

3 *Das Entity-Relationship-Modell*

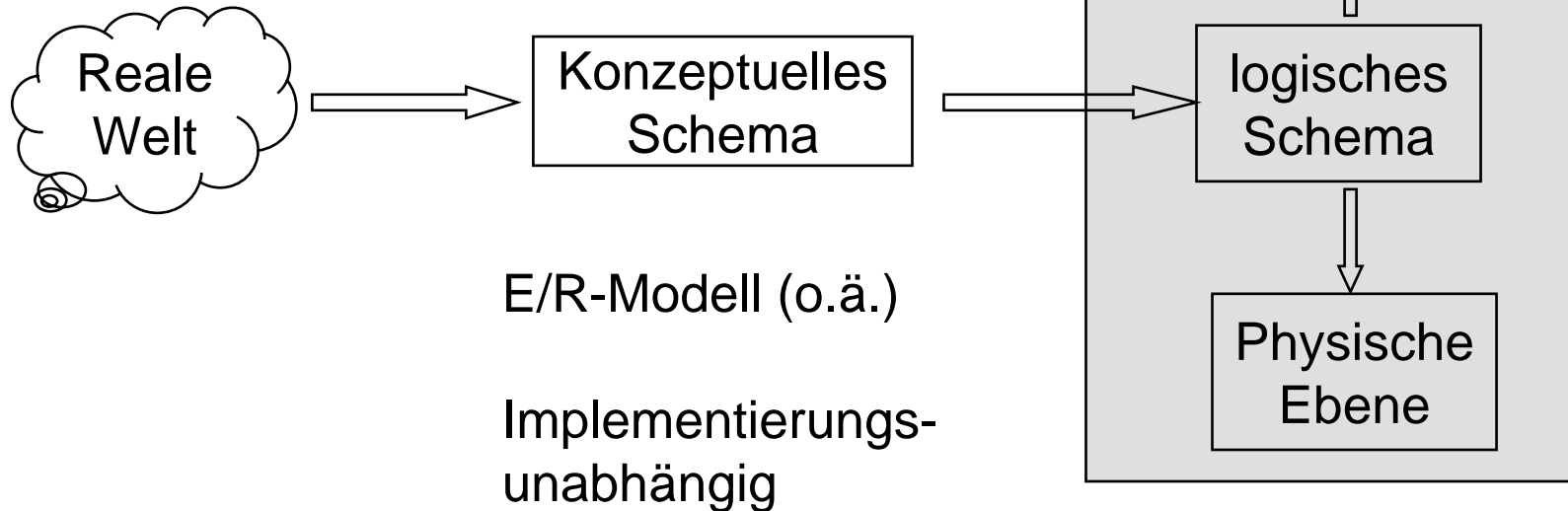
3.0	Einführung	2
3.1	Das Grundmodell	5
3.2	Erweiterungen des ER-Modells	58
3.3	Hinweise für den Aufbau von ER-Schemata	77

3 *Das Entity-Relationship-Modell*

3.0	Einführung	2
3.1	Das Grundmodell.....	5
3.2	Erweiterungen des ER-Modells	58
3.3	Hinweise für den Aufbau von ER-Schemata.....	77



3.0 Einführung



- Trennung von konzeptuellem und logischem Schema
- (Konzeptuelles Schema=) Modell des Anwendungsbereiches nach Analyse aufgrund von Abstraktion und Klassenbildung

Implementierungs-
abhängig;
existierende DBS'e

3.0 Einführung

- Entwickelt von Chen, 1976. Seitdem verschiedene Versionen und Erweiterungen gebräuchlich.
- ER-Modelle werden primär für die Beschreibung der Daten in einem Anwendungsbereich eingesetzt (datenorientierte Modellierung).
- ER-Modelle werden in der Regel von Systemanalytikern und Entwicklern entworfen.
- ER-Modelle sind einfach zu verstehen
 - gut geeignet als Kommunikationsbasis mit Endkunden oder unter Entwicklern.

3 Das Entity-Relationship-Modell

3.0	Einführung	2
3.1	Das Grundmodell	5
3.1.1	Entities, Entity-Sets, Entity-Typen	6
3.1.2	Attribute, Attributwerte	8
3.1.3	Schlüssel, Schlüsselwerte	16
3.1.4	Beziehungen und Beziehungstypen	19
3.1.5	Grafische Darstellung	24
3.1.6	Rollennamen	27
3.1.7	Beziehungstyp mit Grad > 2	28
3.1.8	Weak Entities	36
3.1.9	Kardinalität von Beziehungen	38
3.1.10	Schlüssel von Beziehung(styp)en	52
3.1.11	Zusammenhang Schlüssel / Kardinalitäten mit Grad > 2	57
3.2	Erweiterungen des ER-Modells	58
3.3	Hinweise für den Aufbau von ER-Schemata	77

3.1.1 Entities, Entity-Sets, Entity-Typen

(1|2)

Entity (e):

- Objekt der realen Welt, unterscheidbar von anderen Objekten

Entity-Set (E^t):

- Menge von Objekten zum Zeitpunkt t
(gleichartige Objekte/ charakterisiert durch gewisse Eigenschaften)
- E = Name der Menge
- $e \in E^t$ Zugehörigkeitstest

Entity-Typ (E):

- Objekt-Typ („charakteristische Eigenschaften“)
- Objekte $e \in E^t$ sind Objekte des Typs E
- E = Name des Objekttyps

3.1.1 Entities, Entity-Sets, Entity-Typen

(2|2)

Beispiele:

- Person X
 - bezieht Gehalt:
 - gehört zur Menge der Angestellten;
 - ist vom Typ „Angestellter“

- Person Y
 - hat Artikel gekauft :
 - ist vom Typ „Kunde“

Die Entity-Sets müssen nicht disjunkt sein:

Entity „W. Meyer“: sowohl Angestellter als auch Kunde

3.1.2 Attribute, Attributwerte

(1|8)

Attribut (a):

- Eigenschaft
- a = Name des Attributes

E : < A >

- Ein Entity-Typ E wird charakterisiert durch eine Menge relevanter Attribute:

$$A = \{a_1, \dots, a_n\}$$

- E : < A > ist „Schema“ für Entity-Typ E

3.1.2 Attribute, Attributwerte

(2|8)

Beispiele:

- Angestellte : $\langle \{ \text{ANG-NR, NAME, ORT, GEHALT} \} \rangle$
- Abteilung : $\langle \{ \text{ABT-NR, ABTNAME} \} \rangle$

Vereinfachung der Schreibweise durch Weglassen der Mengenklammern:

- z.B.:

Angestellte: $\langle \text{ANGNR, NAME, ORT, GEHALT} \rangle$

Student: $\langle \text{MatrNr, NAME, W-ORT, S-ORT, St'gang, FachSem} \rangle$

3.1.2 Attribute, Attributwerte

(3|8)

Domain (dom(a))

- „Domain“ = Wertebereich
- Zu jedem Attribut a gehört ein Wertebereich dom(a)
- Domains müssen **nicht disjunkt** sein, sie können **sogar gleich** sein!

Beispiele:

dom(ANG-NR) = Menge von 3-stelligen ganzen Zahlen:
100 ... 999

dom(NAME) = Menge von 25-stelligen
Buchstabenfolgen

dom(MatrnNr) = Menge der 6-stelligen ganzen Zahlen
mit einer 7. Ziffer als Prüfziffer

3.1.2 Attribute, Attributwerte

(4|8)

w_a

- einzelner Wert aus $\text{dom}(a)$
- Zum Betrachtungszeitpunkt t hat Entity $e \in E^t$ für jedes Attribut $a \in A$ einen Wert w_a aus $\text{dom}(a)$.

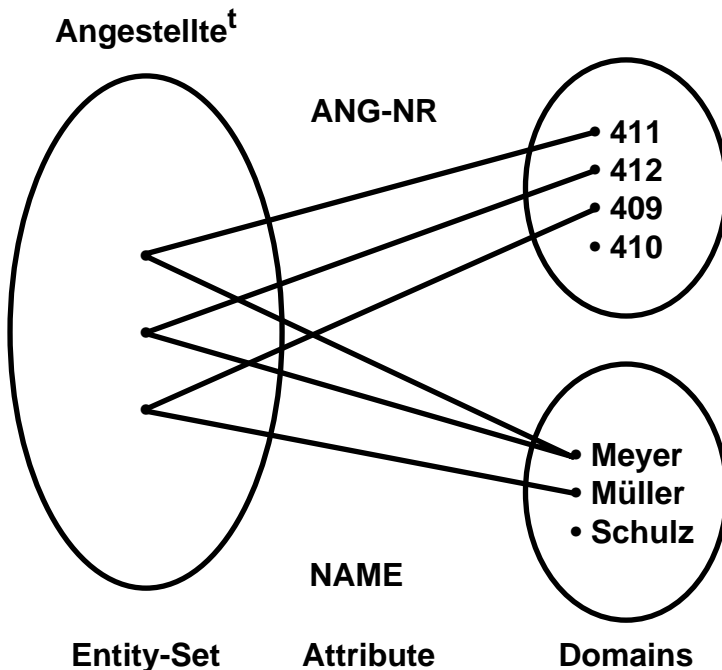
Dieser Wert kann über einen längeren Zeitraum konstant sein, er kann sich aber auch ändern.

3.1.2 Attribute, Attributwerte

(5|8)

Ein Attribut a des Entity-Typs E kann **formal** als eine Abbildung aufgefasst werden, die ein Entity-Set E^t in eine Domain abbildet.

