

Datenbanken

Informatik, ICS und als Wahlfach

4. Logisches (relationales) Modell

Prof. Dr. Markus Goldstein

SoSe 2022

4.1 Überblick Transformation

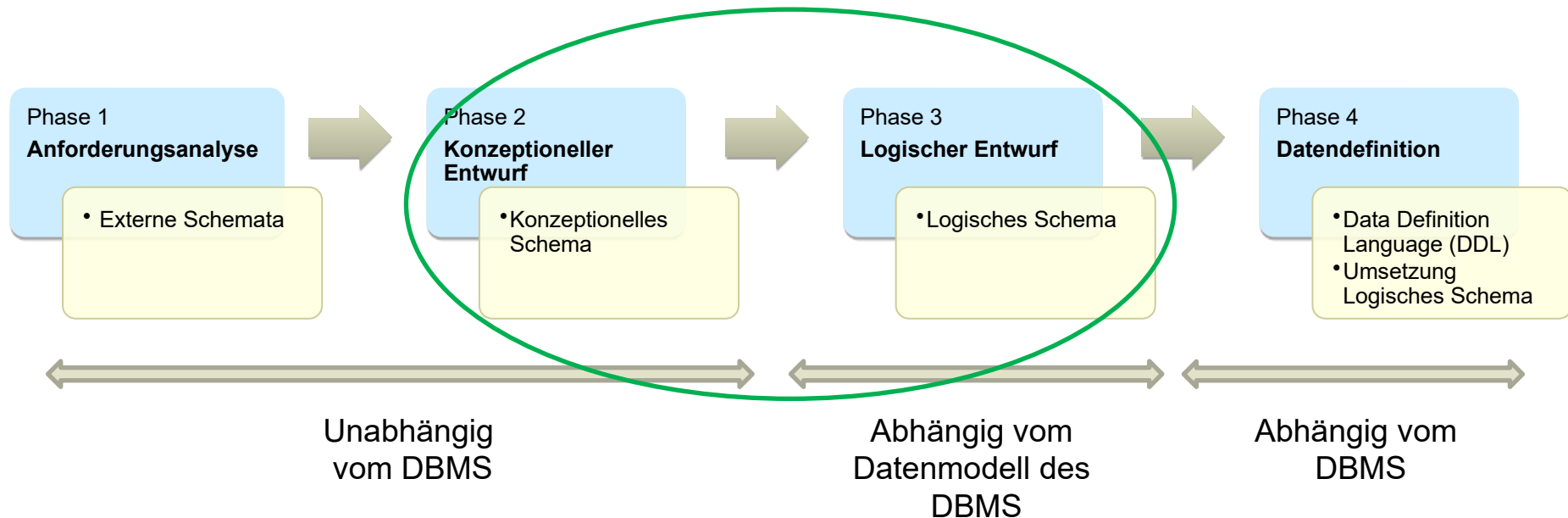
4.2 Duale Beziehungstypen

4.3 Rekursive Beziehungstypen

- Konzeptionelles Schema
 - unabhängig von der Datenbank bzw. dem Datenbankmodell
 - Trennung zur externen Ebene: Logische Datenunabhängigkeit
- Logisches Schema
 - Abhängig vom Datenmodell (z.B. relational)
 - Erstellt aus dem konzeptionellen Schema
 - Physische Datenunabhängigkeit zur internen Ebene
- Entity-Relationship-Modell
 - Beschreibungssprache für **konzeptionelles und logisches** Schema
 - Meta-Modell (Fachkonzept) für Objekt- und Beziehungstypen

Phasenmodell für Datenbankentwurf

- Dieses Kapitel:
Vom konzeptionellen Schema zum logischen Schema

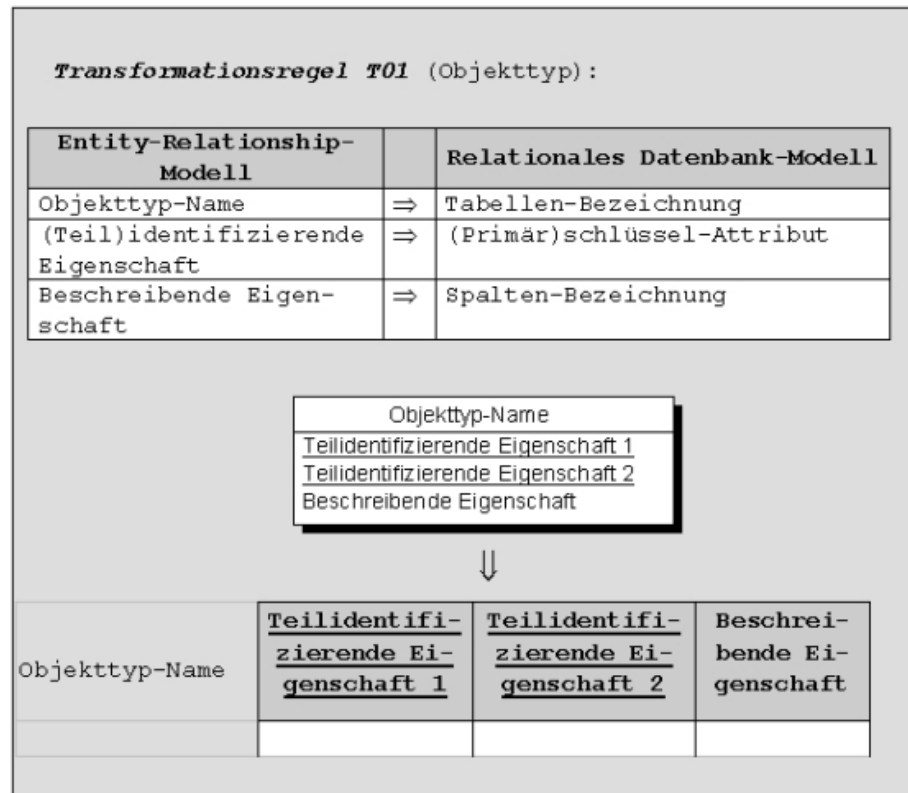


(angelehnt an Thomas Kudraß, Taschenbuch Datenbanken, 2. Auflage, Hanser Verlag 2015, S. 45 ff)

- Sie haben ein **konzeptionelles Datenmodell** für Ihren Kiosk erstellt (z.B. mit dem SAP PowerDesigner).
- Um später mit Hilfe von SQL auf die Daten zugreifen zu können, entscheiden Sie sich für eine **relationale Datenbank**.
- Hierzu muss das konzeptionelle Modell aber zunächst in ein logisches Modell transformiert werden.
- **Hierfür benutzen wir im folgenden Transformationsregeln (Txx)**
– einige sind Ihnen ggf. schon bekannt

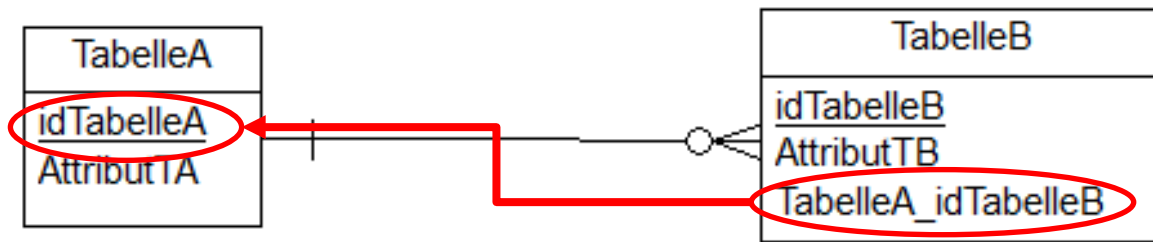
T1: Regel für Entitätstypen

- Jeder Entitätstyp wird im RDBM zu einer Tabelle
 - Attribute werden zu Spalten
 - (Teil) identifizierende Attribute werden zu Schlüsselattributen
→ Primärschlüssel



T1: Regel für Beziehungstypen

- Beziehungstypen werden über **Fremdschlüssel** modelliert (ugs: „Verweis/Link“ auf A)



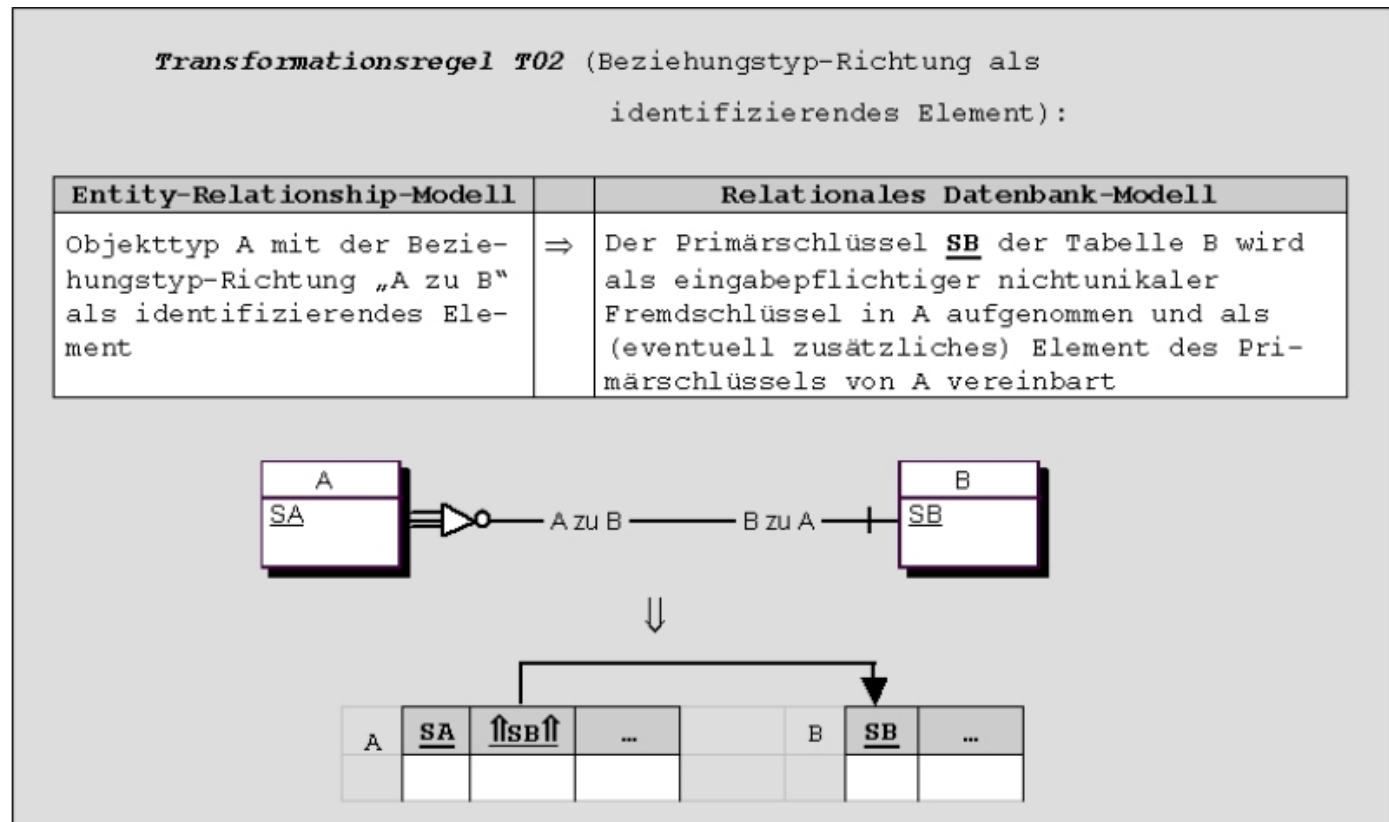
- Fremdschlüssel werden mit Pfeilen dargestellt: $\uparrow f k \uparrow$
- Zusammengesetzte Schlüssel werden mit „+“ dargestellt:
 $\uparrow \text{name} + \text{vorname} + \text{geb_Datum} \uparrow$
- Primärschlüssel werden unterstrichen:
name+vorname+geb_Datum

Im Folgenden werden zwei Charakteristika für Spalten verwendet

- „**Unikal**“: **UNIQUE** – Attributwert darf nur einmal in der Tabelle vorkommen
 - Ein Primärschlüssel ist immer UNIQUE
- „**Eingabepflicht**“: **NOT NULL** – Attributwert darf nicht ausgelassen werden
 - Ist z.B. ein Fremdschlüssel verpflichtend, muss dieser bei jeder Entität vorhanden sein → keine Optionalität möglich

T2: Schwache Entitätstypen

- Bei **schwachem Entitätstyp**
 - Fremdschlüssel wird Teil des Primärschlüssel
 - Identifizierende Beziehung (durchgezogene Linie in MWB)





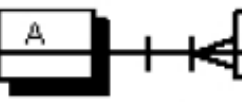
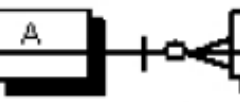


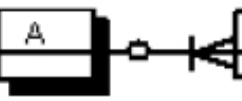
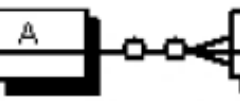








4.1 Überblick Transformation

4.2 Duale Beziehungstypen

4.3 Rekursive Beziehungstypen

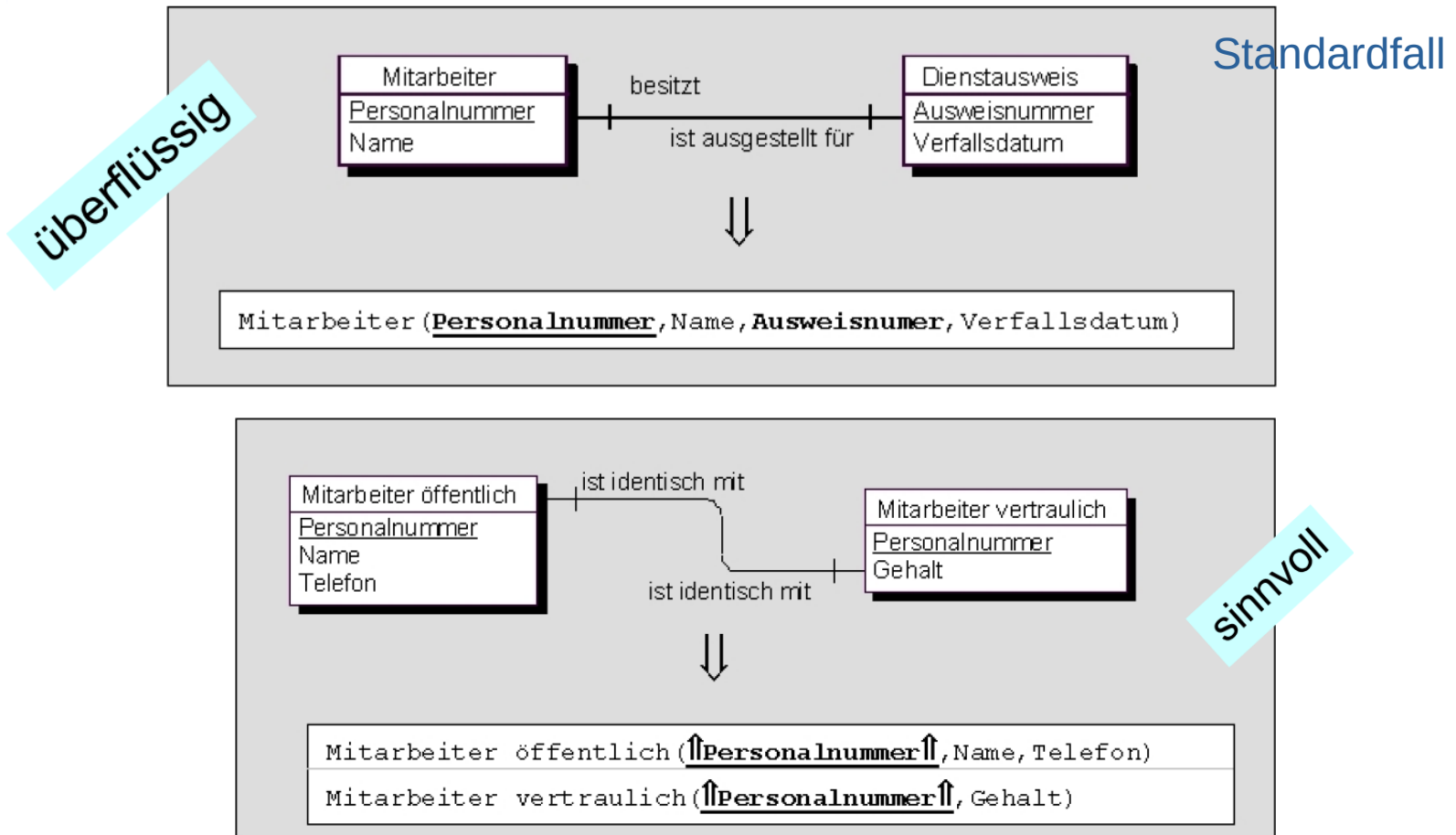
Duale Beziehungstypen

- Wir müssen die folgenden Fälle betrachten ...

