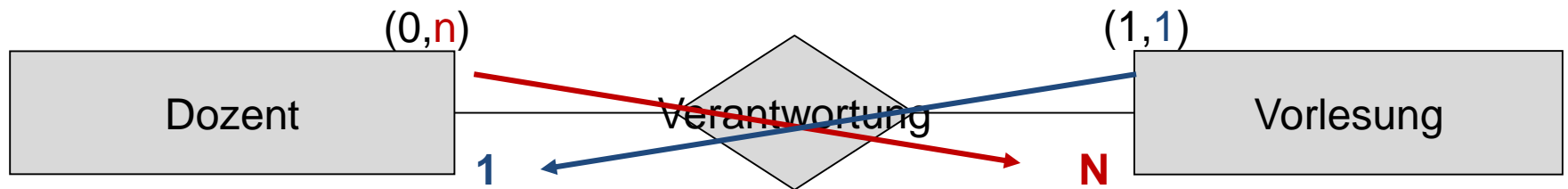
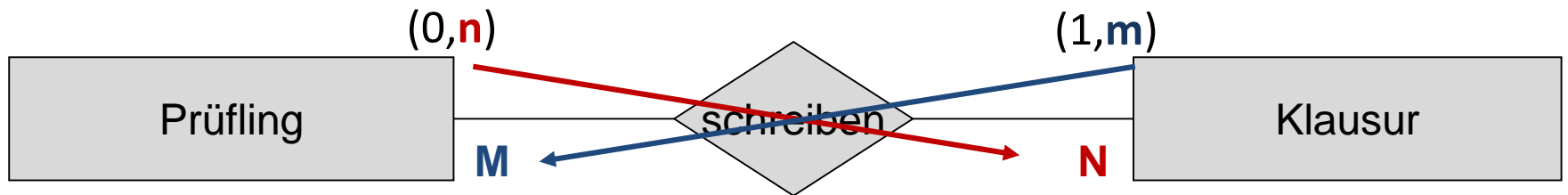
(1,n) : (1,m) - Beziehung



## Kardinalitäten in der min-max Notation (2/2)

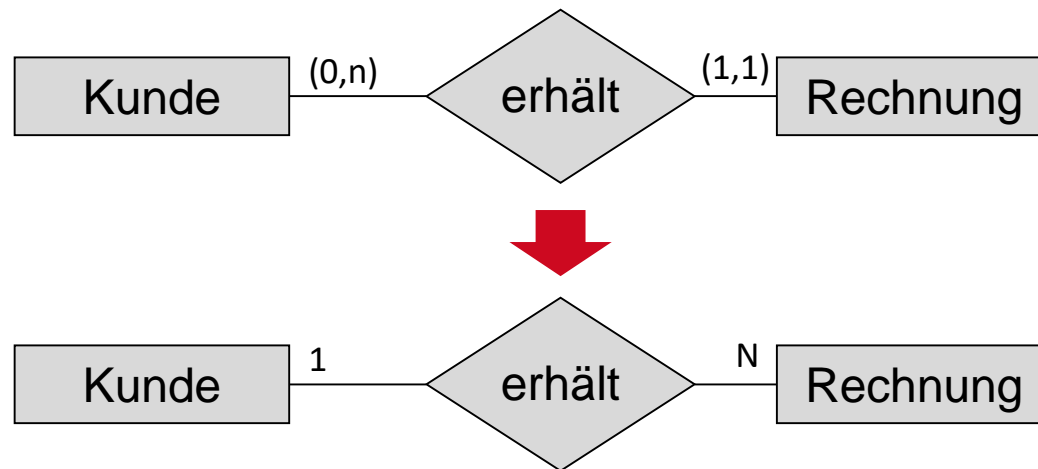


# Kardinalitäten in der vereinfachten Notation nach Chen (1/2)



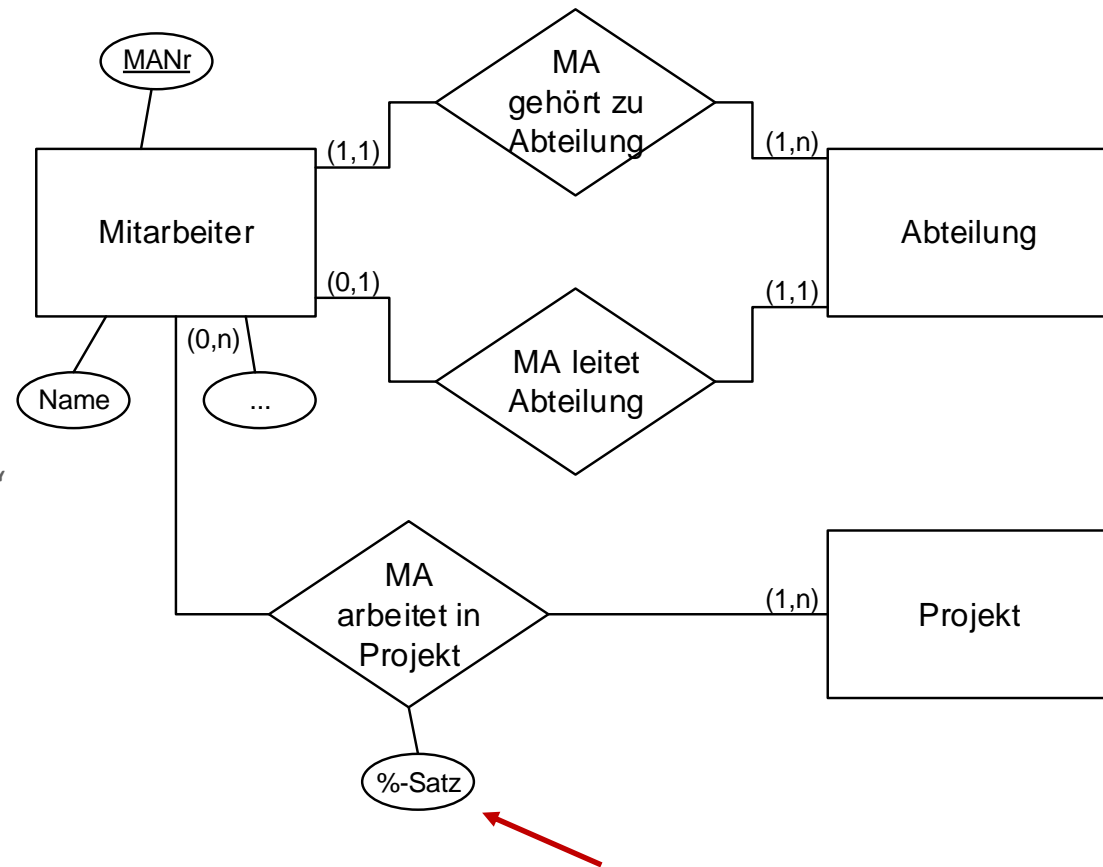
## Kardinalitäten in der vereinfachten Notation nach Chen (2/2)

- Die Chen-Notation ist semantisch ärmer als die (min,max)-Notation.
  - Sie lässt keine Aussagen über die minimale Anzahl korrespondierender Entitäten zu.
  - Dies ist bei der Umwandlung in ein logisches Modell u.U. von Bedeutung.
  - Wir präferieren in diesem Kurs daher die (min,max)-Notation



