

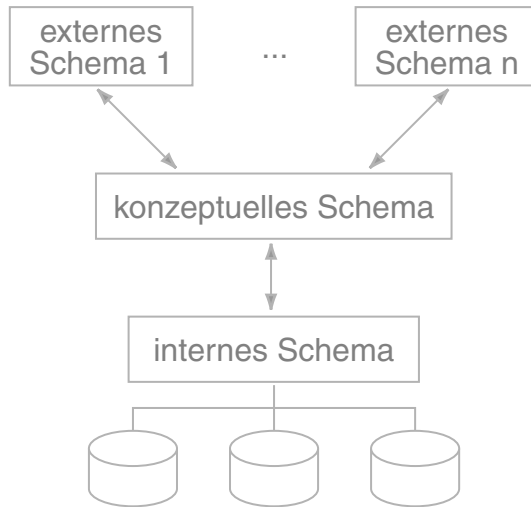
III. Logischer Datenbankentwurf mit dem relationalen Modell

- ❑ Das relationale Modell
- ❑ Integritätsbedingungen
- ❑ Umsetzung ER-Schema in relationales Schema
- ❑ Vergleichende Syntax-Übersicht

Umsetzung ER-Schema in relationales Schema

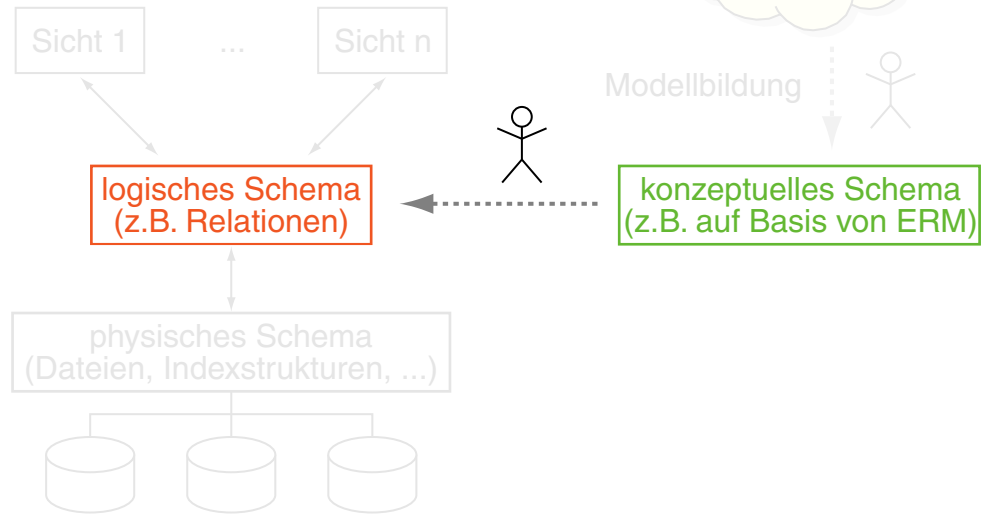
Einordnung

Theorie



Schema-Architektur
nach ANSI/SPARC

Praxis



typische implementierte
Schema-Architektur

Umsetzung ER-Schema in relationales Schema

Einordnung (Fortsetzung)

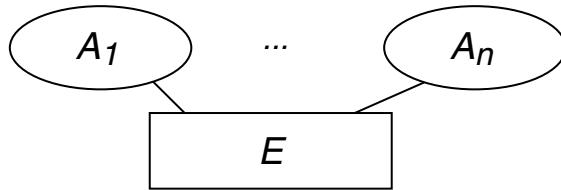
Das ER-Modell besitzt zwei grundlegende Strukturierungskonzepte:

1. Entity-Typen $E(A_1, \dots, A_n)$
2. Beziehungstypen $R(E_1, \dots, E_m; A_1, \dots, A_n)$

Im relationalen Modell werden beide auf das einzige Strukturierungskonzept „Relationenschema“, \mathcal{R} , abgebildet. Hierbei dient das Konzept der Fremdschlüssel zur Abbildung von Beziehungstypen.

Umsetzung ER-Schema in relationales Schema

Reguläre Entity-Typen

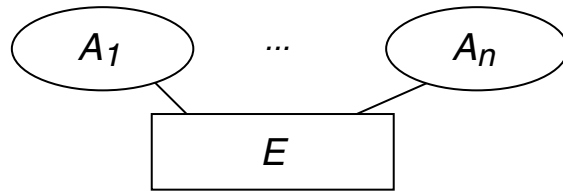


$$\mathcal{R}_E = \{\underline{ID}, A_1, \dots, A_n\}$$

Schlüssel: κ bzw. $\{ID\}$

Umsetzung ER-Schema in relationales Schema

Reguläre Entity-Typen



$$\mathcal{R}_E = \{\underline{ID}, A_1, \dots, A_n\}$$

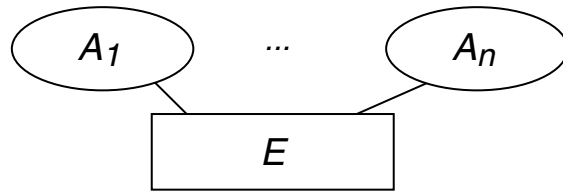
Schlüssel: κ bzw. $\{ID\}$

Umsetzung:

1. Dem Entity-Typ E wird Relationenschema \mathcal{R}_E zugeordnet.
Die Attribute A_1, \dots, A_n von E werden Attribute von \mathcal{R}_E .

Umsetzung ER-Schema in relationales Schema

Reguläre Entity-Typen



$$\mathcal{R}_E = \{\underline{ID}, A_1, \dots, A_n\}$$

Schlüssel: κ bzw. $\{ID\}$

Umsetzung:

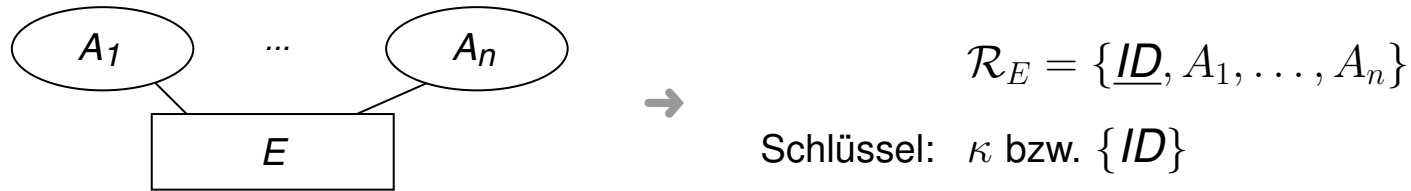
1. Dem Entity-Typ E wird Relationenschema \mathcal{R}_E zugeordnet.
Die Attribute A_1, \dots, A_n von E werden Attribute von \mathcal{R}_E .
2. Der Primärschlüssel $\kappa \subseteq \{A_1, \dots, A_n\}$ von E wird Primärschlüssel von \mathcal{R}_E .

Alternative:

Festlegen eines formalen Primärschlüssels durch Hinzufügen eines Schlüsselattributes ID zur Umsetzung der Eindeutigkeit von Entitäten. Der ursprüngliche Primärschlüssel κ ist dann ein weiterer Schlüssel im Relationenschema \mathcal{R}_E .

Umsetzung ER-Schema in relationales Schema

Reguläre Entity-Typen



Umsetzung:

1. Dem Entity-Typ E wird Relationenschema \mathcal{R}_E zugeordnet.
Die Attribute A_1, \dots, A_n von E werden Attribute von \mathcal{R}_E .
2. Der Primärschlüssel $\kappa \subseteq \{A_1, \dots, A_n\}$ von E wird Primärschlüssel von \mathcal{R}_E .

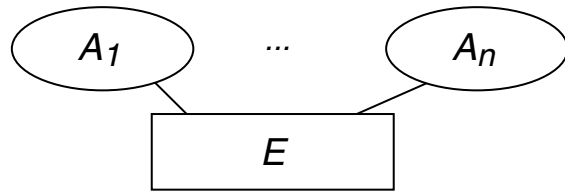
Alternative:

Festlegen eines formalen Primärschlüssels durch Hinzufügen eines Schlüsselattributes ID zur Umsetzung der Eindeutigkeit von Entitäten. Der ursprüngliche Primärschlüssel κ ist dann ein weiterer Schlüssel im Relationenschema \mathcal{R}_E .

3. Der Primärschlüssel wird durch Unterstreichen gekennzeichnet.

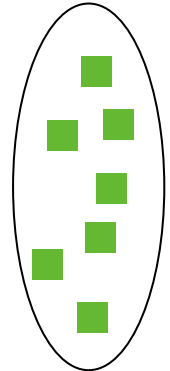
Umsetzung ER-Schema in relationales Schema

Reguläre Entity-Typen



A_1	A_2	A_3	...	A_n

$r(\mathcal{R}_E)$



$state(E)$

$$\mathcal{R}_E = \{ \underbrace{\underline{A_1}, \underline{A_2}, \underline{A_3}, \dots}_{\text{Schlüssel } \kappa}, A_n \}$$

Bemerkungen:

- Die Bezeichnung „regulärer Entity-Typ“ dient als Unterscheidung zu
 - abhängigen bzw. schwachen Entity-Typen,
 - spezialisierten Entity-Typen, die in einer IST-Beziehung stehen.
- Die Tupel in $r(\mathcal{R}_E)$ (zu einem bestimmten Zeitpunkt) entsprechen genau den Instanzen in $state(E)$ (zum gleichen Zeitpunkt).

Umsetzung ER-Schema in relationales Schema

Beziehungstypen

Zwei Umsetzungsstrategien:

- (a) Direkte Abbildung auf ein adäquates Schema.
- (b) Kanonische Umsetzung („Cross-Reference“) mit anschließender Zusammenfassung von Relationenschemata.

Umsetzung ER-Schema in relationales Schema

Beziehungstypen

Zwei Umsetzungsstrategien:

- (a) Direkte Abbildung auf ein adäquates Schema.
- (b) Kanonische Umsetzung („Cross-Reference“) mit anschließender Zusammenfassung von Relationenschemata.

Besondere Behandlung folgender Fälle:

1. 1:n-Beziehung (Formalismus I für Kardinalitäten)
2. 1:1-Beziehung (Formalismus I für Kardinalitäten)
3. $[0,1]$ und $[1,1]$ bei $[min, max]$ -Beschränkung (Formalismus II für Kardinalitäten)
4. existenzabhängige (schwache) Entity-Typen
5. IST-Beziehungstypen
6. reflexive Beziehungstypen

Umsetzung ER-Schema in relationales Schema

Beziehungstypen: Kapazitätserhaltung

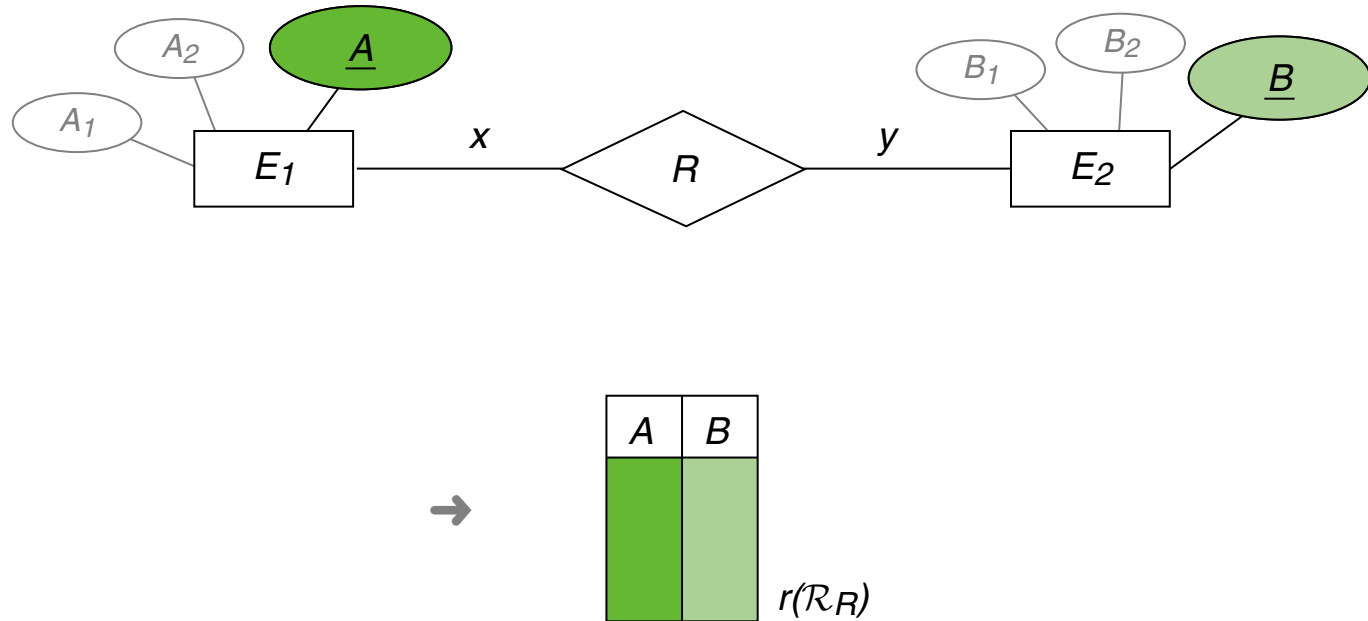
Eine zentrale Forderung bei der Abbildung von Beziehungstypen ist die Kapazitätserhaltung: die möglichen Zustände des ER-Modells sind auch mögliche Instanzen des relationalen Modells – und umgekehrt.