A→B		Kardinalität 1		Kardinalität N	
B		nichtoptional	optional	nichtoptional	optional
1	n	 1:1 (1)	 1:C (2)	 1:N (6)	 1:CN (4)
	o	 C:1	 C:C (3)	 C:N (7)	 C:CN (5)
N	n	 N:1	 N:C	 M:N (10)	 M:CN (9)
	o	 CN:1	 CN:C	 CM:N	 CM:CN (8)

1:1 Beziehungstypen

- Unterscheidung nach Semantik
 - Überflüssig/ sinnvoll**

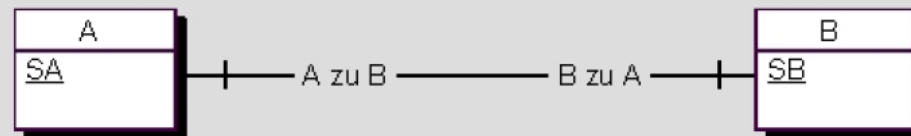


T3: 1:1 Beziehungstyp (überflüssig)

- Beziehungstyp **überflüssig**
 - Eine Tabelle/ Entitätstyp

Transformationsregel T03 (überflüssiger 1:1-Beziehungstyp):

Entity-Relationship-Modell		Relationales Datenbank-Modell
Objekttyp A mit Schlüssel <u>SA</u>	⇒	Tabelle A mit Primärschlüssel <u>SA</u>
Objekttyp B mit Schlüssel <u>SB</u>	⇒	Alle Eigenschaften werden in A aufgenommen. SB wird als eingabepflichtig und unikal vereinbart
1:1-Beziehungstyp	⇒	wird nicht gesondert dargestellt
Speicherbedarf für Beziehungstyp		0



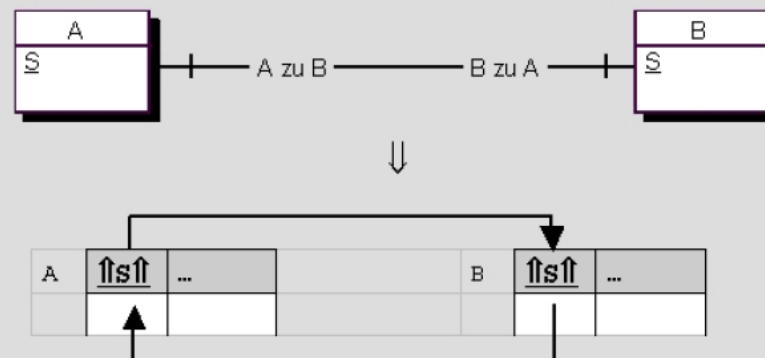
A	<u>SA</u>	...	SB	...

T4: 1:1 Beziehungstyp (sinnvoll)

- Beziehungstyp **sinnvoll**
 - Zwei Tabellen/ Entitätstypen mit gemeinsamen Primär-/Fremdschlüssel

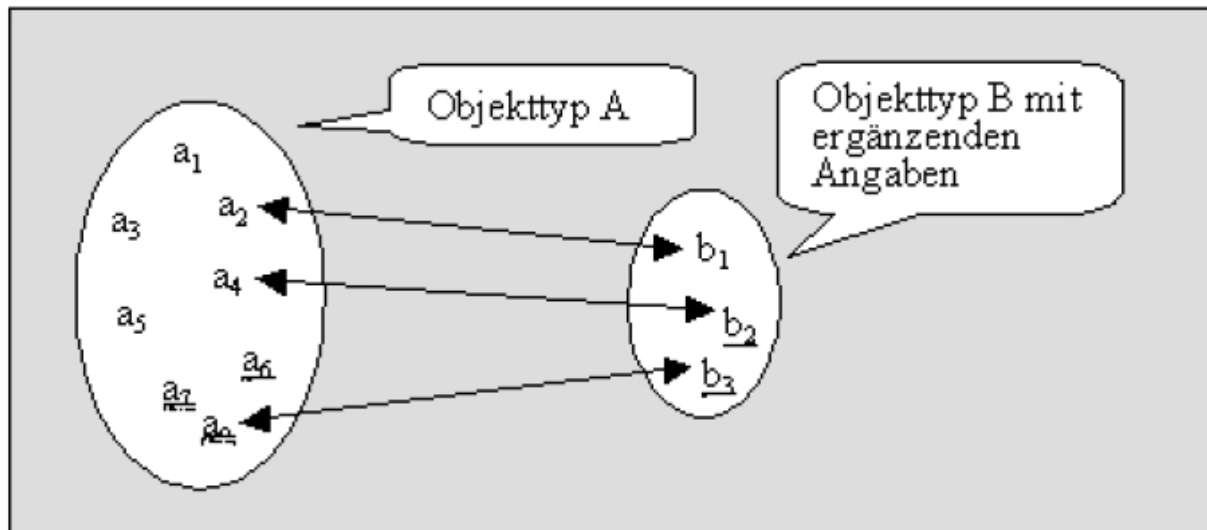
Transformationsregel T04 (sinnvoller 1:1-Beziehungstyp):

Entity-Relationship-Modell		Relationales Datenbank-Modell
Objekttyp A mit Schlüssel <u>S</u>	⇒	Tabelle A mit Primärschlüssel <u>S</u>
Objekttyp B mit Schlüssel <u>S</u>	⇒	Tabelle B mit Primärschlüssel <u>S</u>
1:1-Beziehungstyp	⇒	<u>S</u> wird sowohl in A als auch in B als unikal eingabepflichtiger Fremdschlüssel vereinbart
Speicherbedarf für Beziehungstyp		$\overline{B} \cdot \text{Len}(S)$



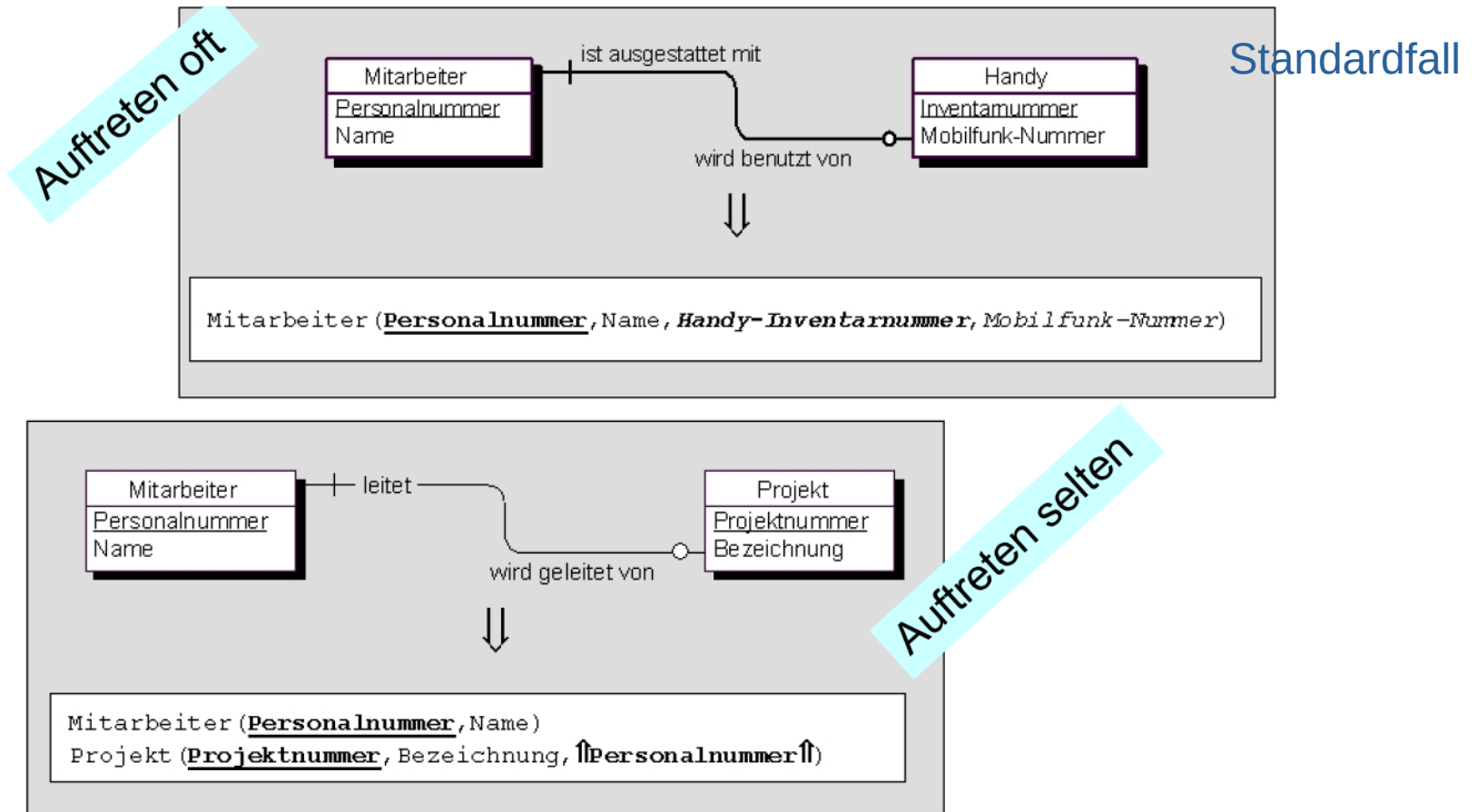
1:C Beziehungstypen

- Unterscheidung nach Auftreten
 - Selten (die meisten A's haben keine Beziehung zu B's)
 - Oft (Fast alle A's haben eine Beziehung)



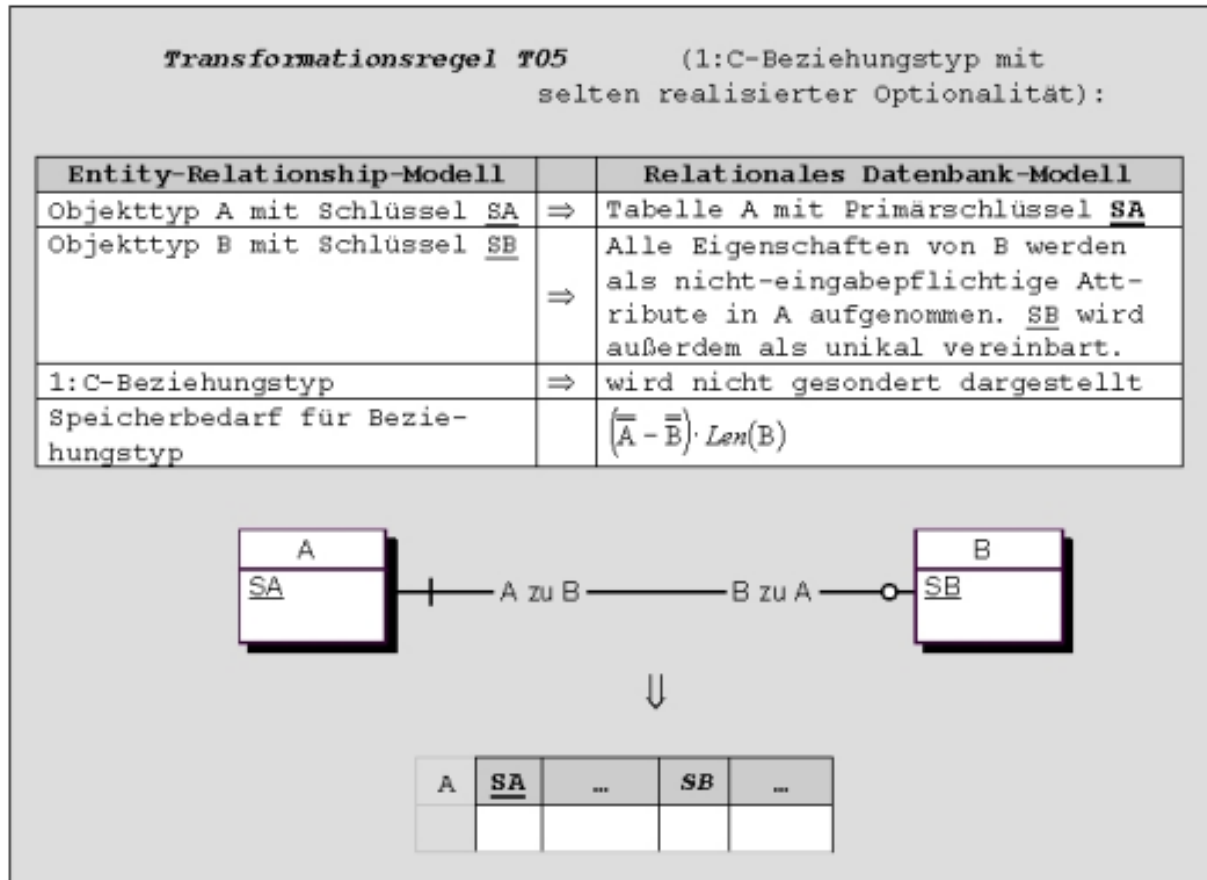
1:C Beziehungstypen

- Beispiele „oft“ vs „selten“



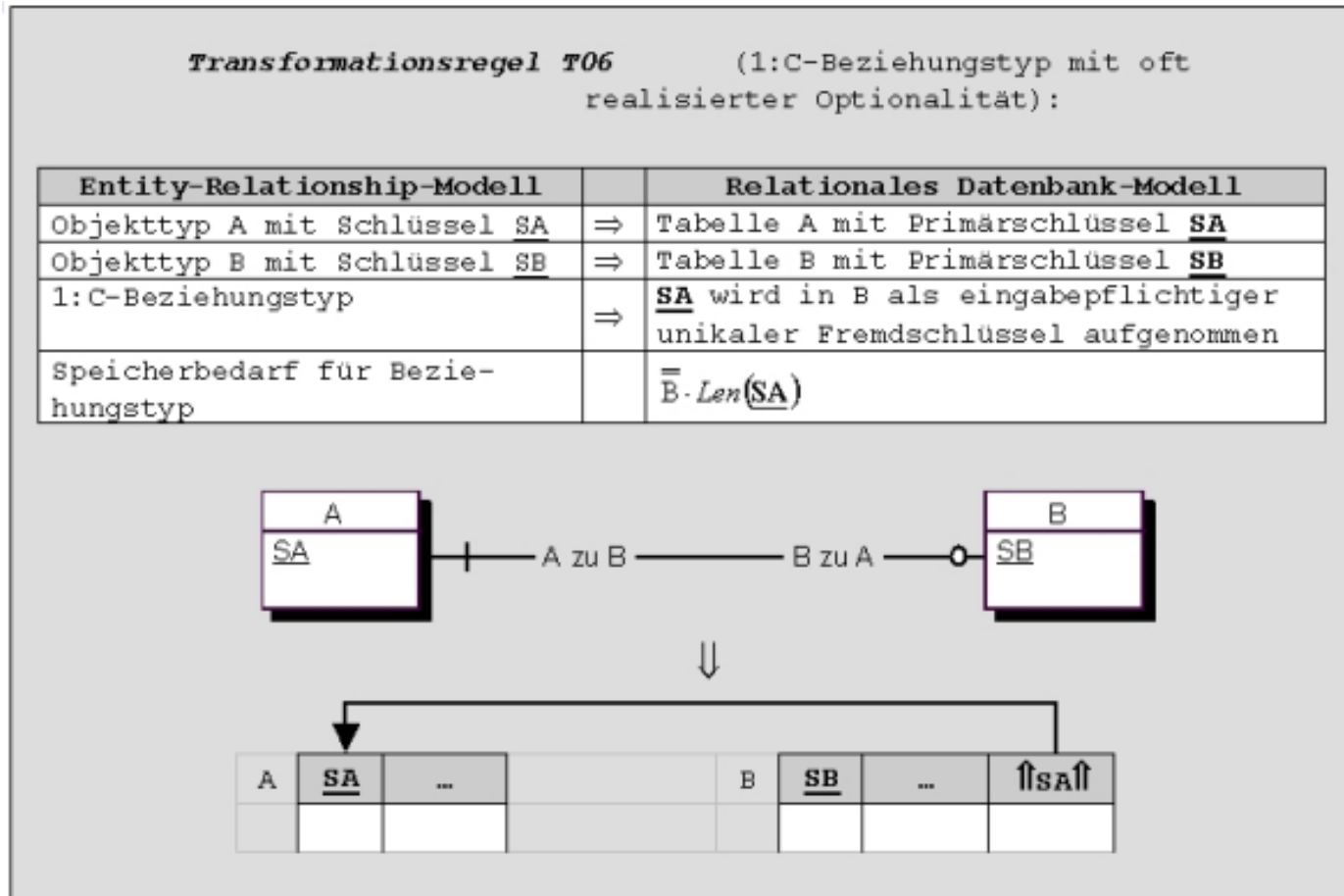
T5: 1:C Beziehungstypen (oft)