Beispiel 7.1: $a: \text{Angestellte}^t \rightarrow \text{dom}(a)$



3.1.2 Attribute, Attributwerte

(6|8)

Zu jedem **Entity-Typ E** gehört eine **Menge A von Attributen**:

E : < A >,

zu jedem **Entity e** dieses Typs eine **Menge von Attributwerten (Attributwertkombination) w** bezüglich aller $a \in A$:

e : w, w \in „dom(A)“

und zu jedem Entity-Set E^t eine Menge X von Attributwertkombinationen w :

$E^t : X, X \subseteq \text{dom}(A); X = \{w \mid e : w \text{ für ein } e \in E^t\}$

3.1.2 Attribute, Attributwerte

(7|8)

Annahme:

feste Reihenfolge für a_i in A :

$$a_1, a_2, \dots, a_n$$

(d.h.: beliebig, aber **fest** gewählt)

Jedes Entity ist dann beschrieben durch ein geordnetes Tupel von Attributwerten w :

$$e : w, \quad w = (w_1, w_2, \dots, w_n)$$

$$\text{dom}(A) ::= \{(w_1, w_2, \dots, w_n) \mid w_i \in \text{dom}(a_i); i = 1, \dots, n\}$$

$$= \underset{\substack{| \\ w_1}}{\text{dom}(a_1)} \underset{\substack{| \\ w_2}}{\text{dom}(a_2)} \times \dots \times \underset{\substack{| \\ w_n}}{\text{dom}(a_n)}$$

3.1.2 Attribute, Attributwerte

(8|8)

Darstellung in **Tabellenform**:

Name	AngNr	Ort	Gehalt
Meyer	411	KA	4000
Meyer	412	KA	4000
Müller	409	MA	5000

e : w

e3 : (Müller, 409, MA, 5000)

3.1.3 Schlüssel, Schlüsselwerte

(1|3)

Gegeben $E: \langle A \rangle$

- Jedes Entity eines Typs ist **eindeutig** durch das zugeordnete Tupel beschrieben.

(sonst wäre A nicht charakteristisch [genug] für E!)

aber

- Oft genügt ein Teil des Tupels zur Identifikation innerhalb dieses Objekt-Typs.

Schlüssel K (engl.: key) für E : $\langle A \rangle$

3.1.3 Schlüssel, Schlüsselwerte

(2|3)

$K \subseteq A$ ist „Schlüssel“ für $E : \langle A \rangle \iff$

- (K1) K ist „identifizierende“ Attributkombination für $E : \langle A \rangle$, (d.h. verschiedene Objekte der realen Welt haben auch verschiedene Attributwerte bzgl. K).
- (K2) Es gibt keine echte Teilmenge $K' \subset K$, für die Eigenschaft K1 gilt (d.h. K ist **minimal** hinsichtlich Eigenschaft (K1)).

Beispiel:

Angestellte : $\langle \text{Name, AngNr, Ort, Gehalt, GebDatum} \rangle$

$\{\text{AngNr}\}$ ist Schlüssel

$\{\text{Name, GebDatum}\}$ *evtl.* Schlüssel

(Schreibweise auch: AngNr; Name, GebDatum ohne $\{ \}$)

3.1.3 Schlüssel, Schlüsselwerte

(3|3)

- Es kann mehrere Schlüssel, d.h. Attribute (Attributkombinationen) mit obigen Eigenschaften (K1) und (K2) geben.
Primärschlüssel (primary key):
einzigster oder fest ausgewählter Schlüssel (häufig: E#)

Üblich ist die Verwendung eines einfachen Schlüsselattributes z.B. AngNr, AbteilungsNr, ArtikelNr, ...
anstelle einer mehrstelligen Attributkombination.

Übliche Schreibweise:

E : <A>; Primärschlüssel für E ist E# (E# ∈ A)

3.1.4 Beziehungen und Beziehungstypen

(1|5)

Beziehung (relationship) (b):

Zwei oder mehr Objekte können miteinander in Beziehung stehen

Beispiele:

- Angestellter 411 *gehört* zu Abteilung 2.
b1 = (Angestellter 411, Abteilung 2)
- Teil x wird *benötigt* zur Herstellung von Teil y.
b2 = (x, y)
- Lieferant Meyer *liefert* Bauteil „Gehäuse“ für Produkt „Getriebe“.
b3 = (Lieferant Meyer, Bauteil „Gehäuse“, Produkt „Getriebe“)

Anzahl n der an einer Beziehung beteiligten Entities ist der Grad dieser Beziehung.

(n = 2: binäre Beziehung).

3.1.4 Beziehungen und Beziehungstypen

(2|5)

Menge von Beziehungen: (B^t)

Beziehungen desselben Grades und derselben Bedeutung zum Zeitpunkt t können (analog zu Entities) zu **Mengen B^t zusammengefaßt** werden.

Beispiele:

- Menge aller Zugehörigkeiten (Angestellte/Abteilung)
{(Ang.47, Abt. 11),(Ang.08, Abt. 15), ...}
- Menge aller Lieferungen (Lieferant, Bauteil, Projekt)

3.1.4 *Beziehungen und Beziehungstypen*

(3|5)

Beziehungstyp (relationship-type) (B):

Charakterisierung der Beziehungsmengen durch geeignete Eigenschaften.

B = Name des Beziehungstyps

Beispiele:

- Zugehörigkeit: Angestellter gehört zu Abteilung.
- Mitarbeit: Angestellter arbeitet an Projekt.

3.1.4 *Beziehungen und Beziehungstypen*

(4|5)

Attribute von Beziehungstypen:

Beziehungstypen können **Attribute** haben

(ebenso wie Entity-Typen).

- konkrete Beziehung: Attributwerte entsprechend

Beispiele:

- Dauer der Zugehörigkeit
- Umfang der Mitarbeit an einem Projekt
- Menge des durch einen Kunden bestellten Artikels

3.1.4 Beziehungen und Beziehungstypen

(5|5)

Beziehungstyp B:

ist gekennzeichnet durch die beteiligten Entity-Typen und ggfs. Attribute des Beziehungstyps:

B : $\langle E_1, E_2, \dots, E_n / Z \rangle$

n = **Grad** des Beziehungstyps

= Anzahl der beteiligten Objekttypen

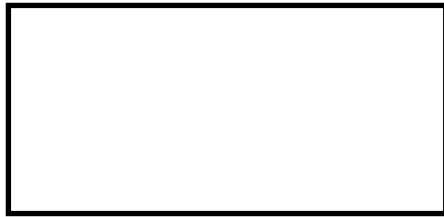
Beispiel:

Bestellung : $\langle \text{Kunde, Artikel} / \text{Menge} \rangle$

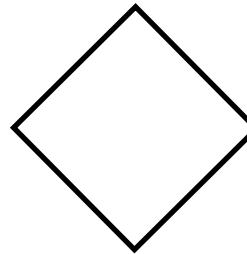
Attributwert wird einer konkreten Beziehung zugeordnet,
d.h. dient nicht zur unterscheidenden Charakterisierung der
Beziehung!

3.1.5 Grafische Darstellung

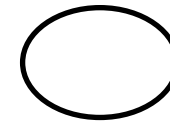
(1|3)



Entity-Typ

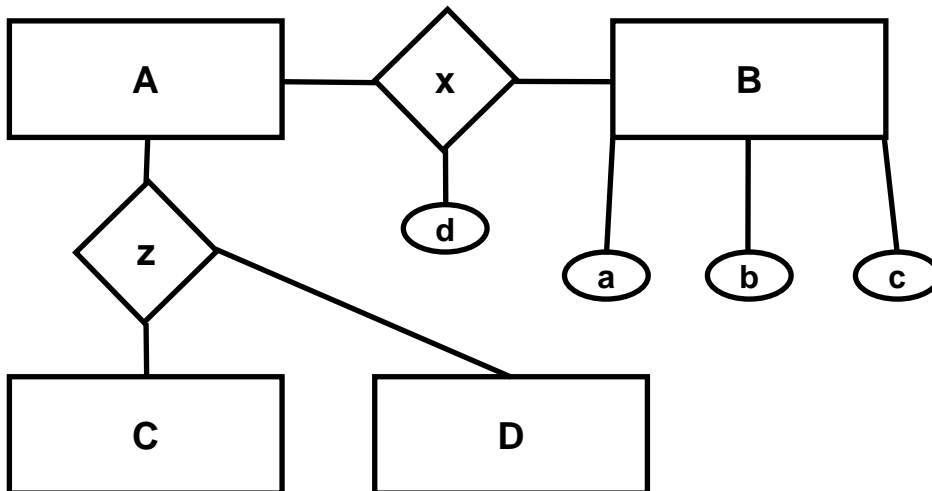


Beziehungstyp



Attribut

„E-R-Diagramm“



3.1.5 Grafische Darstellung

(2|3)

Hinweise für die grafische Darstellung

Angabe aller Attribute bei größeren Diagrammen unübersichtlich;

