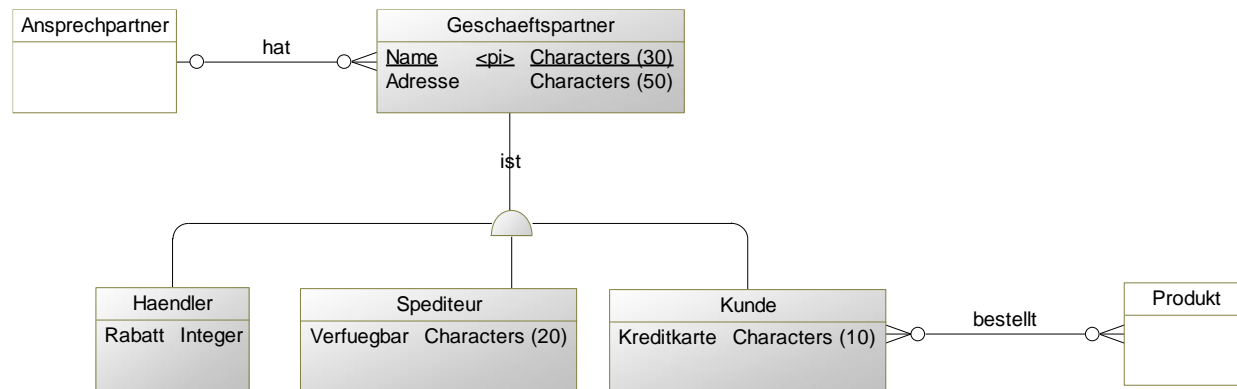




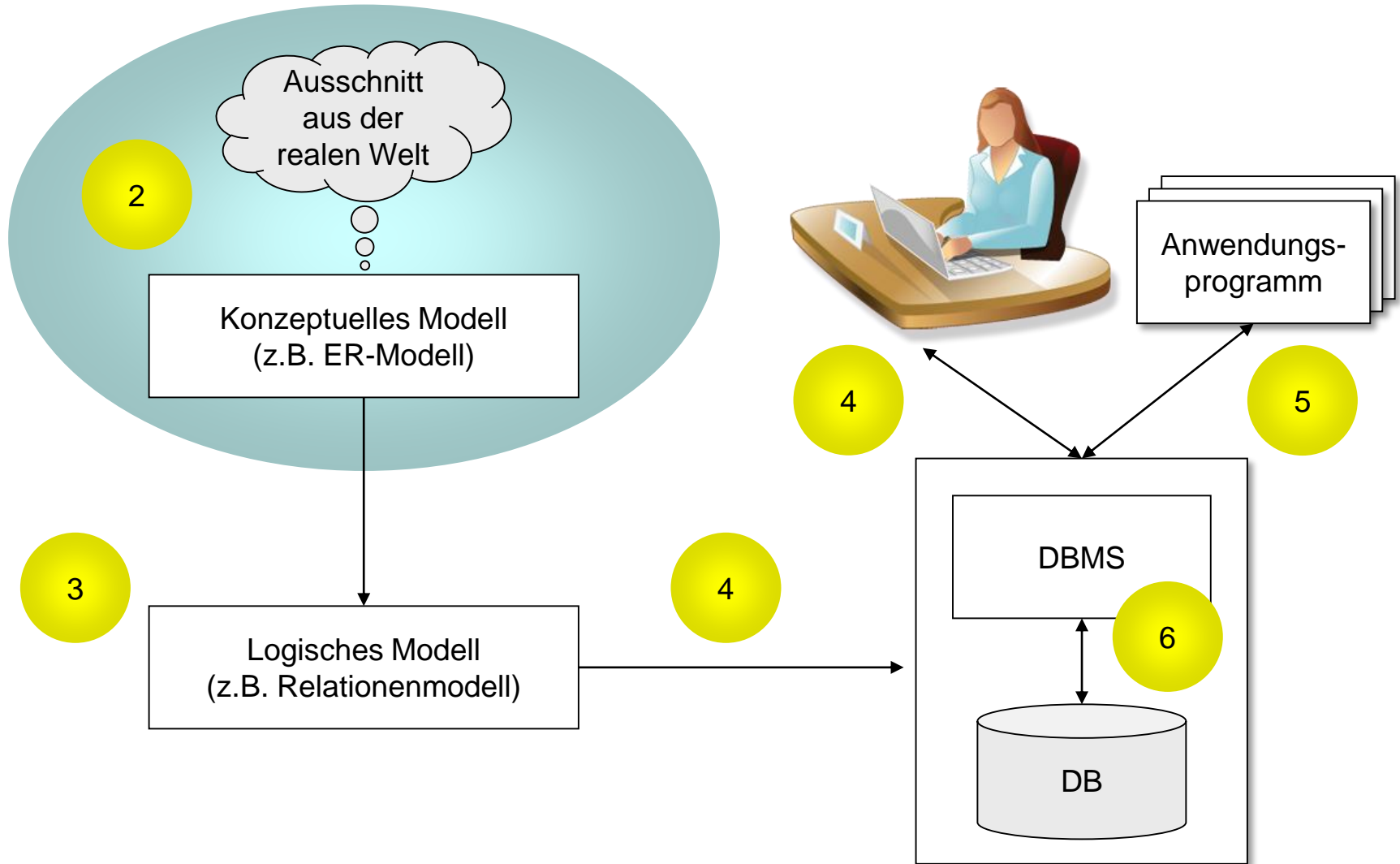
Datenbanken 1

– Kapitel 2: Datenbankentwurf –





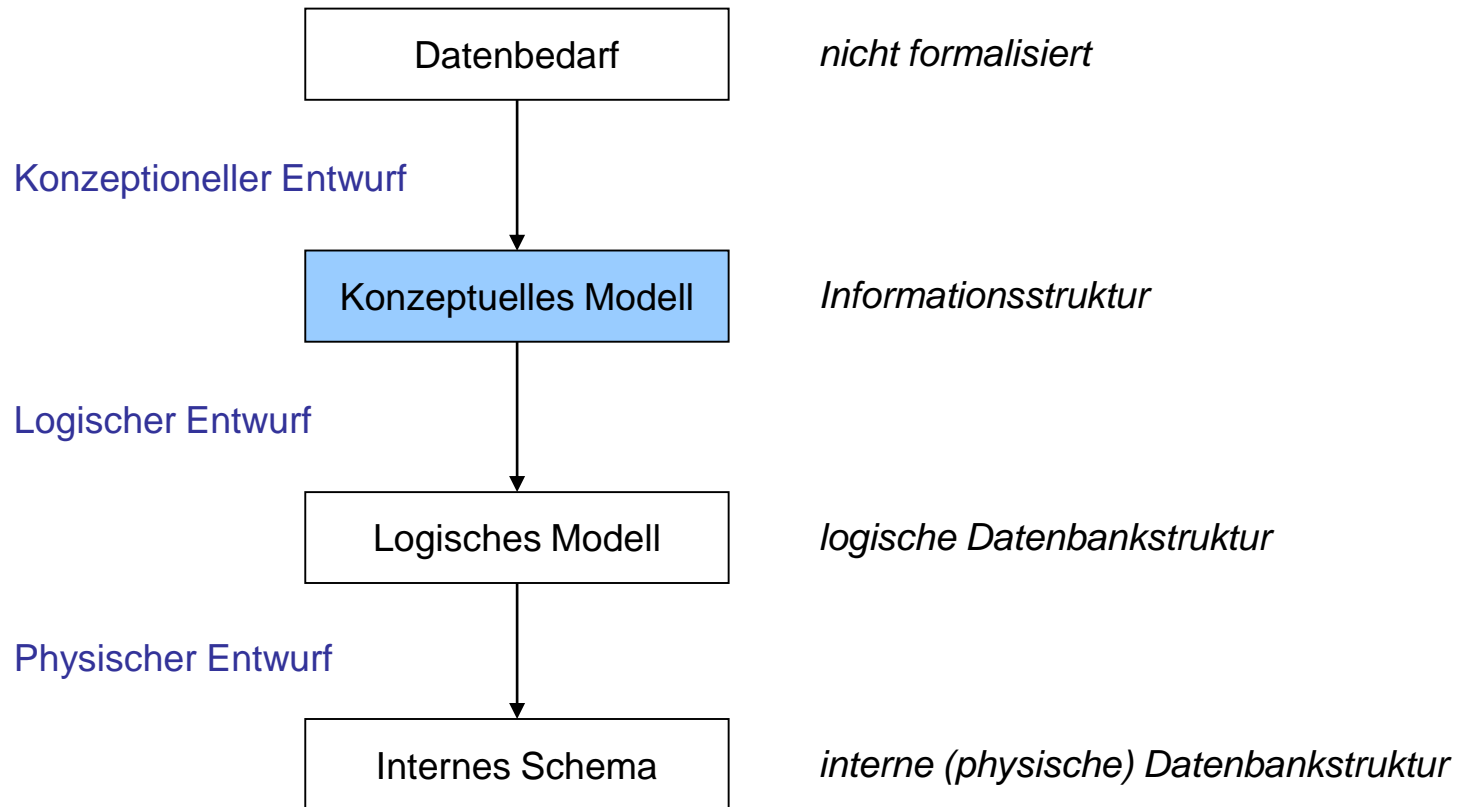
Vorlesung Datenbanken 1





Phasen des Datenbankentwurfs

Anforderungsanalyse





Datenbankentwurf

Inhalt des Kapitels

- Grundkonstrukte des Entity-Relationship-Modell (ER-Modell)
- Erweiterungen des ER-Modells
- Vorgehensweise beim Entwurf

Lernziele

- Kennenlernen der wichtigsten Konstrukte des ER-Modells
- Selbständiges Entwerfen von Datenbankmodellen mit dem ER-Modell



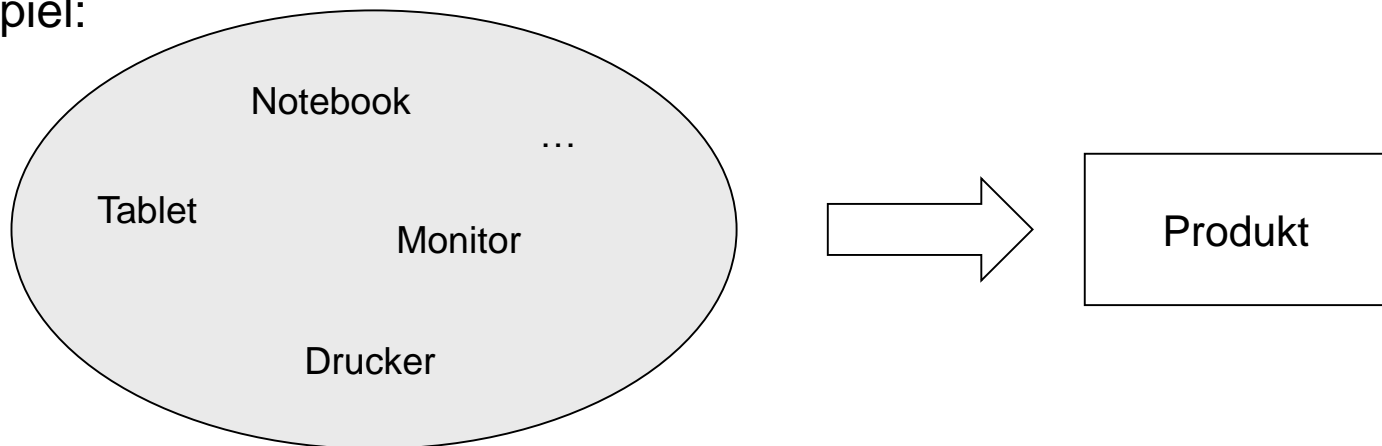
Entities und Entity-Typen

- **Entity**: Objekt der realen (bzw. der zu modellierenden) Welt.
- **Entity-Set**: Menge aller Entities mit gleichen oder ähnlichen Eigenschaften.
- **Entity-Typ**: Repräsentant der Objekte gleichen Typs.