Definition 7 (kapazitätserhaltend)

Gibt es eine bijektive totale Abbildung zwischen den Zuständen eines Entity-Relationship-Modells und den Instanzen eines relationalen Modells, so nennt man die Transformation zwischen den Modellen kapazitätserhaltend.

Umsetzung ER-Schema in relationales Schema

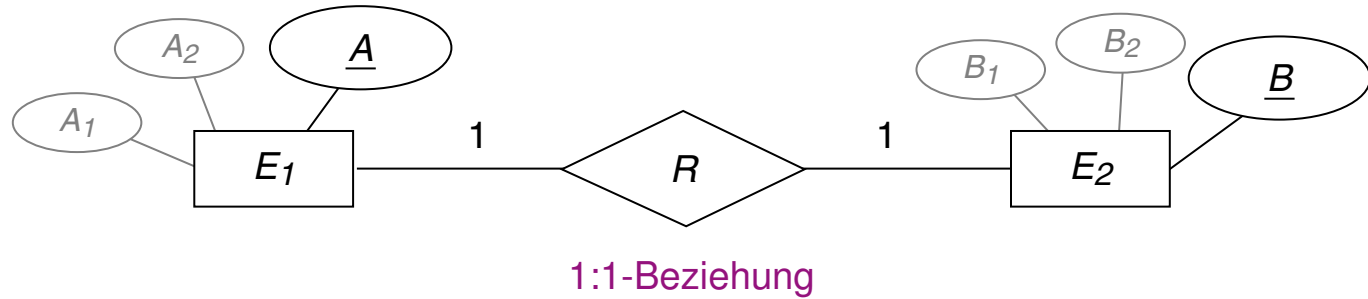
Beziehungstypen: Kapazitätserhaltung (Fortsetzung)



Wie ist der Schlüssel für \mathcal{R}_R bspw. bei einer 1:1-Beziehung zu wählen?

Umsetzung ER-Schema in relationales Schema

Beziehungstypen: Kapazitätserhaltung (Fortsetzung)

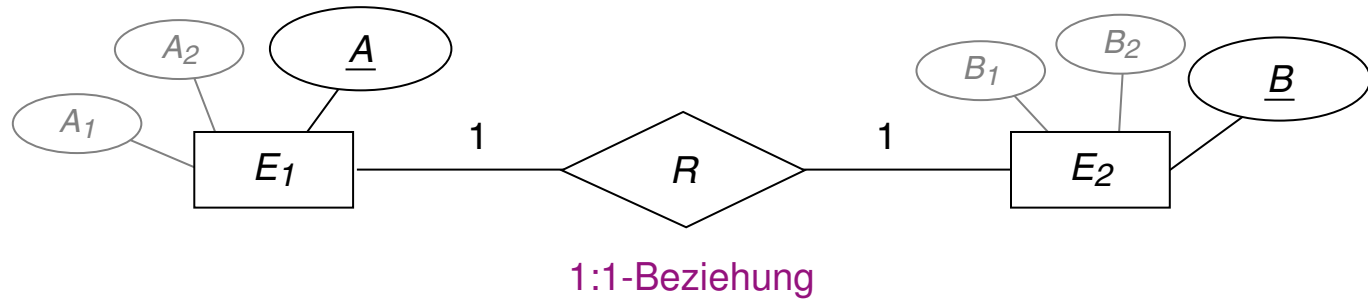


Modellierung (a)

$\mathcal{R}_R = \{\underline{A}, B\}$ mit Schlüssel $\{A\}$

Umsetzung ER-Schema in relationales Schema

Beziehungstypen: Kapazitätserhaltung (Fortsetzung)



Modellierung (a)

$\mathcal{R}_R = \{\underline{A}, B\}$ mit Schlüssel $\{A\}$

mögliche Relationen:

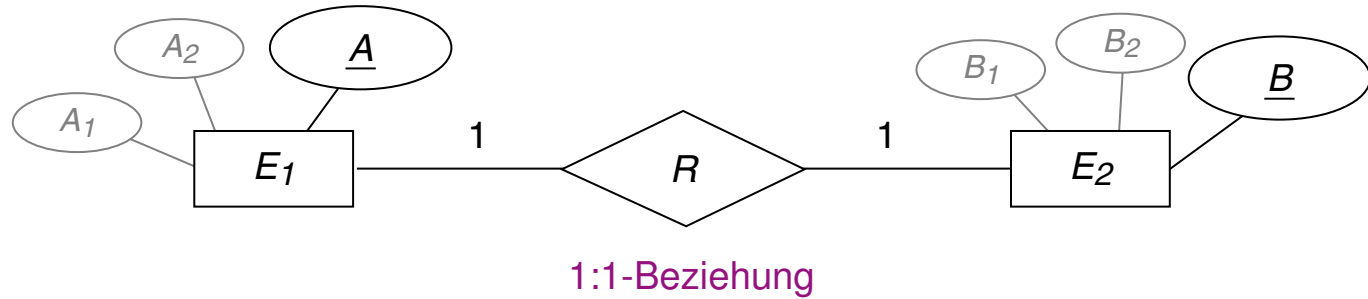
$$r_1(\mathcal{R}_R) = \{(a_1, b_1), (a_2, b_2)\}$$

$$r_2(\mathcal{R}_R) = \{(a_1, b_1), (a_2, b_1)\}$$

(kapazitätserhöhend)

Umsetzung ER-Schema in relationales Schema

Beziehungstypen: Kapazitätserhaltung (Fortsetzung)



Modellierung (a)

$\mathcal{R}_R = \{\underline{A}, B\}$ mit Schlüssel $\{A\}$

Modellierung (b)

$\mathcal{R}_R = \{\underline{A}, B\}$ mit *zwei* Schlüsseln $\{A\}, \{B\}$

mögliche Relationen:

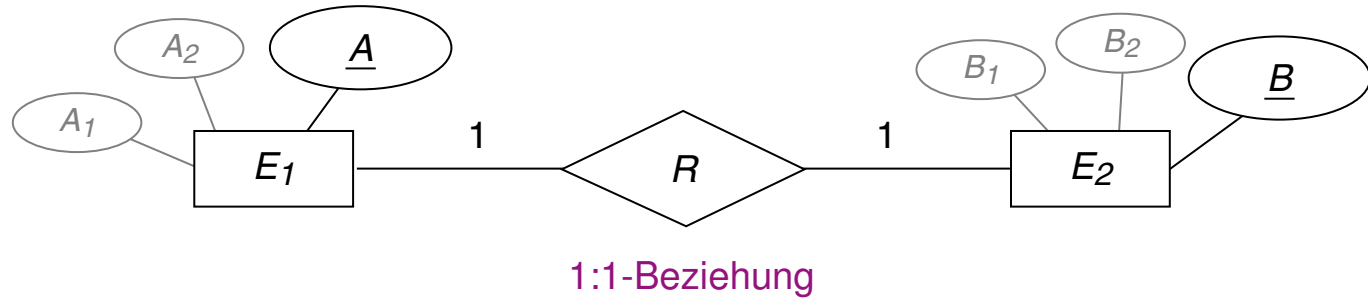
$$r_1(\mathcal{R}_R) = \{(a_1, b_1), (a_2, b_2)\}$$

$$r_2(\mathcal{R}_R) = \{(a_1, b_1), (a_2, b_1)\}$$

(kapazitätserhöhend)

Umsetzung ER-Schema in relationales Schema

Beziehungstypen: Kapazitätserhaltung (Fortsetzung)



Modellierung (a)

$\mathcal{R}_R = \{\underline{A}, B\}$ mit Schlüssel $\{A\}$

mögliche Relationen:

$$r_1(\mathcal{R}_R) = \{(a_1, b_1), (a_2, b_2)\}$$

$$r_2(\mathcal{R}_R) = \{(a_1, b_1), (a_2, b_1)\}$$

(kapazitätserhöhend)

Modellierung (b)

$\mathcal{R}_R = \{\underline{A}, B\}$ mit *zwei* Schlüsseln $\{A\}, \{B\}$

mögliche Relation:

$$r(\mathcal{R}_R) = \{(a_1, b_1), (a_2, b_2)\}$$

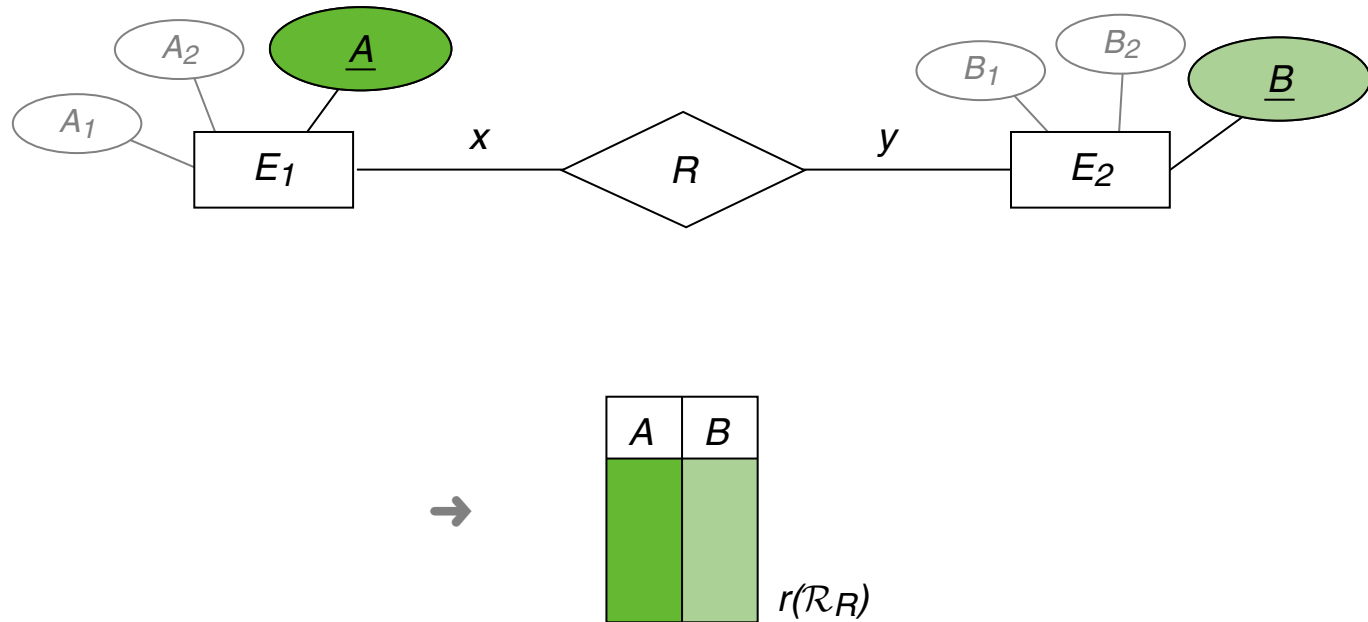
(kapazitätserhaltend)

Bemerkungen:

- Auch wenn wir nur einen Schlüssel als Primärschlüssel auszeichnen und uns auf diesen i.d.R. beziehen, so bleiben die anderen Schlüsselkandidaten in ihrem Wesen als Schlüssel erhalten: je zwei Tupel aus der Relation müssen sich in der Ausprägung von mindestens einem Schlüsselattribut unterscheiden.