- Auftreten „oft“ → eine Tabelle



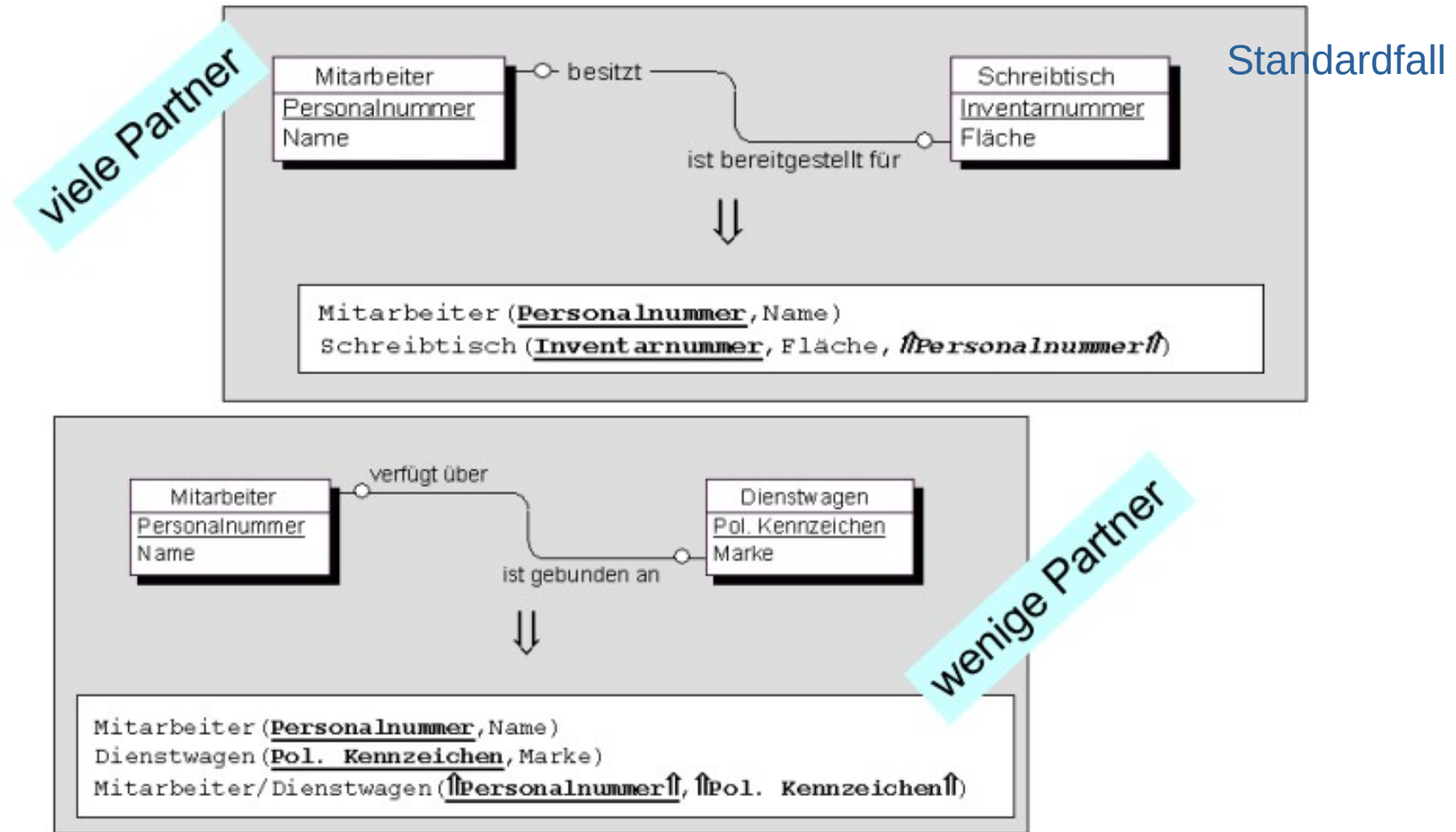
T6: 1:C Beziehungstypen (selten)

- Auftreten „selten“ → zwei Tabellen mit Fremdschlüssel



C:C Beziehungstypen

- Wie viele Beziehungen gibt es? (viele vs. wenige)

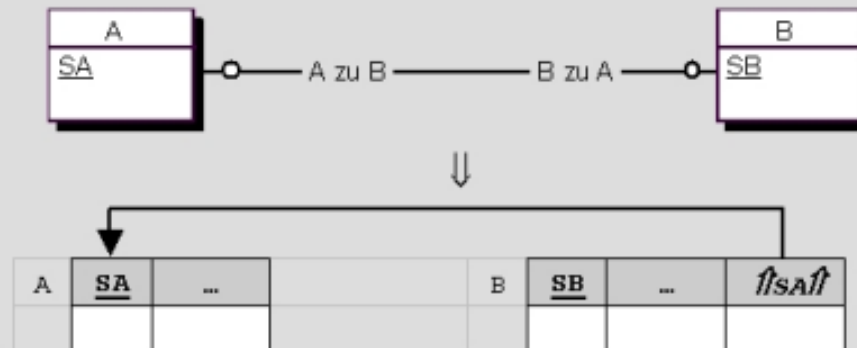


T7: C:C Beziehungstypen (viele Beziehungen)

- **Viele** Beziehungen: zwei Tabellen mit Fremdschlüssel
 - Fremdschlüssel so wählen, dass wenige NULL Werte existieren

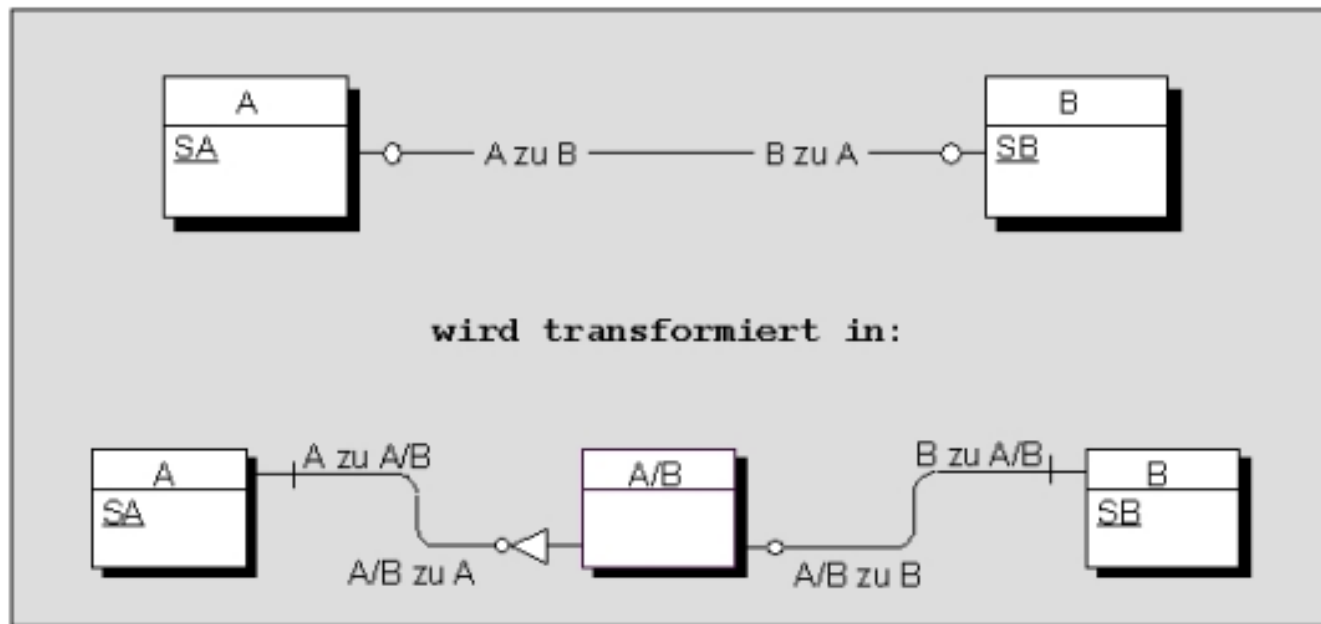
Transformationsregel T07 (C:C-Beziehungstyp mit vielen Partnerbeziehungen):

Entity-Relationship-Modell		Relationales Datenbank-Modell
Objekttyp A mit Schlüssel <u>SA</u>	⇒	Tabelle A mit Primärschlüssel <u>SA</u>
Objekttyp B mit Schlüssel <u>SB</u>	⇒	Tabelle B mit Primärschlüssel <u>SB</u>
C:C-Beziehungstyp	⇒	<u>SA</u> wird in B als nicht-eingabepflichtiger unikalischer Fremdschlüssel aufgenommen
Speicherbedarf für Beziehungstyp		$\overline{B} \cdot \text{Len}(\text{SA})$



T8: C:C Beziehungstypen (wenige Beziehungen)

- **Wenige** Beziehungen: drei Tabellen mit Fremdschlüsseln
 - Einführung einer Koppel-Entität
 - Eher selten verwendet

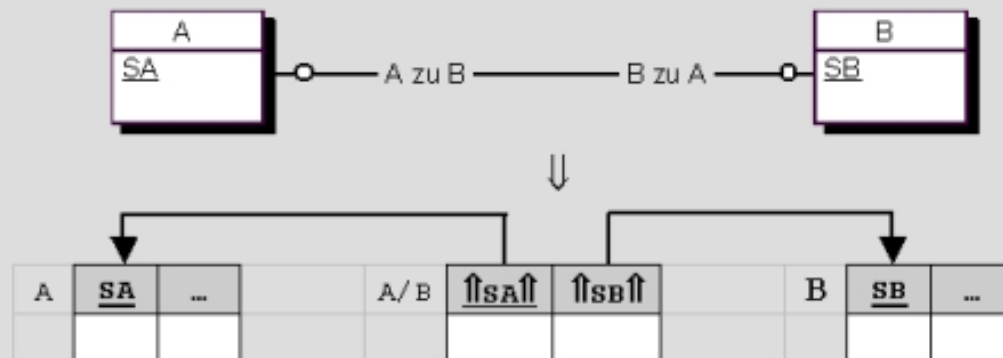


T8: C:C Beziehungstypen (wenige Beziehungen)

- Wenige Beziehungen: drei Tabellen mit Fremdschlüsseln

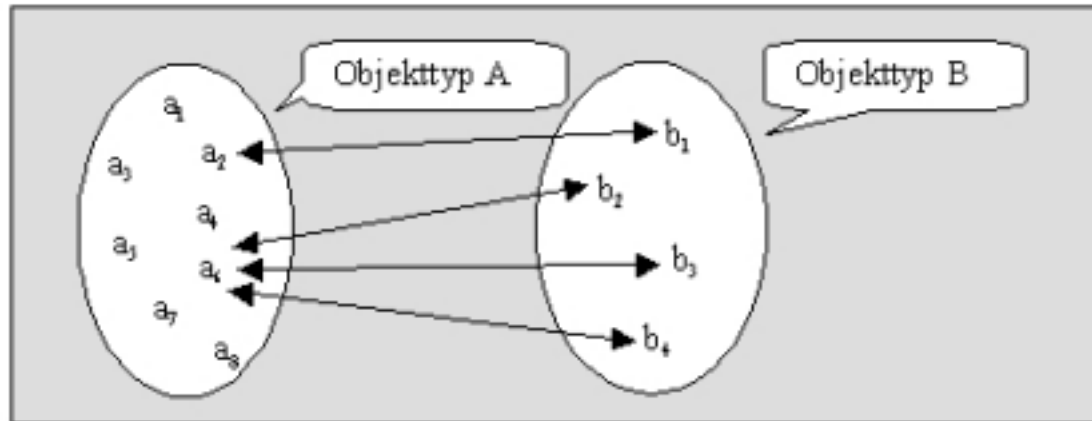
Transformationsregel T08 (C:C-Beziehungstyp mit
wenigen Partnerbeziehungen):

Entity-Relationship-Modell		Relationales Datenbank-Modell
Objekttyp A mit Schlüssel <u>SA</u>	⇒	Tabelle A mit Primärschlüssel <u>SA</u>
Objekttyp B mit Schlüssel <u>SB</u>	⇒	Tabelle B mit Primärschlüssel <u>SB</u>
C:C-Beziehungstyp	⇒	Koppel-Tabelle A/B mit <u>SA</u> und <u>SB</u> als jeweils eingabepflichtiger unikaler Fremdschlüssel. Entweder <u>SA</u> oder <u>SB</u> bildet den Primär- schlüssel von A/B.
Speicherbedarf für Bezie- hungstyp		$\overline{A/B} \cdot (Len(\underline{SA}) + Len(\underline{SB}))$