Notation:

Entity-Typ

Beispiel:





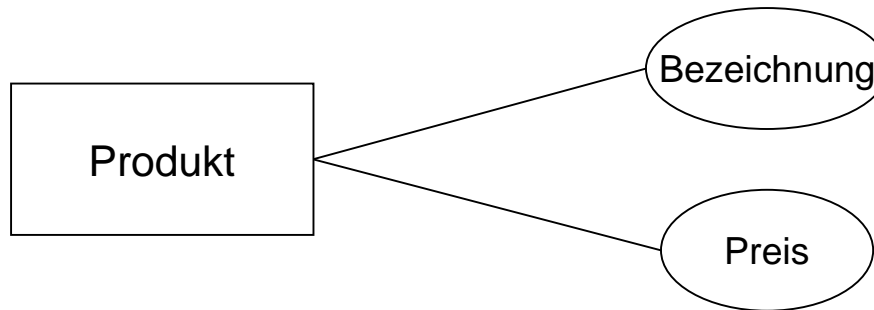
Attribute – 1(2)

- **Attribut:** repräsentiert eine Eigenschaft eines Entity-Typs, d.h. eine Eigenschaft, welche alle Entities dieses Entity-Typs besitzen.

Notation: $E (A_1, A_2, \dots, A_m)$



- **Wertebereich/Domäne:** mögliche bzw. zulässige Werte für ein Attribut
Beispiel:



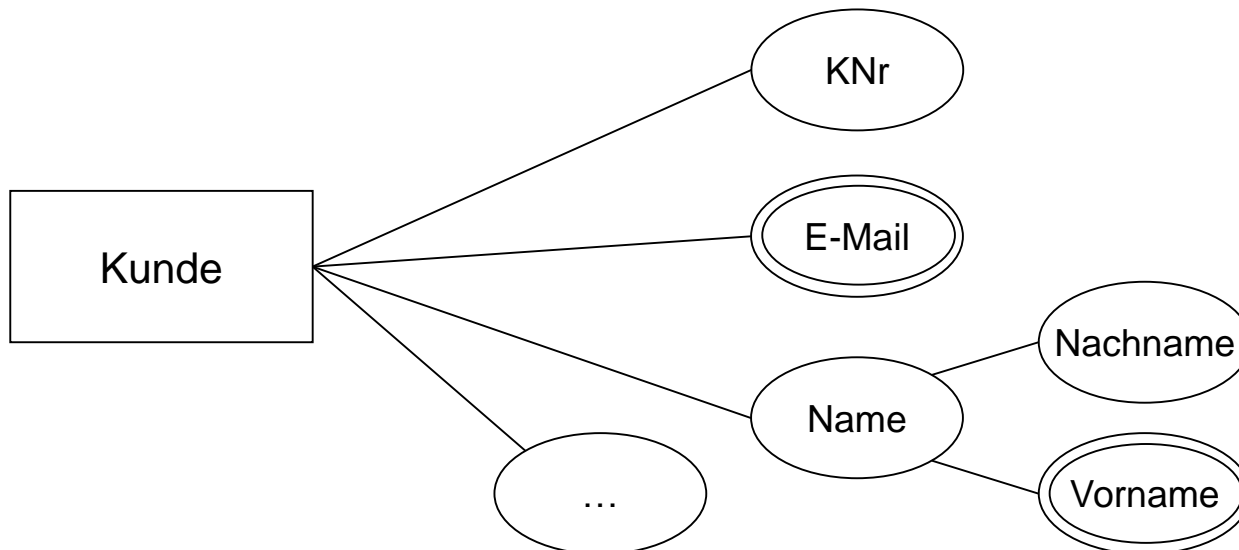
Preis: zwischen 5 und 100 Euro

Notation: $E (A_1:D_1, A_2:D_2, \dots, A_m:D_m)$



Attribute – 2(2)

- Wie wird modelliert, wenn beispielsweise
 - ein Kunde mehrere E-Mail-Adressen hat und
 - Der Name zusammengesetzt ist (Vorname, Nachname)
- ⇒ Erhöhung der Modellierungsgenauigkeit durch
- Mehrwertige Attribute (Notation: Doppeloval)
 - zusammengesetzte Attribute (Notation: hierarchisch angeordnete Attribute)

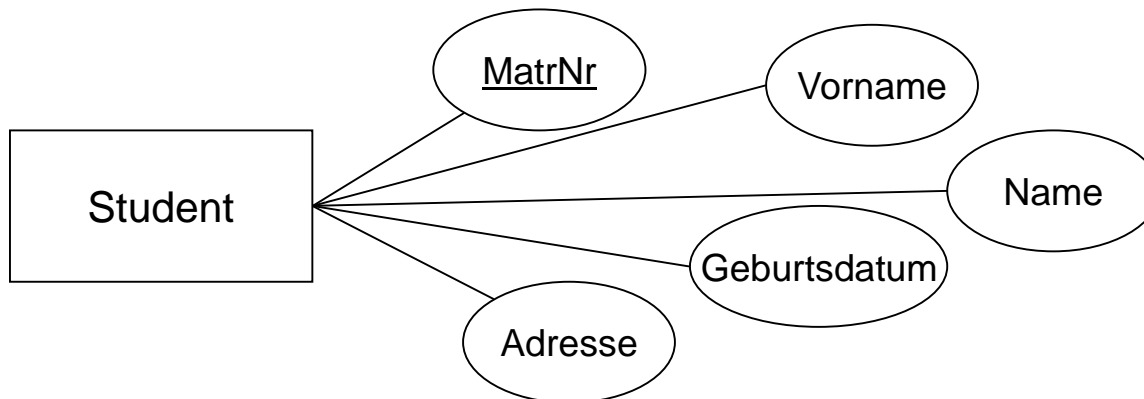


Kunde (KNr, {E-Mail}, Name (Nachname, {Vorname}))



Schlüssel

- Entities müssen „unterscheidbar“, d.h. eindeutig identifizierbar sein
- Informationen über Entities sind in ihren Attributen hinterlegt
- ⇒ Identifikation eines Entities durch Attribut oder Kombination von Attributen
- Ein **Schlüssel** eines Entity-Typs E ist eine Menge K von (einwertigen) Attributen, für die folgendes gilt:
 - Zu jedem Zeitpunkt unterscheiden sich zwei verschiedene Entities aus E bzgl. K, und es gibt keine echte Teilmenge von K, die diese Eigenschaft besitzt.
 - Beispiele: Schlüssel für Buch? ... Produkt? ... Studenten?
- Mehrere Schlüsselkandidaten möglich ⇒ Primärschlüssel auswählen



Student (MatrNr, Vorname, Name, Geburtsdatum, Adresse)

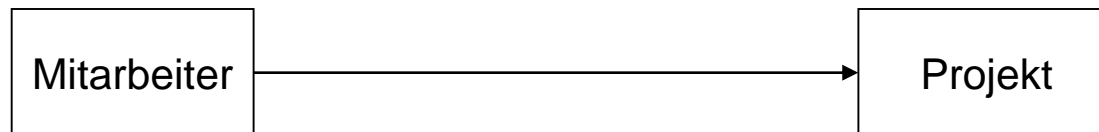


Beziehungstypen – 1(4)

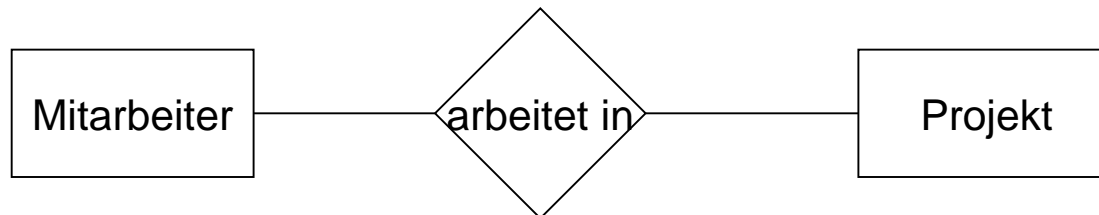
- Wichtige Informationen der realen Welt werden durch Beziehungen zwischen Entities ausgedrückt (insbesondere Prozesse!)
- Beziehungstyp (Relationship-Typ) beschreibt Beziehungen zwischen verschiedenen Entity-Typen

Historie der Notationen

- **Schema-Diagramme** in den 60er Jahren (**Charles Bachmann**)¹⁾
 - gerichtete Pfeile für Beziehungstypen



- **ER-Modell** erstmals 1976 von **Peter Chen**²⁾ beschrieben.
 - Rauten als Repräsentanten für verschiedene Beziehungstypen



1) Bachmann, C.W. *Data Structure Diagrams*. Databases 1, 2 (1969), pp. 4-10

2) Chen, P.P. The Entity-Relationship Model—Toward a Unified View of Data. ACM Trans. Database Systems 1,1 (March 1976) pp. 9-36



Beziehungstypen – 2(4)

Notation:

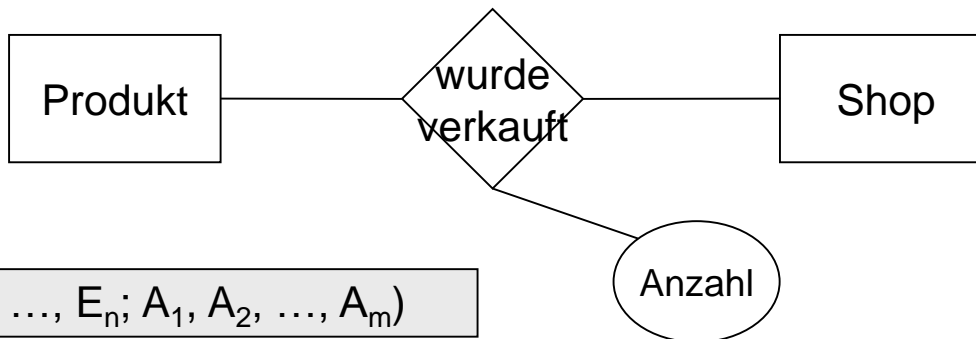
- Raute mit Name des Beziehungstyps (Verb oder Substantiv)

$R (E_1, E_2, \dots, E_n)$

Beispiel:



Beziehungstyp kann auch Attribute haben \Rightarrow **Beziehungsattribute**

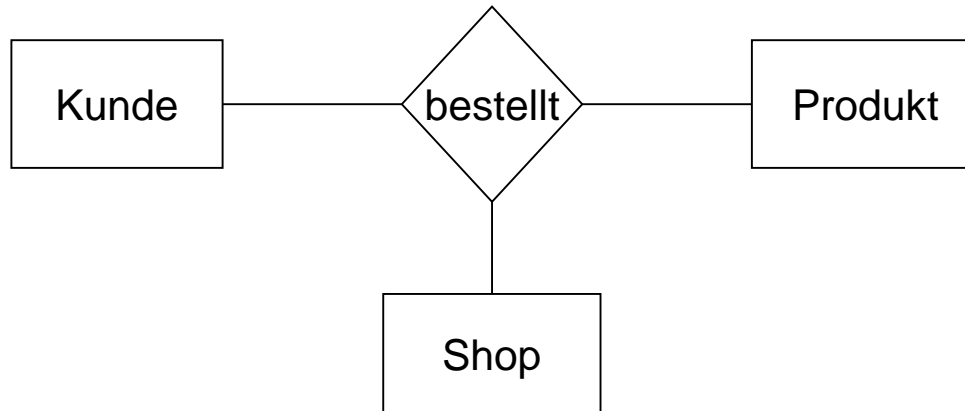


$R (E_1, E_2, \dots, E_n; A_1, A_2, \dots, A_m)$



Beziehungstypen – 3(4)

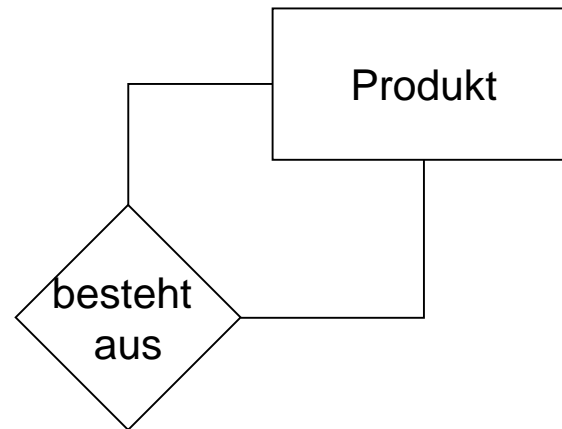
Es können auch **mehrstellige** Beziehungstypen modelliert werden:





Beziehungstypen 4(4)

- Beispiel: Optische Linsen in Fotoapparaten, in der Chip-Fertigung (Beleuchtung) etc. → Eine Linse kann selbst wieder aus anderen Linsen bestehen.
Allgemeiner: Ein Produkt kann selbst wieder aus anderen Produkten bestehen
- Wie wird diese Beziehung zwischen Entities gleichen Typs ausgedrückt?



⇒ **Rekursiver Beziehungstyp**

- Stichwort „Stückliste“



Hörsaalübung

Erstellen Sie ein ER-Modell für die folgende „FH-Miniwelt“

- Jeder Professor hält mehrere Vorlesungen. Mehrere Studenten besuchen jeweils (verschiedene) Vorlesungen.
- Ein Student wird von einem Professor jeweils über eine Vorlesung geprüft und erhält eine Note für diese Prüfung.
- Der Besuch von Vorlesungen setzt i.a. Kenntnisse anderer Vorlesungen voraus.
- Überlegen Sie sich, welche Attribute sie für die jeweiligen Entitäten mit abspeichern würden. Welche Attribute würden Sie als Schlüssel wählen? Wie sind die Beziehungen zwischen den Entity-Typen?



Kardinalitäten

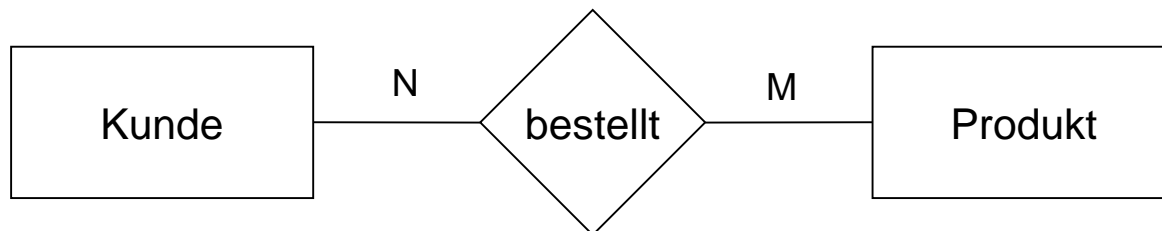
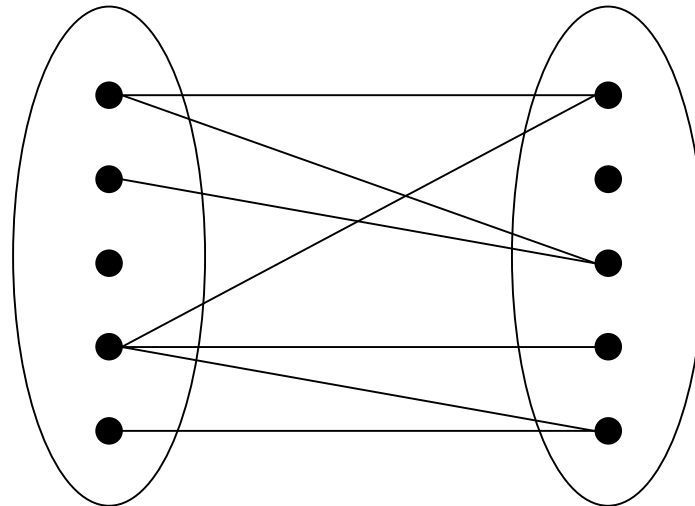
- Motivation: bisher kann nur ausgedrückt werden, dass zwischen bestimmten Entity-Typen eine Beziehung besteht und Eigenschaften dieser Beziehung können durch Beziehungsattribute ausgedrückt werden.
 - Was fehlt?
 - Jede Vorlesung wird von *genau einem* Professor gehalten ...
 - Ein Student kann *mehrere* Vorlesungen hören ...
 - Eine Vorlesung findet nur statt, wenn mindestens *6 Teilnehmer* angemeldet sind ...
- ⇒ Kardinalitäten! = Wichtige Integritätsbedingungen

Achtung! Es gibt verschiedene graphische Notationen für Kardinalitäten im ER-Modell – die wichtigsten (aber nicht alle) werden im folgenden vorgestellt. Das dahinter stehende Grundprinzip (1:1, 1:N und N:M Beziehungen) ist jedoch unabhängig von der graphischen Notation das gleiche!



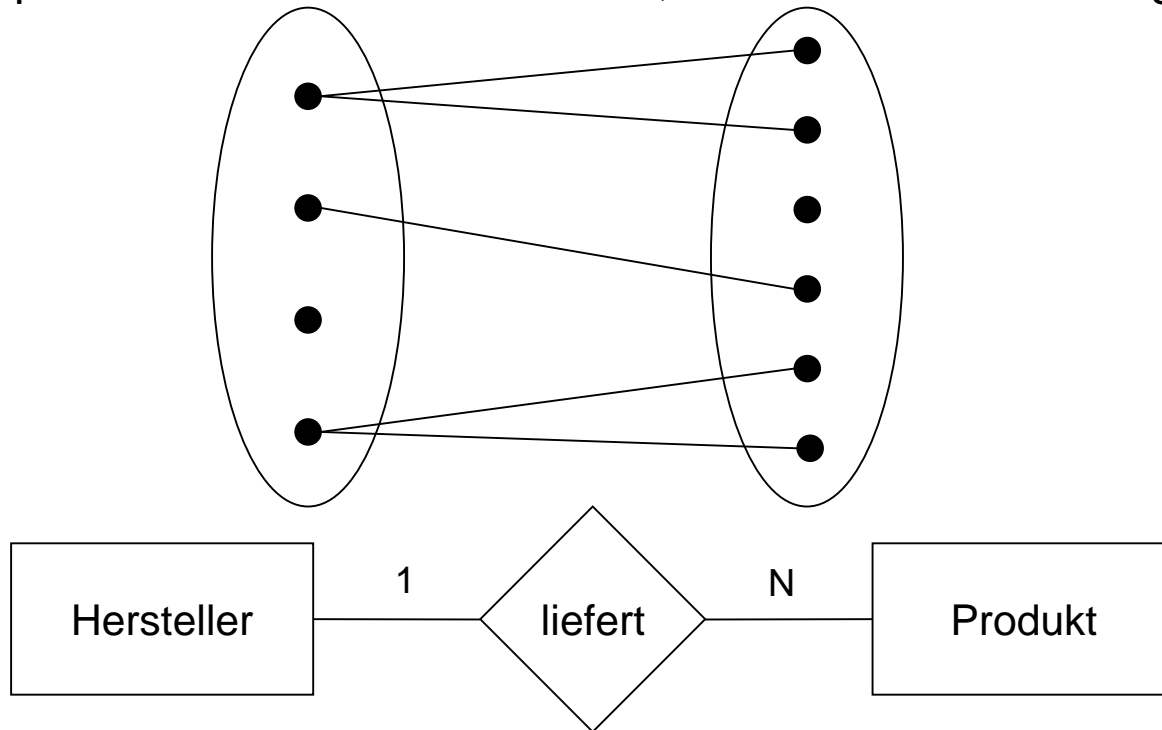
N:M Beziehungen

- keine Restriktionen, d.h. jedem Entity e_1 vom Entity-Typ E_1 können beliebig viele Entities E_2 zugeordnet sein und jedem Entity e_2 können beliebig viele Entities E_1 zugeordnet sein.
- Beispiele: *Kunde bestellt Produkt; Flächen begrenzen Kanten*