Umsetzung ER-Schema in relationales Schema

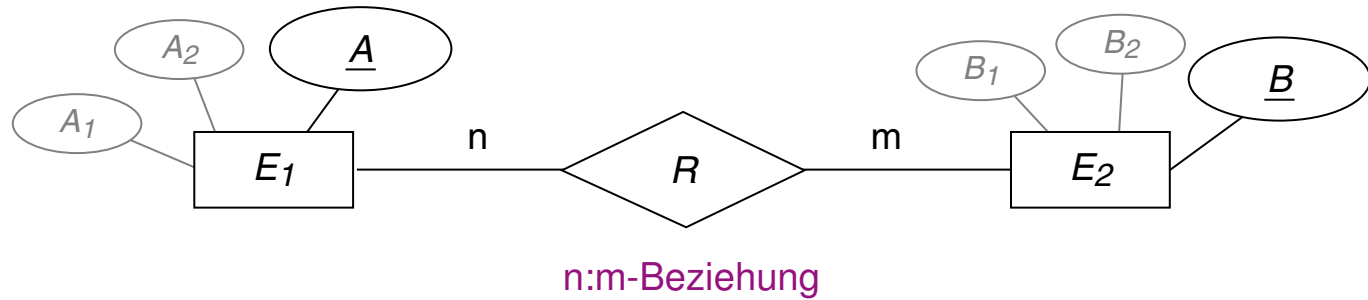
Kapazitätserhaltung (Fortsetzung)



Wie ist der Schlüssel für \mathcal{R}_R bei einer n:m-Beziehung zu wählen?

Umsetzung ER-Schema in relationales Schema

Kapazitätserhaltung (Fortsetzung)

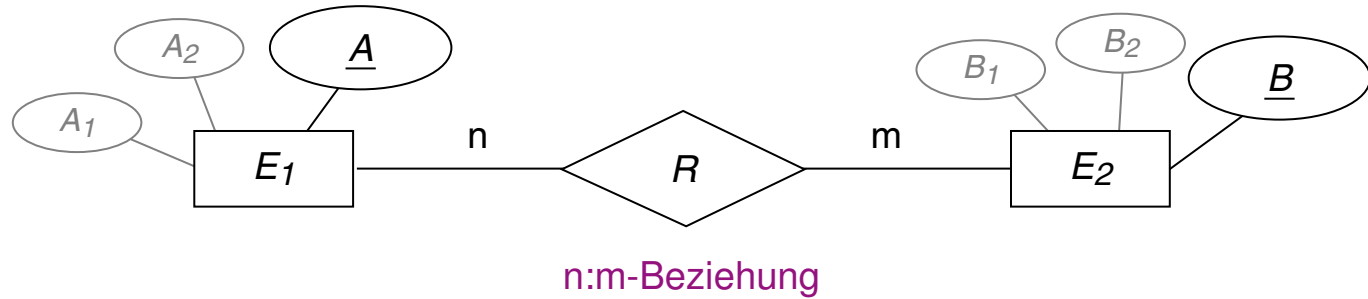


Modellierung (a)

$\mathcal{R}_R = \{\underline{A}, B\}$ mit Schlüssel $\{A\}$

Umsetzung ER-Schema in relationales Schema

Kapazitätserhaltung (Fortsetzung)



Modellierung (a)

$\mathcal{R}_R = \{\underline{A}, B\}$ mit Schlüssel $\{A\}$

mögliche Relation:

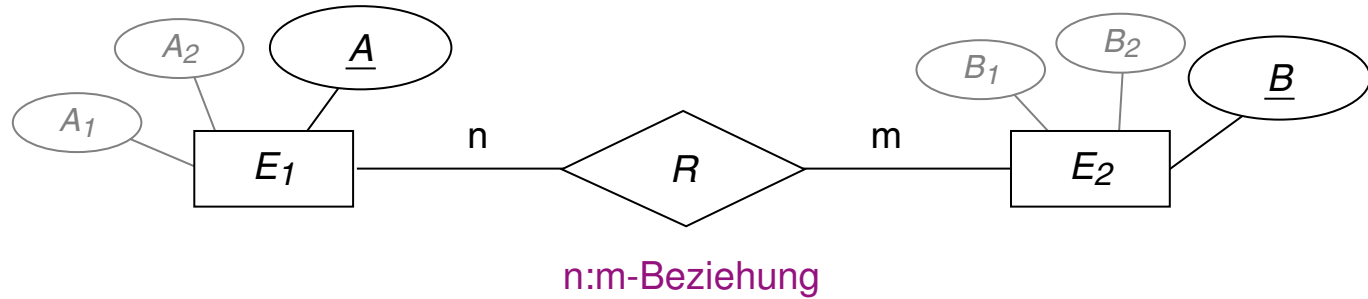
$$r(\mathcal{R}_R) = \{(a_1, b_1), (a_2, b_1)\}$$

$$\cancel{r(\mathcal{R}_R) = \{(a_1, b_1), (a_1, b_2), (a_2, b_2)\}}$$

(kapazitätsvermindernd)

Umsetzung ER-Schema in relationales Schema

Kapazitätserhaltung (Fortsetzung)



Modellierung (a)

$\mathcal{R}_R = \{\underline{A}, B\}$ mit Schlüssel $\{A\}$

mögliche Relation:

$$r(\mathcal{R}_R) = \{(a_1, b_1), (a_2, b_1)\}$$

$$\cancel{r(\mathcal{R}_R) = \{(a_1, b_1), (a_1, b_2), (a_2, b_2)\}}$$

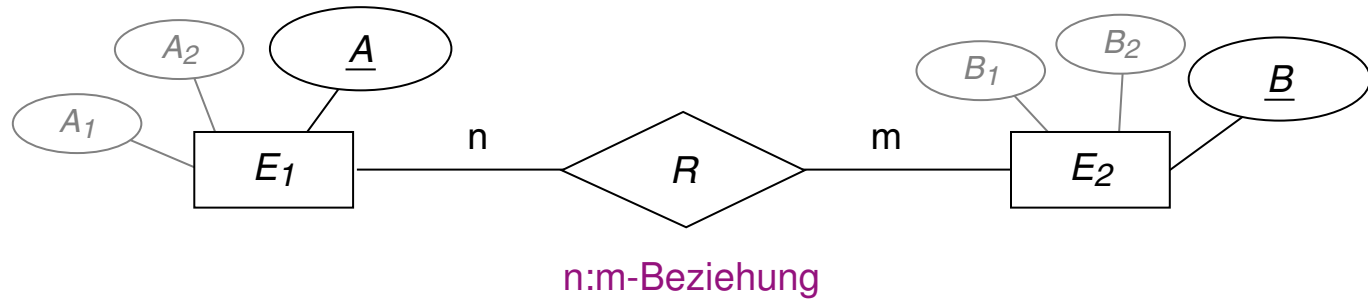
(kapazitätsvermindernd)

Modellierung (b)

$\mathcal{R}_R = \{\underline{A}, \underline{B}\}$ mit Schlüssel $\{A, B\}$

Umsetzung ER-Schema in relationales Schema

Kapazitätserhaltung (Fortsetzung)



Modellierung (a)

$\mathcal{R}_R = \{\underline{A}, B\}$ mit Schlüssel $\{A\}$

mögliche Relation:

$$r(\mathcal{R}_R) = \{(a_1, b_1), (a_2, b_1)\}$$

~~$$r(\mathcal{R}_R) = \{(a_1, b_1), (a_1, b_2), (a_2, b_2)\}$$~~

(kapazitätsvermindernd)

Modellierung (b)

$\mathcal{R}_R = \{\underline{A}, \underline{B}\}$ mit Schlüssel $\{A, B\}$

mögliche Relationen:

$$r_1(\mathcal{R}_R) = \{(a_1, b_1), (a_2, b_2)\}$$

$$r_2(\mathcal{R}_R) = \{(a_1, b_1), (a_1, b_2), (a_2, b_2)\}$$

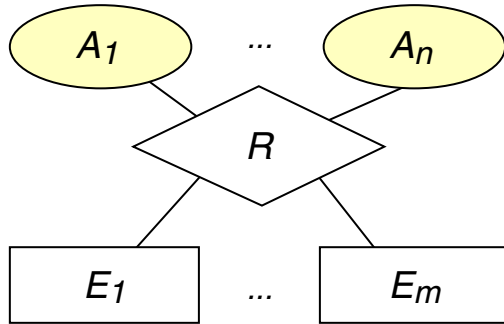
(kapazitätserhaltend)

Bemerkungen:

- Weil A und B gemeinsam den Schlüssel $\{A, B\}$ bilden, kann in der Relation r *eine* Attributausprägung von A mit *mehreren* Ausprägungen von B vorkommen (und umgekehrt), ohne dass die Schlüsselintegrität (= die eindeutige Identifizierbarkeit von Tupeln) für r verletzt ist.

Umsetzung ER-Schema in relationales Schema

Reguläre Beziehungstypen und beliebige Entity-Anzahl



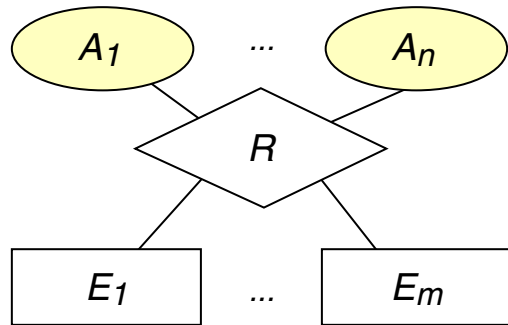
$$\mathcal{R}_R = \{ID_1, \dots, ID_m, A_1, \dots, A_n\}$$

Schlüssel: $\alpha \subseteq \kappa_1 \cup \dots \cup \kappa_m$ bzw.

$$\alpha \subseteq \{ID_1, \dots, ID_m\}$$

Umsetzung ER-Schema in relationales Schema

Reguläre Beziehungstypen und beliebige Entity-Anzahl



$$\mathcal{R}_R = \{ID_1, \dots, ID_m, A_1, \dots, A_n\}$$

Schlüssel: $\alpha \subseteq \kappa_1 \cup \dots \cup \kappa_m$ bzw.

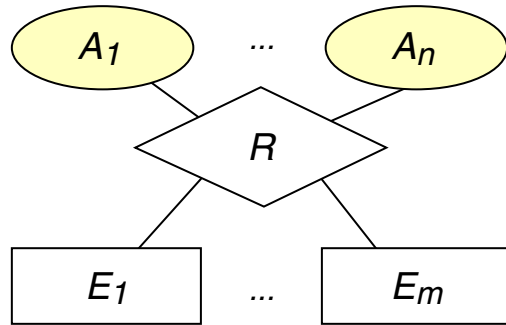
$$\alpha \subseteq \{ID_1, \dots, ID_m\}$$

Cross-Reference [Elmasri/Navathe 2016] :

1. Dem Beziehungstyp R wird Relationenschema \mathcal{R}_R zugeordnet.
Die Attribute A_1, \dots, A_n von R werden Attribute von \mathcal{R}_R .

Umsetzung ER-Schema in relationales Schema

Reguläre Beziehungstypen und beliebige Entity-Anzahl



$$\mathcal{R}_R = \{ID_1, \dots, ID_m, A_1, \dots, A_n\}$$

Schlüssel: $\alpha \subseteq \kappa_1 \cup \dots \cup \kappa_m$ bzw.

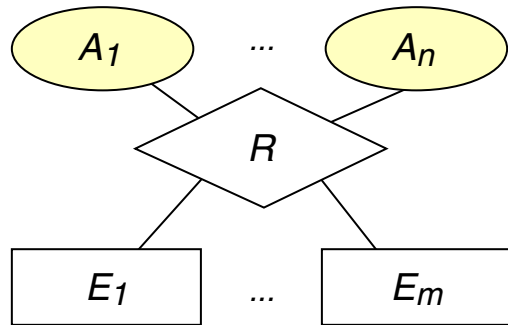
$$\alpha \subseteq \{ID_1, \dots, ID_m\}$$

Cross-Reference [Elmasri/Navathe 2016] :

1. Dem Beziehungstyp R wird Relationenschema \mathcal{R}_R zugeordnet.
Die Attribute A_1, \dots, A_n von R werden Attribute von \mathcal{R}_R .
2. Die Attribute in den κ_i (bzw. die ID_i) von \mathcal{R}_{E_i} werden Attribute von \mathcal{R}_R .