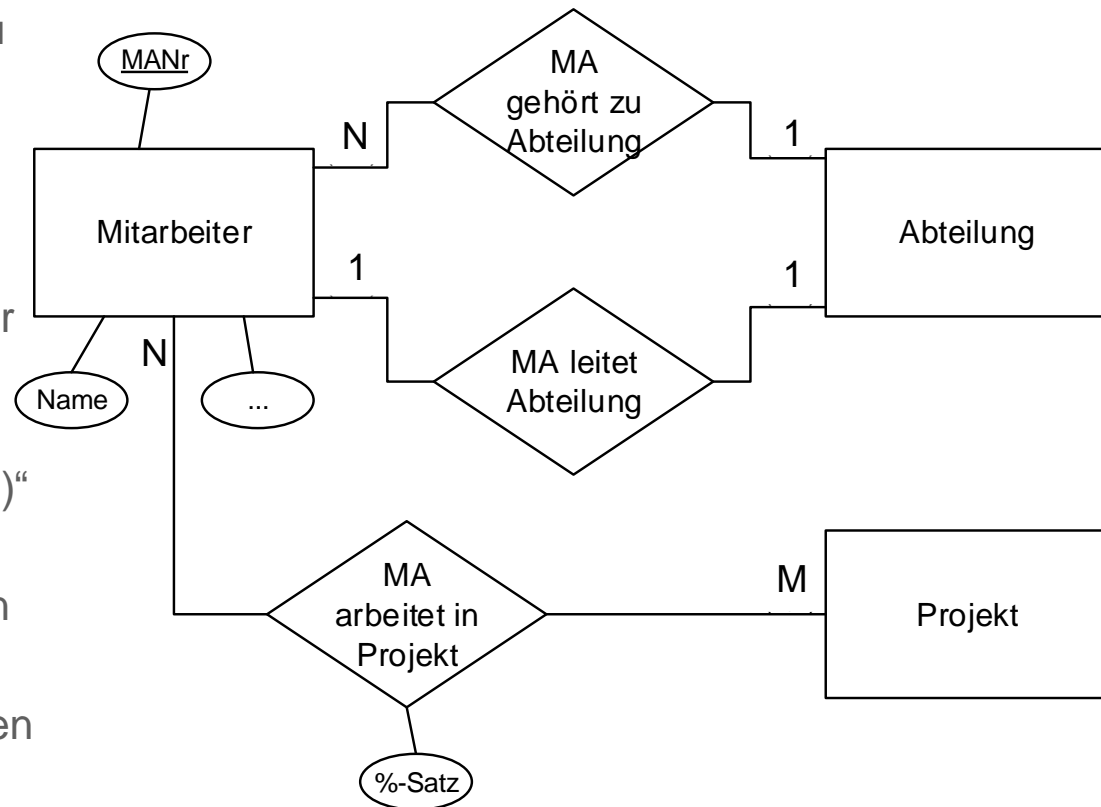
# Kardinalitäten: Beispiel (min-max Notation)

- Ein Mitarbeiter gehört zu genau einer Abteilung: „(1,1)“
- In einer Abteilung arbeiten mindestens ein Mitarbeiter, maximal n Mitarbeiter: „(1,n)“
- Ein Mitarbeiter kann Leiter einer Abteilung sein: „(0,1)“
- Eine Abteilung wird von genau einem Mitarbeiter geleitet: „(1,1)“
- Ein Mitarbeiter kann in keinem, einem oder mehreren Projekten arbeiten: „(0,n)“
- Ein Projekt hat mindestens einen aktiven Mitarbeiter: „(1,n)“



# Kardinalitäten: Beispiel (Notation nach Chen)

- Ein Mitarbeiter gehört zu genau einer Abteilung: „(1,1)“
- In einer Abteilung arbeiten mindestens ein Mitarbeiter, maximal n Mitarbeiter: „(1,n)“
- Ein Mitarbeiter kann Leiter einer Abteilung sein: „(0,1)“
- Eine Abteilung wird von genau einem Mitarbeiter geleitet: „(1,1)“
- Ein Mitarbeiter kann in keinem, einem oder mehreren Projekten arbeiten: „(0,n)“
- Ein Projekt hat mindestens einen aktiven Mitarbeiter: „(1,n)“





## Fallbeispiel „Vereinsportal“

Bitte helfen Sie einem Verein bei der Errichtung eines Mitglieder-Portals. Der Vereinsvorstand beschreibt Ihnen die Anforderungen wie folgt:

- In unserem Portal müssen wir die Mitglieder speichern können. Zu jedem Mitglied soll der Name, Geburtsdatum und eine Telefonnummer erfasst werden. Eindeutig identifizieren wir Mitglieder über den Namen und das Geburtsdatum.
  - Wir sind ein lokaler Verein und nehmen bisher nur Mitglieder aus unserer Stadt auf. Daher soll zu jedem Mitglied aus einer Liste der Stadtteil zugeordnet werden können (eindeutig über den Namen des Stadtteils identifizierbar).
  - Ein Mitglied kann eine oder mehrere Rollen im Verein übernehmen. Bei den Rollen haben wir uns RollenIDs zur Identifikation und Bezeichnungen ausgedacht.
- **Modellieren Sie das Fallbeispiel als ERM. Verwenden Sie die (min,max)-Notation für Kardinalitäten.**



## Fallbeispiel „Baumarkt“

Bitte helfen Sie einer Baumarktkette bei der Datenmodellierung folgender Domäne:

- Die „SchraubeLocker AG“ betreibt Baumärkte und unterhält in Deutschland einige Regionalgesellschaften (Nord, Ost, Süd, West).
  - Diese Regionalgesellschaften bestehen jeweils aus mehreren Filialen.
  - Artikel werden in den Filialen angeboten. Nicht alle Artikel sind jedoch in allen Filialen im Angebot.
  - Die Artikel werden jeweils von genau einem Lieferanten bezogen. Jeder Lieferant hat einen oder mehrere Artikel im Angebot.
  - Lieferungen von Artikeln werden mit Lieferdatum gespeichert und gehen direkt an die einzelnen Filialen.
  - Kunden können sich bei ihrer Regionalgesellschaft registrieren lassen und erhalten dann Rabatt.
- **Modellieren Sie das Fallbeispiel als ERM. Verwenden Sie die (min,max)-Notation für Kardinalitäten.**
- **Achtung: man kann dieses Fallbeispiel auf verschiedene Arten lösen! Es gibt nicht das eine richtige Modell.**



## Fallbeispiel „Hochschule“

Bitte helfen Sie der Hochschulverwaltung bei der Datenmodellierung folgender Domäne:

- Wir speichern von Studierenden den Namen und Vornamen. Sie werden über ihre Matrikelnummer identifiziert.
  - Studierende können Vorlesungen hören, die durch eine Vorlesungsnummer eindeutig identifiziert werden. Zusätzlich möchten wir im System hinterlegen, wie viele SWS die Vorlesung umfasst und wie der Titel lautet.
  - Professoren haben eine Personalnummer und wir speichern zusätzlich ihre Namen und Vornamen. Dies ist identisch zu den Wissenschaftlichen Mitarbeitern, die für Professoren arbeiten. Professoren und Mitarbeiter haben einen eigenen Raum in unseren Hochschulgebäuden. Räume können aber auch geteilt werden, weshalb wir neben Raum- und Gebäudenummer auch noch die Anzahl Arbeitsplätze je Raum speichern möchten.
  - Vorlesungen werden von Professoren verantwortet.
  - Studierende werden in Vorlesungen von einem Professor geprüft (nicht zwangsläufig die-/derselbe, die/der die Vorlesung verantwortet). Neben der erreichten Note möchten wir auch das Semester hinterlegen, in dem der/die Student/in geprüft wurde. So können wir eine Historie der Prüfungen aufbauen.
- **Modellieren Sie das Fallbeispiel als ERM. Verwenden Sie die (min,max)-Notation für Kardinalitäten.**
- **Fallen Ihnen fehlende Elemente der Modellierung auf, die zur besseren Beschreibung / Darstellung des Sachverhalts im Hinblick auf die Daten helfen könnten?**