1:N Beziehungen

- jedem Entity e_1 vom Entity-Typ E_1 können beliebig viele Entities E_2 zugeordnet sein, aber zu jedem Entity e_2 gibt es maximal ein e_1 aus E_1
- Beispiele: *Hersteller liefert Produkt*, *Produkt hat Lackierung* ($n:1$)

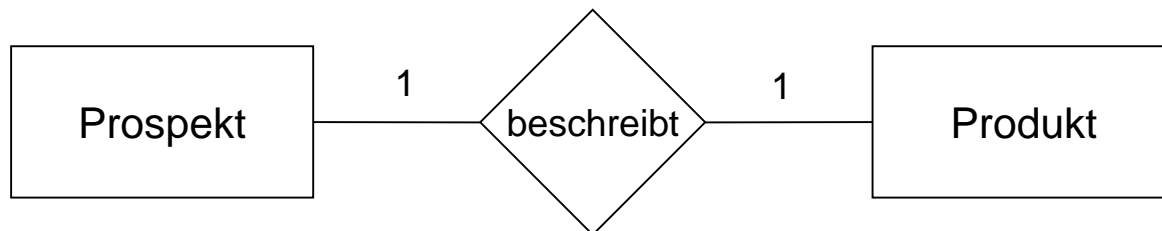
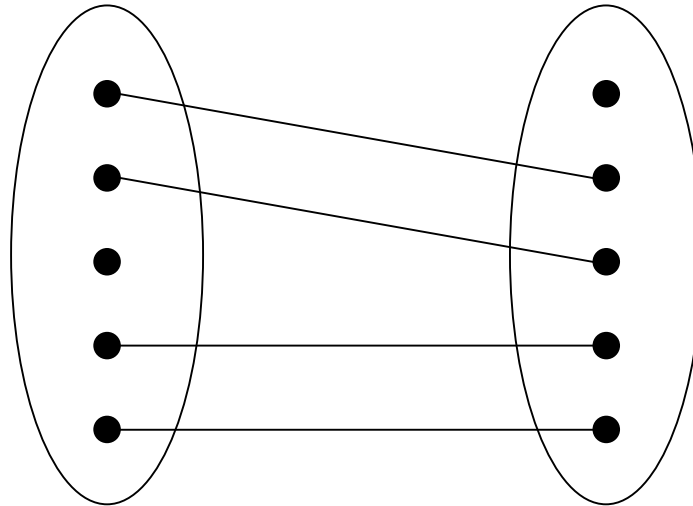


- Lesweise: „*ein* Hersteller liefert **N** Produkte“ bzw. „*ein* Produkt wird von höchstens **1** Hersteller geliefert“



1:1 Beziehungen

- jedem Entity e_1 vom Entity-Typ E_1 ist maximal ein Entity e_2 aus E_2 zugeordnet und umgekehrt
- Beispiele: *Prospekt beschreibt Produkt*; *Kennzeichen gehört zu Fahrzeug*





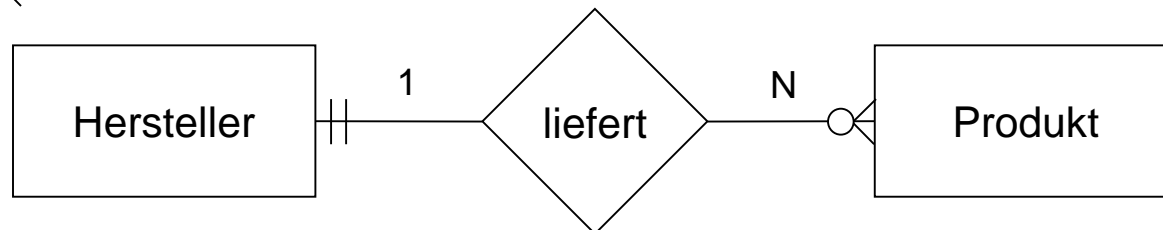
Verfeinerung der Kardinalitätsrestriktionen

- Limitation des 1:1, 1:N bzw. N:M Modells:
 - 1 bedeutet „höchstens“ eins – der Unterschied zu „genau eins“ ist nicht ausdrückbar
 - Präzisere Angaben der Kardinalitätsrestriktionen:
 - statt **1**: (0, 1) oder (1, 1)
 - statt **N**: (0, n) oder (1, n)

⇒ „**Krähenfuss-Notation**“ oder „**Martin-Notation**“ (James Martin)

- **Graphische Darstellung**

○ = 0
| = 1
◁ = n



- Diese Notation beeinflusste auch die heutige UML-Notation stark und ist in Modellierungstools sehr populär.

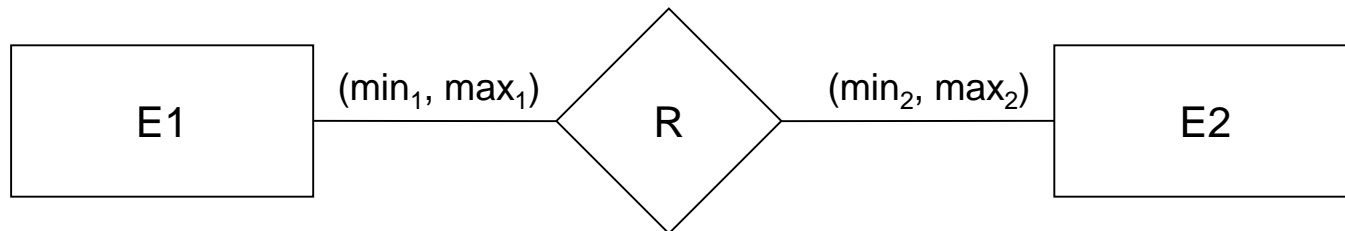


Weitere Verfeinerung der Kardinalitätsrestriktionen

- Spezifischere Restriktionen (ein Auto hat genau 4 Räder; ein Fahrzeug hat mindestens 2, höchstens 4 Räder) nicht ausdrückbar

⇒ (min, max)-Notation

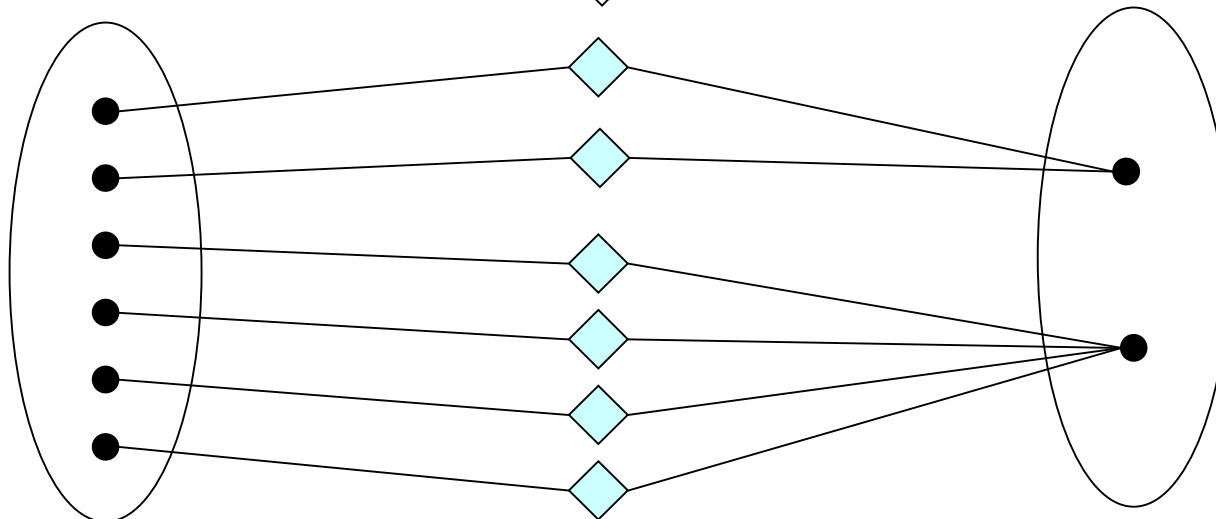
- schränkt die möglichen **Teilnahmen von Instanzen der beteiligten Entity-Typen an der Beziehung ein**, indem ein minimaler und ein maximaler Wert vorgegeben wird



- e_1 nimmt an mindestens \min_1 und höchstens \max_1 Beziehungen vom Typ R teil
 - e_2 nimmt an mindestens \min_2 und höchstens \max_2 Beziehungen vom Typ R teil
- **Achtung: Die Angabe der (min, max)-Kardinalitäten erfolgt genau gespiegelt zur UML-Notation!**



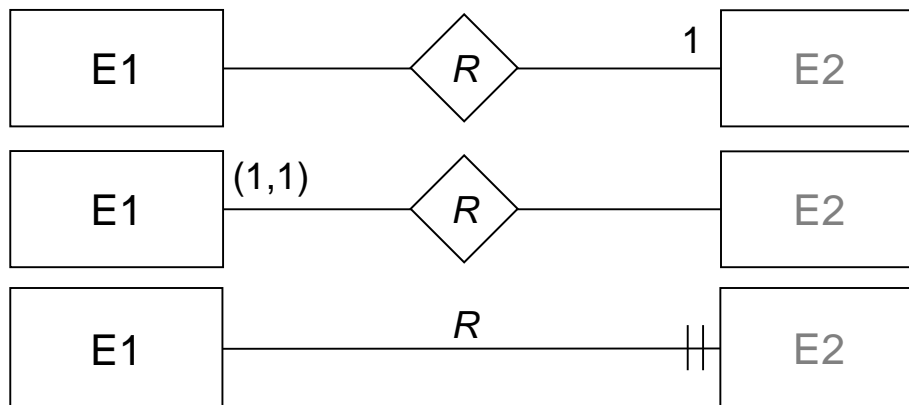
Beispiel für (min, max)-Notation



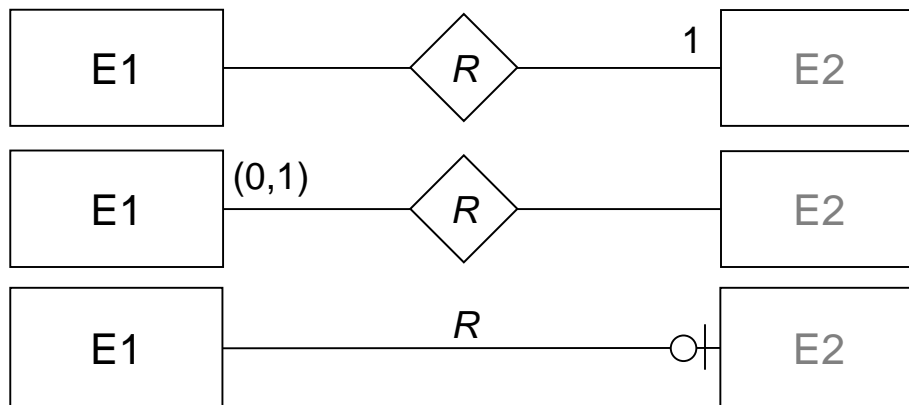


Verschiedene Kardinalitätsnotationen – 1(2)

- Jedes Element von E1 steht zu **genau einem** von E2 in Beziehung.