Umsetzung ER-Schema in relationales Schema

Reguläre Beziehungstypen und beliebige Entity-Anzahl



$$\mathcal{R}_R = \{ID_1, \dots, ID_m, A_1, \dots, A_n\}$$

Schlüssel: $\alpha \subseteq \kappa_1 \cup \dots \cup \kappa_m$ bzw.

$$\alpha \subseteq \{ID_1, \dots, ID_m\}$$

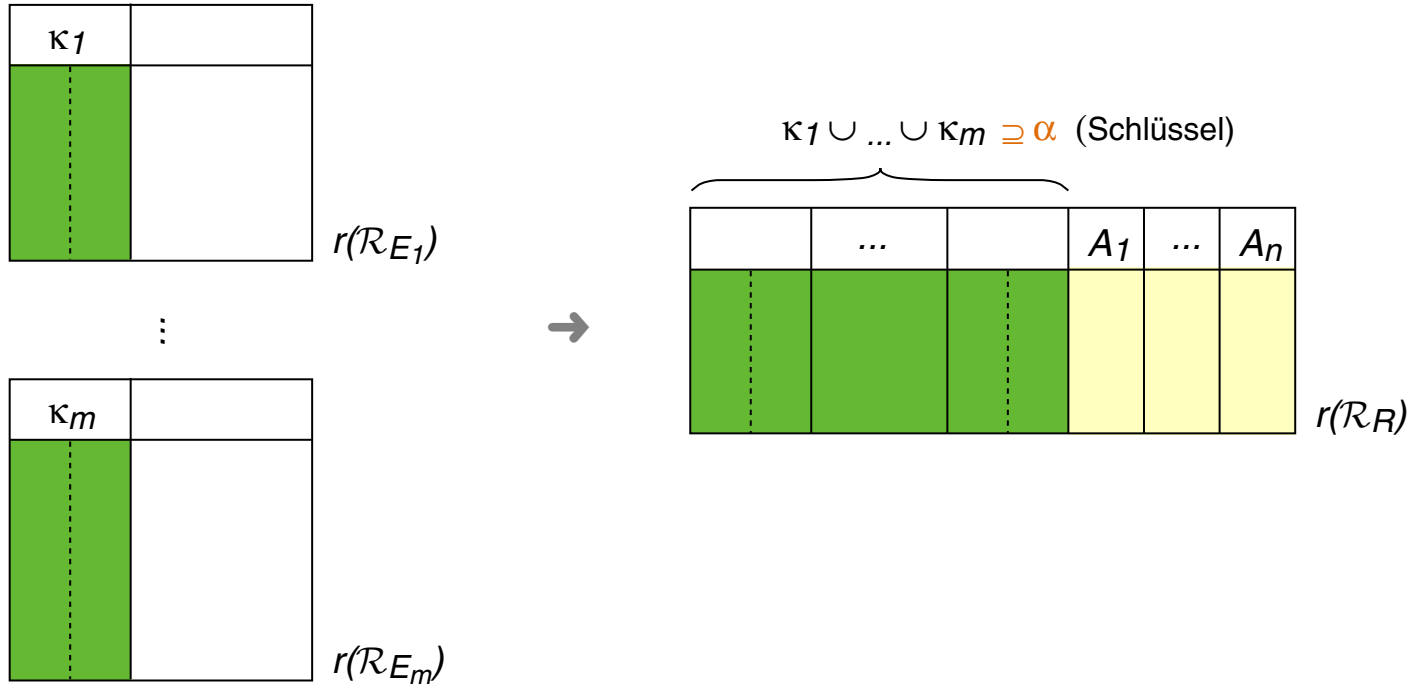
Cross-Reference [Elmasri/Navathe 2016] :

1. Dem Beziehungstyp R wird Relationenschema \mathcal{R}_R zugeordnet.
Die Attribute A_1, \dots, A_n von R werden Attribute von \mathcal{R}_R .
2. Die Attribute in den κ_i (bzw. die ID_i) von \mathcal{R}_{E_i} werden Attribute von \mathcal{R}_R .
3. Der Schlüssel von \mathcal{R}_R ist eine **Teilmenge** der Vereinigungsmenge der κ_i (bzw. der Menge aller ID_i).

Umsetzung ER-Schema in relationales Schema

Reguläre Beziehungstypen und beliebige Entity-Anzahl (Fortsetzung)

Cross-Reference:



Bemerkungen:

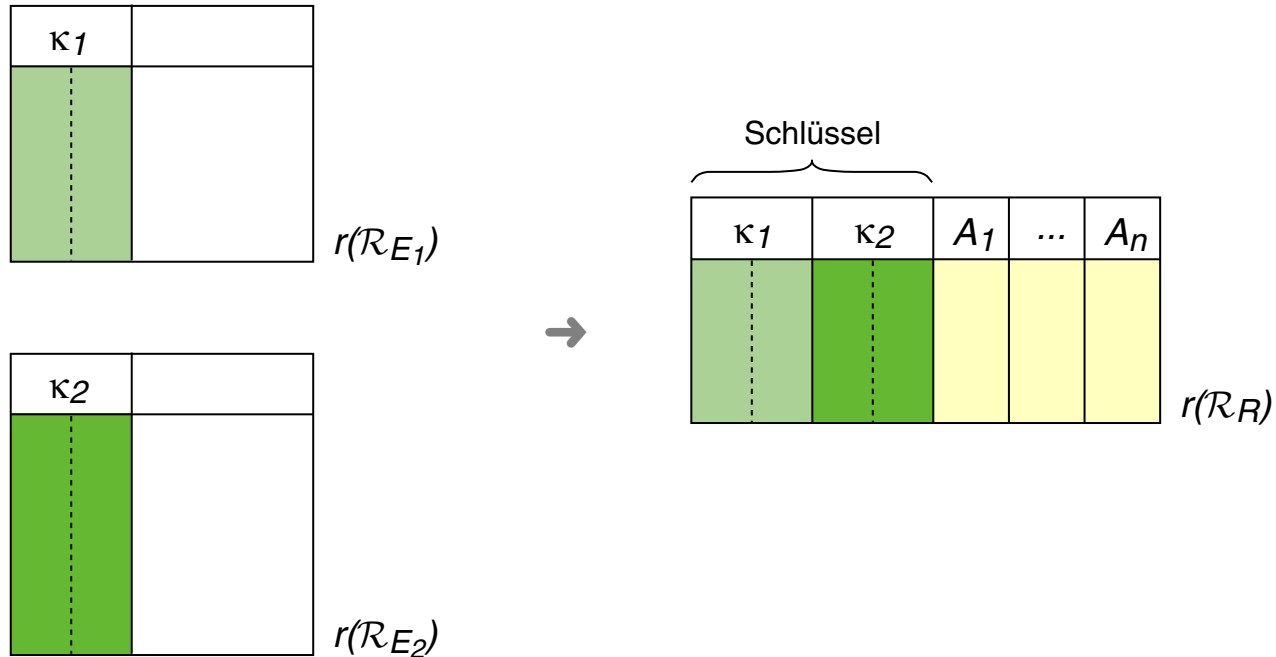
- Die Bezeichnung „regulärer Beziehungstyp“ dient als Unterscheidung zu
 - Beziehungstypen für abhängige bzw. schwache Entity-Typen,
 - IST-Beziehungstypen.
- Die $\kappa_i \subset \mathcal{R}_R$ (bzw. die $\{ID_i\} \subset \mathcal{R}_R$) sind Fremdschlüssel in \mathcal{R}_R bzgl. κ_i (bzw. $\{ID_i\}$) in \mathcal{R}_{E_i} .
- Es stellt sich die Frage, wie die Teilmenge aus der Vereinigungsmenge der κ_i gebildet wird, so dass ein Schlüssel für die Relation \mathcal{R}_R entsteht. Man kann diese Frage nicht in der Allgemeinheit beantworten.

Vergleiche hierzu die möglichen funktionalen Beziehungen, die beispielsweise von einer $x : y : z$ -Relation, $x, y, z \in \{1, n, m\}$, impliziert sein können: falls keine funktionale Beziehung gegeben ist, also x und y und $z \neq 1$, so bilden nur alle Schlüsselattribute der drei Entity-Typen zusammen einen Schlüssel für \mathcal{R}_R . Gibt es einen funktionalen Zusammenhang, also x oder y oder $z = 1$, so bildet die Vereinigungsmenge der Schlüsselattribute der beiden Entity-Typen des Urbildbereiches einen Schlüssel für \mathcal{R}_R .

Umsetzung ER-Schema in relationales Schema

n:m-Beziehungstypen

Cross-Reference:



Die Primärschlüssel der beteiligten Relationenschemata \mathcal{R}_{E_1} und \mathcal{R}_{E_2} bilden zusammen den Schlüssel im Relationenschema \mathcal{R}_R des n:m-Beziehungstyps.

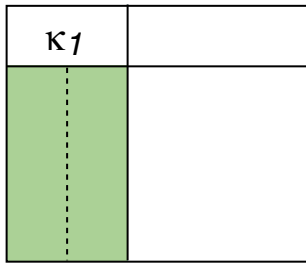
[Kapazitätserhaltung]

Umsetzung ER-Schema in relationales Schema

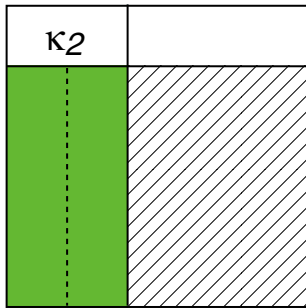
1:n-Beziehungstypen

Funktionaler Zusammenhang $E_2 \rightarrow E_1$:

1-Entity-Typ

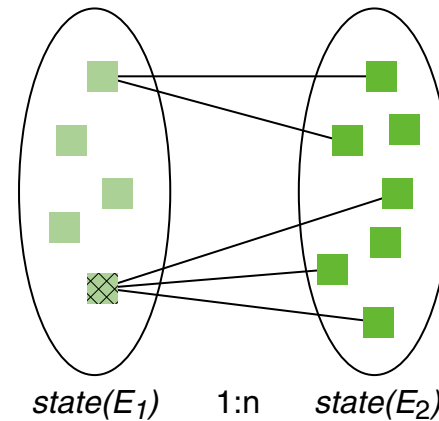


$r(\mathcal{R}_{E_1})$



$r(\mathcal{R}_{E_2})$

n-Entity-Typ



Umsetzung ER-Schema in relationales Schema

1:n-Beziehungstypen (Fortsetzung)

Cross-Reference:

1-Entity-Typ

κ_1	

$r(\mathcal{R}_{E_1})$

κ_2	

$r(\mathcal{R}_{E_2})$

n-Entity-Typ



Schlüssel

κ_1	κ_2	A_1	\dots	A_n

$r(\mathcal{R}_R)$

Umsetzung ER-Schema in relationales Schema

1:n-Beziehungstypen (Fortsetzung) [[Sonderfall 1](#)]