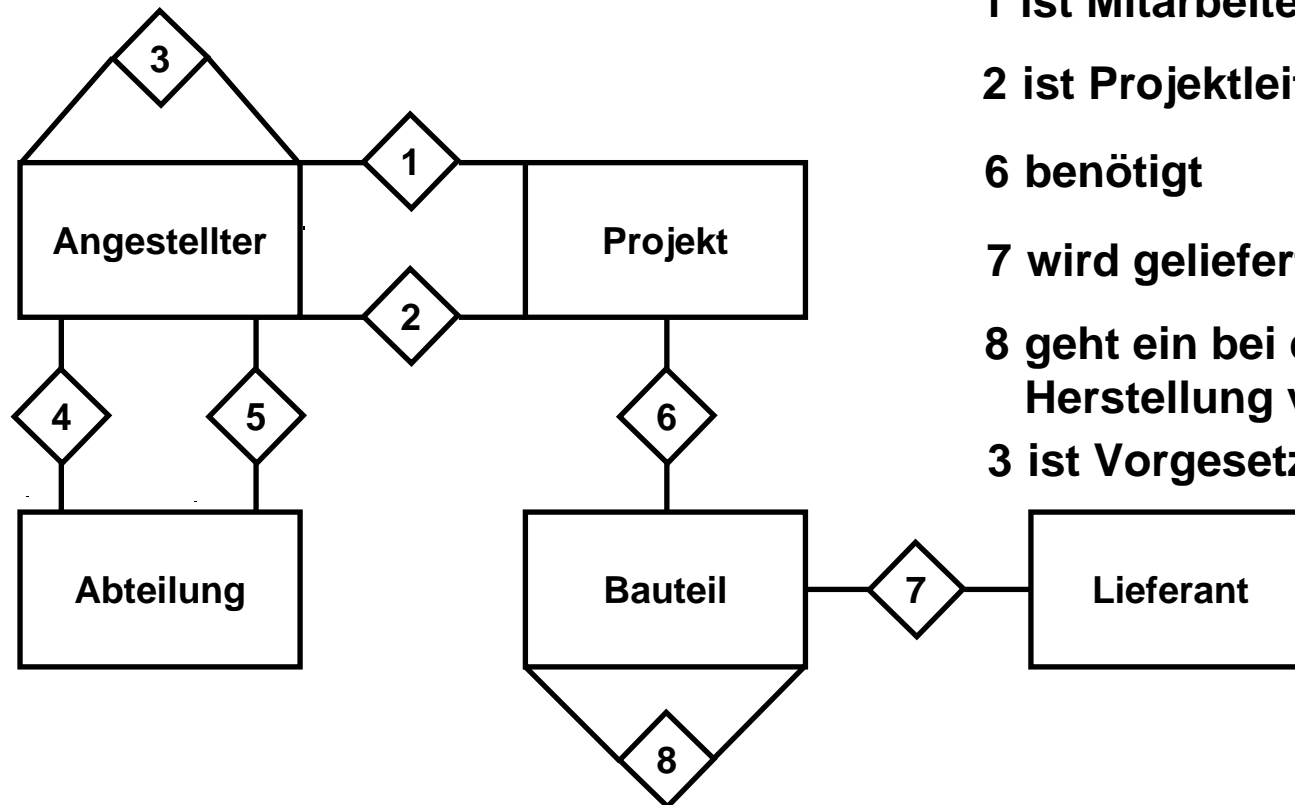
Empfehlung:

- Anfertigen von Auszügen mit Angabe der Attribute
- Im Gesamt-Diagramm eventuell lediglich Schlüsselattribute und Attribute von Beziehungstypen angeben (ohne Symbole)
- Beziehungstypen durchnummerieren und gesondert auflisten mit Angabe ihrer Bedeutung bzw. des Namens.

3.1.5 Grafische Darstellung

(3|3)

Beispiel 7.2: „ER-Diagramm“
(Ausschnitt aus einem Unternehmen)



Beziehungstypen :

4 hat Angestellte

5 wird geleitet von

1 ist Mitarbeiter an

2 ist Projektleiter von

6 benötigt

7 wird geliefert von

**8 geht ein bei der
Herstellung von**

3 ist Vorgesetzter von

3.1.6 Rollennamen

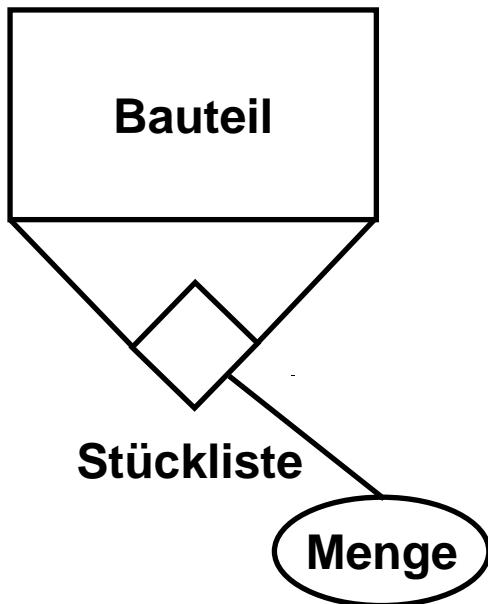
(1|1)

Ist ein **Beziehungstyp B** wie folgt definiert

B: <E1, E1>

(d.h. B ist Beziehung zwischen Entities desselben Typs), dann müssen sogenannte **Rollennamen** vergeben werden.

Beispiel:



Stückliste:

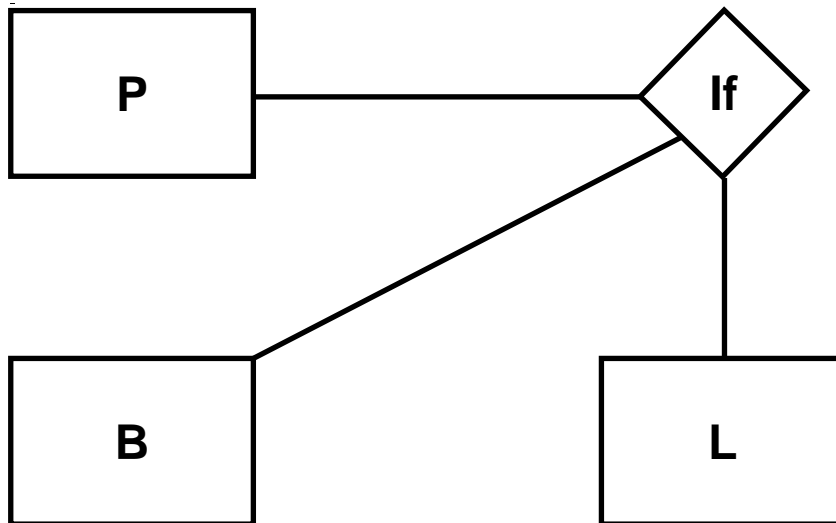
<übergeordnetes Bauteil, untergeordnetes Bauteil / Menge>

3.1.7 Beziehungstyp mit Grad > 2

(1|8)

Diskussion am **Beispiel 7.3:**

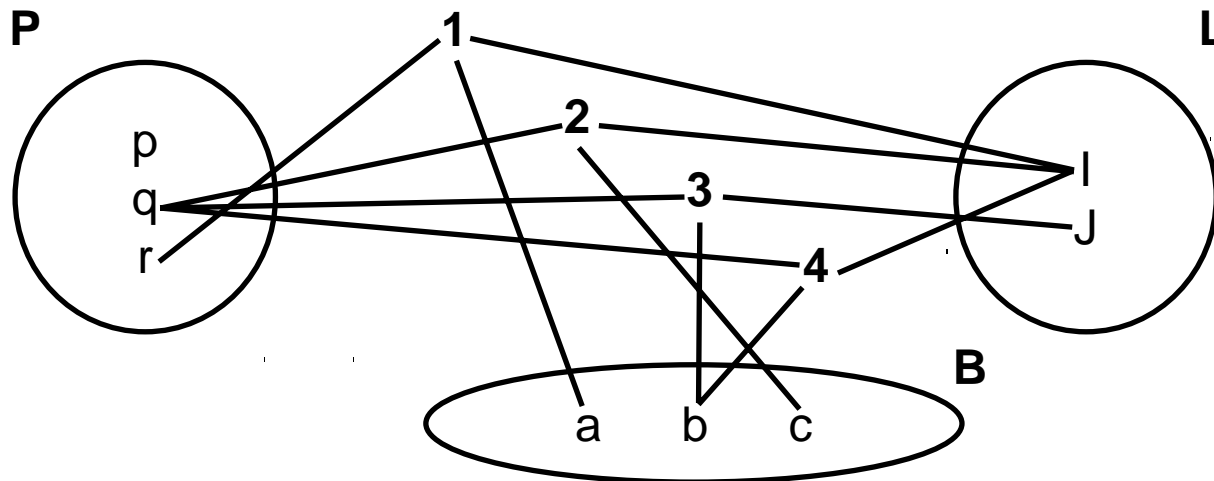
(If: L liefert B für P)



3.1.7 Beziehungstyp mit Grad > 2

(2|8)

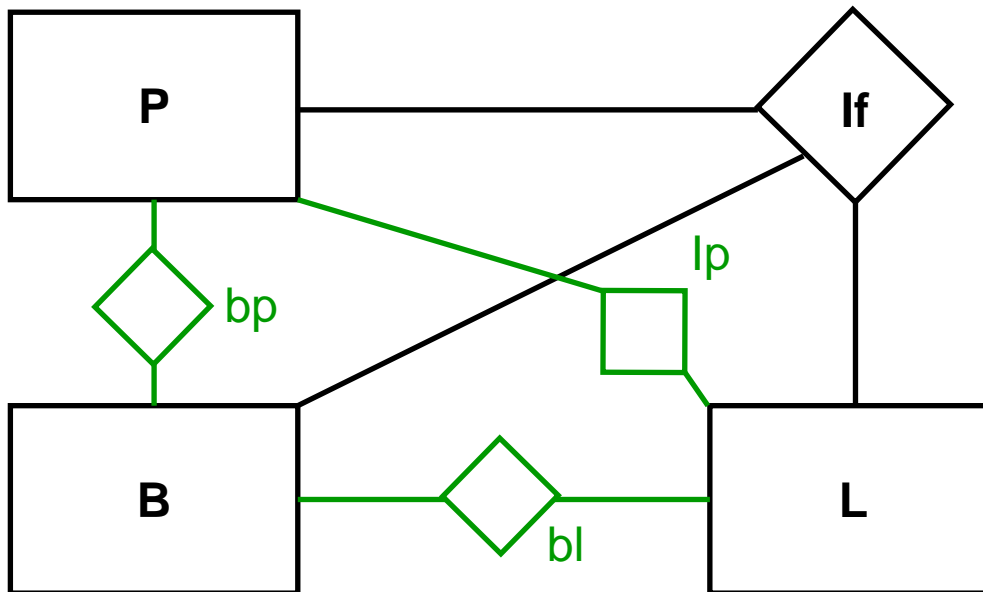
Ift	B	L	P
1	a	I	r
2	c	I	q
3	b	J	q
4	b	I	q



3.1.7 Beziehungstyp mit Grad > 2

(3|8)

Frage: Ist es möglich, die Dreier-Beziehung durch mehrere Zweierbeziehungen zu ersetzen?