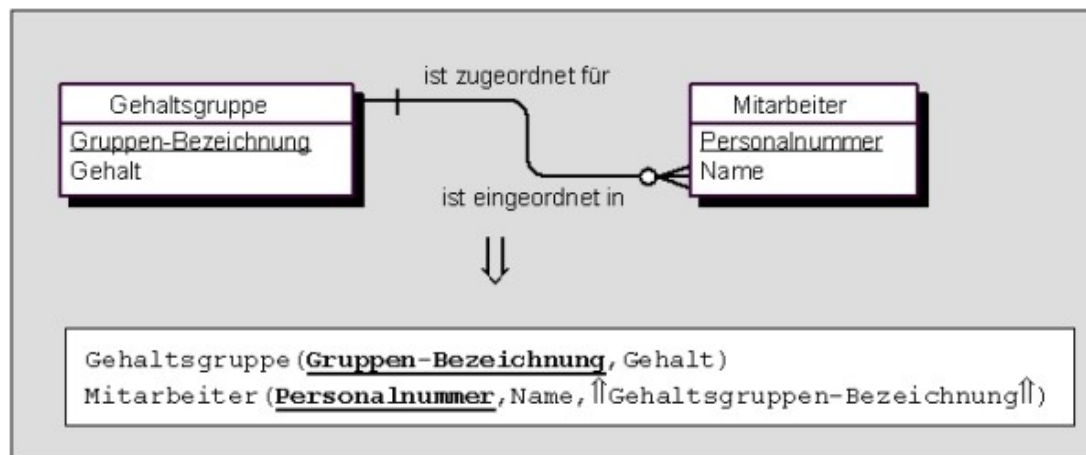


1:CN Beziehungstypen

- Fremdschlüssel kann in B mehrfach vorkommen

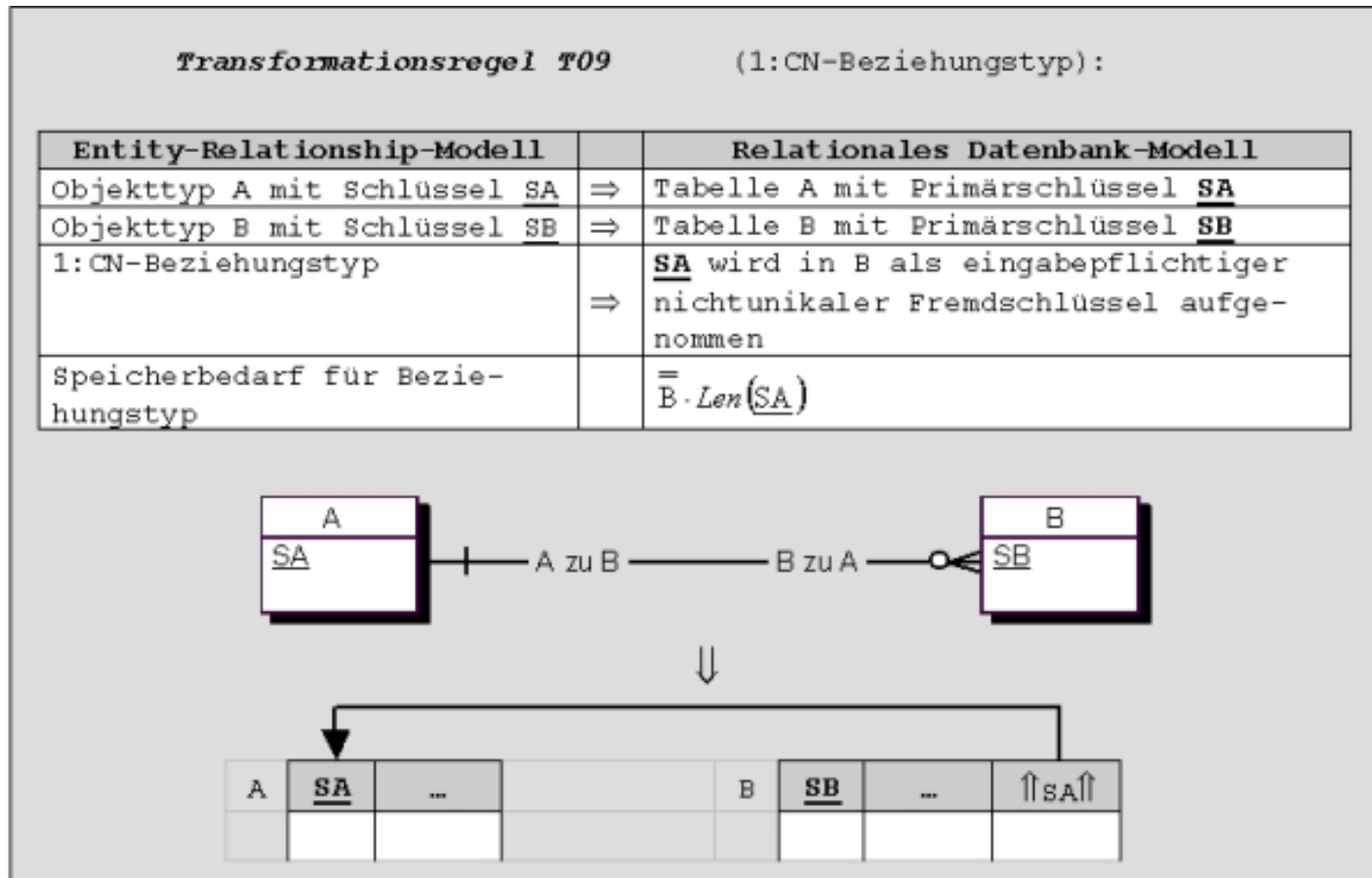


- Beispiel



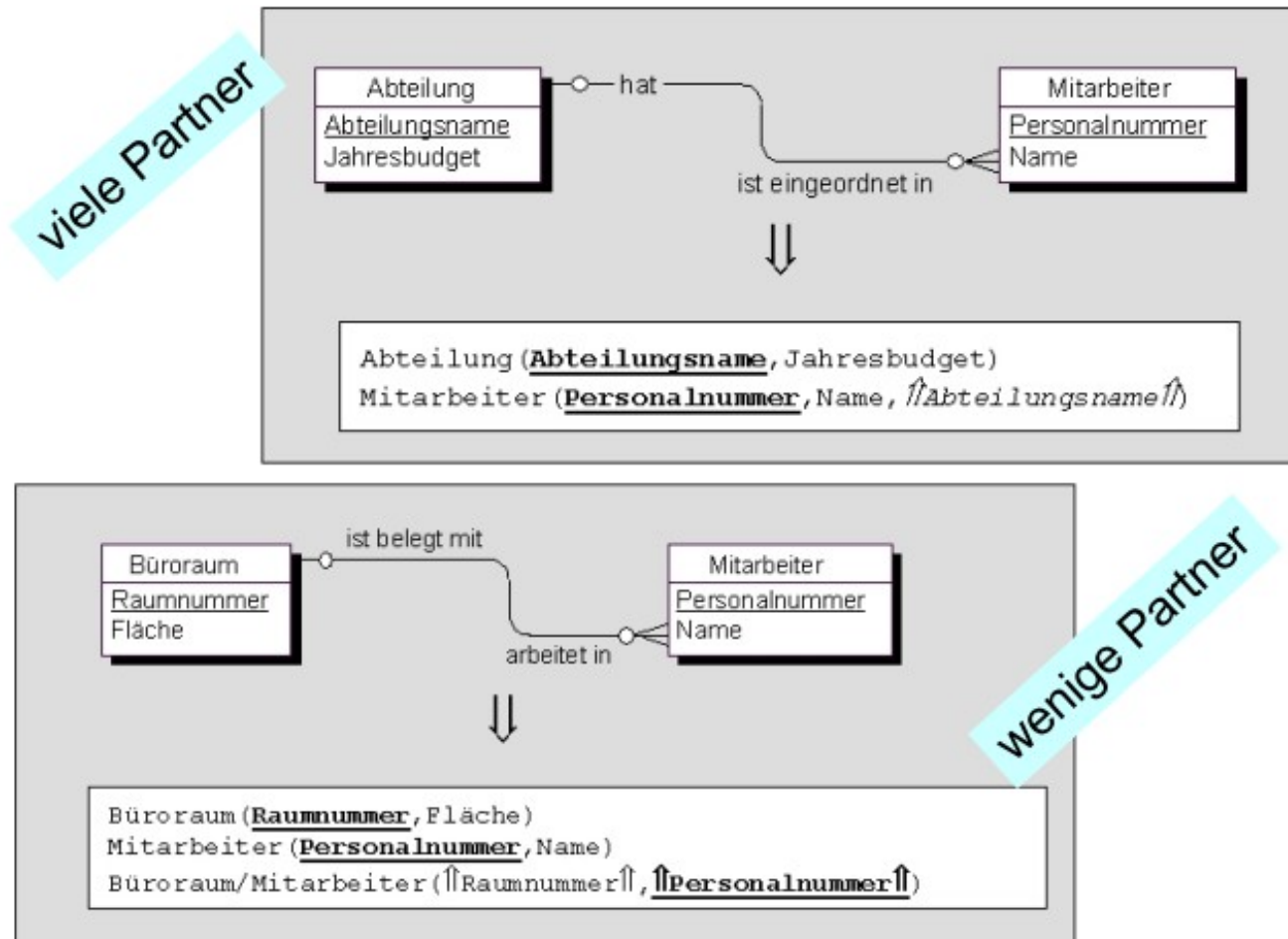
T9: 1:CN Beziehungstypen

- Zwei Tabellen mit Fremdschlüssel



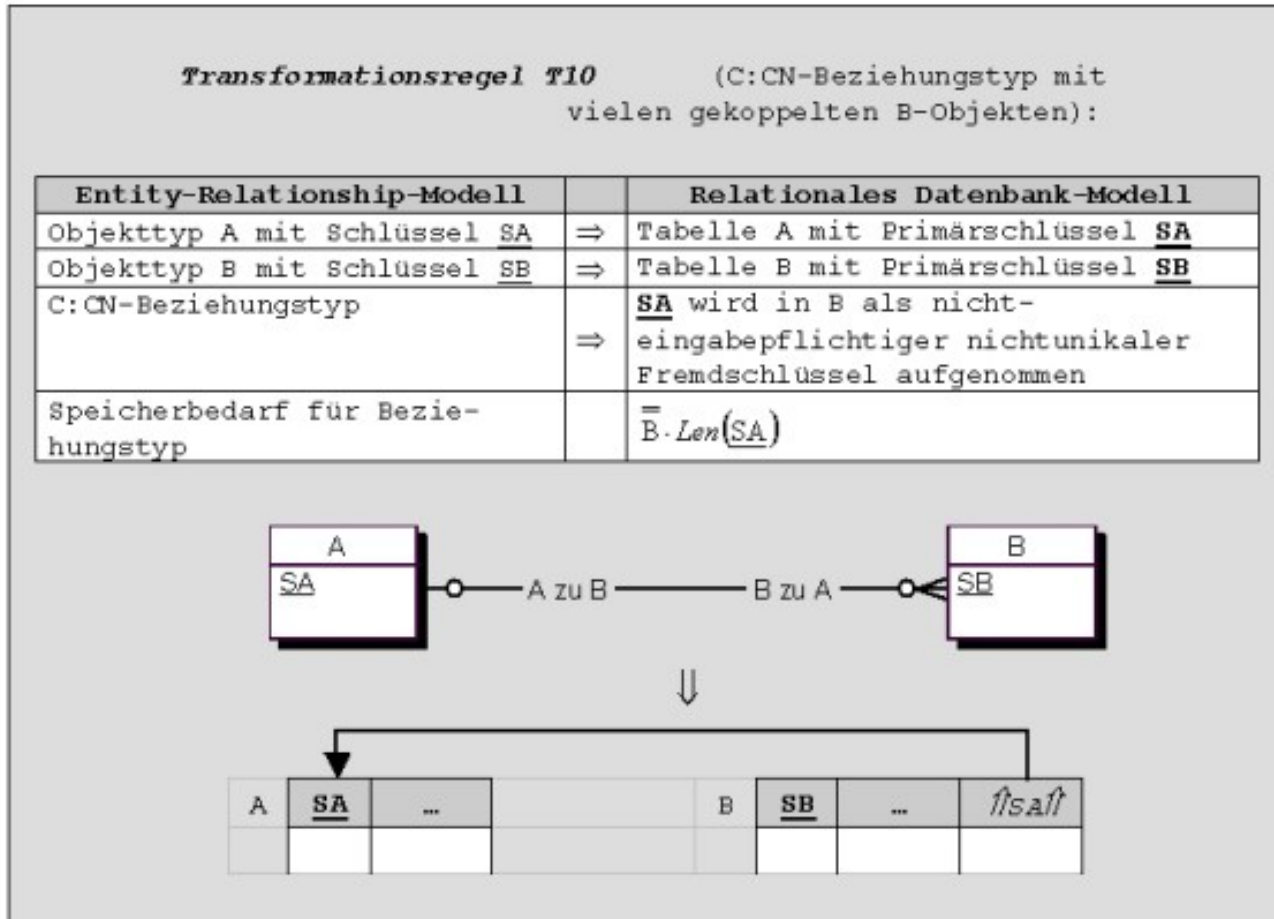
C:CN Beziehungstypen

- Unterscheidung
 - Viele vs wenige Beziehungen



T10: C:CN Beziehungstypen (viele Partner)

- Realisierung über Fremdschlüssel und zwei Tabellen

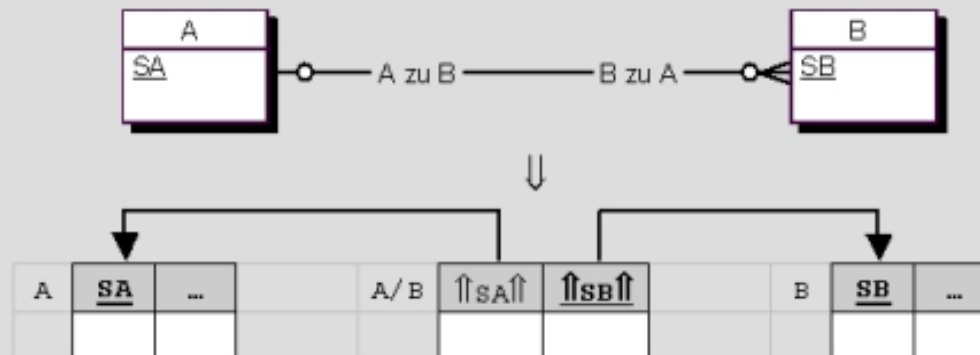


T11: C:CN Beziehungstypen (wenige Partner)

- Realisierung über drei Tabellen mit Koppel-Entität

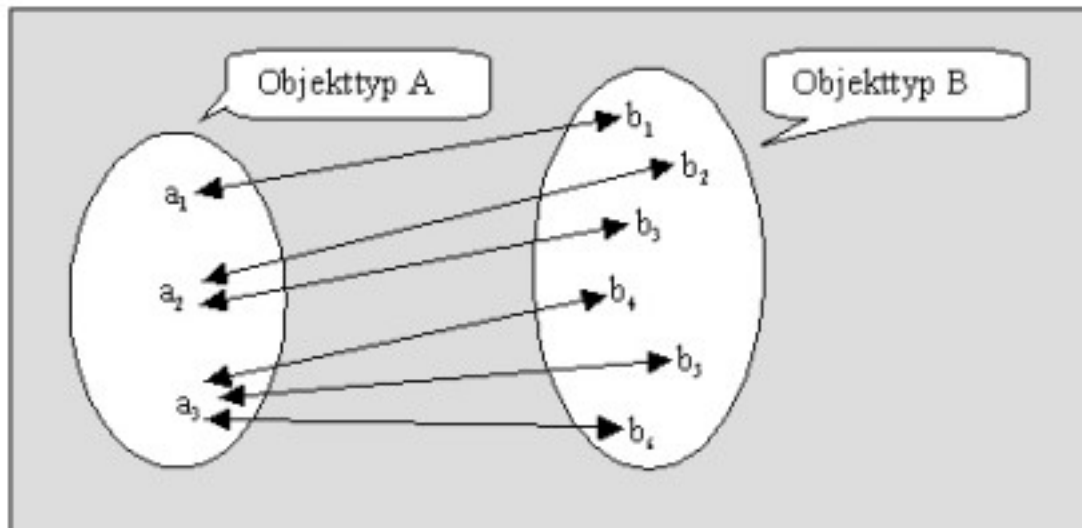
Transformationsregel T11 (C:CN-Beziehungstyp mit wenigen gekoppelten B-Objekten):

Entity-Relationship-Modell		Relationales Datenbank-Modell
Objekttyp A mit Schlüssel <u>SA</u>	⇒	Tabelle A mit Primärschlüssel <u>SA</u>
Objekttyp B mit Schlüssel <u>SB</u>	⇒	Tabelle B mit Primärschlüssel <u>SB</u>
C:CN-Beziehungstyp	⇒	Koppel-Tabelle A/B mit <u>SA</u> und <u>SB</u> als jeweils eingabepflichtiger Fremdschlüssel. <u>SA</u> wird als nichtunikal vereinbart. <u>SB</u> wird als unikal vereinbart und bildet den Primärschlüssel von A/B.
Speicherbedarf für Beziehungstyp		$\overline{A/B} \cdot (Len(\underline{SA}) + Len(\underline{SB}))$



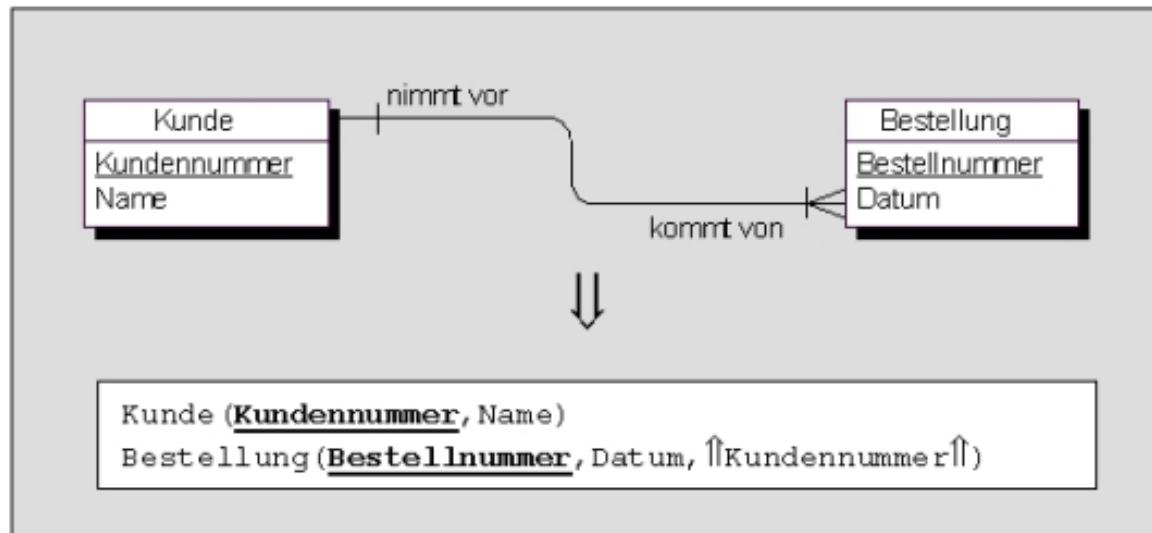
1:N Beziehungstypen

- Jeder Primärschlüssel aus A **muss** als Fremdschlüssel in B referenziert werden
 - Referentielle Integrität beachten!