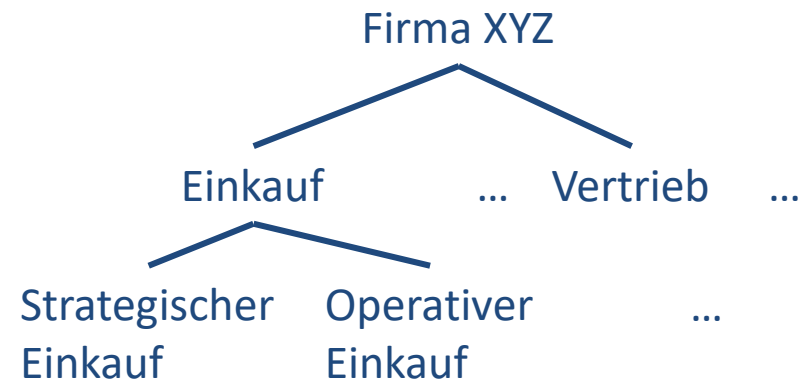
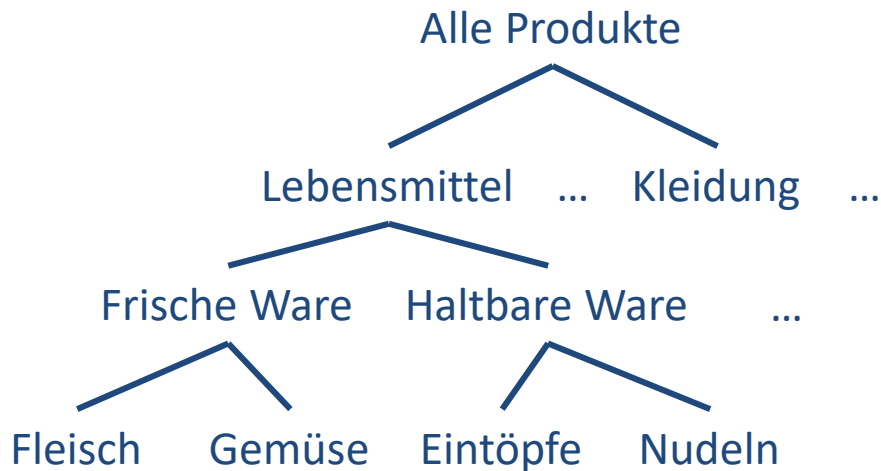


# Agenda

- ☐ Grundlagen der Datenmodellierung und Datenbanktypen
- ☐ Konzeptionelle Modellierung mit ERM – Grundlagen
- ☒ Konzeptionelle Modellierung mit eERM – fortgeschrittene Konzepte
- ☐ Fallstudie
- ☐ Zusammenfassung & Ausblick

# Hierarchien im ERM

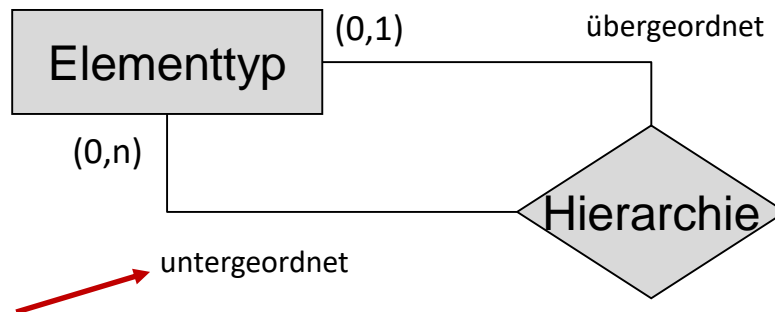
- Wir lernen nun die erweiterte ERM (eERM) Notation kennen.
  - Wir starten mit Hierarchien.
- Hierarchien setzen Entities in eine Über- /Unterordnungsbeziehung.
  - Damit werden Baumstrukturen abgebildet, bspw. Produktgruppenhierarchien, Organigramme, Stücklisten, ...



# Hierarchien im ERM

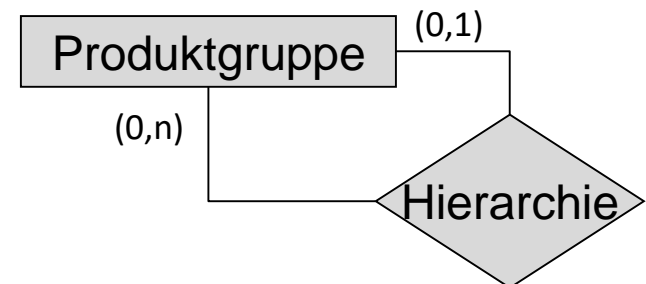
## ■ Hierarchie = Baumstruktur

- Einem Element darf maximal ein anderes Element (gleichen Typs) übergeordnet werden, einem Element können aber beliebig viele Elemente (gleichen Typs) untergeordnet werden.
- Entsprechend lautet die Kardinalität: (0,1) – (0,n)



Optional: Rollenbezeichner können  
die Lesbarkeit erhöhen!

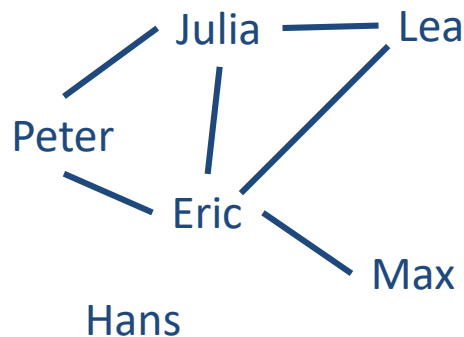
Beispiel:



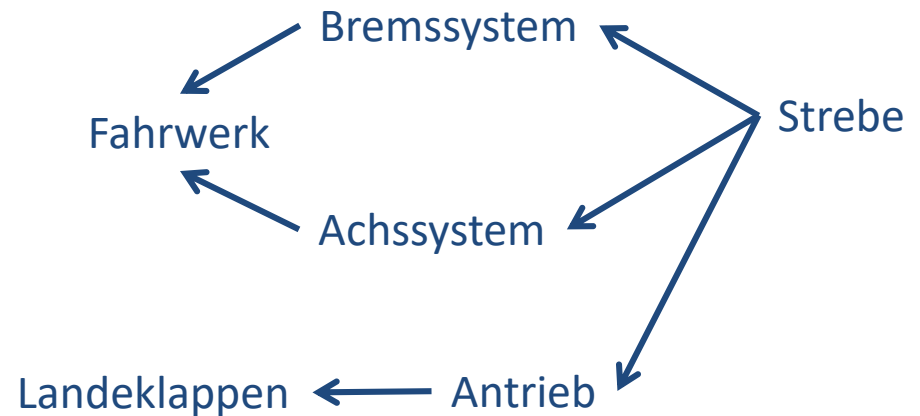
# Strukturen im ERM

- Strukturen im ERM setzen Entity-Typen mit beliebig vielen Entity-Typen gleichen Typs in Verbindung.
  - Beispiele: soziales Netzwerk, Teilestrukturen