3.1.7 Beziehungstyp mit Grad > 2

(4|8)

Drei Zweier-Beziehungen:

bl^t	B	L
	a	l
	c	l
	b	J
	b	l

lp^t	L	P
	l	r
	l	q
	J	q

bp^t	B	P
	a	r
	c	q
	b	q

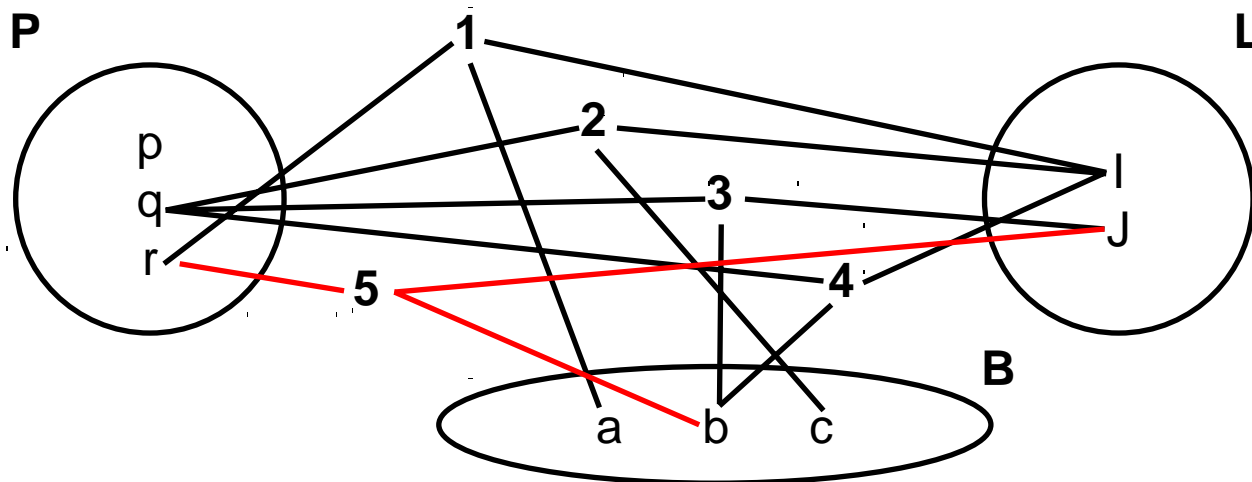
Daraus resultierende **mögliche** Dreierbeziehungen:

	B	L	P	
$bl, lp \Rightarrow bl^t * lp^t$	a	l	r	
	a	l	q	geht nicht wegen bp^t
	c	l	r	geht nicht wegen bp^t
	c	l	q	
	b	J	q	
	b	l	r	geht nicht wegen bp^t
	b	l	q	

3.1.7 Beziehungstyp mit Grad > 2

(5|8)

If ^t	B	L	P
1	a	I	r
2	c	I	q
3	b	J	q
4	b	I	q
5	b	J	r



3.1.7 Beziehungstyp mit Grad > 2

(6|8)

Drei Zweier-Beziehungen:

bl ^t	B	L
	a	l
	c	l
	b	J
	b	l

lp ^t	L	P
	l	r
	l	q
	J	q
	J	r

bp ^t	B	P
	a	r
	c	q
	b	q
	b	r

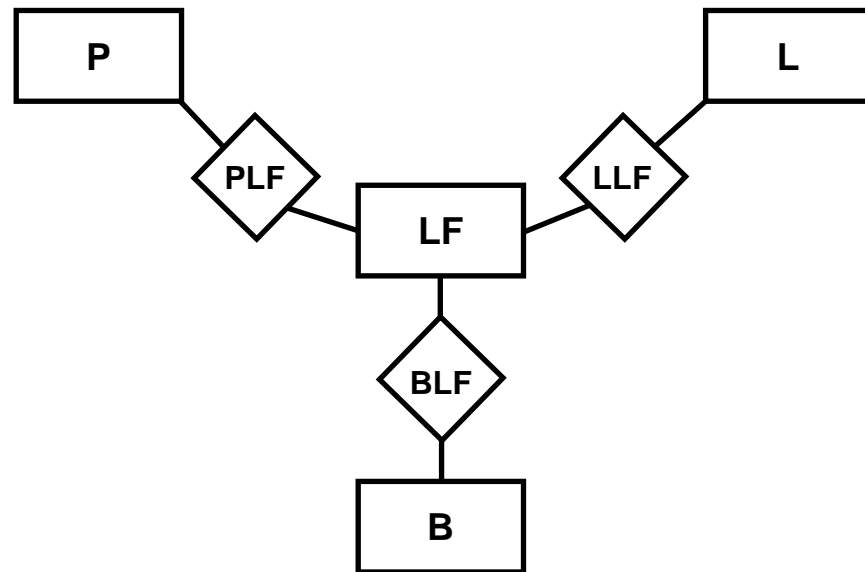
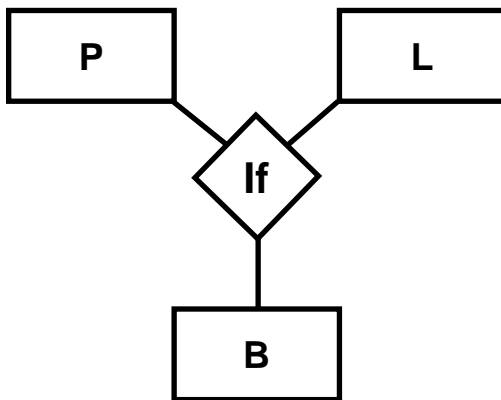
Daraus resultierende **mögliche** Dreierbeziehungen:

	B	L	P	
bl,lp ⇒ bl ^t * lp ^t	a	l	r	
	a	l	q	geht nicht wegen bp ^t
	c	l	r	geht nicht wegen bp ^t
	c	l	q	
	b	J	q	
	b	l	r	geht nicht wegen bp ^t
	b	l	q	
	b	J	r	
	b	l	r	nicht in lf ^t !

3.1.7 Beziehungstyp mit Grad > 2

(7|8)

Einzige Alternative:
Ersetze If durch LF, BLF, PLF und LLF



3.1.7 Beziehungstyp mit Grad > 2

(8|8)

LF ^t	lfd_Nr
	1
	2
	3
	4
	5

BLF ^t	B	LF
	a	1
	b	3
	b	4
	b	5
	c	2

LLF ^t	L	LF
	l	1
	l	2
	l	4
	J	3
	J	5

PLF ^t	P	LF
	q	2
	q	3
	q	4
	r	1
	r	5

Zusätzliche Semantische Integritätsbedingungen Einfügen:

a) $|BLF^t| = |LLF^t| = |PLF^t| = |LF^t|$

b) $x \in LF^t \Rightarrow x$ kommt ebenfalls vor in PLF^t, BLF^t, LLF^t

3.1.8 Weak Entities

(1|2)

Es gibt Objekte, die im Rahmen des vorliegenden konzeptuellen Schemas **(nicht selbst identifizierbar sind, sondern)** nur in Zusammenhang mit einem anderen Objekt **(dem sie zugeordnet sind)** identifizierbar sind;

z.B. „Benutzerhandbuch“ zu Projekt „LuG-Abrechnung“

Man bezeichnet diese nicht selbst identifizierbaren Objekte als **weak entities**.