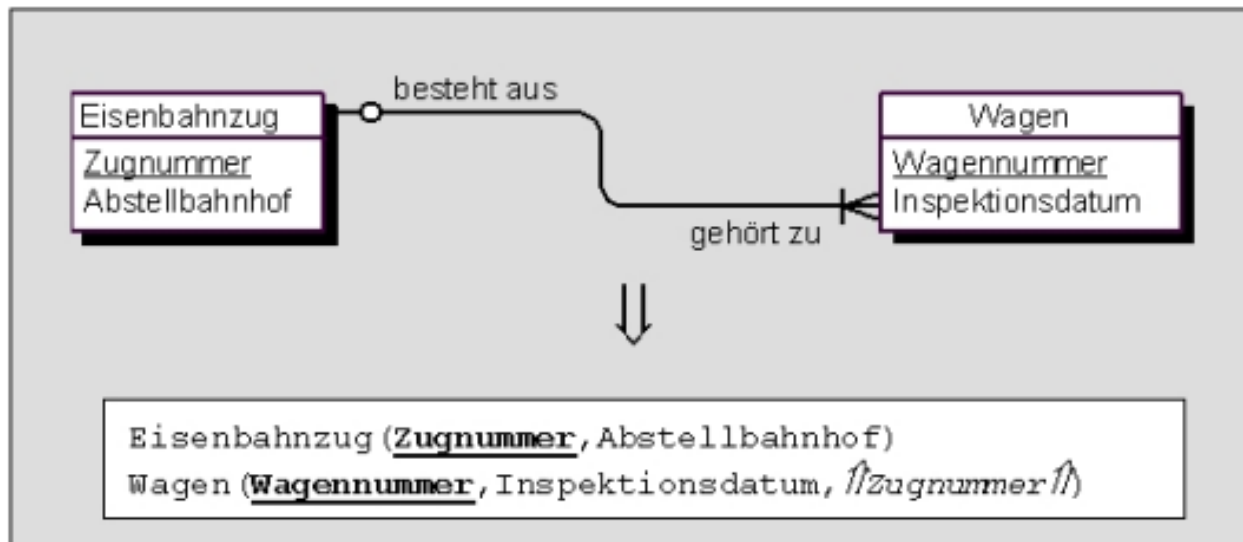
1:N Beziehungstypen

- Jeder Primärschlüssel aus A **muss** als Fremdschlüssel in B referenziert werden
 - **Bedingung lässt sich so im (logischen) relationalen Modell nicht abbilden!**
 - T9 mit „Eingabepflicht eines B in der Anwendung“

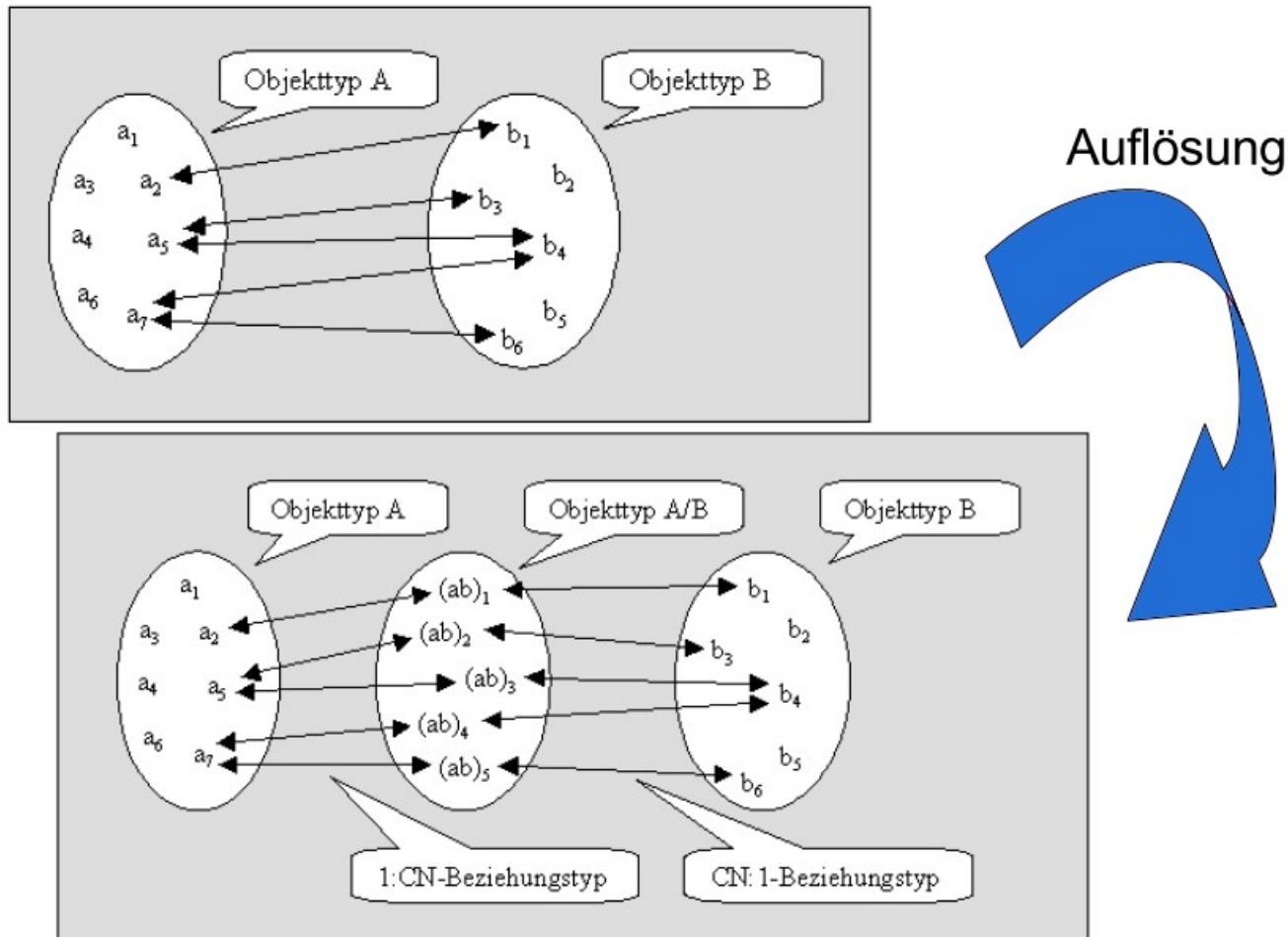


C:N Beziehungstypen

- Jeder Primärschlüssel aus A **muss** als Fremdschlüssel in B referenziert werden
 - **Bedingung lässt sich so im (logischen) relationalen Modell nicht abbilden!**
 - T10 mit „Eingabepflicht eines B in der Anwendung“
 - Beispiel:

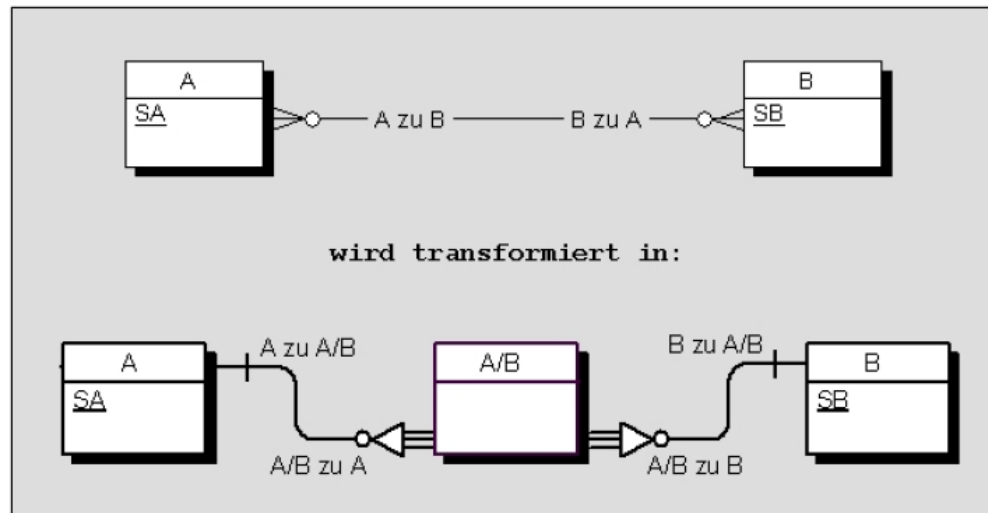


- Dies lässt sich im logischen Modell nur mit Hilfe einer Koppel-Entität lösen (3. Tabelle verknüpft A zu B)

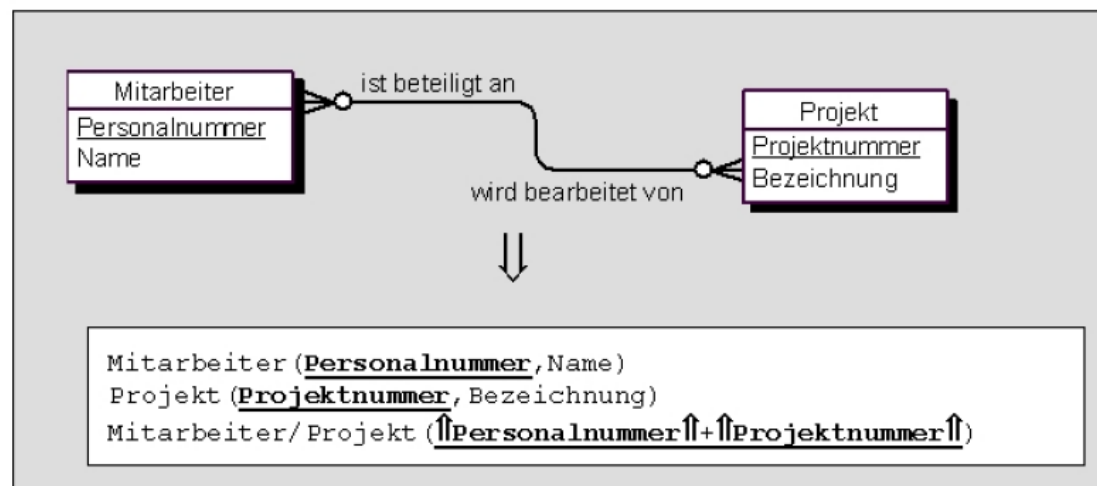


T12: CM:CN Beziehungstypen

- Einführung einer Koppel-Entität mit **identifizierender Beziehung**



- Beispiel:



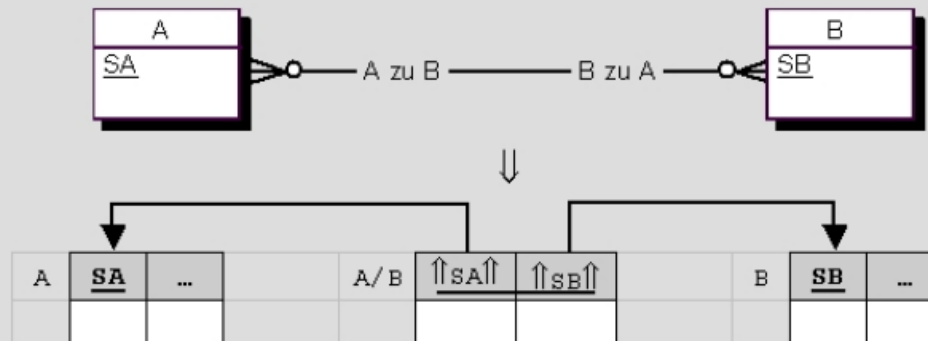
T12: CM:CN Beziehungstypen

- Einführung einer Koppel-Entität

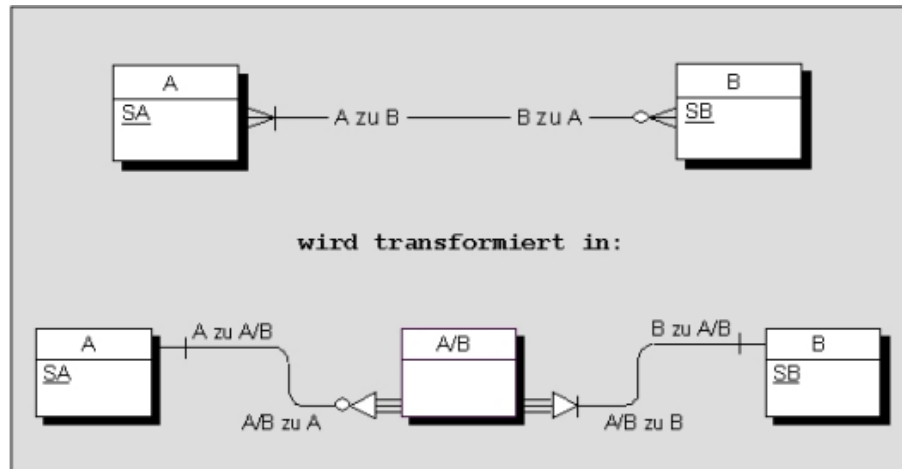
Transformationsregel T12

(CM:CN-Beziehungstyp):

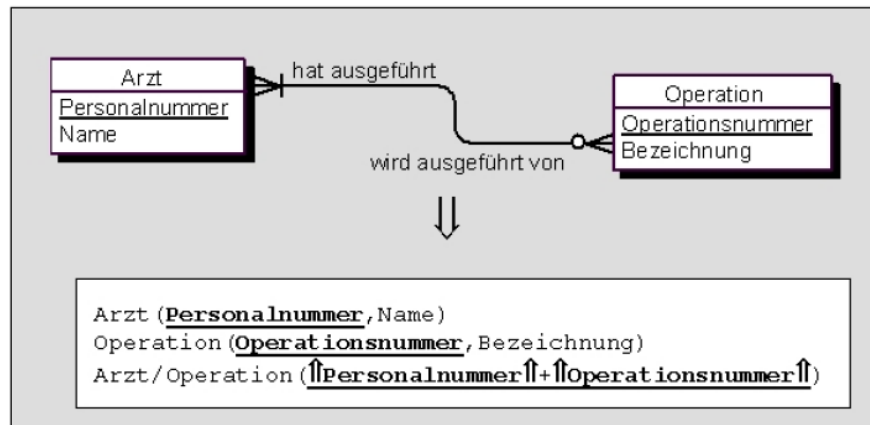
Entity-Relationship-Modell		Relationales Datenbank-Modell
Objekttyp A mit Schlüssel <u>SA</u>	⇒	Tabelle A mit Primärschlüssel <u>SA</u>
Objekttyp B mit Schlüssel <u>SB</u>	⇒	Tabelle B mit Primärschlüssel <u>SB</u>
CM:CN-Beziehungstyp	⇒	Koppel-Tabelle A/B. <u>SA</u> und <u>SB</u> werden als eingabepflichtige nichtunikale Fremdschlüssel in A/B aufgenommen. Die Kombination der Fremdschlüssel $\uparrow\uparrow SA \uparrow\uparrow$ und $\uparrow\uparrow SB \uparrow\uparrow$ wird als unikal vereinbart. Sie bildet den Primärschlüssel von A/B.
Speicherbedarf für Beziehungstyp		$\overline{A/B} \cdot (Len(\underline{SA}) + Len(\underline{SB}))$



- Nach Regel T12 (erneut Verpflichtung von ≥ 1 nicht modellierbar!)

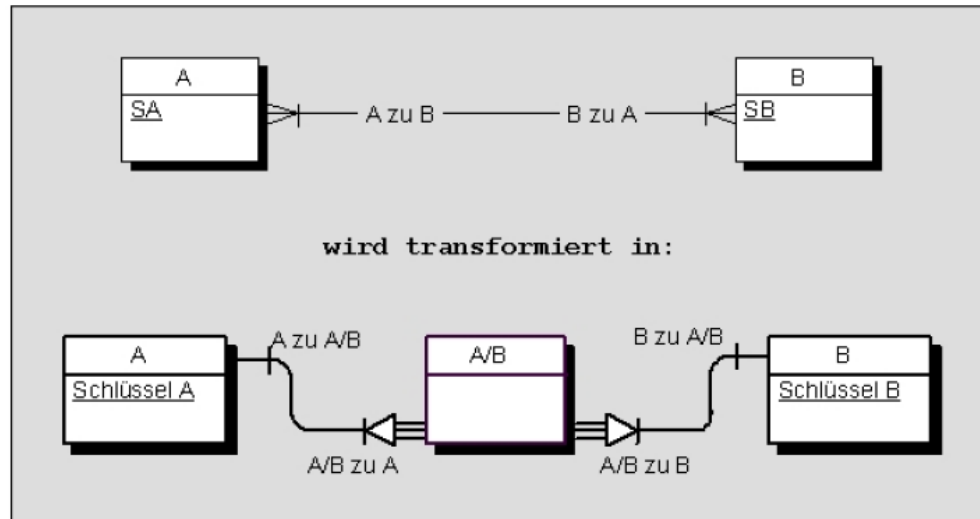


- Beispiel:

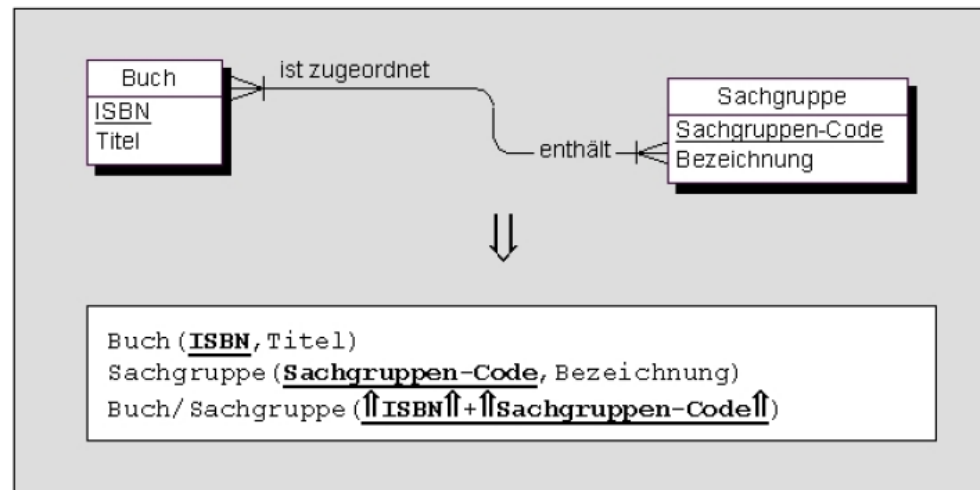


M:N Beziehungstypen

- Regel T12 (erneut Verpflichtungen von ≥ 1 nicht modellierbar!)