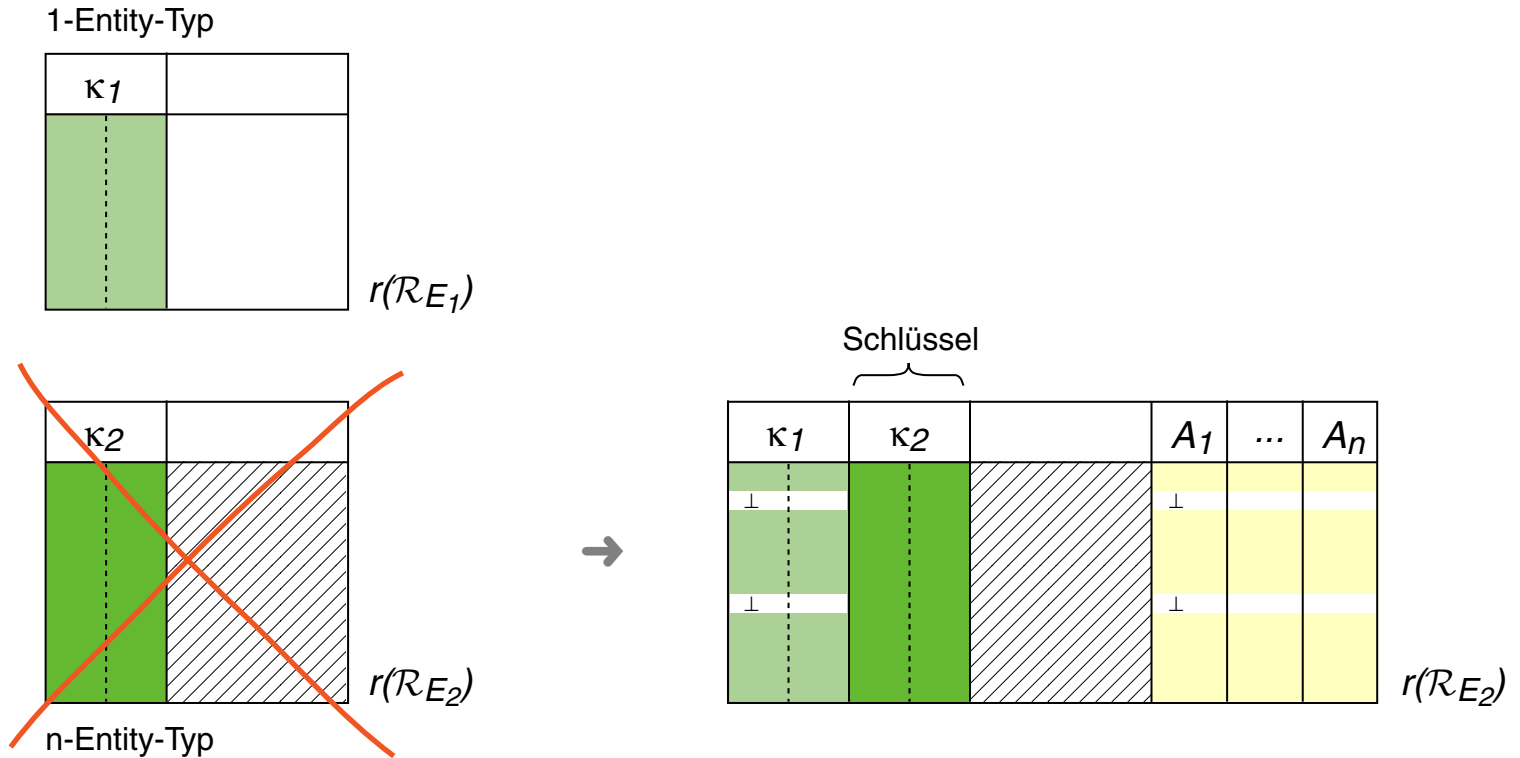
Als Verfeinerung der Cross-Reference kann man bei 1:n-Beziehungstypen das Relationenschema \mathcal{R}_R mit dem Relationenschema \mathcal{R}_{E_2} , das den **n-Entity-Typ** im 1:n-Beziehungstyp repräsentiert, zusammenfassen (\mathcal{R}_R fällt dann weg):

1. Die Attribute des Primärschlüssels in \mathcal{R}_{E_1} werden Attribute in \mathcal{R}_{E_2} und stellen dort einen entsprechenden Fremdschlüssel dar.
2. Die Attribute des 1:n-Beziehungstyps werden Attribute in \mathcal{R}_{E_2} .
3. Der Primärschlüssel des n-Entity-Typs wird Schlüssel im zusammengefassten Relationenschema.

Umsetzung ER-Schema in relationales Schema

1:n-Beziehungstypen (Fortsetzung)

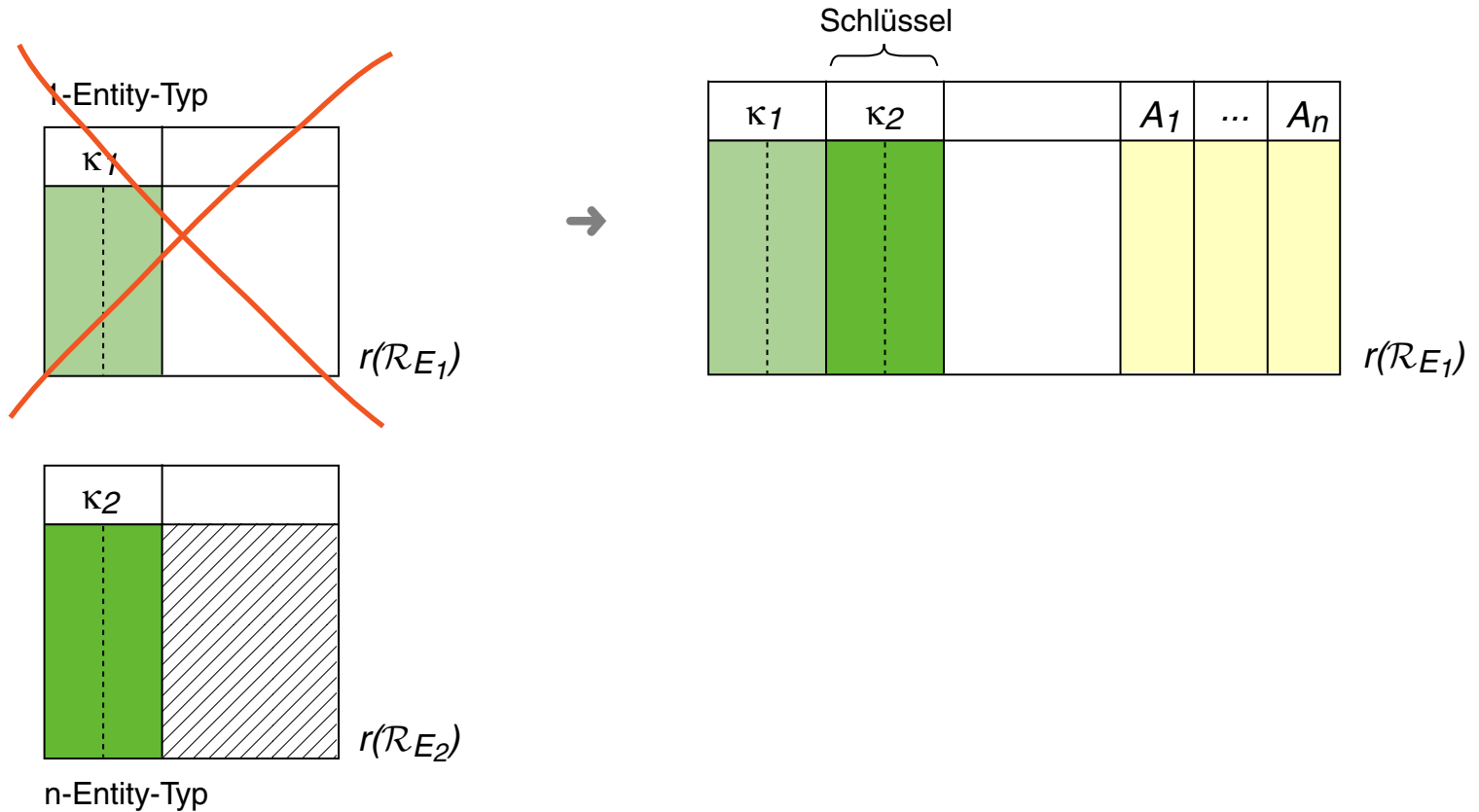
Erlaubte Zusammenfassung von \mathcal{R}_R und \mathcal{R}_{E_2} :



Umsetzung ER-Schema in relationales Schema

1:n-Beziehungstypen (Fortsetzung)

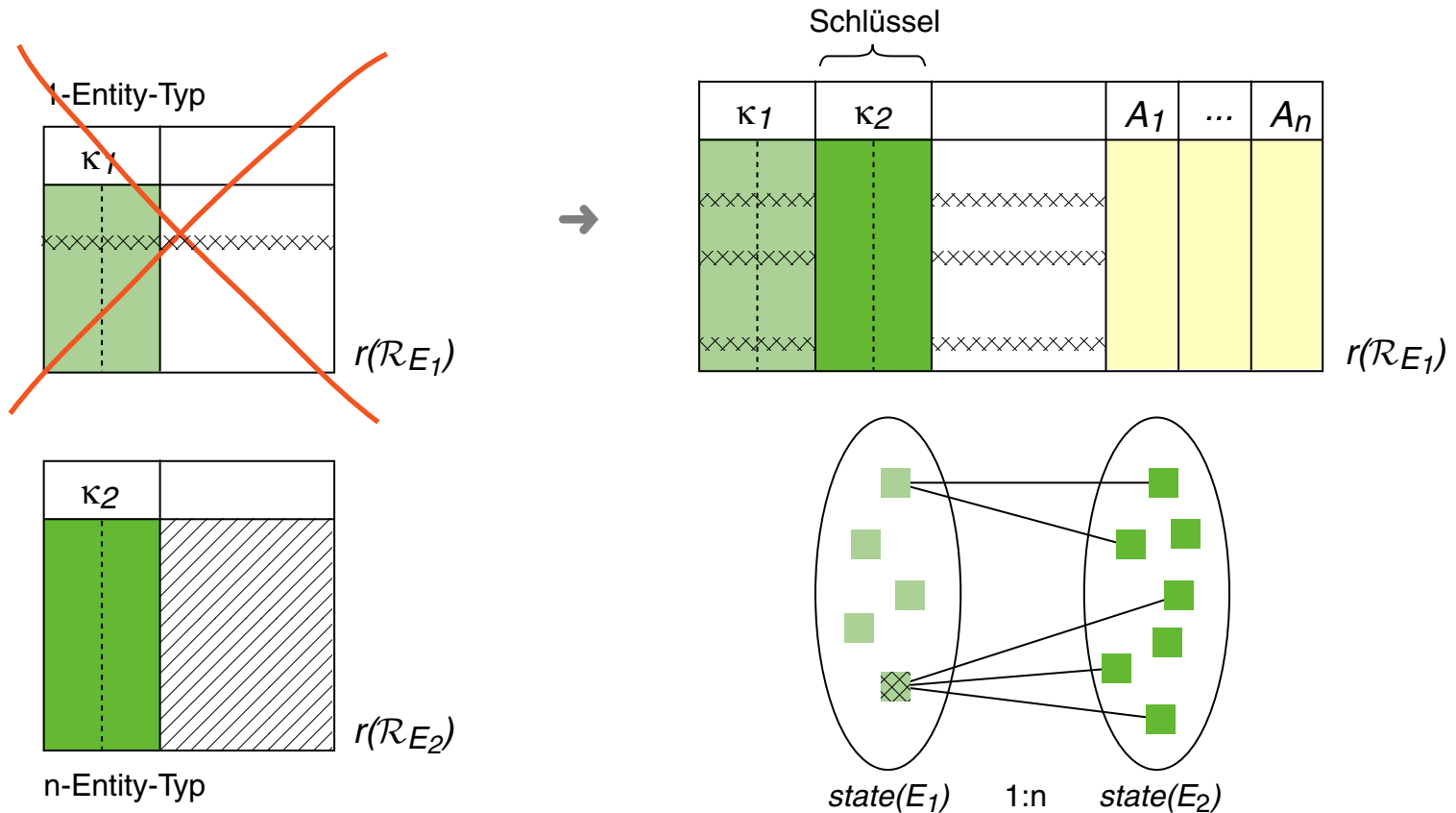
Unerlaubte Zusammenfassung von \mathcal{R}_R und \mathcal{R}_{E_1} :



Umsetzung ER-Schema in relationales Schema

1:n-Beziehungstypen (Fortsetzung)

Unerlaubte Zusammenfassung von \mathcal{R}_R und \mathcal{R}_{E_1} :



Bemerkungen:

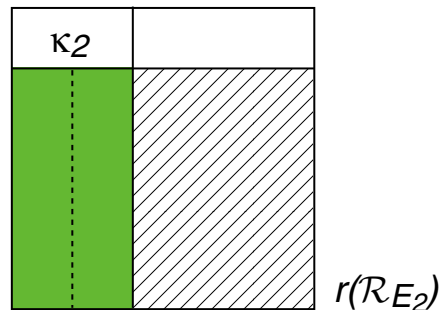
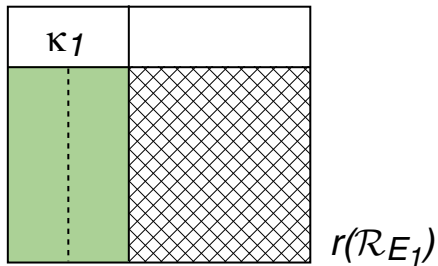
- ❑ [Kemper/Eickler 2015] gibt folgende Regel als Hilfe bei der Zusammenfassung von Relationen an: „Nur Relationen mit gleichem Schlüssel zusammenfassen.“
In der Illustration sind das die beiden Relationen \mathcal{R}_R und \mathcal{R}_{E_2} ; beide haben den Schlüssel κ_2 .
- ❑ Bei der „erlaubten“ Zusammenfassung entstehen Nullwerte (\perp) bei allen Entitäten des Typs E_2 , die nicht in Beziehung mit einer Entität des Typs E_1 stehen.
- ❑ Bei der „unerlaubten“ Zusammenfassung werden alle Daten der Entitäten des Typs E_1 , die mit mehr als einer Entität des Typ E_2 in Beziehung stehen, redundant gespeichert. Der eindeutige Zugriff auf eine Entität des Typs E_1 ist nicht möglich.
- ❑ Die Konsistenz wird in beiden Zusammenfassungen erhalten.