Weak entity

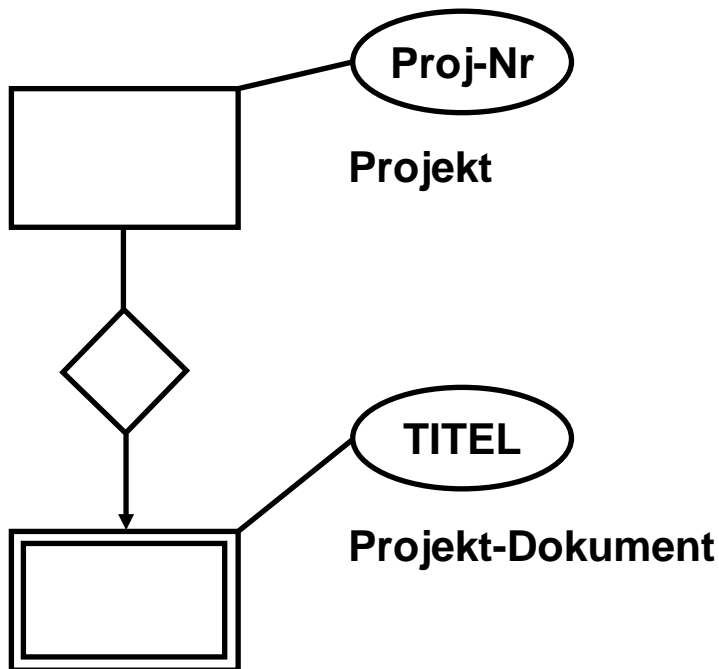
- nur mit dem übergeordneten Entity identifizierbar.
- hat keinen selbständigen Schlüssel

3.1.8 Weak Entities

(2|2)

Beispiel 7.4: Weak-Entity-Typ

Entitytyp Projekt Schlüssel: Proj-Nr
Entitytyp Projekt-Dokument Schlüssel: TITEL



Schlüssel:
Proj-Nr, TITEL

3.1.9 Kardinalität von Beziehungen

(1|15)

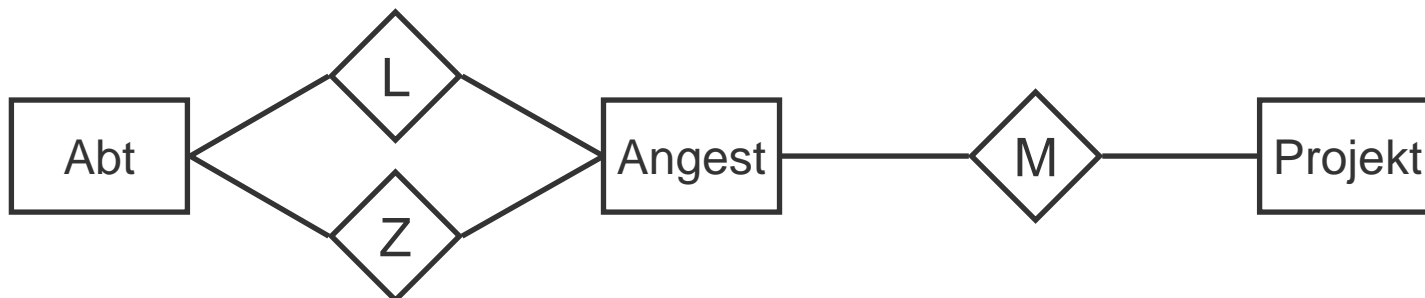
Beziehungen vom Grad 2

Üblicherweise gelten für Beziehungstypen bestimmte Bedingungen, die die möglichen Kombinationen zwischen Objekten in einer Beziehung beschränken:

Die Komplexität einer Beziehung wird durch Angabe von Kardinalitäten bestimmt.

Hier: (1,n)-Notation (auch 1:n-Notation)

Beispiel 7.5:



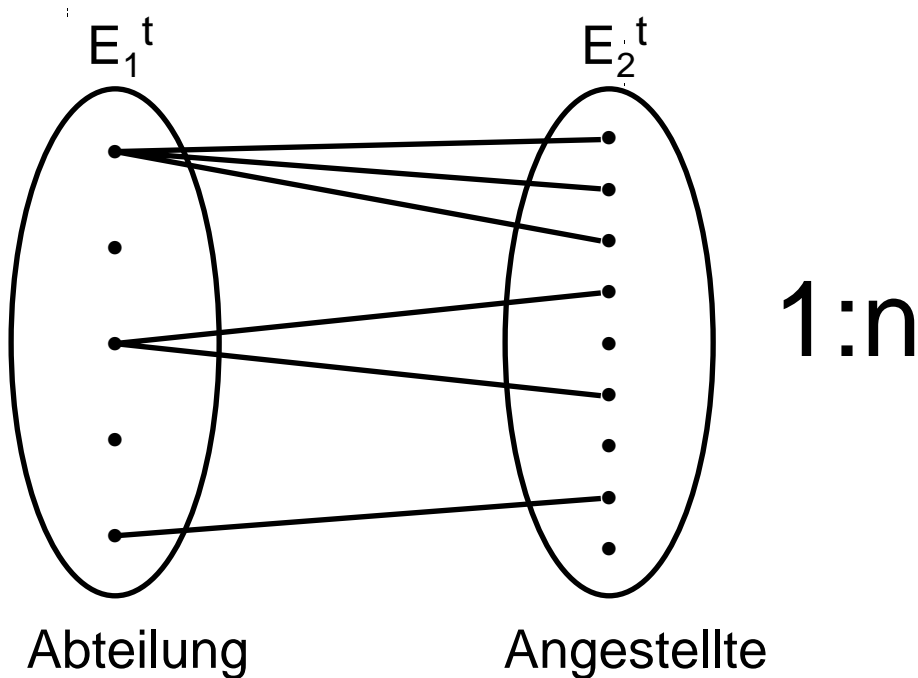
3.1.9 Kardinalität von Beziehungen

(3|15)

„ $\langle E_1, E_2 \rangle$ ist $(x : y)$ - Beziehung“ ::

(1 : n) - Beziehung zwischen E_1 und E_2 :

z.B. Abteilung - Angestellter („Abteilung hat Angestellte“)



(„Abteilung hat Angestellte“)

3.1.9 Kardinalität von Beziehungen

(4|15)

„1 : n“ ::=

- Ein Objekt des Typs E_1 kann mit einer beliebigen Anzahl von Objekten des Typs E_2 in Beziehung stehen. Ein Objekt des Typs E_2 kann mit höchstens einem Objekt des Typs E_1 in Beziehung stehen.

(„n“ steht für beliebige ganze Zahl ≥ 0 („beliebig viele“))

(n : 1) - Beziehung:

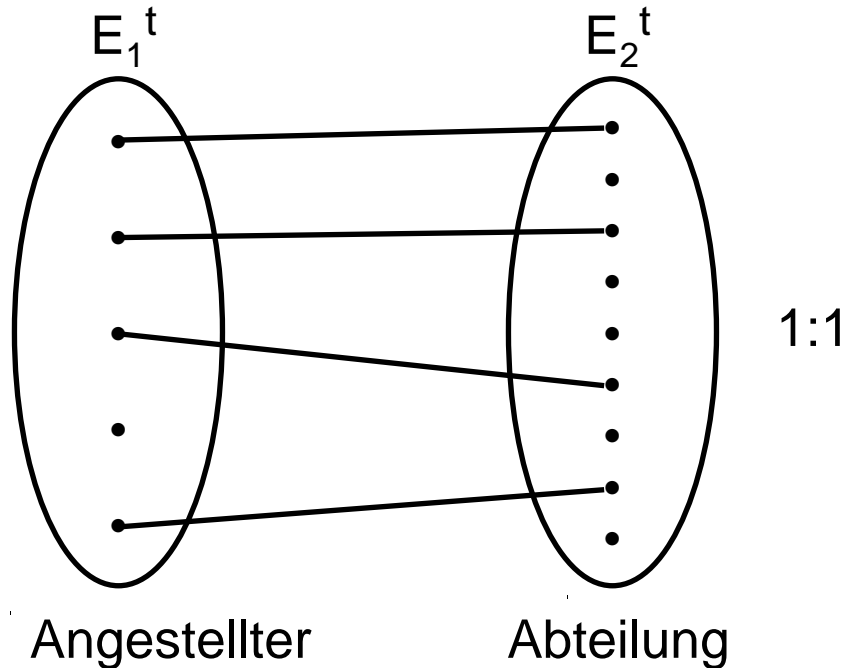
„n : 1“ ::=

- Umkehrung von 1: n

3.1.9 Kardinalität von Beziehungen

(5|15)

(1 : 1) - Beziehung zwischen E_1 und E_2 :



(„Angestellter leitet Abteilung...“)

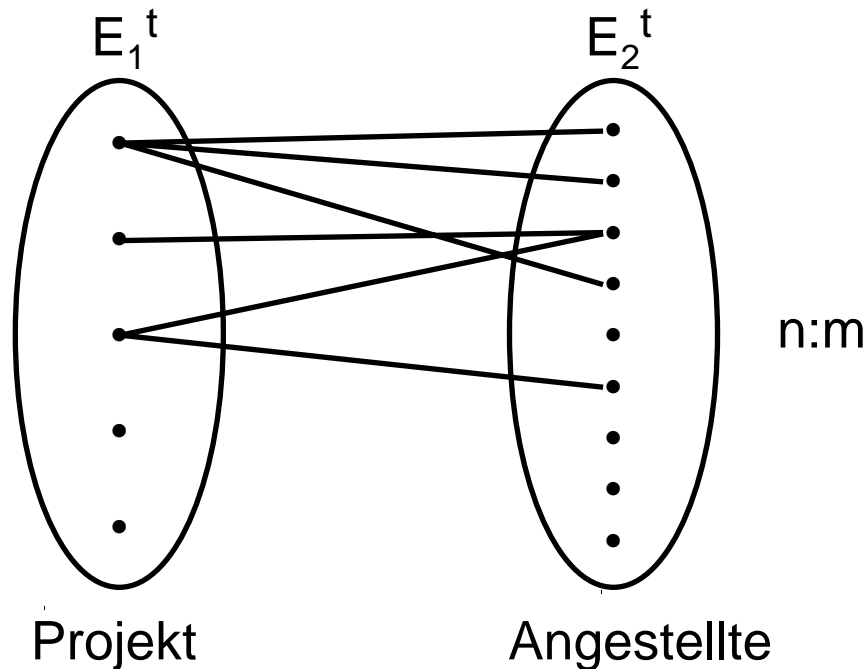
„1 : 1“ ::=

- Ein Objekt des Typs E_1 steht mit höchstens einem Objekt des Typs E_2 in Beziehung (und umgekehrt).