- Beispiel:



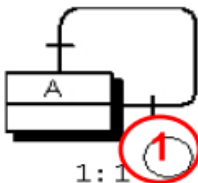
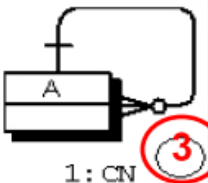
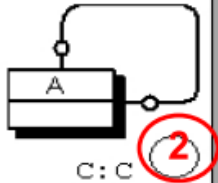
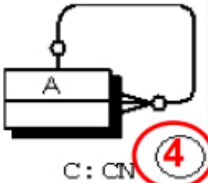
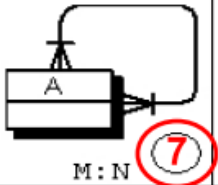
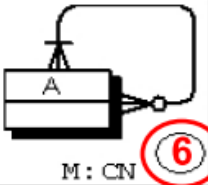
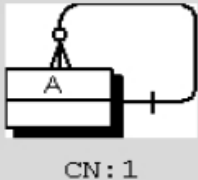
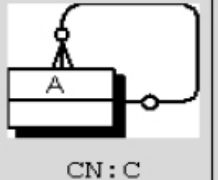
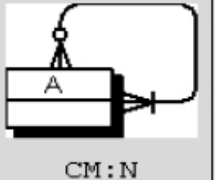
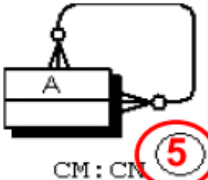
4.1 Überblick Transformation

4.2 Duale Beziehungstypen

4.3 Rekursive Beziehungstypen

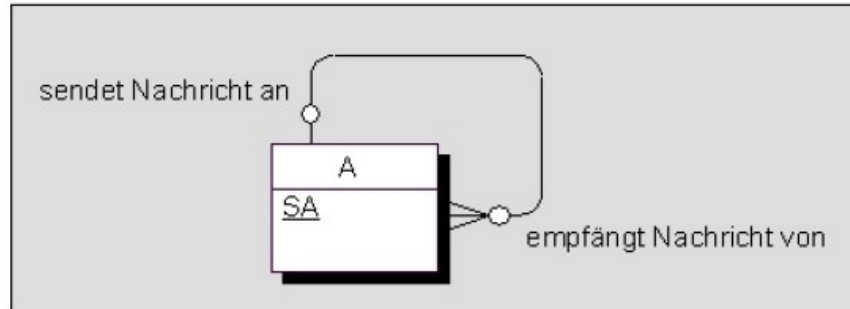
Rekursive Beziehungstypen

■ Überblick

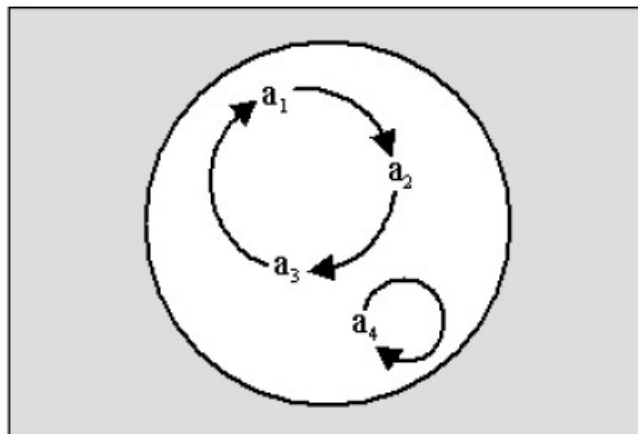
1. Beziehungstyp-Richtung		Kardinalität 1		Kardinalität N	
2. Beziehungstyp-Richtung		nichtoptional	optional	nichtoptional	optional
Kardinalität 1	nichtoptional	 1:1 (1)			 1:CN (3)
	optional		 C:C (2)		 C:CN (4)
Kardinalität N	nichtoptional			 M:N (7)	 M:CN (6)
	optional	 CN:1	 CN:C	 CM:N	 CM:CN (5)

Rekursiv-Beziehungstypen

- Im Folgenden: Sender-Empfänger-Modell



- Dabei können (Entitäts-)Zyklen entstehen:

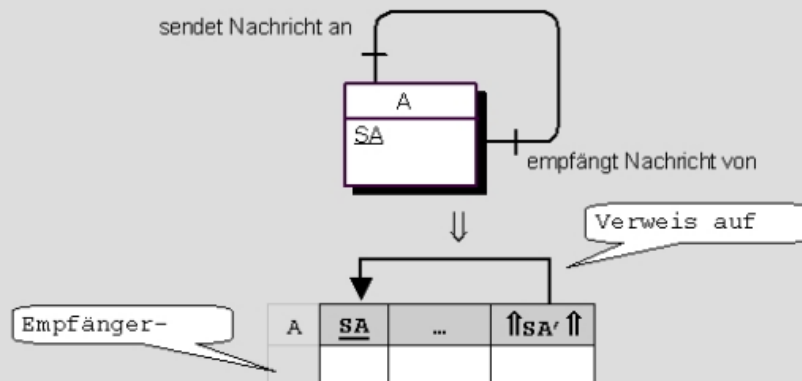


T13: 1:1 Rekursiv-Beziehungstypen

- Eine Tabelle
- Fremdschlüssel verweist auf einen Primärschlüssel
- Guten Rollennamen wählen (aussagekräftig)!

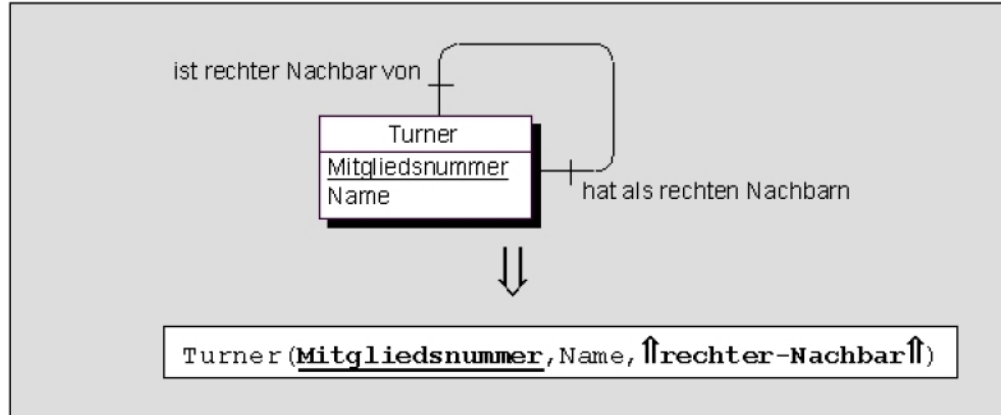
Transformationsregel T13 (1:1-Rekursiv-Beziehungstyp):

Entity-Relationship-Modell		Relationales Datenbank-Modell
Objekttyp A mit Schlüssel <u>SA</u>	⇒	Tabelle A mit Primärschlüssel <u>SA</u>
1:1-Rekursiv-Beziehungstyp	⇒	<u>SA</u> wird in A mit anderen Attributbezeichnungen „gedoppelt“ und als eingabepflichtiger unikaler Fremdschlüssel $\uparrow\text{SA}'\uparrow$ vereinbart, der auf den Sender verweist.
Speicherbedarf für Beziehungstyp		$\overline{A} \cdot \text{Len}(\text{SA})$

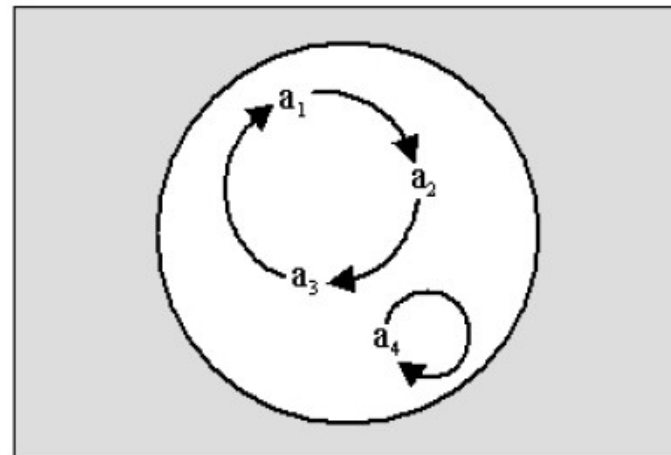


T13: 1:1 Rekursiv-Beziehungstypen

- Beispiel: Turner stehen im Kreis

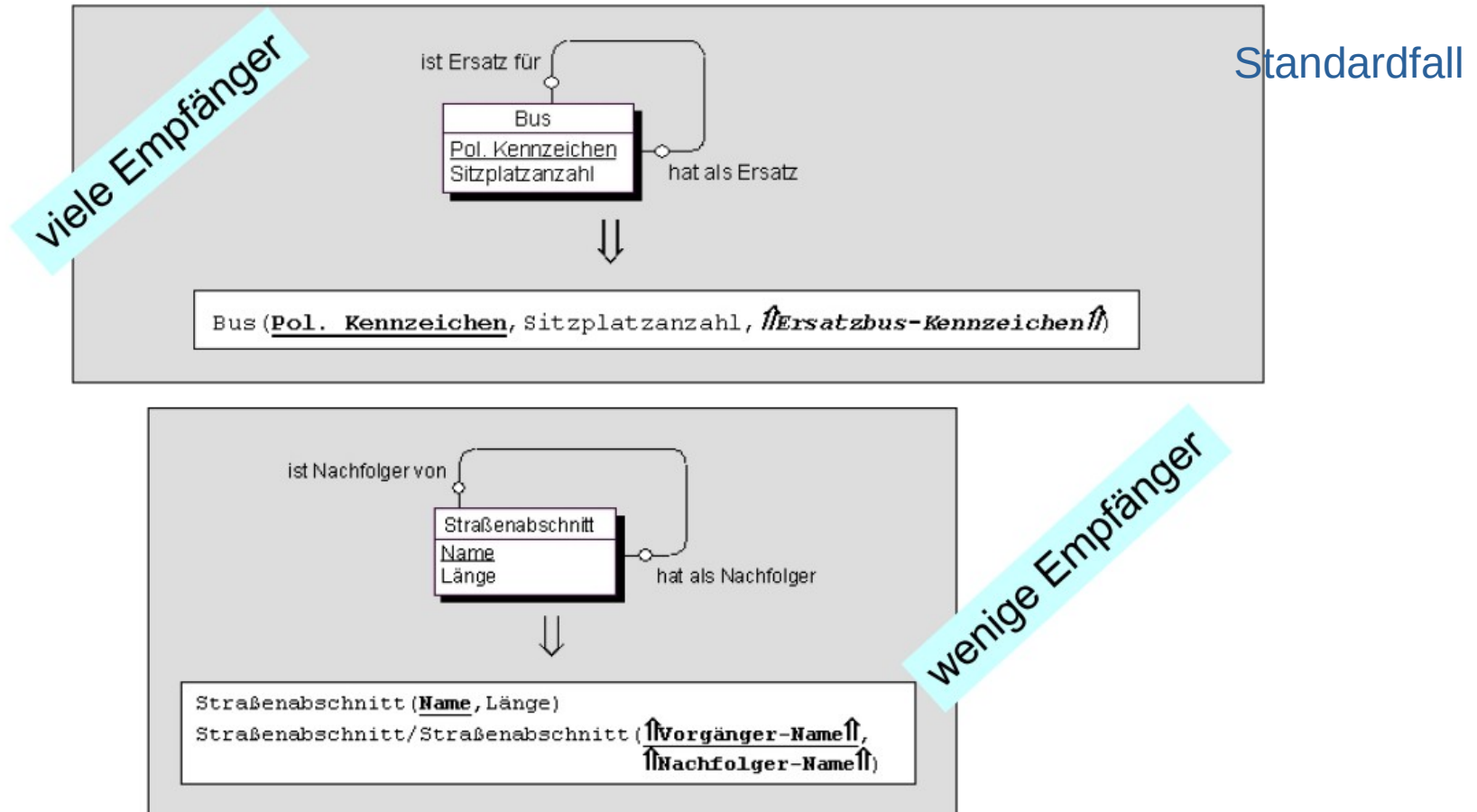


- Zyklus von Entitäten entstehen



C:C Rekursiv-Beziehungstypen

- Unterscheidung: viele/wenige Beziehungen

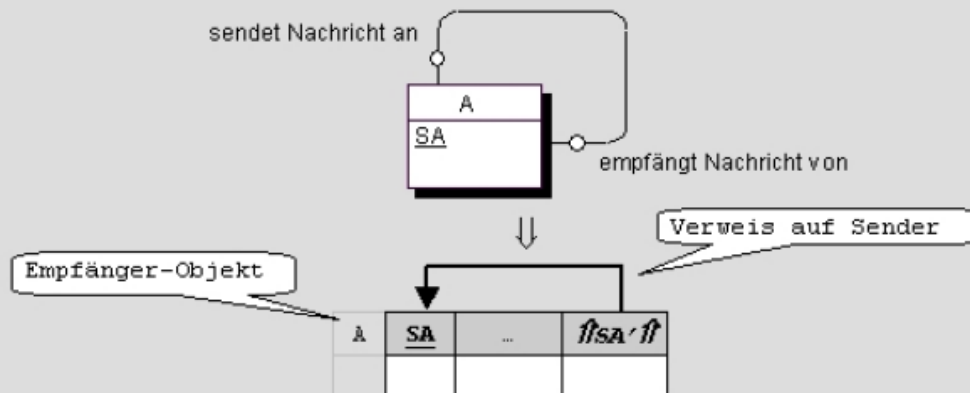


T14: C:C Rekursiv-Beziehungstypen (viele Sender)

- Fremdschlüssel in einer Tabelle (wie T7)