Umsetzung ER-Schema in relationales Schema

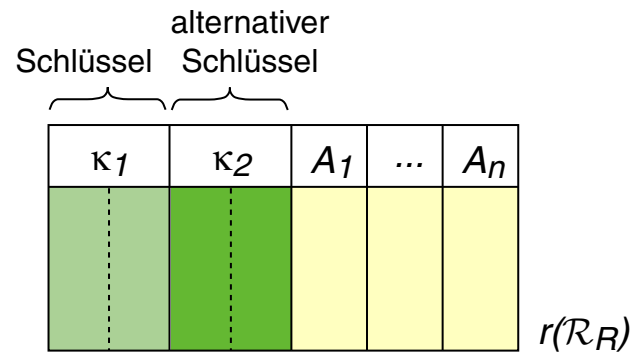
1:1-Beziehungstypen [Kapazitätserhaltung]

Cross-Reference:

1-Entity-Typ



1-Entity-Typ



Umsetzung ER-Schema in relationales Schema

1:1-Beziehungstypen (Fortsetzung) [[Sonderfall 2](#)]

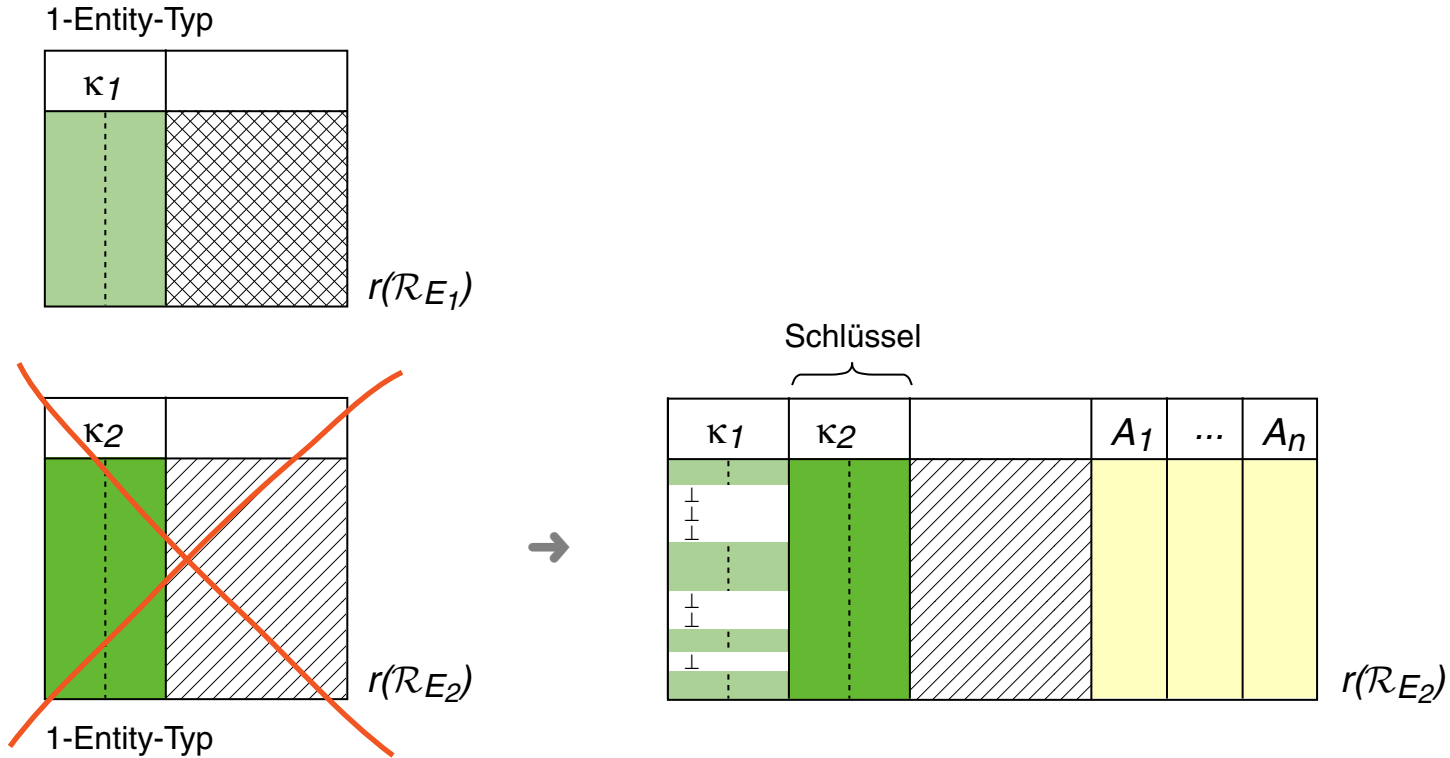
Als Verfeinerung der Cross-Reference kann man bei 1:1-Beziehungstypen das Relationenschema \mathcal{R}_R mit einem der beiden Relationenschemata der beteiligten Entity-Typen, \mathcal{R}_{E_2} oder \mathcal{R}_{E_1} , zusammenfassen:

1. Die Attribute des Primärschlüssels in \mathcal{R}_{E_1} (\mathcal{R}_{E_2}) werden Attribute in \mathcal{R}_{E_2} (\mathcal{R}_{E_1}) und stellen dort einen entsprechenden Fremdschlüssel dar.
2. Die Attribute des 1:1-Beziehungstyps werden Attribute in \mathcal{R}_{E_2} (\mathcal{R}_{E_1}).
3. Der Primärschlüssel von \mathcal{R}_{E_2} (\mathcal{R}_{E_1}) wird Schlüssel im zusammengefassten Relationenschema.

Umsetzung ER-Schema in relationales Schema

1:1-Beziehungstypen (Fortsetzung)

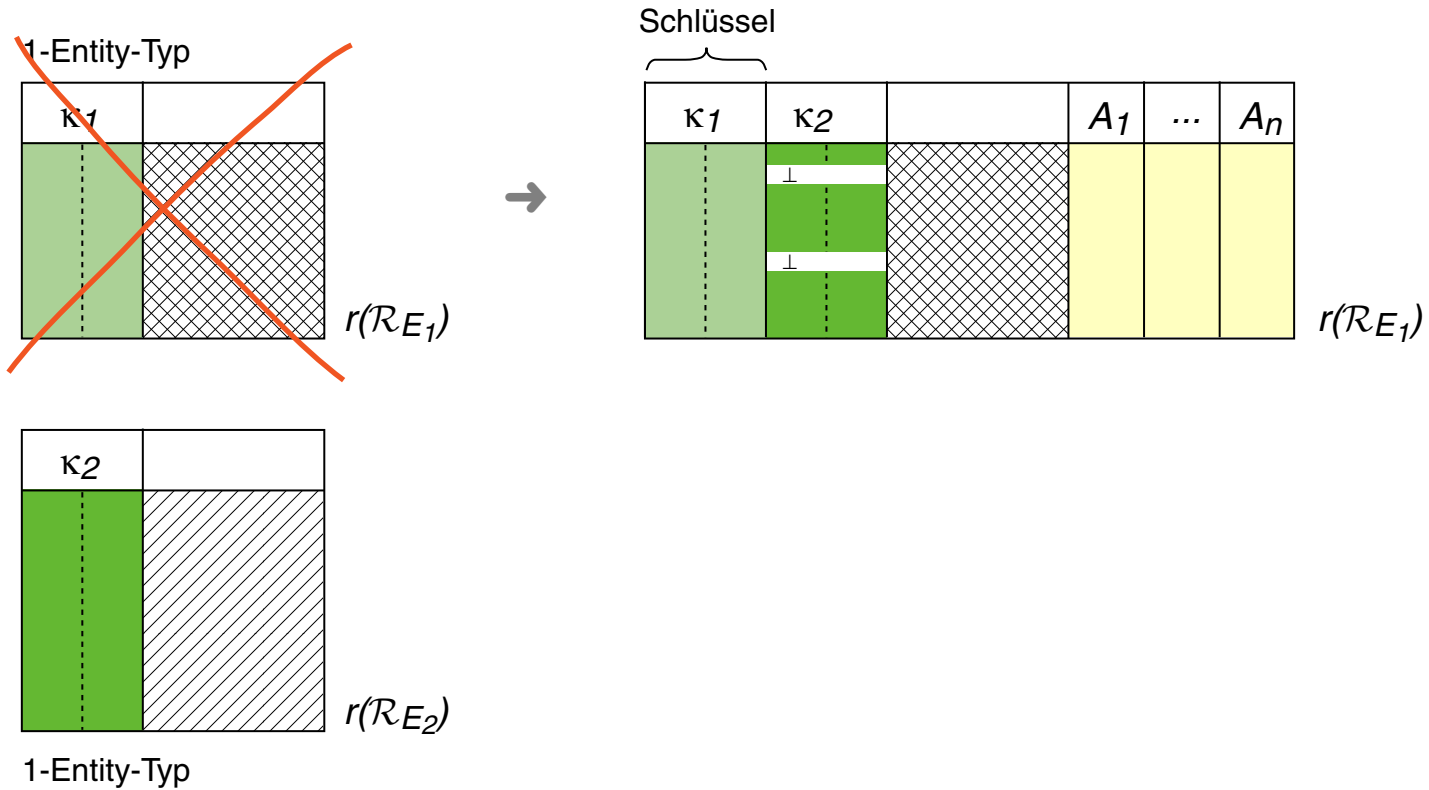
Zusammenfassung von \mathcal{R}_R und \mathcal{R}_{E_2} :



Umsetzung ER-Schema in relationales Schema

1:1-Beziehungstypen (Fortsetzung)

Zusammenfassung von \mathcal{R}_R und \mathcal{R}_{E_1} :



Bemerkungen:

- ❑ Die dargestellten Zusammenfassungen bei der Umsetzung von 1:1-Beziehungstypen finden sich so auch in der Literatur; sie sind aber mit Vorsicht zu genießen:
Im Gegensatz zu der Cross-Reference-Umsetzung ist die Kapazitätserhaltung nur bei einer totalen Teilnahme des aufnehmenden Entity-Typs gegeben. Liegt dieser Sachverhalt nicht vor, enthält der Fremdschlüssel Nullwerte – mit der Folge, dass er im zusammengefassten Schema keinen alternativen Schlüssel mehr darstellt.
- ❑ Manche DBMS stellen Datentypen zu Verfügung, mittels derer die Eindeutigkeit aller Nicht-Null-Werte vereinbart werden kann und gleichzeitig beliebig viele Nullwerte zugelassen sind. Damit kann die Kapazitätserhaltung sichergestellt werden, auch wenn der aufnehmende Entity-Typ (E_2 im ersten bzw. E_1 im zweiten Beispiel) nicht total teilnimmt.
- ❑ Nehmen nur wenige Instanzen der beiden Entity-Typen an der Beziehung teil, sollte auf eine Zusammenfassung verzichtet werden.