3.1.9 Kardinalität von Beziehungen

(6|15)

(n : m) - Beziehung zwischen E_1 und E_2 :



(„Projekt wird von Angestelltem bearbeitet...“)

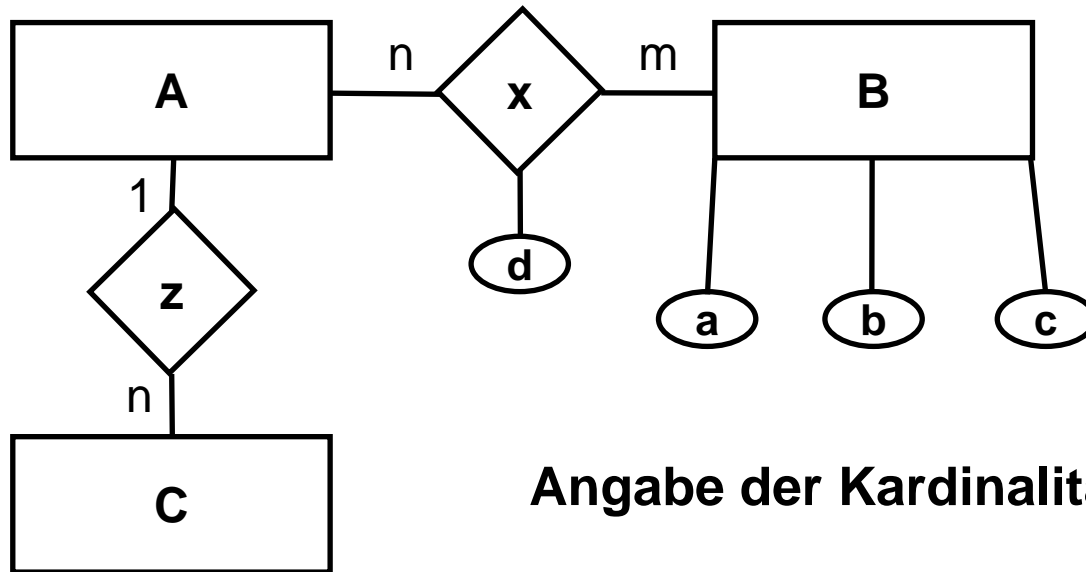
„n : m“ ::= („n“, „m“ beide für „beliebig viele“)

- Ein Entity des Typs E_1 kann mit mehreren anderen Entities des Typs E_2 in Beziehung stehen (und umgekehrt).

3.1.9 Kardinalität von Beziehungen

(7|15)

ER-Diagramm mit Kardinalitäten



Angabe der Kardinalitäten

3.1.9 Kardinalität von Beziehungen

(8|15)

Verallgemeinerbar auf **Kardinalität von Attributen:**

Attribut a - Entity-Typ E

entspricht Beziehung $B: \langle a, E \rangle$

„1:1“ ::= Zu einem Attributwert gibt es höchstens ein Entity mit diesem Wert (Schlüsseleigenschaft!)

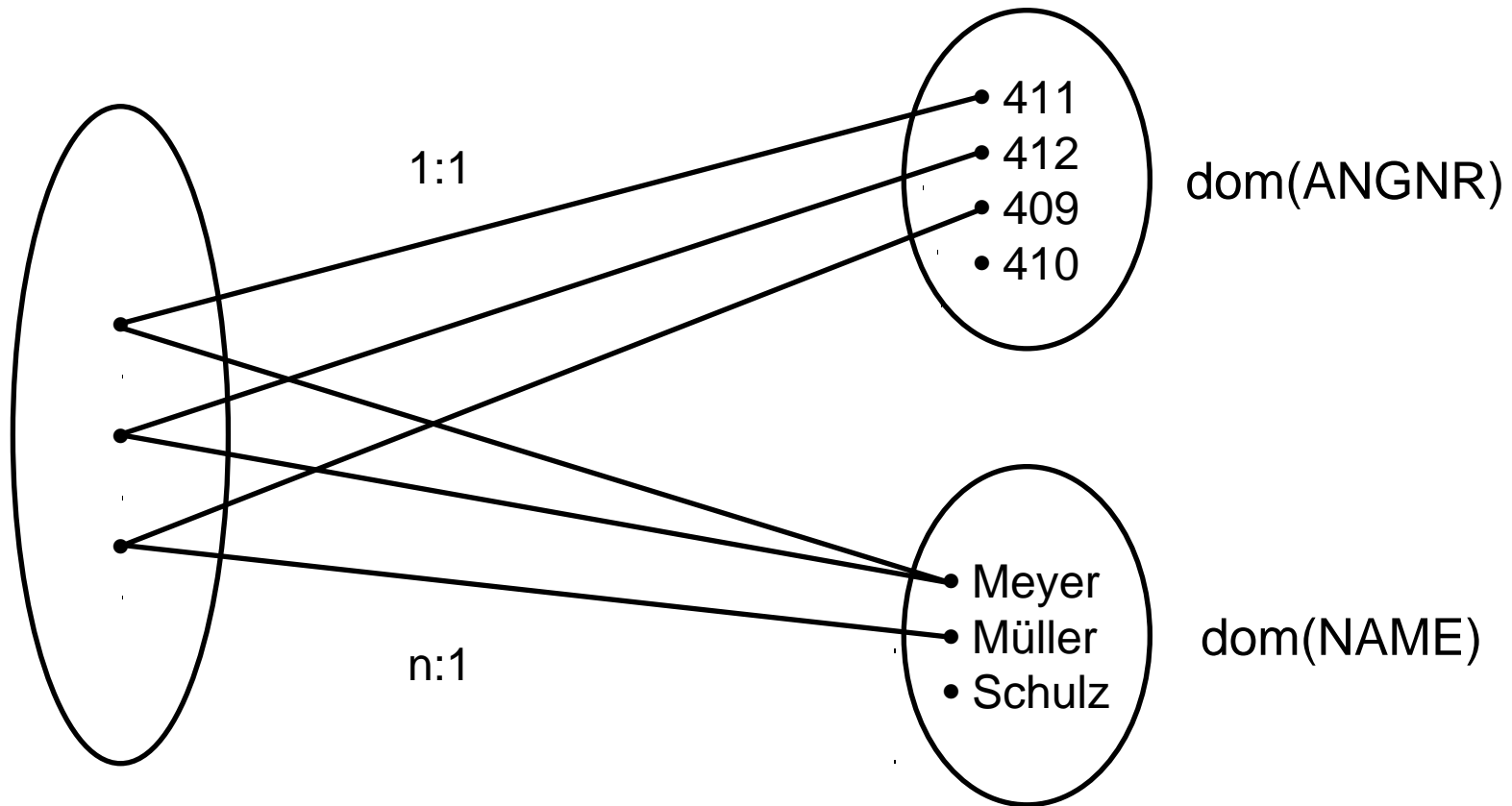
Beispiel: ANG-NR — Angestellte

„1:n“ ::= Zu einem Attributwert kann es mehrere Entities mit diesem Wert geben.

Beispiel: NAME — Angestellte

3.1.9 Kardinalität von Beziehungen

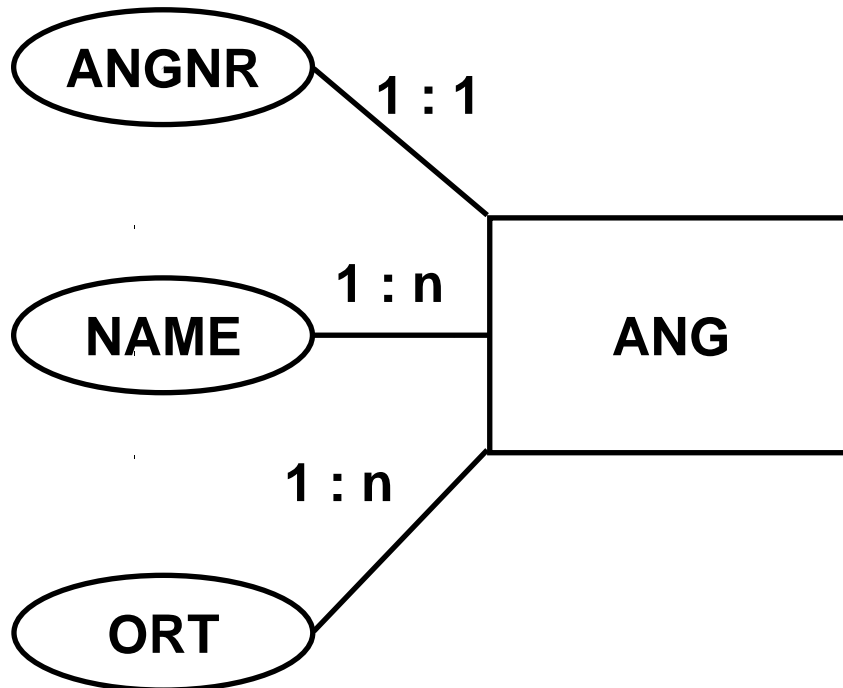
(9|15)