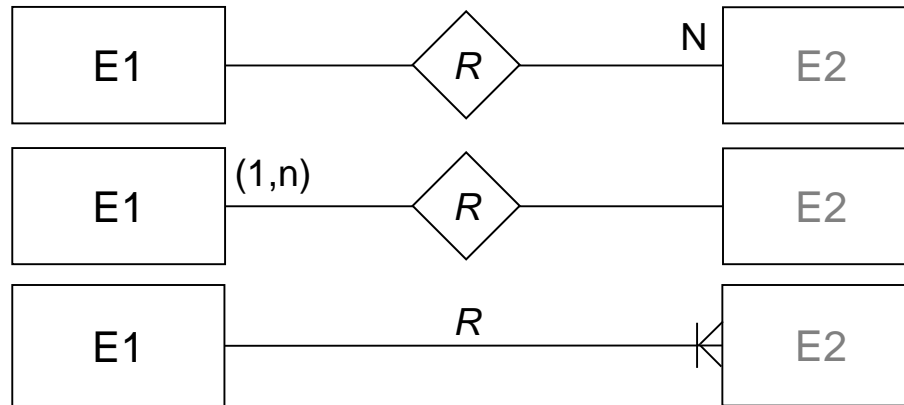
- Jedes Element von E1 steht zu **höchstens einem** von E2 in Beziehung.



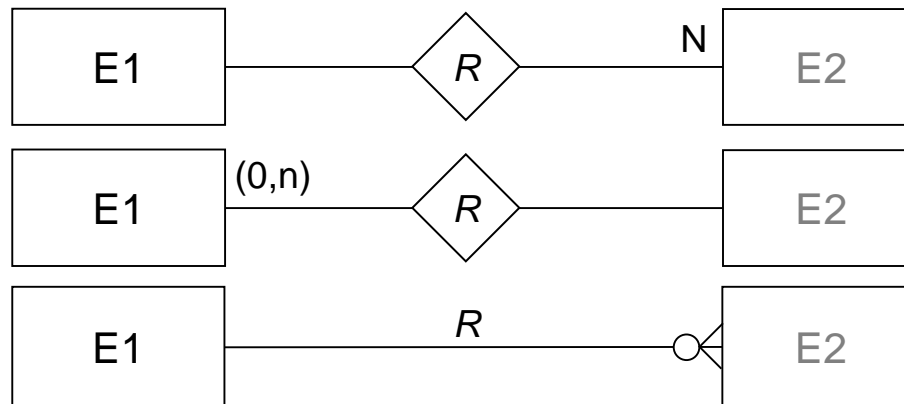


Verschiedene Kardinalitätsnotationen – 2(2)

- Jedes Element von E1 steht zu **mindestens einem** von E2 in Beziehung.

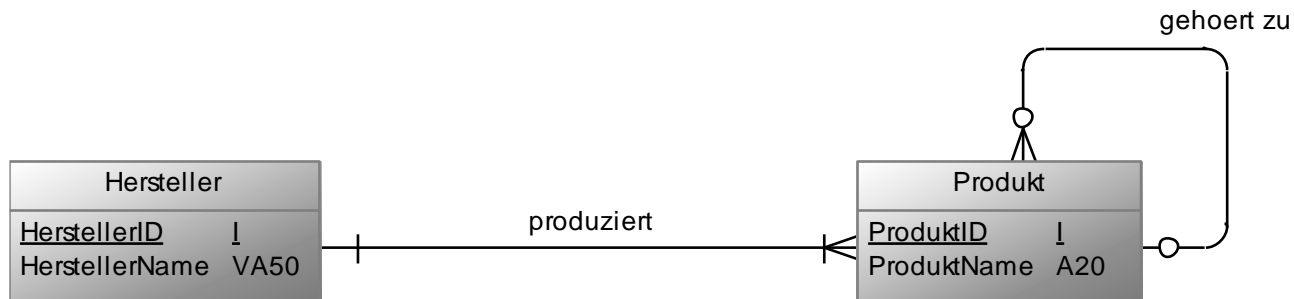
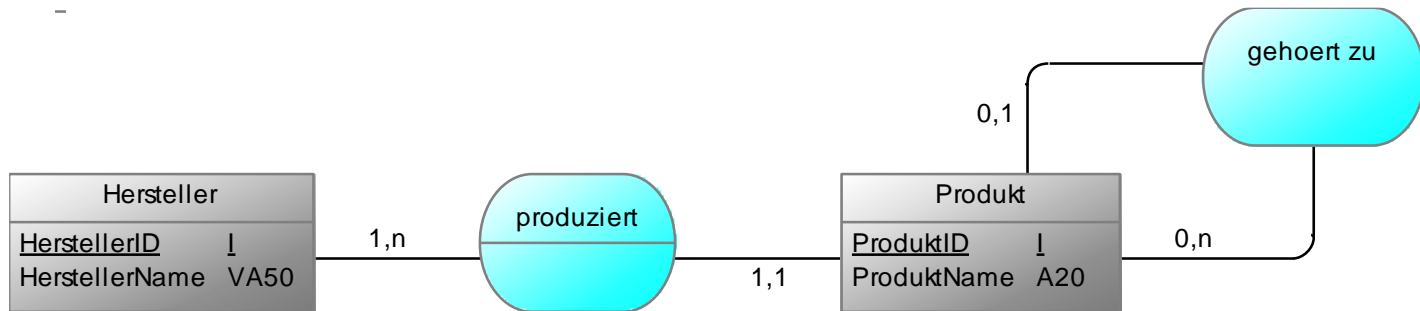


- Jedes Element von E1 steht zu **beliebig vielen** von E2 in Beziehung.





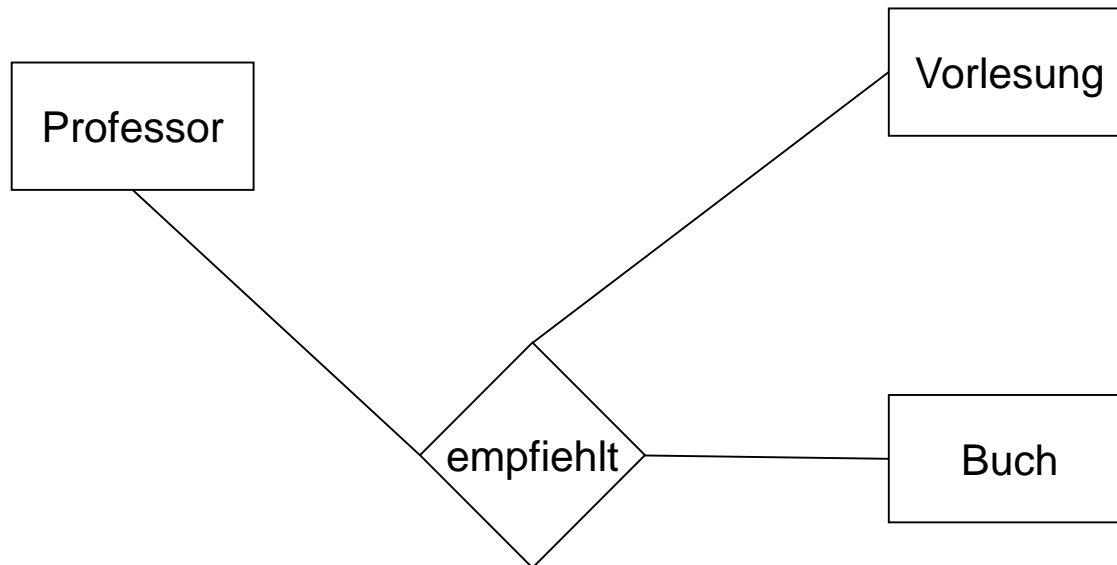
Modellierungstool PowerDesigner





Bemerkung zu n-stelligen Beziehungen – 1(3)

- n-stellige Beziehungen lassen sich in Modellierungswerkzeugen oft nicht darstellen (u.a. da – außer bei Verwendung der (min, max) „Teilnahme“-Kardinalitätsnotation – nicht eindeutig ist, auf welches Entity sich die Kardinalitätsangabe bezieht)



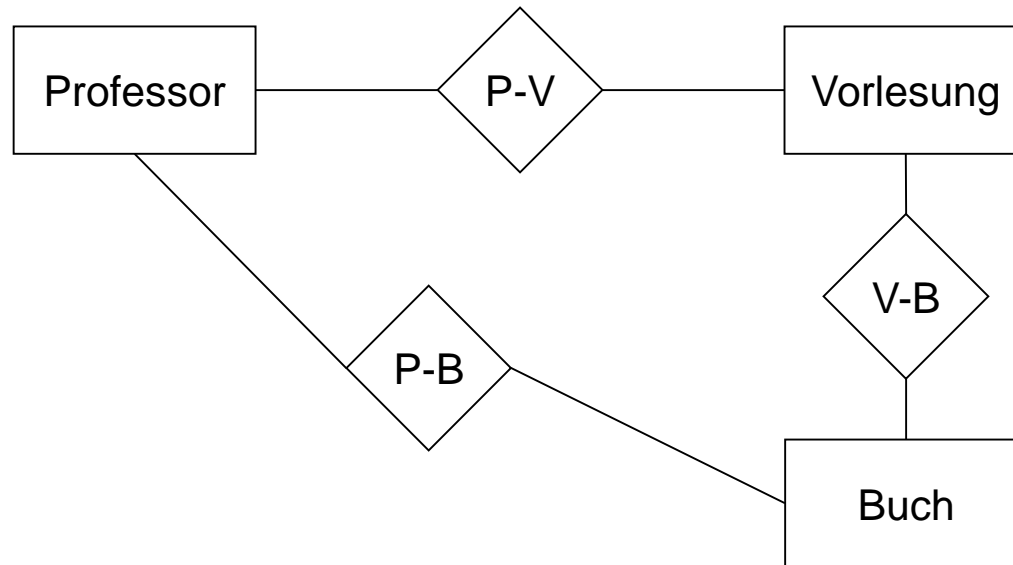
empfiehl

| | | |
|-------|-----|-----|
| Heuer | DB1 | ABC |
| Heuer | DB2 | DEF |
| Saake | DB1 | DEF |
| Saake | DB2 | DEF |



Bemerkung zu n-stelligen Beziehungen – 2(3)

- Was passiert bei Auflösung in drei zweistellige Beziehungen?
- Wird die Information korrekt wiedergegeben?