**Beispiel „Freundschaften“**  
(ungerichtet)

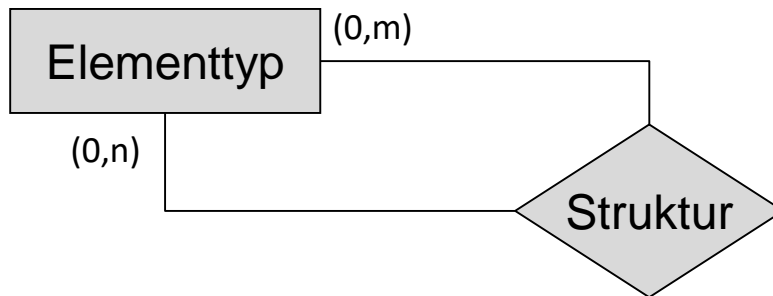


**Beispiel „Teilestruktur“**  
(gerichtet)

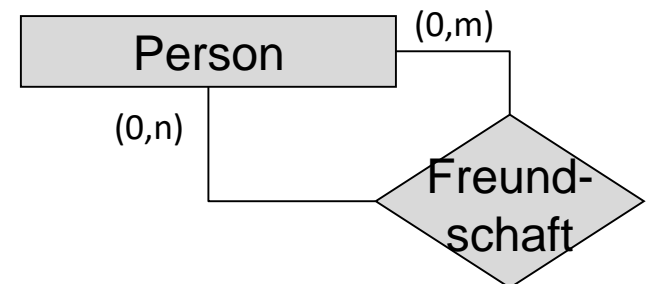


# Strukturen im ERM

- Strukturen im ERM setzen Entity-Typen mit beliebig vielen Entity-Typen gleichen Typs in Verbindung.
  - → Kardinalität:  $(0, n) - (0, m)$

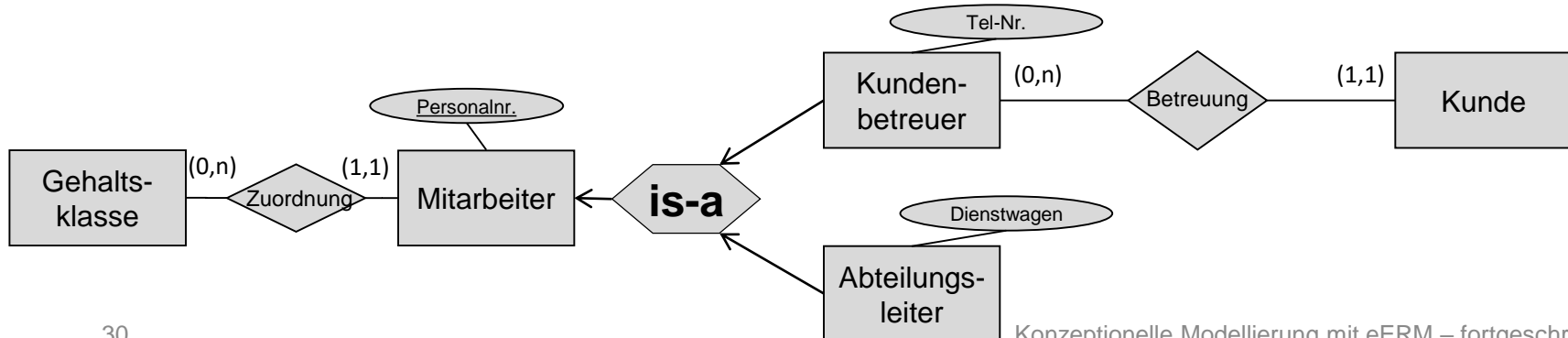
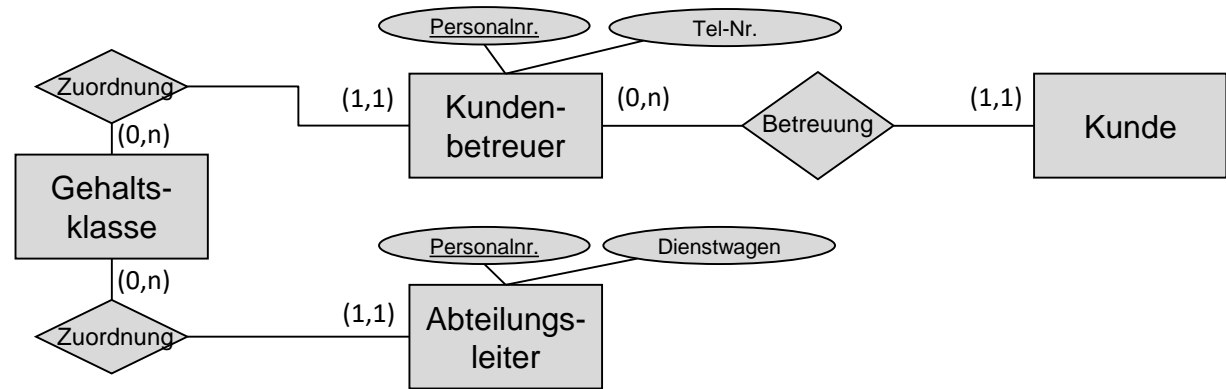
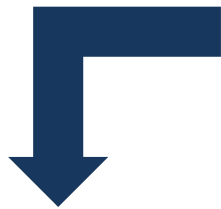


Beispiel:

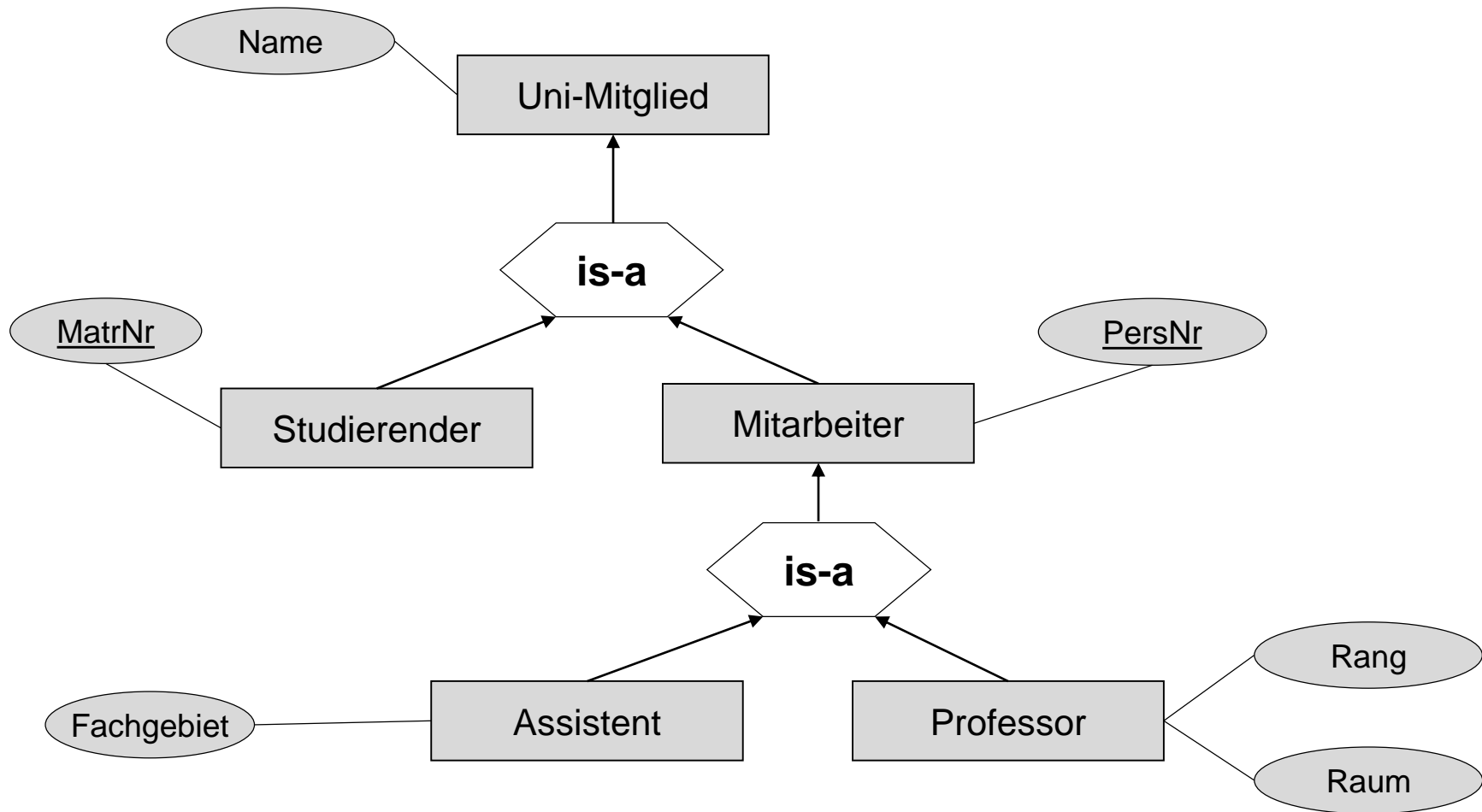


# Generalisierung / Spezialisierung im ERM

- Die Generalisierung und Spezialisierung von Entitytypen dient dazu, gemeinsame Eigenschaften ähnlicher Entitytypen zusammenzufassen.
  - Gemeinsame Attribute und Beziehungen aller ähnlichen Typen gehen auf den generalisierten Typ über.
  - Spezifische Attribute und Beziehungen verbleiben in den spezialisierten Typen.

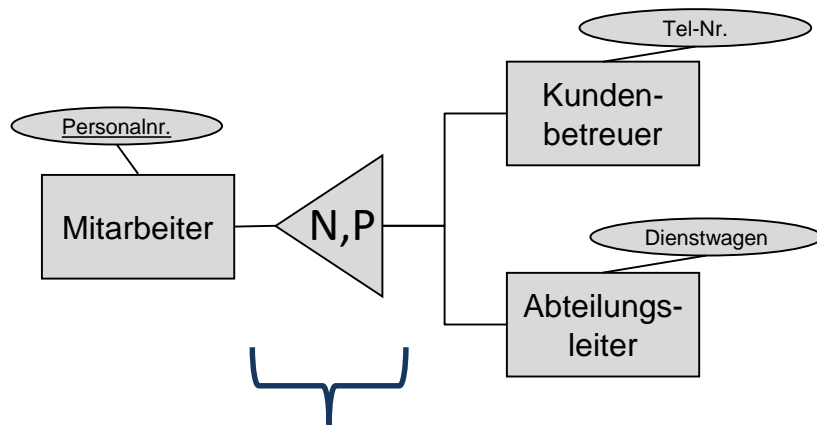


# Generalisierung / Spezialisierung im ERM: Beispiel

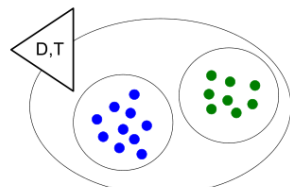
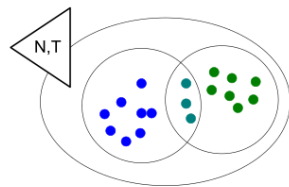
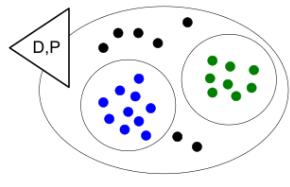
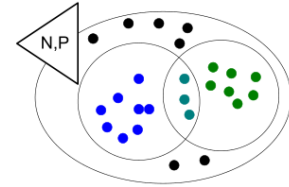


# Generalisierung / Spezialisierung: Zerlegung & Vollständigkeit

- Alternative Modellierungsmöglichkeit zur detaillierteren Darstellung:



Spezifikation der  
Zerlegung und  
Vollständigkeit  
(vier Möglichkeiten, siehe Tabelle)

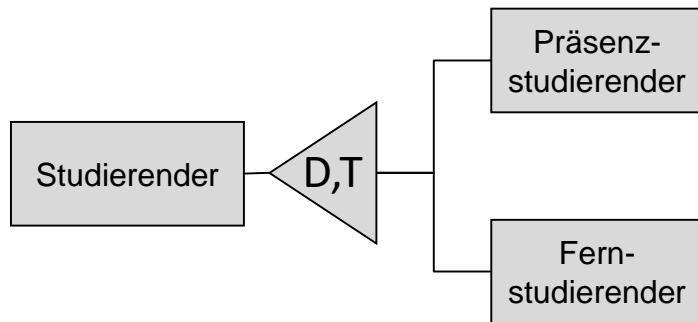
	<b>D - disjunkt</b> Ein Entity darf nur <i>maximal</i> einem Spezialfall angehören	<b>N - nicht disjunkt</b> Ein Entity darf <i>beliebig vielen</i> Spezialfällen angehören
<b>T - total</b> Jedes Entity <i>muss mindestens</i> einem Spezialfall angehören	 <p><b>Disjunkt-Total:</b> Jedes Entity gehört <i>immer genau einem</i> Spezialfall an</p>	 <p><b>Nichtdisjunkt-Total:</b> Jedes Entity gehört <i>immer mindestens einem</i> Spezialfall an.</p>
<b>P - partiell</b> Ein Entity <i>darf</i> einem Spezialfall angehören, muss aber nicht	 <p><b>Disjunkt-Partiell:</b> Jedes Entity <i>kann maximal einem</i> Spezialfall angehören.</p>	 <p><b>Nichtdisjunkt-Partiell:</b> Jedes Entity <i>kann einem oder mehreren</i> Spezialfällen angehören</p>