3.1.9 Kardinalität von Beziehungen

(10|15)

Beispiel 7.6: Attributangabe



Angabe von
Attributen

(nach Chen)

3.1.9 Kardinalität von Beziehungen

(11|15)

Beispiel 7.7: „ER-Diagramm“
(Ausschnitt aus einem Unternehmen)

Beziehungstypen :

4 hat Angestellte

5 wird geleitet von

1 ist Mitarbeiter an

2 ist Projektleiter von

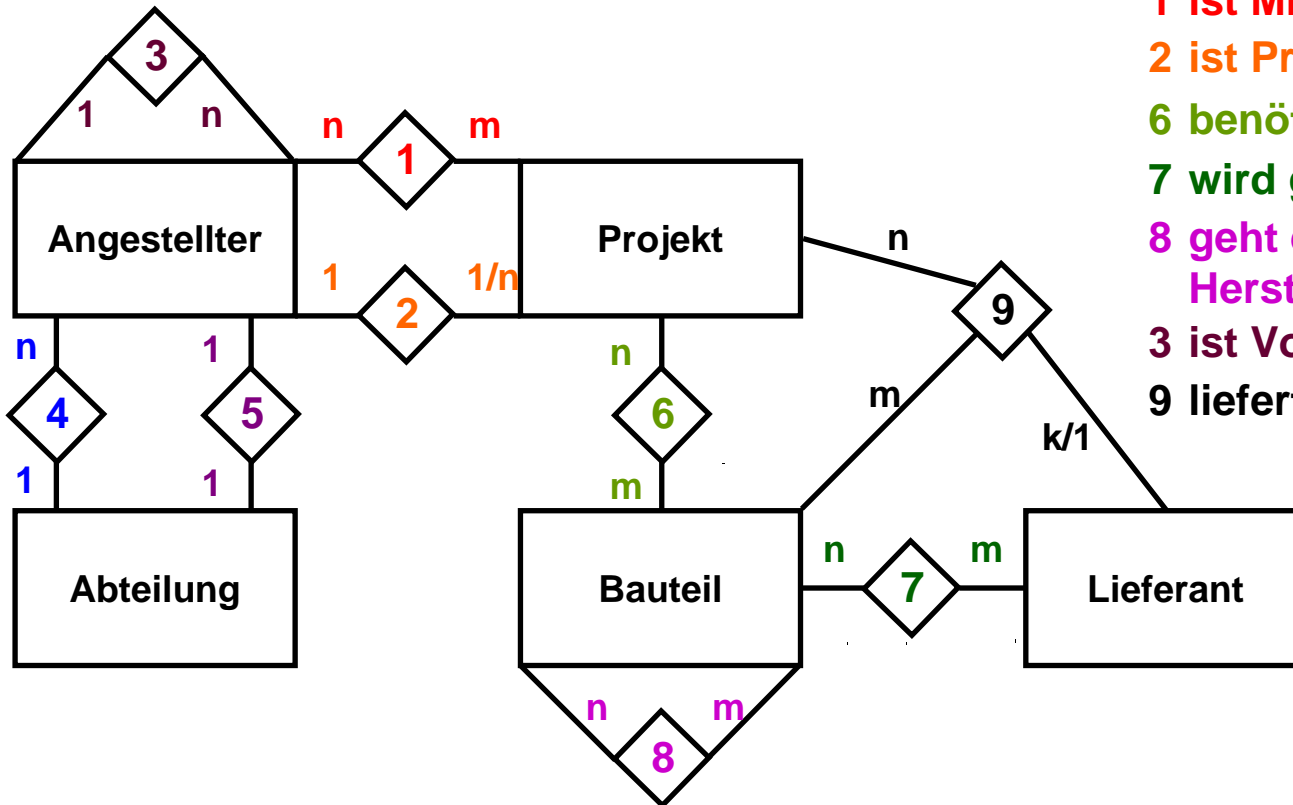
6 benötigt

7 wird geliefert von

8 geht ein bei der
Herstellung von

3 ist Vorgesetzter von

9 liefert für

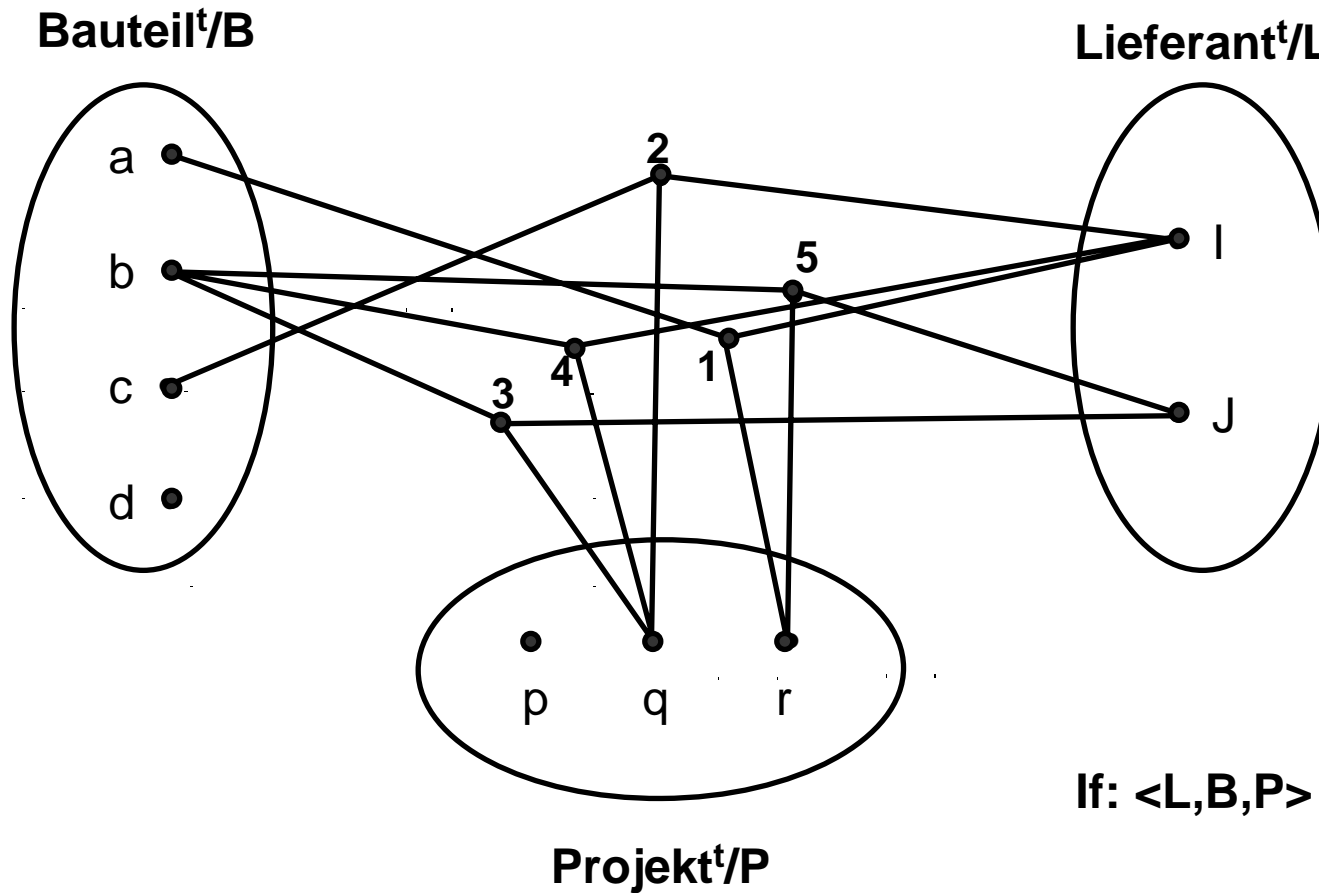


3.1.9 Kardinalität von Beziehungen

(12|15)

Dreier-Beziehung aus Beispiel 7.3:

(Verallgemeinerung auf Grad > 2)



3.1.9 Kardinalität von Beziehungen

(13|15)

If : < Projekt, Bauteil, Lieferant >

(1) Keine Einschränkung der Lieferbeziehung:

(n : m : k) – Beziehung

(2) Einschränkung:

„Für jedes Projekt darf ein- und dasselbe Bauteil nur von einem einzigen Lieferanten geliefert werden.“

(n : m : 1) - Beziehung

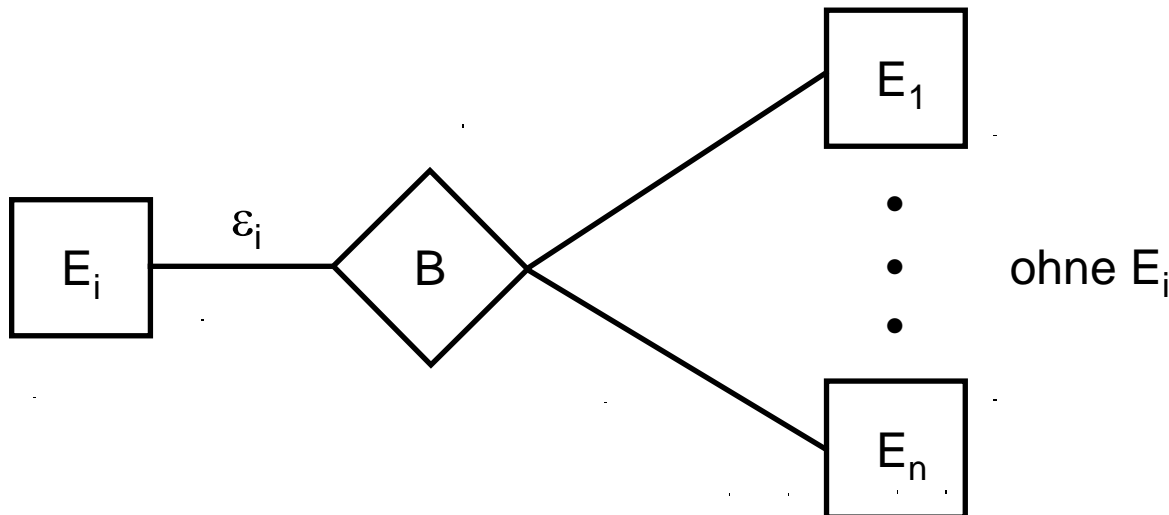
3.1.9 Kardinalität von Beziehungen

(14|15)