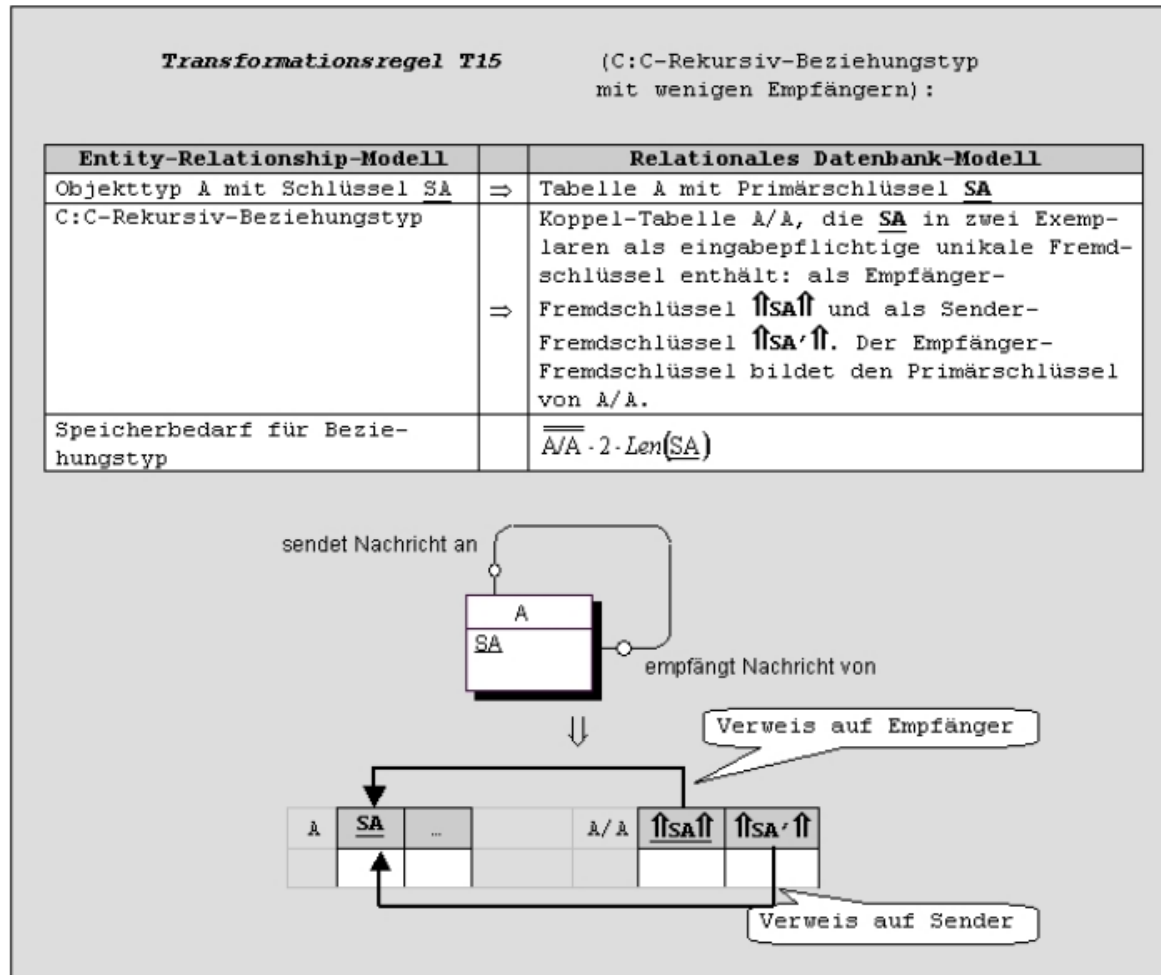
Transformationsregel T14 (C:C-Rekursiv-Beziehungstyp mit vielen Empfängern):

Entity-Relationship-Modell		Relationales Datenbank-Modell
Objekttyp A mit Schlüssel SA	⇒	Tabelle A mit Primärschlüssel SA
C:C-Rekursiv-Beziehungstyp		SA wird in A mit anderen Attributbezeichnungen „gedoppelt“ und als nicht-eingabepflichtiger unikaler Fremdschlüssel <i>ffSA'ff</i> vereinbart, der auf den Sender verweist.
Speicherbedarf für Beziehungstyp		$\overline{A} \cdot \text{Len}(\underline{SA})$



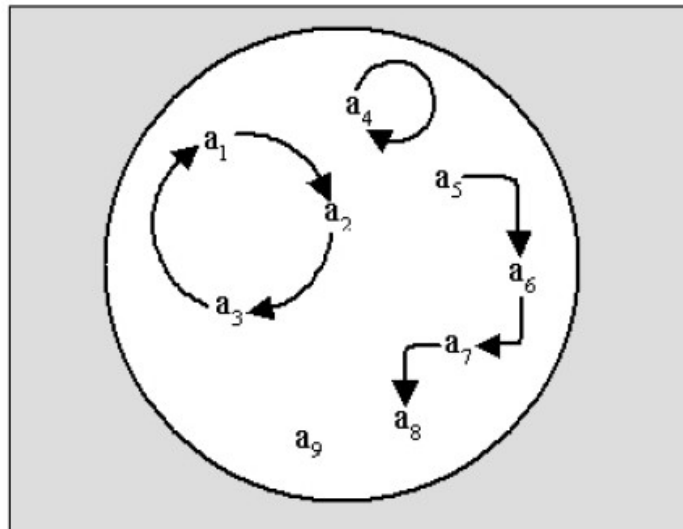
T15: C:C Rekursiv-Beziehungstypen (wenige Sender)

- Realisierung mit zusätzlicher Tabelle (wie T8)



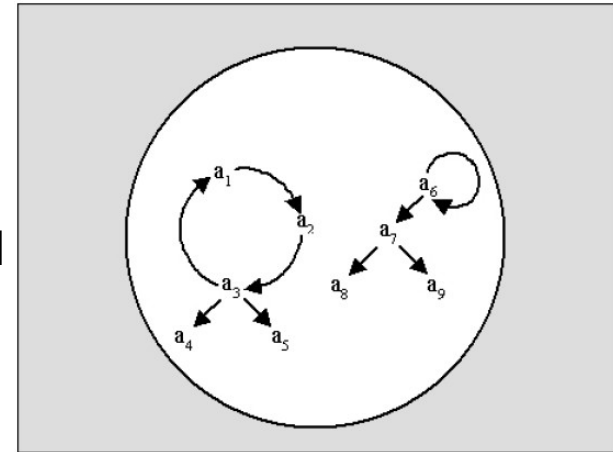
C:C Rekursiv-Beziehungstypen

- Optionalität ermöglicht
 - Entitätszyklen
 - Entitätsketten
 - Keine Partner

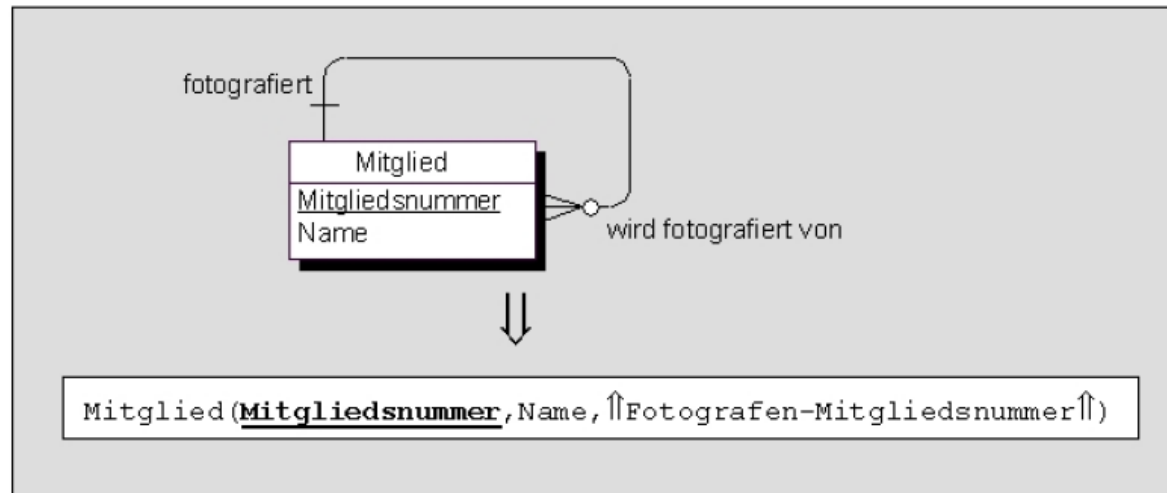


1:CN Rekursiv-Beziehungstypen

- Optionalität ermöglicht
 - Entitätszyklen (eigentl. 1:1)
 - Monohierarchien mit Zyklus an der Wurzel



- Beispiel

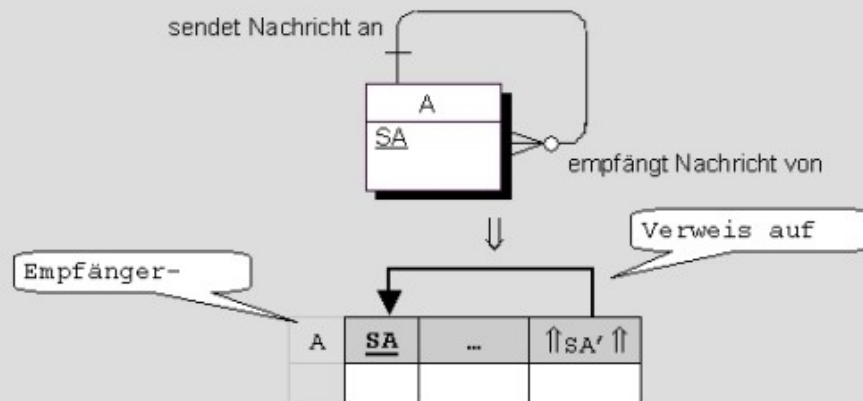


T16: 1:CN Rekursiv-Beziehungstypen

- Realisierung über Fremdschlüssel (NOT NULL)

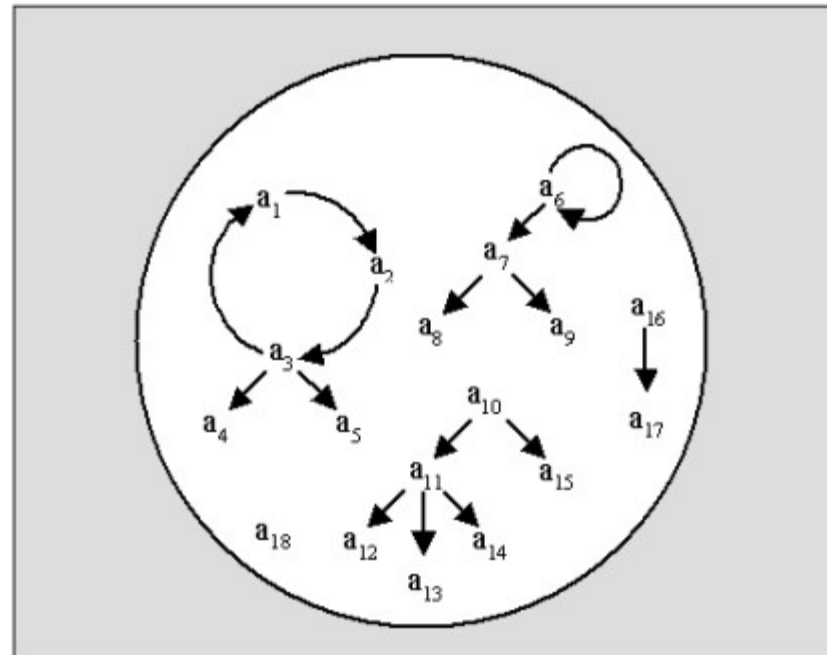
Transformationsregel T16 (1:CN-Rekursiv-Beziehungstyp):

Entity-Relationship-Modell		Relationales Datenbank-Modell
Objekttyp A mit Schlüssel <u>SA</u>	⇒	Tabelle A mit Primärschlüssel <u>SA</u>
1:CN-Rekursiv-Beziehungstyp	⇒	<u>SA</u> wird in A mit anderen Attributbezeichnungen „gedoppelt“ und als eingabepflichtiger nichtunktaler Fremdschlüssel $\uparrow\uparrow SA' \uparrow\uparrow$ vereinbart, der auf den Sender verweist.
Speicherbedarf für Beziehungstyp		$= A \cdot Len(\underline{SA})$



C:CN Rekursiv-Beziehungstypen

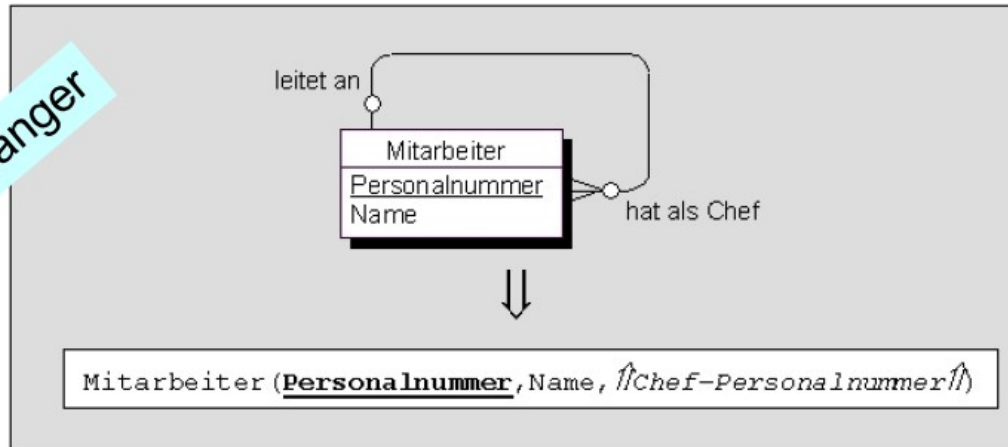
- Möglich sind hiermit
 - Lineare Listen
 - Hierarchien mit Wurzel (ohne „Wurzelzyklus“)
- Wieder Unterscheidung nach
 - Viele Empfänger
 - Wenige Empfänger



C:CN Rekursiv-Beziehungstypen

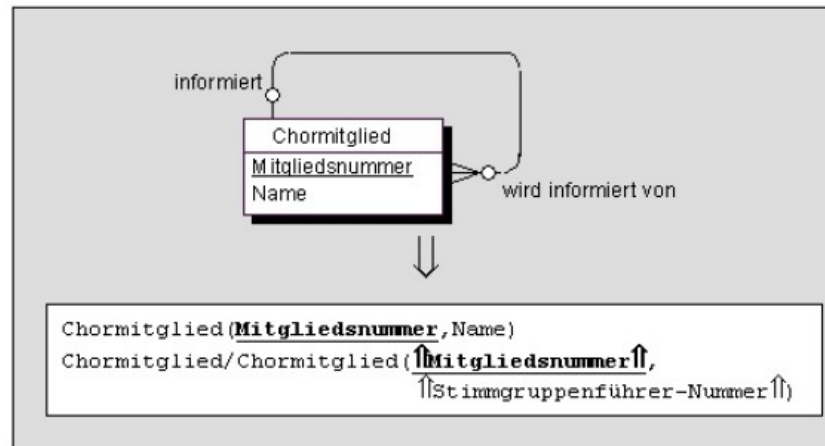
- Zusätzliche Tabelle für wenige Empfänger

viele Empfänger



Standardfall

wenige Empfänger

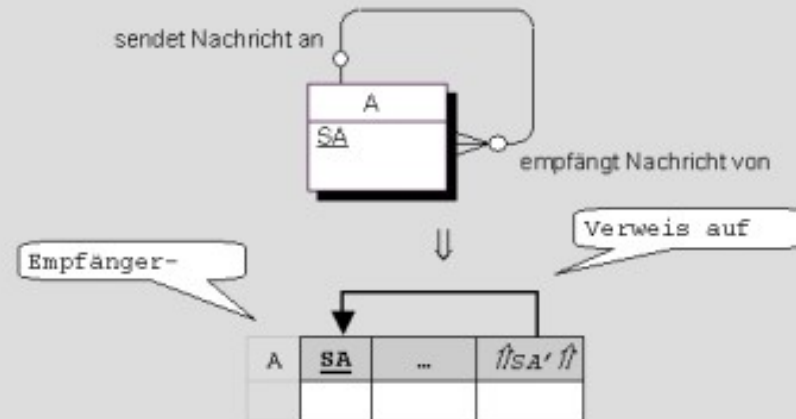


T17: C:CN Rekursiv-Beziehungstypen (viele Partner)

- Realisierung über Fremdschlüssel (wie T10)