P-V

| | |
|-------|-----|
| Heuer | DB1 |
| Heuer | DB2 |
| Saake | DB1 |
| Saake | DB2 |

P-B

| | |
|-------|-----|
| Heuer | ABC |
| Heuer | DEF |
| Saake | DEF |

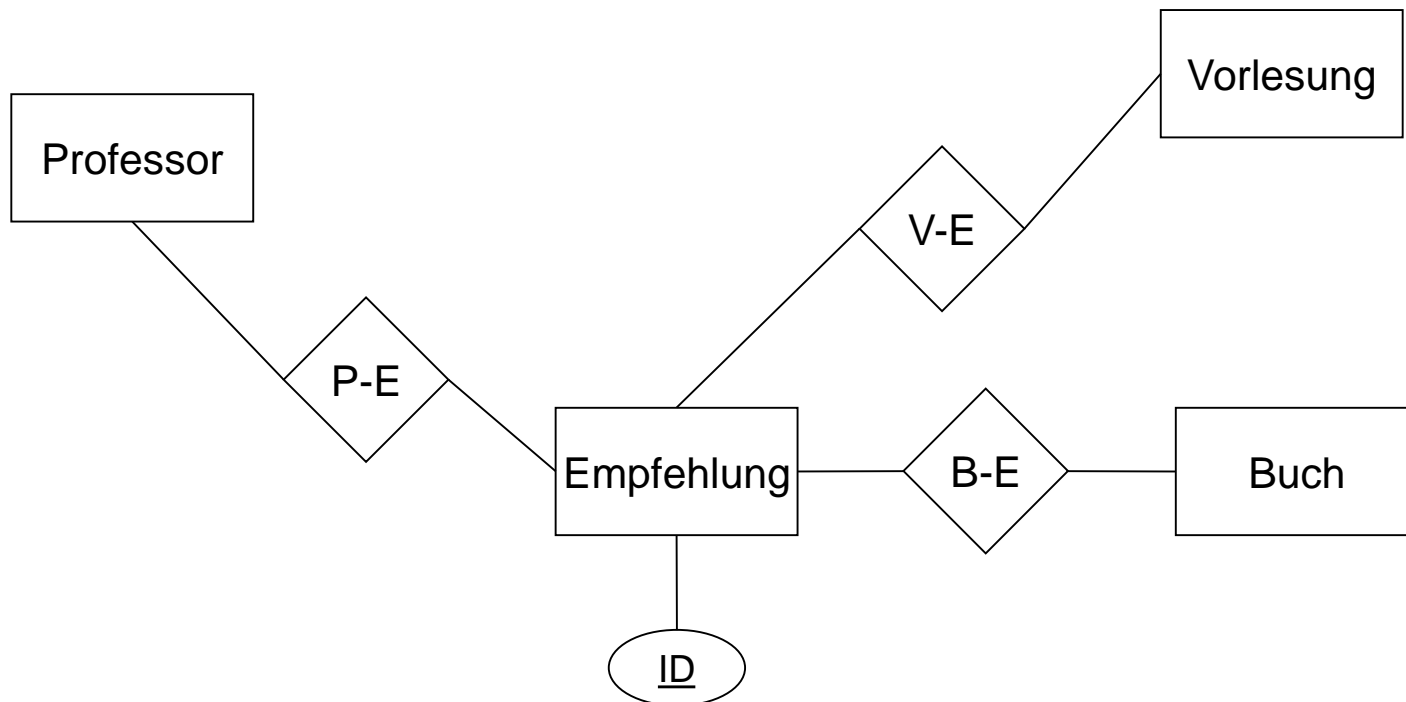
V-B

| | |
|-----|-----|
| DB1 | ABC |
| DB2 | DEF |
| DB1 | DEF |



Bemerkung zu n-stelligen Beziehungen – 3(3)

- Direkte Umsetzung n-stelliger Beziehungstypen in zweistellige kann zu unerwünschten Effekten führen, d.h. die Ausdrucksfähigkeit ist in diesem Fall geringer als bei n-stelligen Beziehungen
- Alternative: n-stelligen Beziehungstypen in (künstlichen) Entity-Typen umwandeln und n zweistellige Beziehungen definieren.





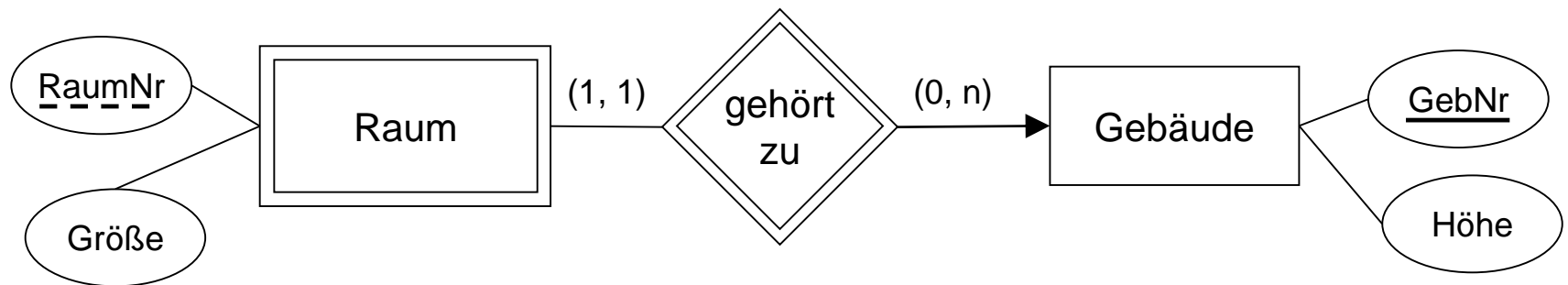
Entity-Relationship-Modell

- Bisher eingeführt
 - Entity-Typen
 - Attribute
 - Beziehungstypen
 - Kardinalitäten von Beziehungstypen
- Jetzt: Einführung weiterer Konstrukte, um bestimmte Gegebenheiten der zu modellierenden „Miniwelt“ adäquat auszudrücken.



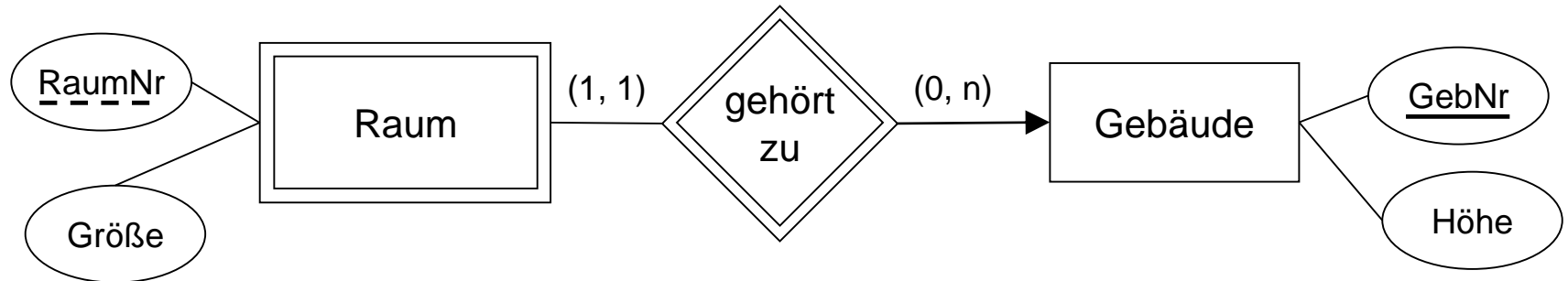
Existenzabhängige Entities – 1(2)

- Annahme bisher: Entities existieren autonom und sind über ihren Schlüssel eindeutig identifizierbar
 - Aber: in der Realität existieren oft Entities, welche
 - in ihrer Existenz von einem anderen, übergeordneten Entity abhängig sind und
 - oft nur in Kombination mit dem Schlüssel des übergeordneten Entities eindeutig identifizierbar sind
- ⇒ Existenzabhängige Entities (auch *schwache Entities* oder *weak entities* genannt)

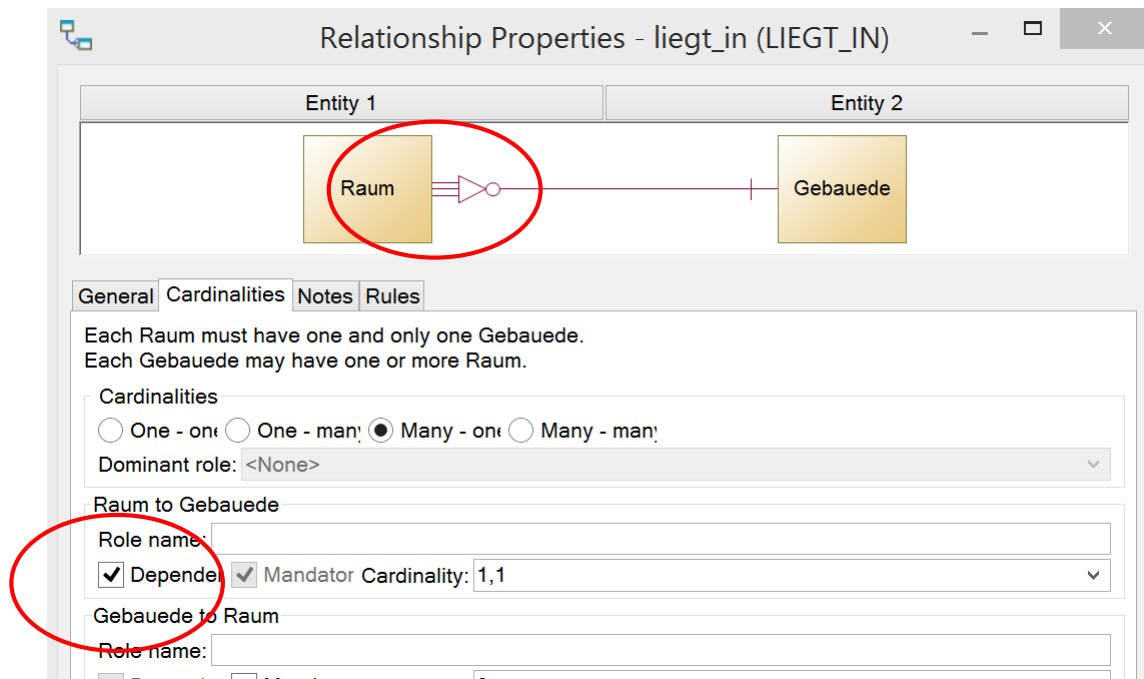




Existenzabhängige Entities – 2(2)