Kardinalität eines Entity-Typs in einer Beziehung vom Grad $n \geq 2$

$B: \langle E_1, \dots, E_i, \dots, E_n \mid Z \rangle \quad n \geq 2$

$E = \{E_1, \dots, E_i, \dots, E_n\}$ alle E_i verschieden



3.1.9 Kardinalität von Beziehungen

(15|15)

(“:“)-Kardinalität:

- B ist eine $(\varepsilon_1 : \varepsilon_2 : \dots : \varepsilon_i : \dots : \varepsilon_n)$ -Beziehung mit $\varepsilon_i = 1$ oder x ($x \in \{n, m, \dots\}$)

$$\varepsilon_i = 1 \Leftrightarrow \forall t \forall e_j \in E_j^t (j = 1, \dots, n, j \neq i)$$

\exists höchst. ein $e_i \in E_i^t: (e_1, \dots, e_{i-1}, e_i, e_{i+1}, \dots, e_n) \in B^t$

3.1.10 Schlüssel von Beziehung(styp)en

(1|5)

Jede Beziehung eines Typs ist eindeutig durch die beteiligten Entities beschrieben.

Der **Schlüssel K** eines Beziehungstyps

$$B: \langle E_1, E_2, \dots, E_n \mid Z \rangle$$

ist wie folgt definiert:

$K \subseteq \{ E_1, E_2, \dots, E_n \}$ mit

- (K1) K ist identifizierend für $B: \langle E_1, E_2, \dots, E_n \mid Z \rangle$,
d.h. verschiedene Beziehungen der realen Welt haben auch verschiedene beteiligte Entities bzgl. K.
- (K2) Es gibt keine echte Teilmenge $K' \subset K$, für die Eigenschaft (K1) gilt (d.h. K ist minimal hinsichtlich Eigenschaft (K1)).

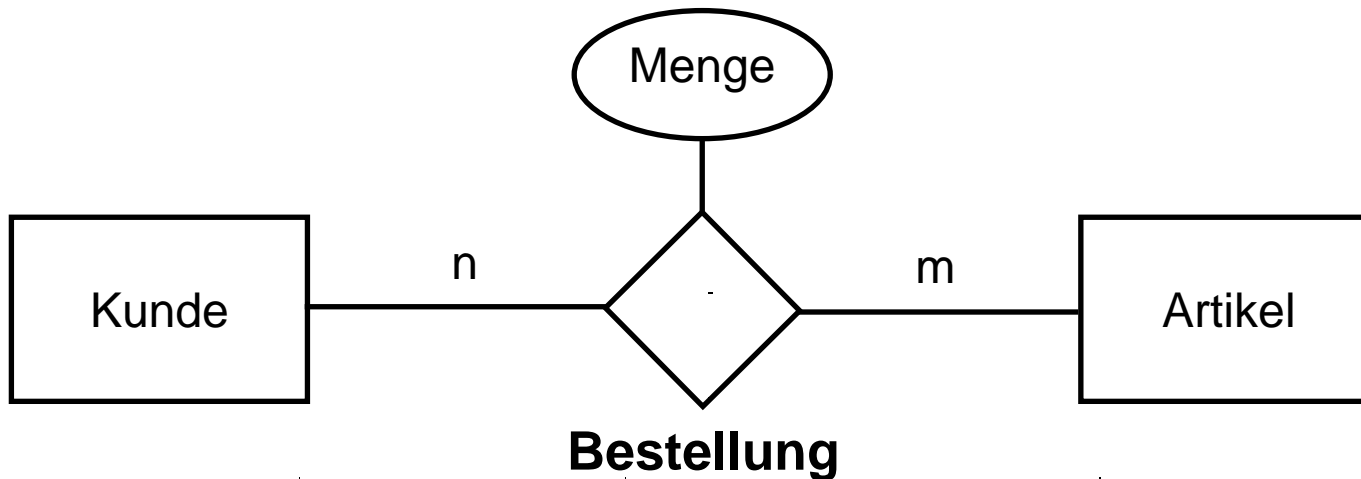
3.1.10 Schlüssel von Beziehung(styp)en

(2|5)

Beispiel: Bestellung: <Kunde, Artikel | Menge>

B: <E₁, E₂, ..., E_n | Z>

E₁, ..., E_n identifizierend für B; Schlüssel ?






Anmerkung:

Wie Entity-Typen kann man auch **Beziehungstypen** in Tabellenform darstellen.



3.1.10 Schlüssel von Beziehung(styp)en

(3|5)

Kunde ^t	K#	Name	Ort
	100	M	K
	101	W	M
	102	M	M









Schlüssel:

K#

Artikel ^t	A#	Bezeichnung
	45	X
	50	Y

Schlüssel:

A#

Bestellung ^t	Kunde	Artikel	Menge
			3
			2
			3
			4

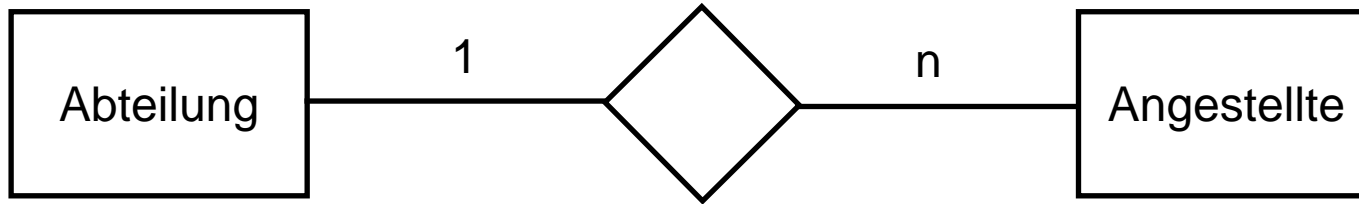
Schlüssel:

Kunde, Artikel

(„n:m“)

3.1.10 Schlüssel von Beziehung(styp)en

(4|5)



Abteilung ^t	Abt#	Abtname
	1	X
	2	Y

Angestellte ^t	A#	Name	Gehalt
	101	W	4000
	102	M	4000
	103	M	5000
	104	A	4000

Zugehörigkeit ^t	Angestellte	Abteilung

Schlüssel:
Angestellte

3.1.10 Schlüssel von Beziehung(styp)en

(5|5)

Zusammenfassung:

- (1) Bei **n : m - Beziehungen** (n : m : p - Beziehungen) setzt sich der Schlüssel aus allen beteiligten Entity-Typen zusammen.
- (2) $\langle E_1, E_2 \rangle$ sei **1 : n - Beziehung**:
Schlüssel von B ist E_2
- (3) $\langle E_1, E_2 \rangle$ sei **1 : 1 - Beziehung**:
sowohl E_1 als auch E_2 ist Schlüssel.

z.B.: Betreuung : $\langle \text{Vertreter, Bezirk} \rangle$
 : $\langle V, B \rangle$

V oder B ist Schlüssel.

Anmerkung:

Sei $B : \langle E_1, E_2, \dots, E_n \mid Z \rangle$ Beziehungstyp.

Zum Primärschlüssel von B gehört **nie** ein Attribut aus Z!

3.1.11 Zusammenhang Schlüssel / Kardinalitäten mit Grad > 2

Lemma:

- (1) $\varepsilon_i = 1 \quad \Leftrightarrow \quad E - \{E_i\}$ ist identifizierend für B
- (2) $\quad \quad \quad \Leftrightarrow \quad \exists K \subseteq E: K$ Schlüssel für B, $E_i \notin K$
- (3) $\varepsilon_i = m (\neq 1) \Leftrightarrow \forall K \subseteq E: (K$ Schlüssel für B $\Rightarrow E_i \in K)$

Beweis:

$$(1) \quad \varepsilon_i = 1 \quad \Leftrightarrow \quad \left(\left((e_1, \dots, e_{i-1}, e_i, e_{i+1}, \dots, e_n) \in B^t \text{ und} \right. \right. \\ \left. \left. (e_1, \dots, e_{i-1}, e'_i, e_{i+1}, \dots, e_n) \in B^t \right) \Rightarrow e_i = e'_i \right) \\ \Leftrightarrow \quad E - \{E_i\} \text{ ist identifizierend für B}$$

- (2) $E - \{E_i\}$ ist identifizierend für B
 $\Rightarrow E - \{E_i\}$ enthält Schlüssel K; d.h. $E_i \notin K$.
K Schlüssel für B, $E_i \notin K \Rightarrow K \subseteq E - \{E_i\}$

- (3) Negation von (2)