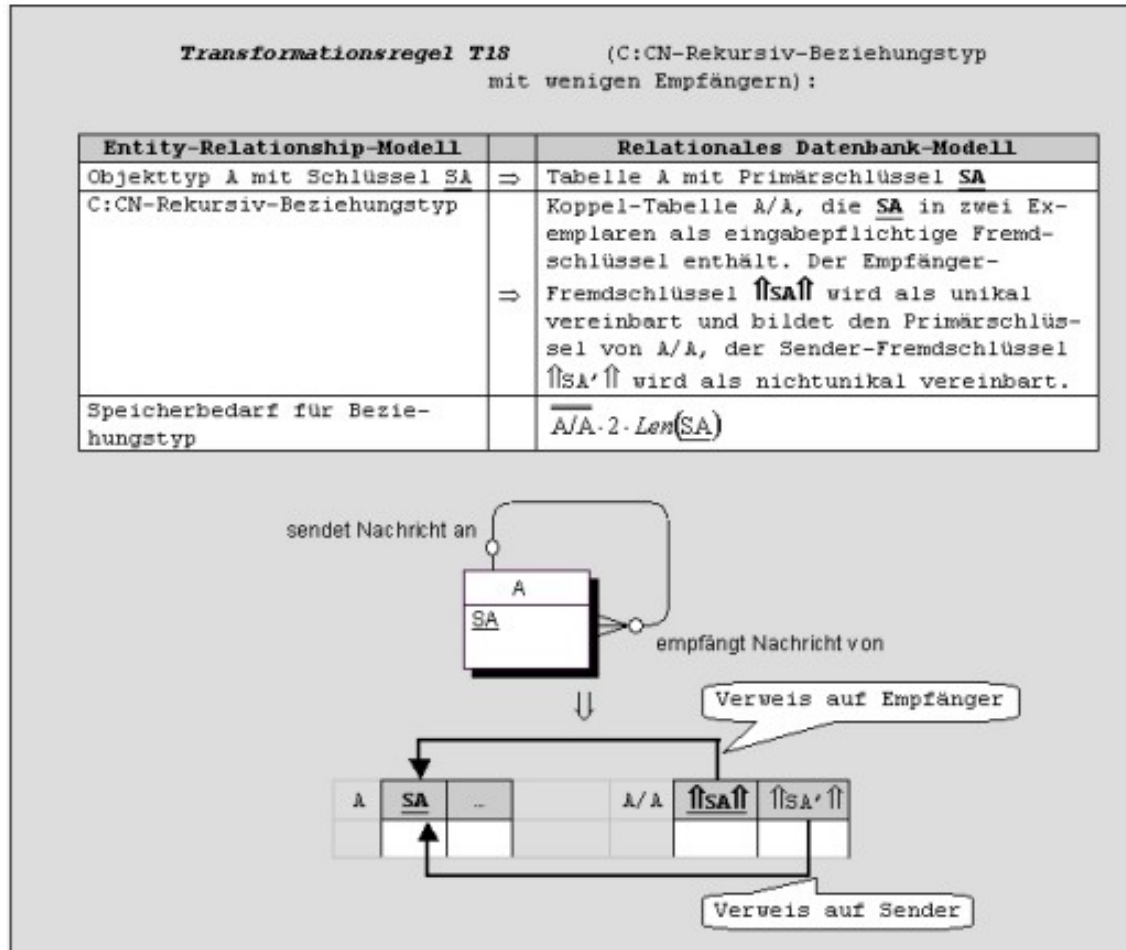
Transformationsregel T17 (C:CN-Rekursiv-Beziehungstyp
mit vielen Empfängern):

Entity-Relationship-Modell		Relationales Datenbank-Modell
Objekttyp A mit Schlüssel SA	⇒	Tabelle A mit Primärschlüssel <u>SA</u>
C:CN-Rekursiv-Beziehungstyp	⇒	<u>SA</u> wird in A mit anderen Attributbezeichnungen „gedoppelt“ und als nicht-eingabepflichtiger nichtunikaler Fremdschlüssel $\uparrow\uparrow SA' \uparrow\uparrow$ vereinbart, der auf den Sender verweist.
Speicherbedarf für Beziehungstyp		$= A \cdot Len(SA)$



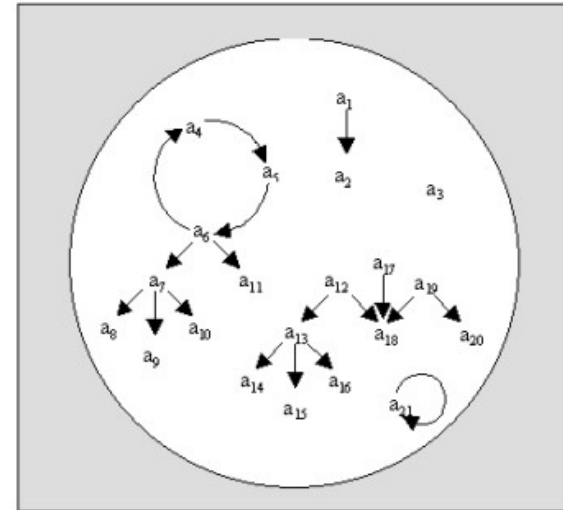
T18: C:CN Rekursiv-Beziehungstypen (wenige Partner)

- Realisierung über Koppel-Entität

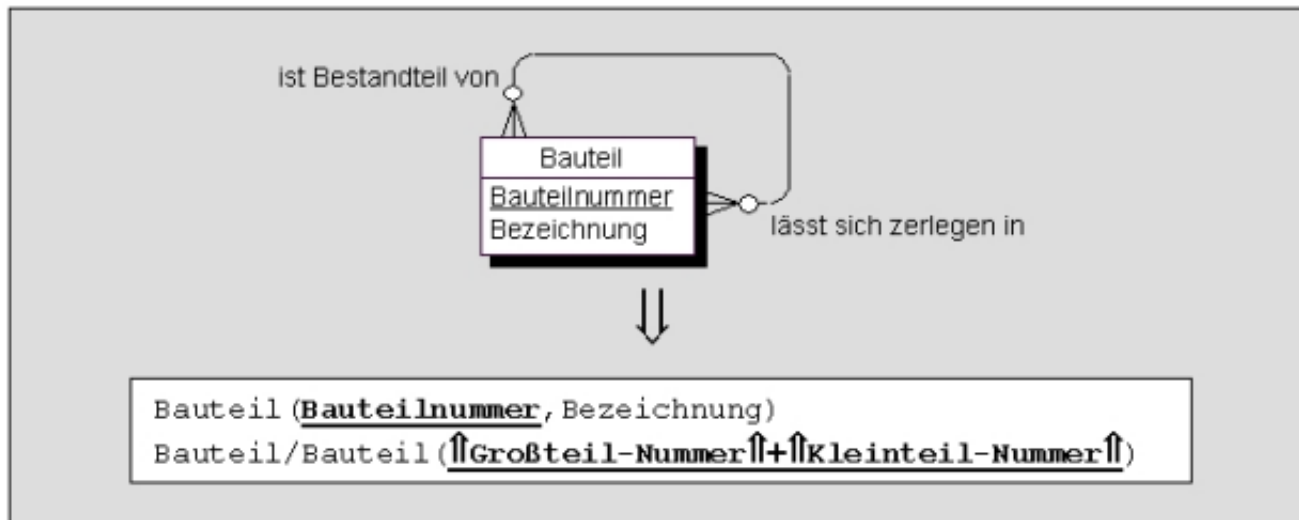


CM:CN Rekursiv-Beziehungstypen

- Auch Polyhierarchien möglich

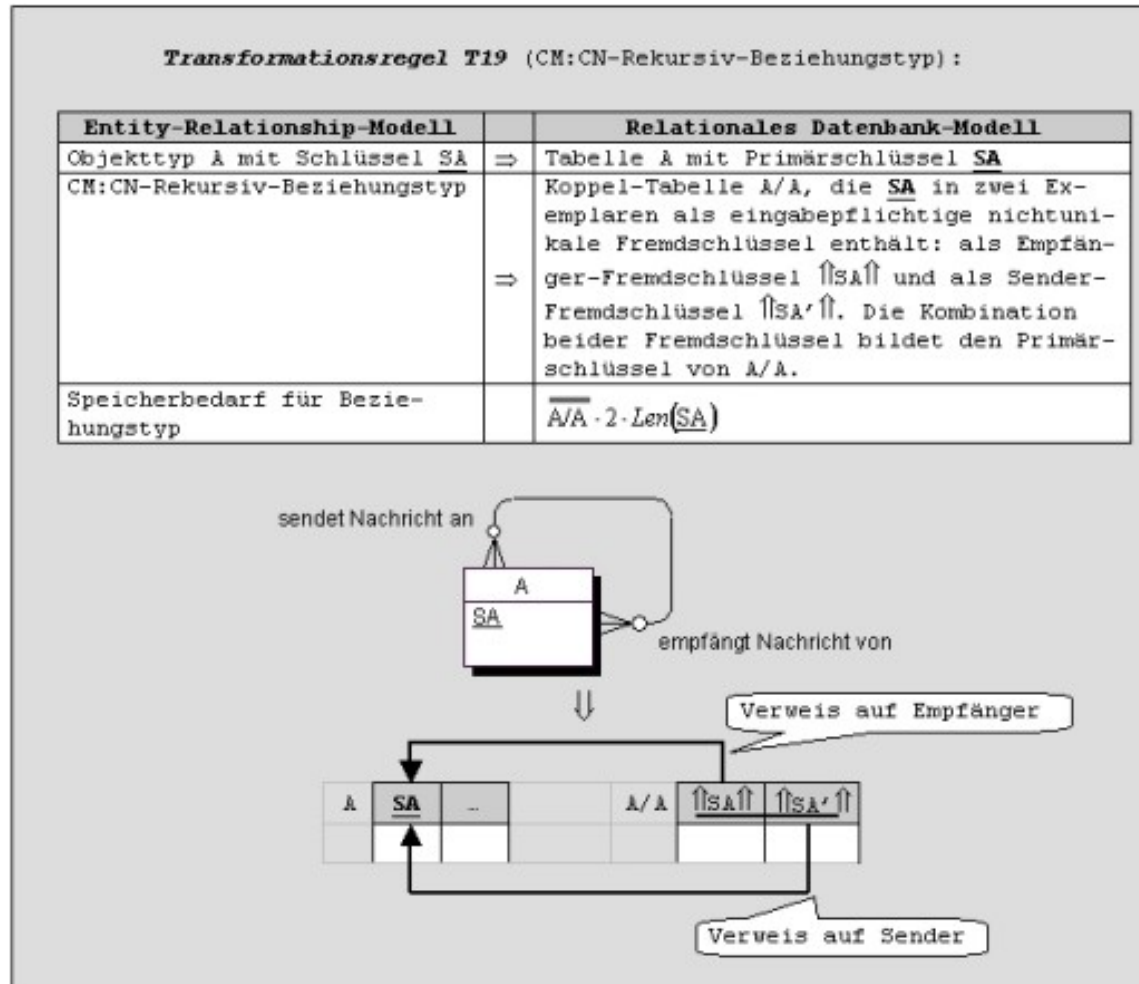


- Beispiel



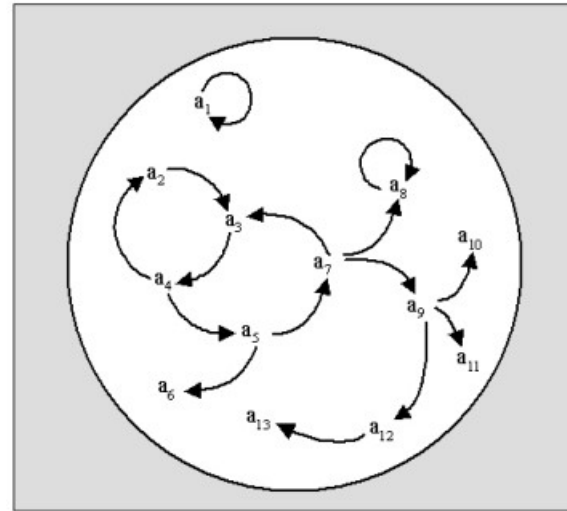
T19: CM:CN Rekursiv-Beziehungstypen

- Realisierung über zweite Tabelle

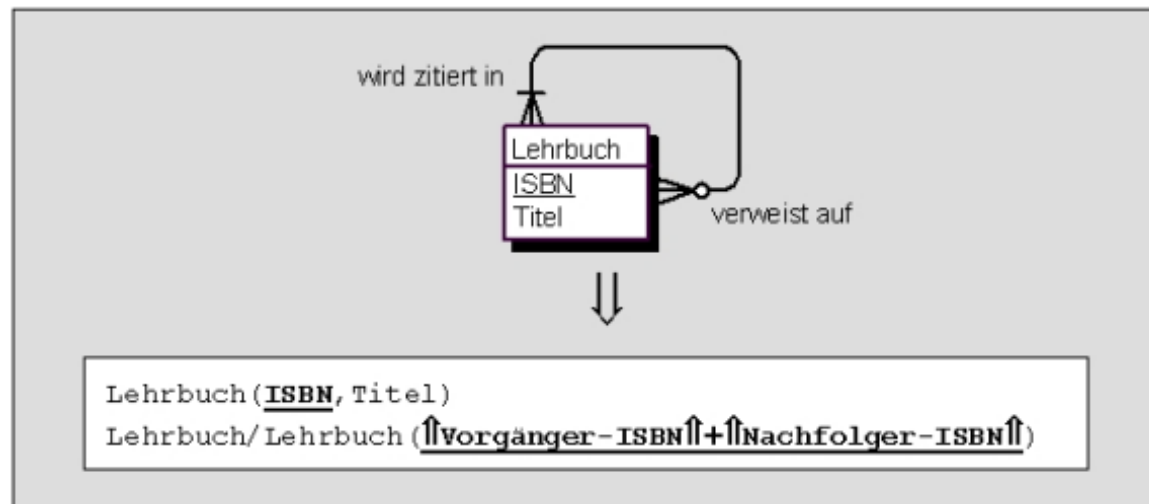


M:CN Rekursiv-Beziehungstypen

- Mit T19
- Auch wieder Verlust der ≥ 1 Bedingung (nicht-modellierbar)
- „Eingabeverpflichtung“ im Anwendungsprogramm

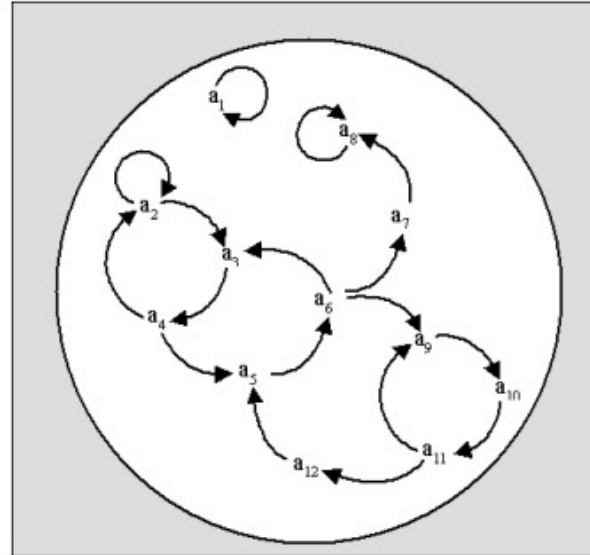


- Beispiel

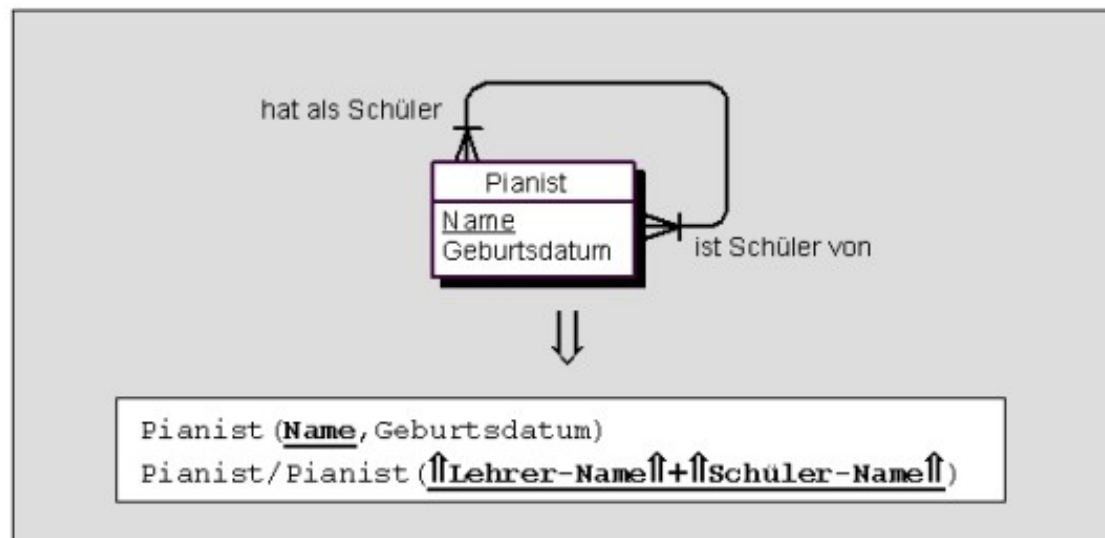


M:N Rekursiv-Beziehungstypen

- Mit T19
- Auch wieder Verlust der ≥ 1 Bedingung (nicht-modellierbar)
- „Eingabeverpflichtung“ im Anwendungsprogramm



- Beispiel



- **Konzeptionelles Datenmodell** in **logisches Datenbankmodell** überführbar
- Bei mehrfach/mehrfach Kardinalitäten wird eine Koppel-Entität zwingend erforderlich
- Teilweise Verlust der ≥ 1 Bedingung (muss dann durch das Anwendungsprogramm überprüft werden)

Aufgaben

Bitte bearbeiten Sie jetzt die Aufgaben in Moodle zum Kapitel 4.

- **Teil A**