Quelle: Becker et al. (2009) – Script Datenmanagement

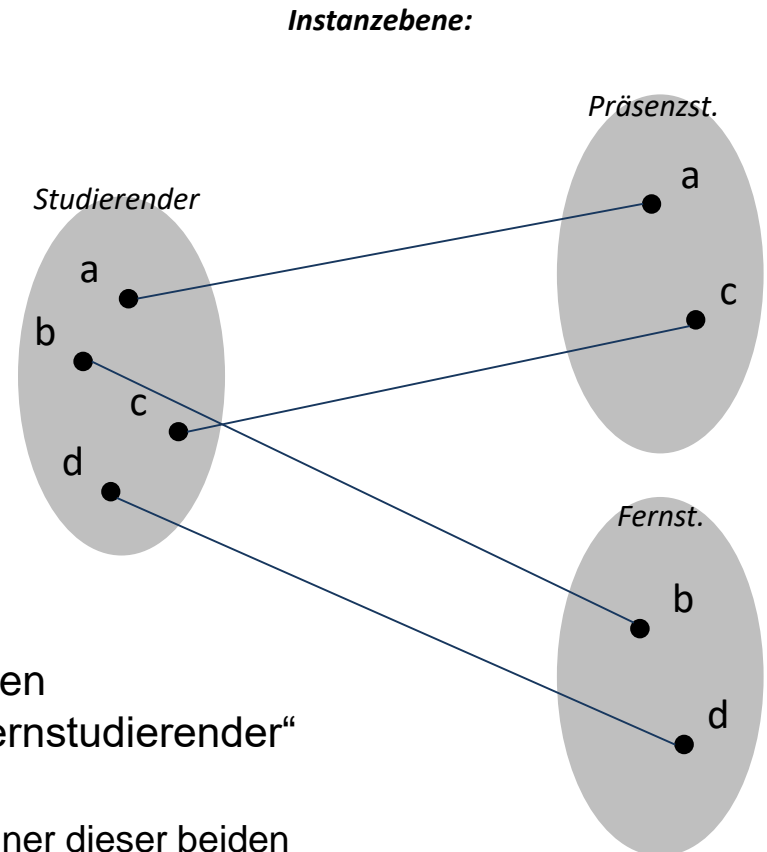


# Generalisierung / Spezialisierung: Zerlegung & Vollständigkeit

## ■ Beispiel: Disjunkt, Total

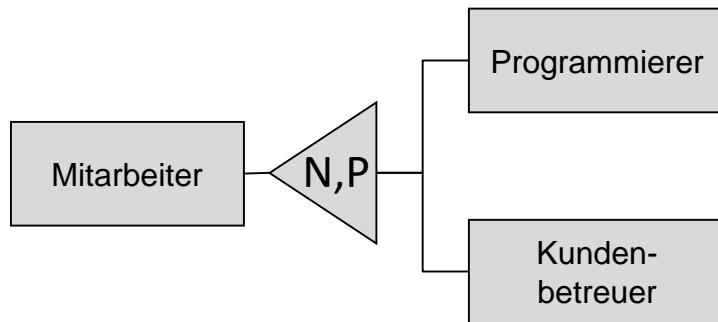


- Jeder Studierender muss einer der beiden Klassen „Präsenzstudierender“ oder „Fernstudierender“ zugeordnet sein.
  - Es gibt keinen Studierenden, der nicht einer dieser beiden Spezialisierungen zuzuordnen ist.

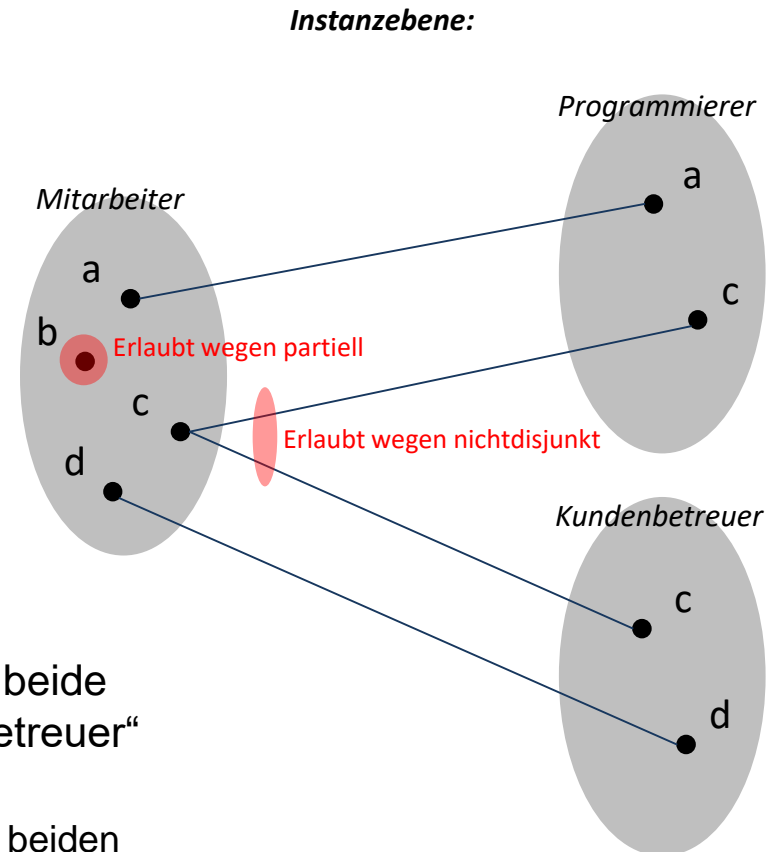


# Generalisierung / Spezialisierung: Zerlegung & Vollständigkeit

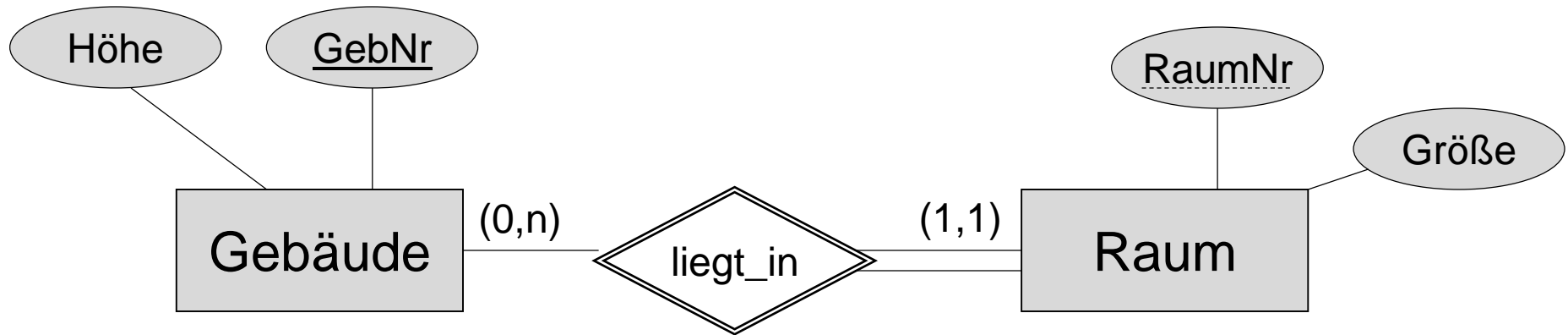
## ■ Beispiel: Nichtdisjunkt, Partiell



- Jeder Mitarbeiter kann keine, eine oder beide Rollen „Programmierer“ bzw. „Kundenbetreuer“ einnehmen.
  - Es gibt Mitarbeiter, die nicht einer dieser beiden Spezialisierungen zuzuordnen ist (Instanz b).
  - Es gibt Mitarbeiter, die beiden Spezialisierungen zuzuordnen sind (Instanz c)