Umsetzung ER-Schema in relationales Schema

1:1-Beziehungstypen (Fortsetzung)

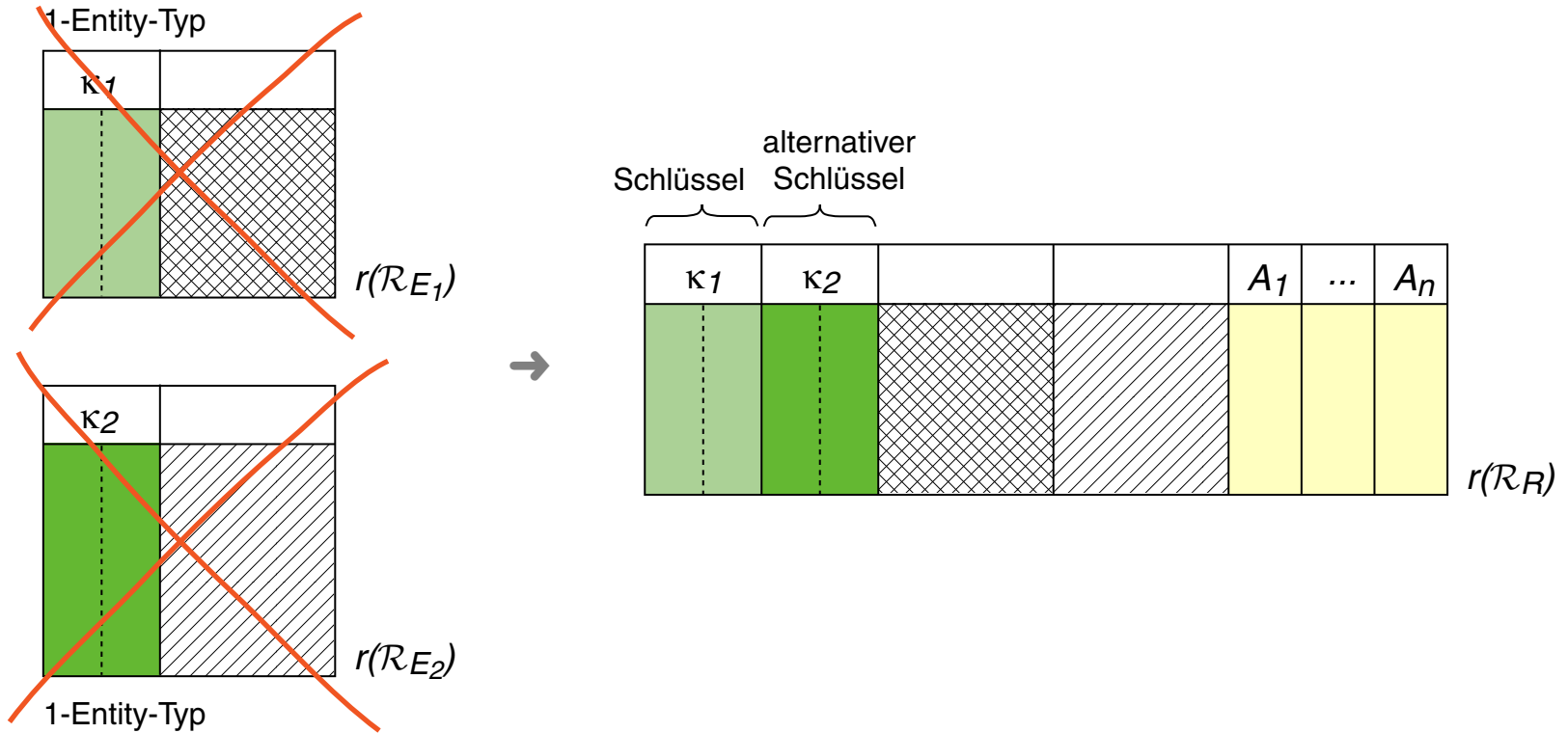
Ist die Teilnahme beider Entity-Typen am Beziehungstyp total – existiert also eine bijektive totale Abbildung zwischen E_1 und E_2 – lassen sich \mathcal{R}_{E_1} und \mathcal{R}_{E_2} in *einem* Relationenschema zusammenfassen. Merged-Relation [Elmasri/Navathe 2016] :

1. ...
2. ...
3. Die Primärschlüssel **beider** Entity-Typen sind Schlüssel im zusammengefassten Relationenschema; von ihnen wird **einer** als Primärschlüssel gewählt. [[Kapazitätserhaltung](#)]

Umsetzung ER-Schema in relationales Schema

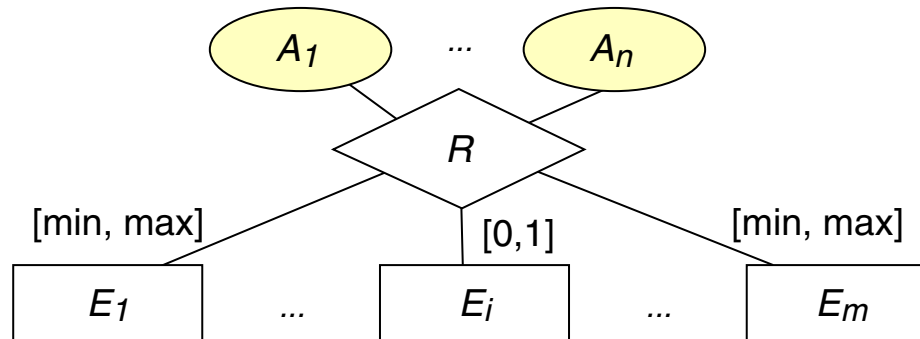
1:1-Beziehungstypen (Fortsetzung)

Merged-Relation:



Umsetzung ER-Schema in relationales Schema

Beziehungstypen mit $[min, max]$ -Beschränkung [Sonderfall 3]

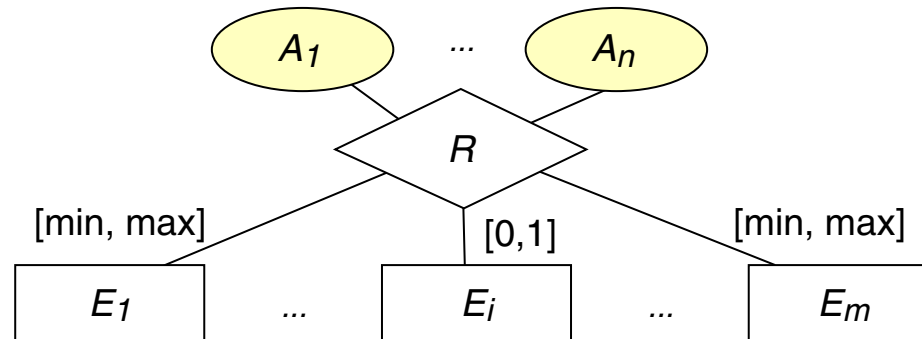


(a) m -äre Beziehungstypen mit $[0, 1]$ -Beschränkung für Entity-Typ E_i :
 $R(E_1[min_1, max_1], \dots, E_i[0, 1], \dots, E_m[min_m, max_m])$

□ Der Primärschlüssel von \mathcal{R}_{E_i} wird ein Schlüssel von \mathcal{R}_R .

Umsetzung ER-Schema in relationales Schema

Beziehungstypen mit $[min, max]$ -Beschränkung [Sonderfall 3]



(a) m -äre Beziehungstypen mit $[0, 1]$ -Beschränkung für Entity-Typ E_i :
 $R(E_1[*min_1*, *max_1*], \dots, E_i[0, 1], \dots, E_m[*min_m*, *max_m*])$

□ Der Primärschlüssel von \mathcal{R}_{E_i} wird ein Schlüssel von \mathcal{R}_R .

(b) m -äre Beziehungstypen mit $[1, 1]$ -Beschränkung für Entity-Typ E_i :
 $R(E_1[*min_1*, *max_1*], \dots, E_i[1, 1], \dots, E_m[*min_m*, *max_m*])$

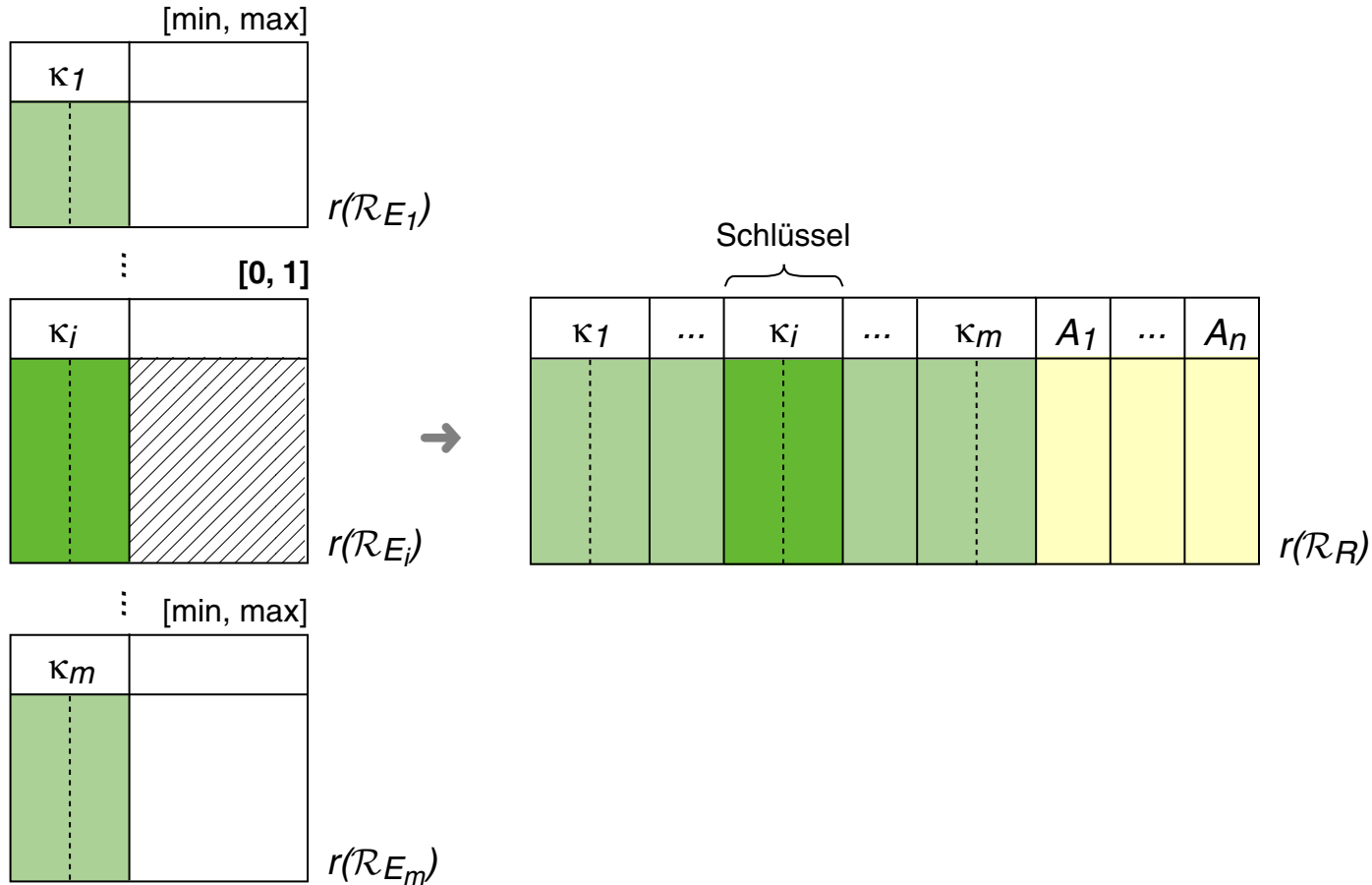
□ Der Primärschlüssel von \mathcal{R}_{E_i} wird ein Schlüssel von \mathcal{R}_R .

□ Die Relationenschemata \mathcal{R}_R und \mathcal{R}_{E_i} können zusammengefasst werden. Alle Schlüssel von \mathcal{R}_{E_i} werden auch Schlüssel von \mathcal{R}_R .