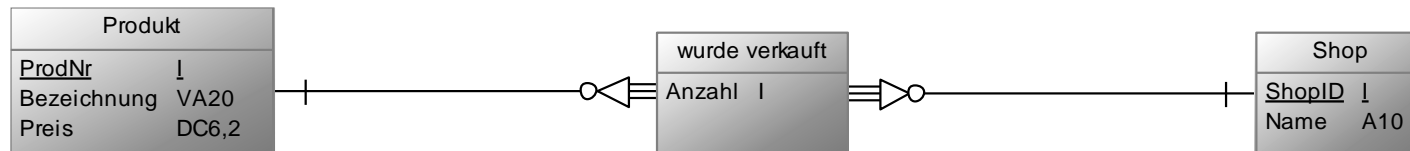
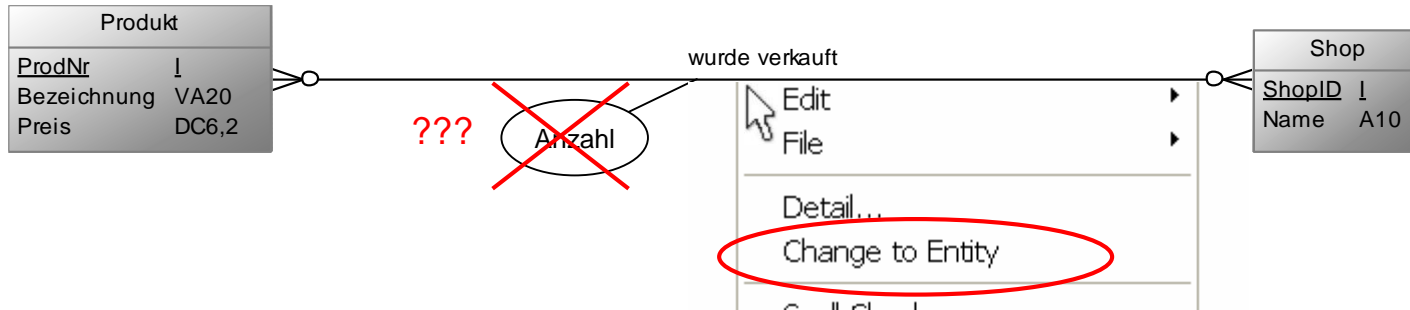


- Modellierung im PowerDesigner





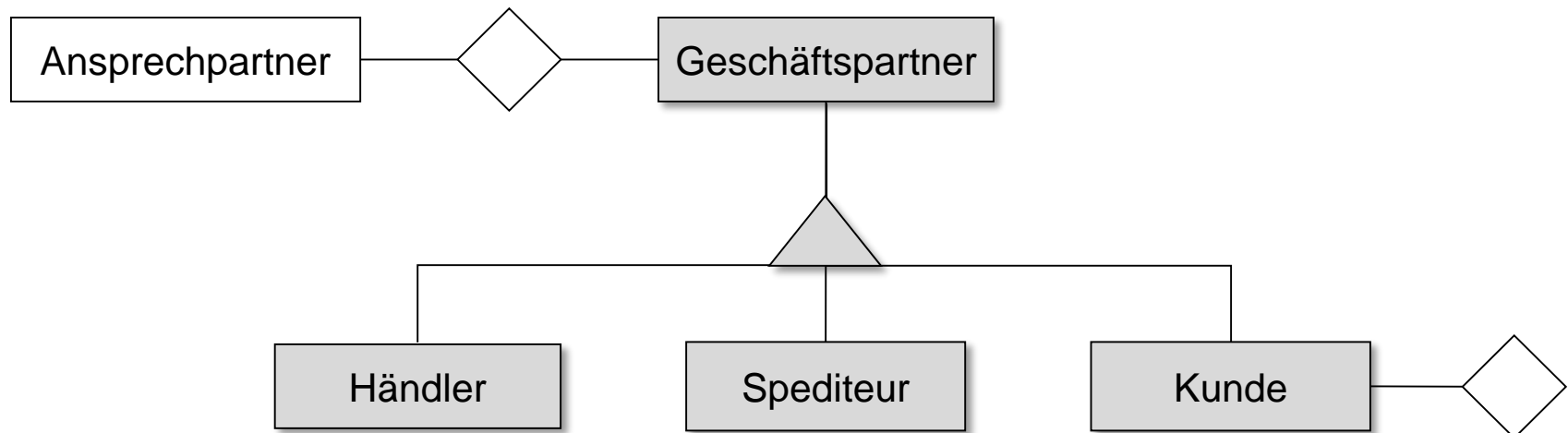
Anmerkung zur Abbildung von Beziehungsattributen im PowerDesigner





Generalisierung / Spezialisierung – 1(3)

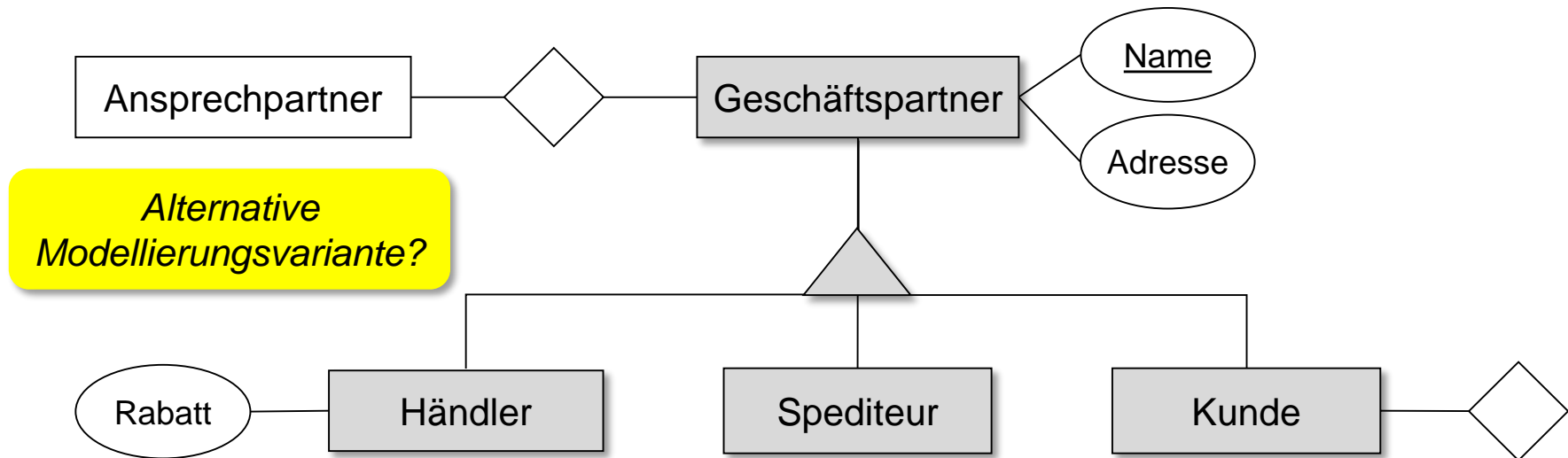
- Anordnung der Typen in eine Typhierarchie
 - Generalisierung: Übergang zu einem allgemeineren (Super-)Typ
 - Bsp.: Spediteur \Rightarrow Geschäftspartner
 - Spezialisierung: Übergang zu einem oder mehreren speziellen (Sub-)Typen
 - Bsp.: Geschäftspartner \Leftarrow Spediteur
Geschäftspartner \Leftarrow Händler
- *Spezialisierungs-/Generalisierungsbeziehung* wird auch *IST-Beziehung* genannt (engl. *is-a relationship*)





Generalisierung / Spezialisierung – 2(3)

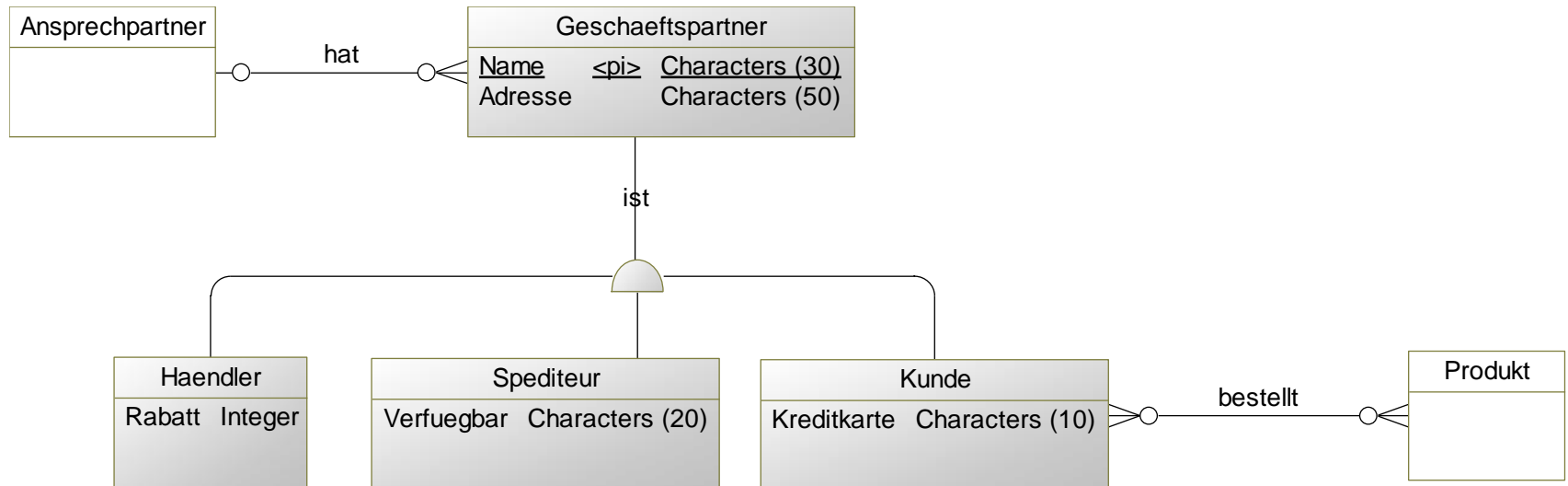
- Eigenschaften
 - Jeder Händler-Instanz ist genau eine Geschäftspartner-Instanz zugeordnet.
 - Händler-Instanzen werden durch die funktionale IST-Beziehung identifiziert.
 - Attribute des Entity-Typs Geschäftspartner treffen auch auf Händler zu = „vererbte“ Attribute.
 - nicht nur die Attributdeklarationen vererben sich, sondern auch jeweils die aktuellen Werte für eine Instanz.