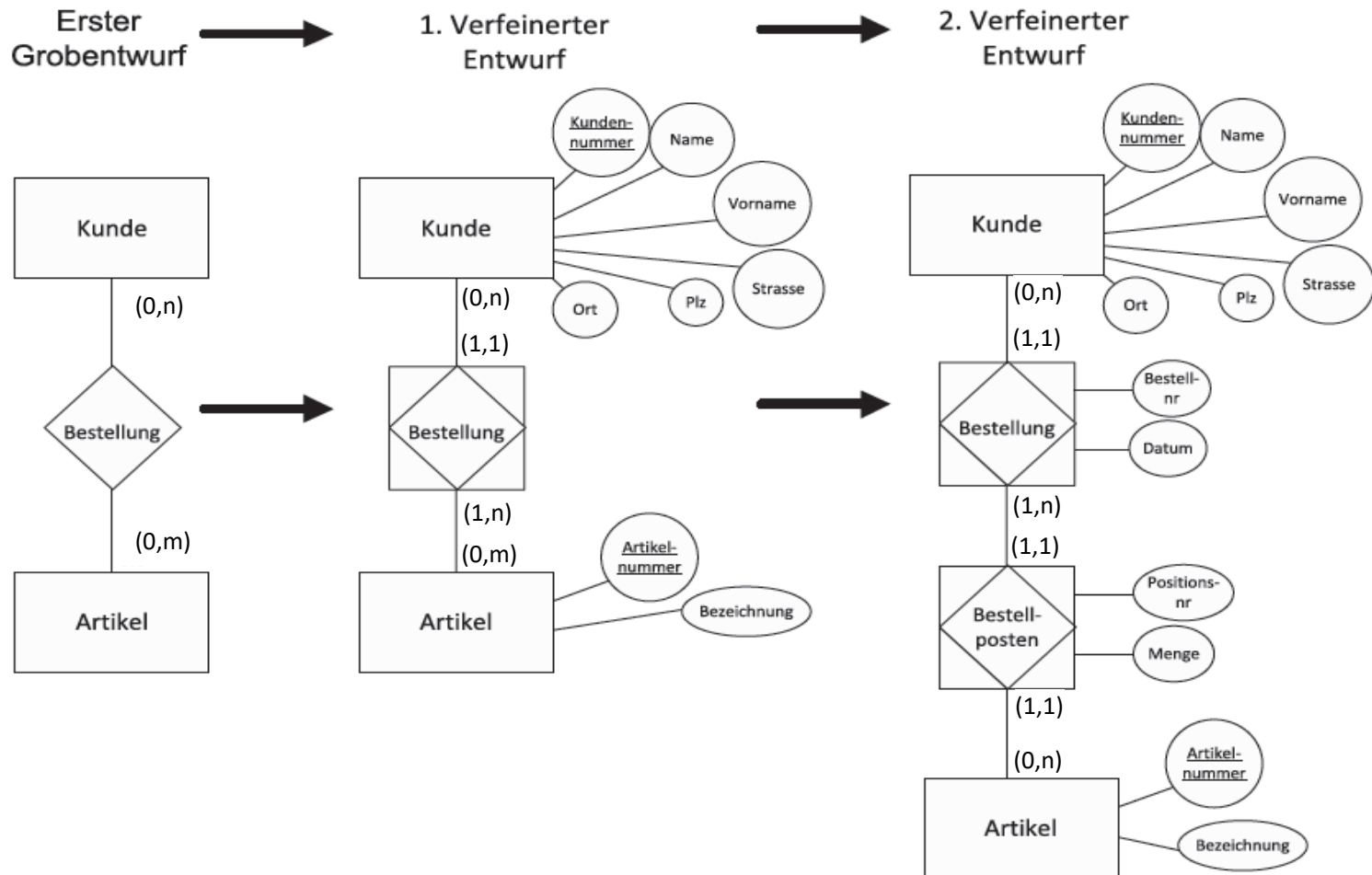
# Schwache Entitytypen



- Schwache Entities können **nicht autonom** existieren.
  - Sie sind in ihrer Existenz von einem anderen, übergeordneten Entitytyp abhängig.
  - Sie sind nur in Kombination mit dem Schlüssel des übergeordneten Entity eindeutig identifizierbar.
  - Schwache Entities werden durch doppelt gerahmte Rechtecke repräsentiert und ihre Beziehung zum übergeordneten Entity-Typ durch eine Verdoppelung der Raute und der von dieser Raute zum schwachen Entity-Typ ausgehenden Kante markiert.
- Im Beispiel:
  - RaumNr ist nur innerhalb eines Gebäudes eindeutig (gestrichelt-unterstrichen!)
  - Schlüssel ist: GebNr und RaumNr

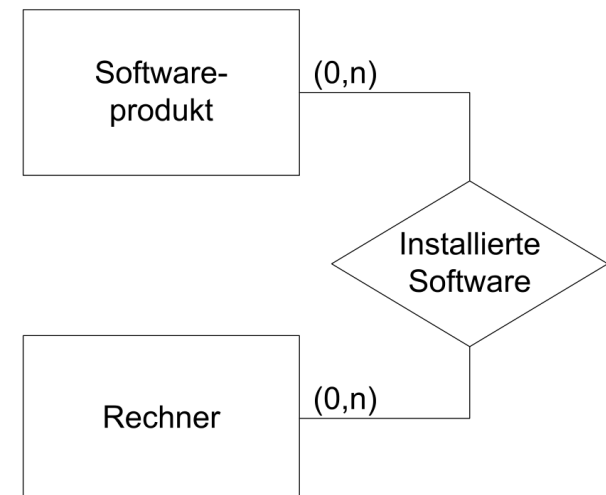
# Iterative Verfeinerung von ERM

- Modellierung ist oft ein iterativer Prozess, vom Grobentwurf zu den Details.



# Uminterpretierte Relationstypen

- Relationstypen beschreiben Beziehungen zwischen Instanzen zweier oder mehrerer Entities.
  - Manche Fälle können es erfordern, dass ein Relationstyp wiederum mit einem Entitytypen eine Beziehung eingehen muss.
  - Relationstypen dürfen nicht direkt mit anderen Relationstypen verbunden werden → der betreffende Relationstyp muss „uminterpretiert“ werden
- Beispiel:
  - „In einem Computerpool wird den Nutzern eine Anzahl von Rechnern zur Verfügung gestellt. Auf den Rechnern werden verschiedene Softwareprodukte installiert.“

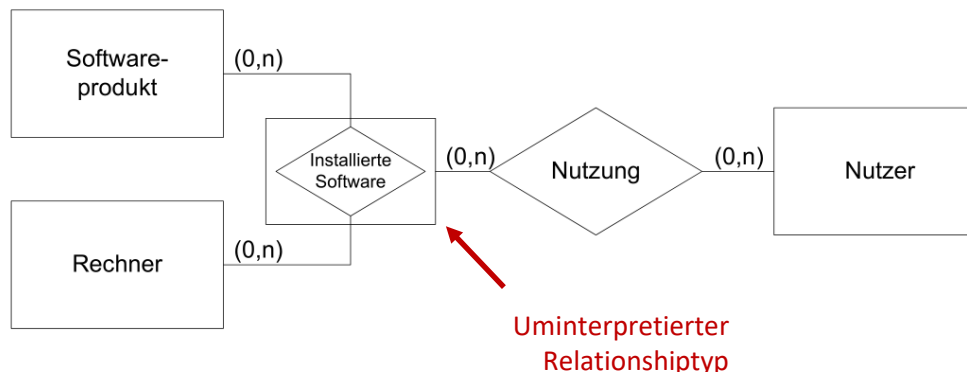
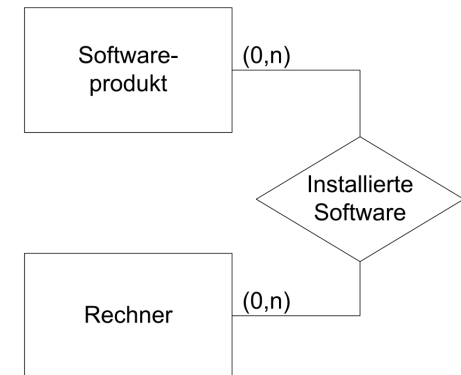


Quelle: Becker et al. (2009) – Script Datenmanagement

# Uminterpretierte Relationshiptypen

## ■ Beispiel (Fortsetzung):

- „In einem Computerpool wird den Nutzern eine Anzahl von Rechnern zur Verfügung gestellt. Auf den Rechnern werden verschiedene Softwareprodukte installiert.“
- Ergänzung: „Es soll nun protokolliert werden, welche Nutzer die verschiedenen installierten Softwareprodukte auf welchen Rechnern nutzen. Jede Nutzung soll protokolliert werden“.
  - Der Text verleiht dem ursprünglich als Beziehung modellierten Konzept Installierte Software nun eine „entity-artige“ Qualität. „Nutzung“ bezieht sich auf eine konkrete Kombination aus Softwareprodukt und Rechner (also „Installierter Software“):
  - Lösung: Uminterpretation des Relationshiptypen „Installierte Software“ als Entitytyp:



Quelle: Becker et al. (2009) – Script Datenmanagement



# Fallbeispiel „Hochschule“

## **Betrachten wir noch einmal dieses Fallbeispiel (s.o.):**

Bitte helfen Sie der Hochschulverwaltung bei der Datenmodellierung folgender Domäne:

- Wir speichern von Studierenden den Namen und Vornamen. Sie werden über ihre Matrikelnummer identifiziert.
- Studierende können Vorlesungen hören, die durch eine Vorlesungsnummer eindeutig identifiziert werden. Zusätzlich möchten wir im System hinterlegen, wie viele SWS die Vorlesung umfasst und wie der Titel lautet.
- Professoren haben eine Personalnummer und wir speichern zusätzlich ihre Namen und Vornamen. Dies ist identisch zu den Wissenschaftlichen Mitarbeitern, die für Professoren arbeiten. Professoren und Mitarbeiter haben einen eigenen Raum in unseren Hochschulgebäuden. Räume können aber auch geteilt werden, weshalb wir neben Raum- und Gebäudenummer auch noch die Anzahl Arbeitsplätze je Raum speichern möchten.
- Vorlesungen werden von Professoren verantwortet.
- Studierende werden in Vorlesungen von einem Professor geprüft (nicht zwangsläufig die-/derselbe, die/der die Vorlesung verantwortet). Neben der erreichten Note möchten wir auch das Semester hinterlegen, in dem der/die Student/in geprüft wurde. So können wir eine Historie der Prüfungen aufbauen.

→ **Modellieren Sie das Fallbeispiel als eERM. Verwenden Sie die (min,max)-Notation für Kardinalitäten.**

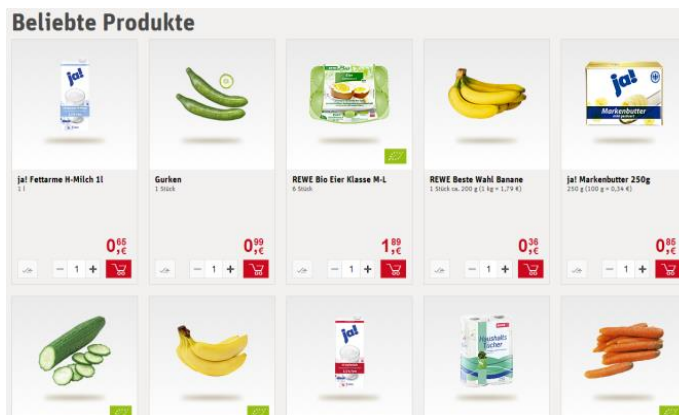
# Agenda

- ☐ Grundlagen der Datenmodellierung und Datenbanktypen
- ☐ Konzeptionelle Modellierung mit ERM – Grundlagen
- ☐ Konzeptionelle Modellierung mit eERM – fortgeschrittene Konzepte
- Fallstudie
- ☐ Zusammenfassung & Ausblick



# Fallstudie

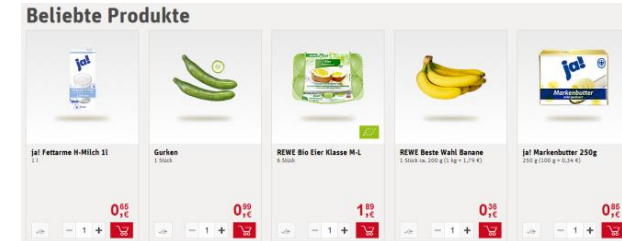
- Ein Unternehmen möchte in den Markt der Online-Supermärkte einsteigen und von Ihnen einen entsprechenden Online-Shop entwickeln lassen.
  - Grundlage hierfür stellt eine entsprechende Datenbank dar.



- Grobe Darstellung der Anforderungen
  - Kunden sollen sich als Nutzer zunächst registrieren. Auch Mitarbeiter sind als Nutzer des Systems abzubilden. Kunden legen Produkte in ihren Warenkorb mit entsprechender Mengenangabe. Bei Bestellung wird der Warenkorb in eine Bestellung überführt (alle Produkte werden aus dem Warenkorb in die Bestellung übertragen). Bestellungen werden von Mitarbeitern bearbeitet und haben einen Bestellstatus (bspw. „offen“, „ausgeliefert“, ...). Mitarbeiter gehören zu Abteilungen, die in einer Abteilungshierarchie abzubilden ist. Die Produkte gehören zu Kategorien.

## ■ Detaillierte Anforderungen

- Das System wird von Benutzern (Kunden & Mitarbeiter), die durch ihre E-Mailadresse identifiziert werden, bedient.
  - Alle Benutzer registrieren sich mit Name, Vorname und Passwort
  - Kunden sollen darüber hinaus in der Lage sein, bei der Registrierung ein Geburtsdatum zu hinterlegen. Für Kunden wird automatisch eine ID (Kundennummer) angelegt.
  - Mitarbeiter hinterlegen ihre eindeutige Personalnummer und ihre Tel.-Nr. Zusätzlich sollen Sie ihrer Abteilung, die einer Hierarchie angehört, zugeordnet werden.
- Registrierte Kunden sollen Produkte in ihren Warenkorb „legen“ können (mit Mengenangabe, bspw. 3 Äpfel).
  - Jeder Kunde hat genau einen Warenkorb.
  - Produkte haben eine eindeutige Produktnummer und in der Datenbank sollen zusätzlich die Bezeichnung, die aktuelle Verfügbarkeit, der Listenpreis und das Anlagedatum des Produkts in den Stammdaten gespeichert werden.
  - Produkte können zu Kategorien gehören. Für diese soll wiederum gespeichert werden können, welche Kategorien oft zusammen gekauft werden (bspw. Wurst → Grillzubehör).
- Wenn ein Kunde einen Warenkorb bestellen möchte, wird eine entsprechende Bestellung im System angelegt.
  - Dabei werden alle in den Warenkorb gelegten Produkte der Bestellung zugeordnet (inklusive Mengen).
  - So kann eine Historie der Bestellungen aller Kunden aufgebaut werden.
  - Bestellungen werden von genau einem Mitarbeiter bearbeitet und der Kunde kann sich über die Website über den aktuellen Status seiner Bestellung informieren.



# Agenda

- ☐ Grundlagen der Datenmodellierung und Datenbanktypen
- ☐ Konzeptionelle Modellierung mit ERM – Grundlagen
- ☐ Konzeptionelle Modellierung mit eERM – fortgeschrittene Konzepte
- ☐ Fallstudie
- ☒ Zusammenfassung & Ausblick



- Sie haben die Grundlagen der konzeptionellen Datenmodellierung kennengelernt und können mittels ERM und eERM die datenbezogenen Zusammenhänge in verschiedenen Domänen modellieren.
- Sie haben die „Basiskonzepte“ der ERM Notation und können auch erweiterte Konzepte (eERM) anwenden.
- Damit haben Sie eine ideale Grundlage, um im Unternehmenskontext über Daten zu diskutieren und Zusammenhänge abstrakt zu notieren und zu präsentieren.
- Als nächstes werden wir lernen, wie wir konzeptionelle Modelle in das relationale Modell überführen, um die Umsetzung in einer Datenbank vorzubereiten.

**Vielen Dank für Ihre Aufmerksamkeit!**