Generalisierung / Spezialisierung – 3(3)

- Modellierung im PowerDesigner





Datenbankentwurf

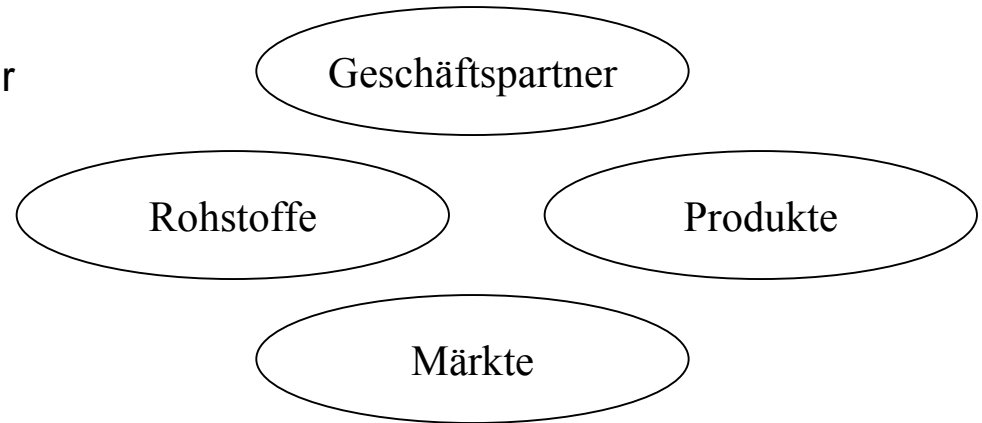
- Entity-Relationship-Modell
 - Entity-Typen und Attribute
 - Beziehungstypen
 - Kardinalitäten von Beziehungstypen
 - Existenzabhängige Entities
 - Generalisierung / Spezialisierung
- Was fehlt?
 - Vorgehensweise beim Konzeptionellen Entwurf



Vorgehensweise beim Entwurf

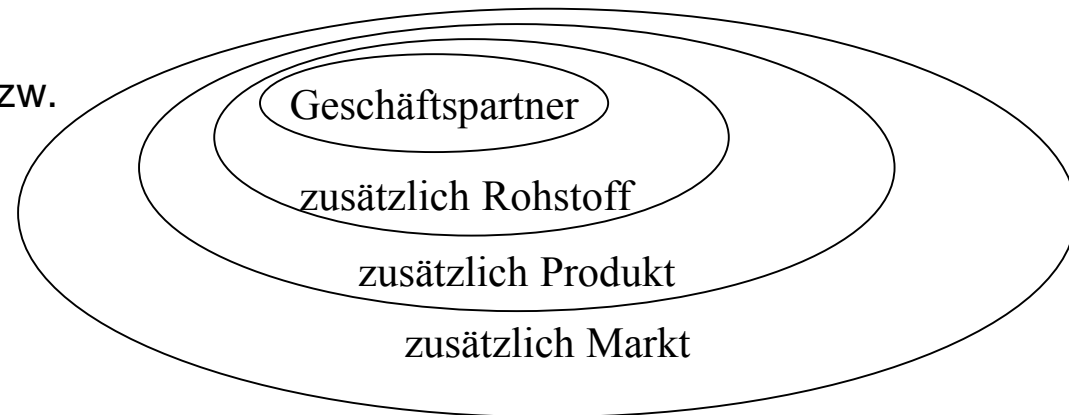
Variante 1

- Entwurf der Teilsysteme
- Anschließend: Zusammenfügen der Teilsysteme



Variante 2

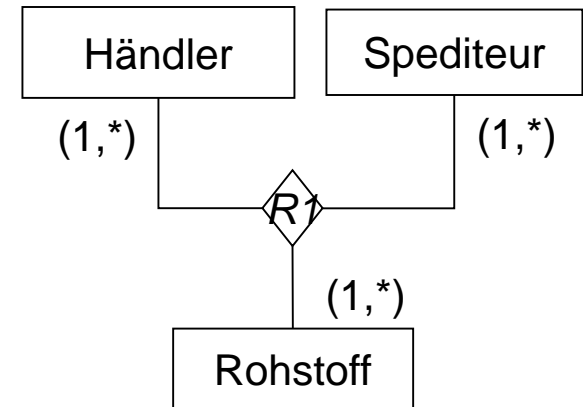
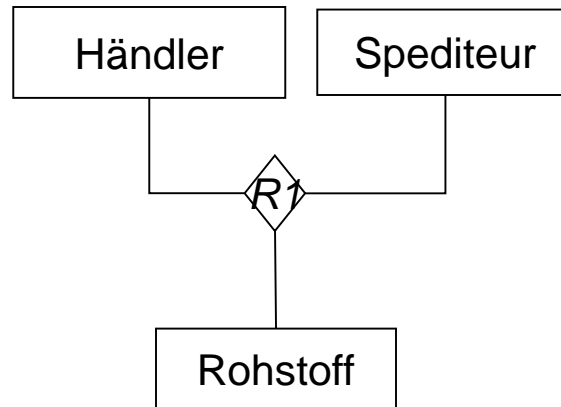
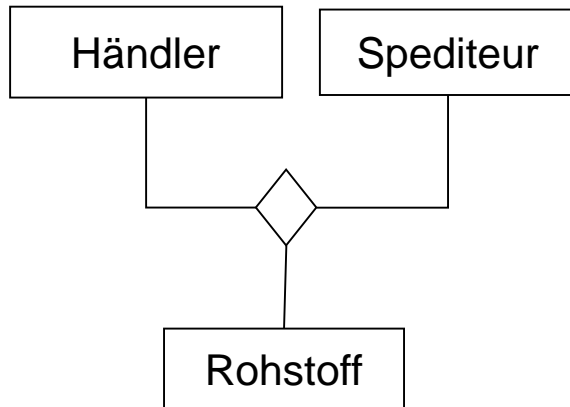
- Ausgehend von einem Objekt bzw. einem Teilsystem: Entwurf des Gesamtsystems



...



Vorgehensweise bei der ER-Modellierung



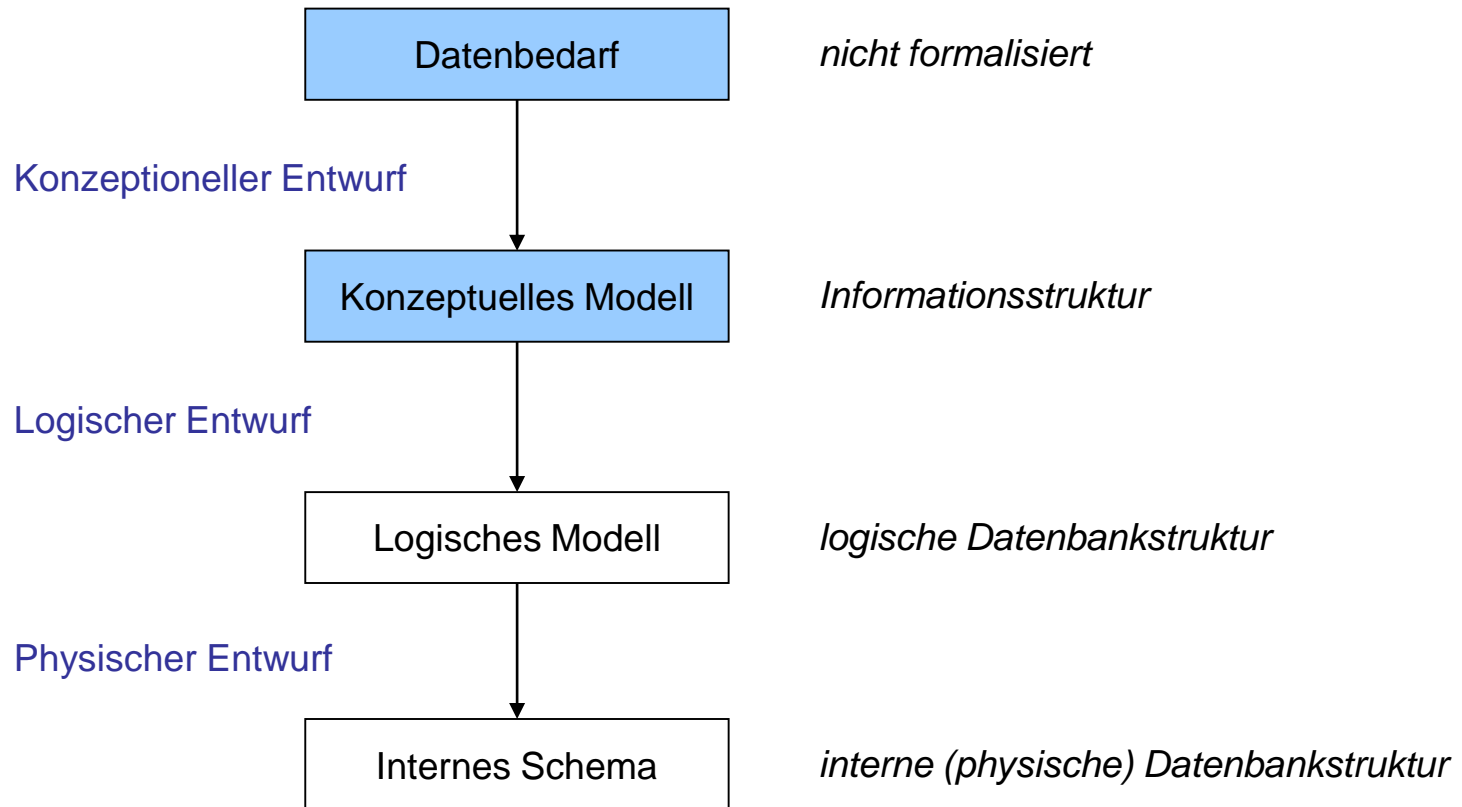
R1: liefert (Händler, Spediteur, Rohstoff;
Bestelldatum, Lieferdatum, Preis, Menge)

1. Definition der Entity-Typen (inkl. Attribute und Schlüssel) und Analyse, welche Entity-Typen miteinander in Beziehung stehen
2. Analyse, wie die Beziehungstypen bezeichnet werden und welche Attribute sie besitzen
3. Analyse, welche Kardinalitäten die Beziehungstypen haben



Phasen des Datenbankentwurfs

Anforderungsanalyse





Vorlesung Datenbanken 1