Umsetzung ER-Schema in relationales Schema

Beziehungstypen mit $[min, max]$ -Beschränkung

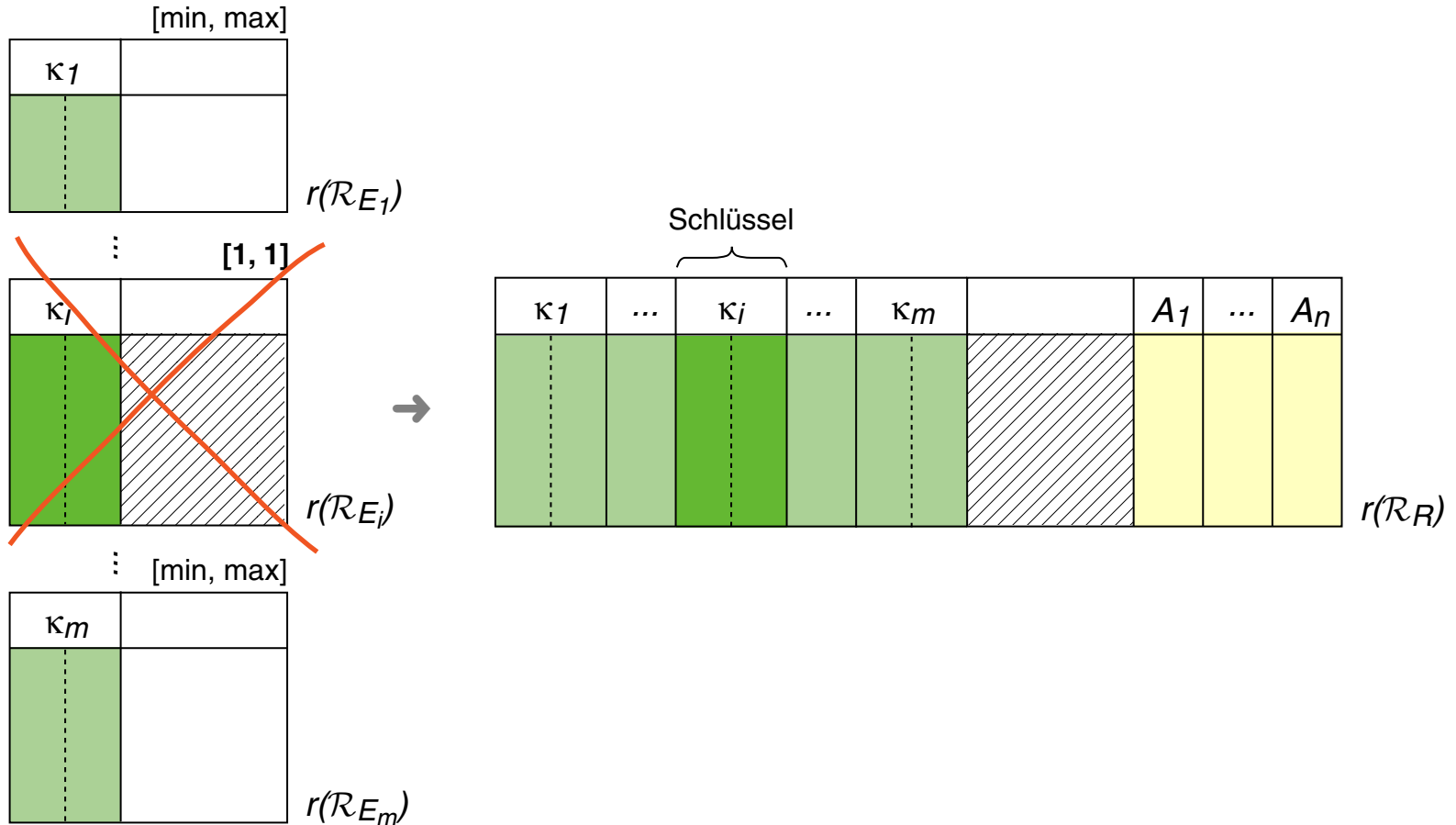
zu (a) Cross-Reference:



Umsetzung ER-Schema in relationales Schema

Beziehungstypen mit $[min, max]$ -Beschränkung

zu (b) Zusammenfassung von \mathcal{R}_R und \mathcal{R}_{E_i} :

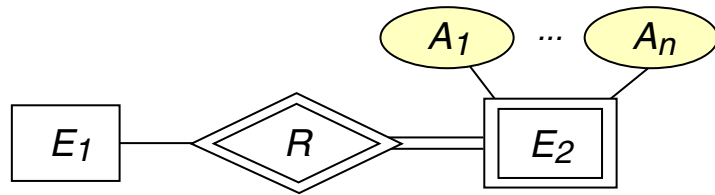


Bemerkungen:

- ❑ Eine $[0, 1]$ - bzw. $[1, 1]$ -Beschränkung qualifiziert den Schlüssel des zugehörigen Entity-Typs E_i offensichtlich als Schlüssel für den Beziehungstyp R , denn jedes Tupel vom Typ R ist höchsten bzw. genau mit einer Instanz von E_i assoziiert.
- ❑ Für $m = 2$ und Vorliegen einer $[0, 1]$ -Beschränkung bei einem Entity-Typ entspricht die Umsetzung der Cross-Reference für binäre 1:n-Beziehungen.
- ❑ Für $m = 2$ und Vorliegen einer $[1, 1]$ -Beschränkung bei einem Entity-Typ entspricht die Umsetzung der Zusammenfassung für binäre 1:n-Beziehungen.
- ❑ Für $m = 2$ und Vorliegen einer $[1, 1]$ -Beschränkung bei *beiden* Entity-Typen ist eine Umsetzung als Merged-Relation wie bei binären 1:1-Beziehungen möglich.

Umsetzung ER-Schema in relationales Schema

Existenzabhängige Entity-Typen [Sonderfall 4]

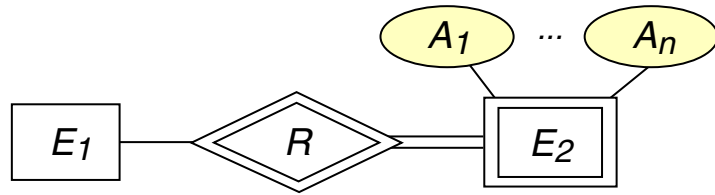


$$\mathcal{R}_{E_2} = \{\underline{ID_1}, A_1, \dots, A_n\}$$

Schlüssel: $\kappa_1 \cup \kappa_2$ bzw. $\{ID_1\} \cup \kappa_2$

Umsetzung ER-Schema in relationales Schema

Existenzabhängige Entity-Typen [Sonderfall 4]



$$\mathcal{R}_{E_2} = \{\underline{ID_1}, A_1, \dots, A_n\}$$

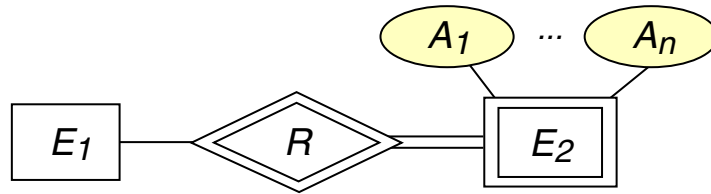
Schlüssel: $\kappa_1 \cup \kappa_2$ bzw. $\{ID_1\} \cup \kappa_2$

Umsetzung:

1. Dem abhängigen Entity-Typ E_2 wird Relationenschema \mathcal{R}_{E_2} zugeordnet.
Die Attribute A_1, \dots, A_n von E_2 werden Attribute von \mathcal{R}_{E_2} .

Umsetzung ER-Schema in relationales Schema

Existenzabhängige Entity-Typen [Sonderfall 4]



$$\mathcal{R}_{E_2} = \{\underline{ID_1}, A_1, \dots, A_n\}$$

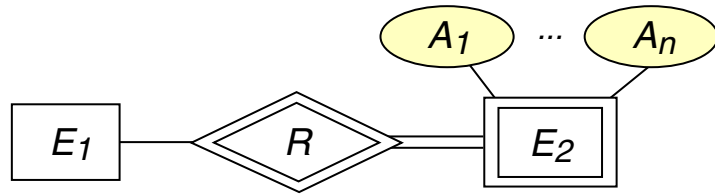
Schlüssel: $\kappa_1 \cup \kappa_2$ bzw. $\{ID_1\} \cup \kappa_2$

Umsetzung:

1. Dem abhängigen Entity-Typ E_2 wird Relationenschema \mathcal{R}_{E_2} zugeordnet. Die Attribute A_1, \dots, A_n von E_2 werden Attribute von \mathcal{R}_{E_2} .
2. Die Attribute in κ_1 (bzw. ID_1) von \mathcal{R}_{E_1} werden Attribute von \mathcal{R}_{E_2} und stellen dort einen entsprechenden Fremdschlüssel dar.

Umsetzung ER-Schema in relationales Schema

Existenzabhängige Entity-Typen [Sonderfall 4]



$$\mathcal{R}_{E_2} = \{\underline{ID_1}, A_1, \dots, A_n\}$$

Schlüssel: $\kappa_1 \cup \kappa_2$ bzw. $\{ID_1\} \cup \kappa_2$

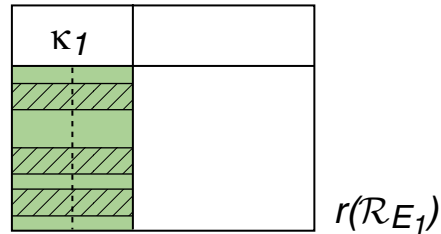
Umsetzung:

1. Dem abhängigen Entity-Typ E_2 wird Relationenschema \mathcal{R}_{E_2} zugeordnet.
Die Attribute A_1, \dots, A_n von E_2 werden Attribute von \mathcal{R}_{E_2} .
2. Die Attribute in κ_1 (bzw. ID_1) von \mathcal{R}_{E_1} werden Attribute von \mathcal{R}_{E_2} und stellen dort einen entsprechenden Fremdschlüssel dar.
3. Die Vereinigung des partiellen Schlüssels κ_2 von E_2 mit dem Primärschlüssel κ_1 (bzw. $\{ID_1\}$) von E_1 bildet den Schlüssel für \mathcal{R}_{E_2} .

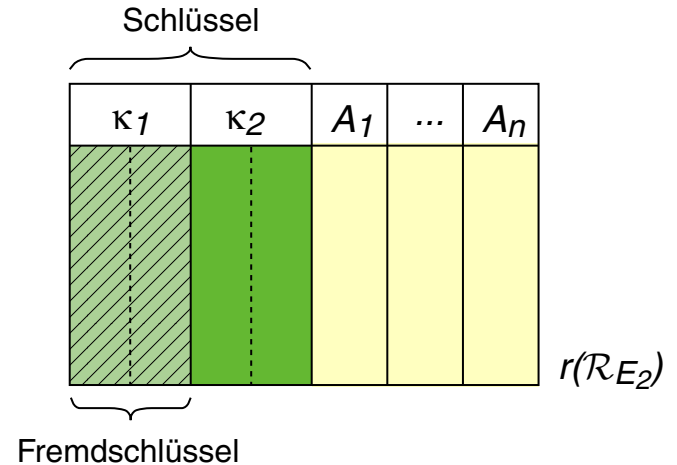
Umsetzung ER-Schema in relationales Schema

Existenzabhängige Entity-Typen

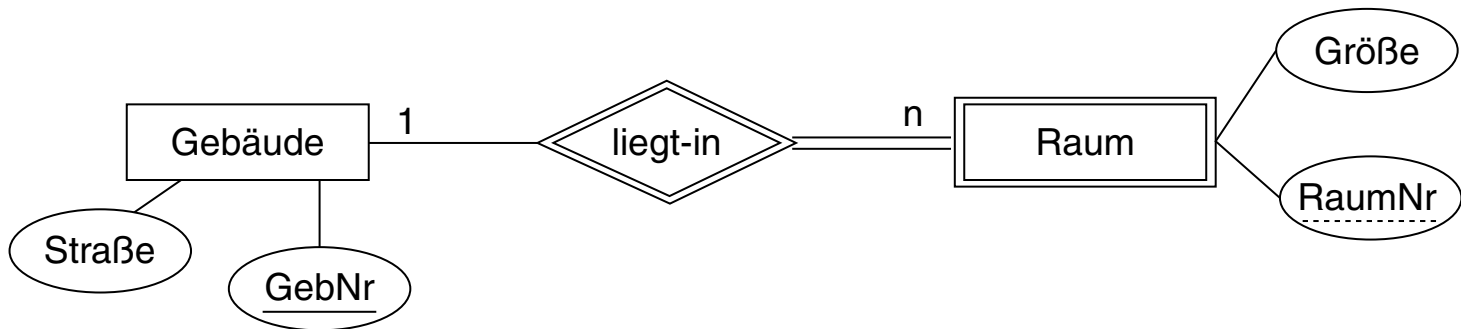
Regulärer Entity-Typ:



Abhängiger Entity-Typ:

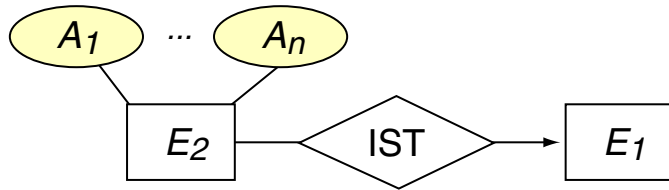


Beispiel:



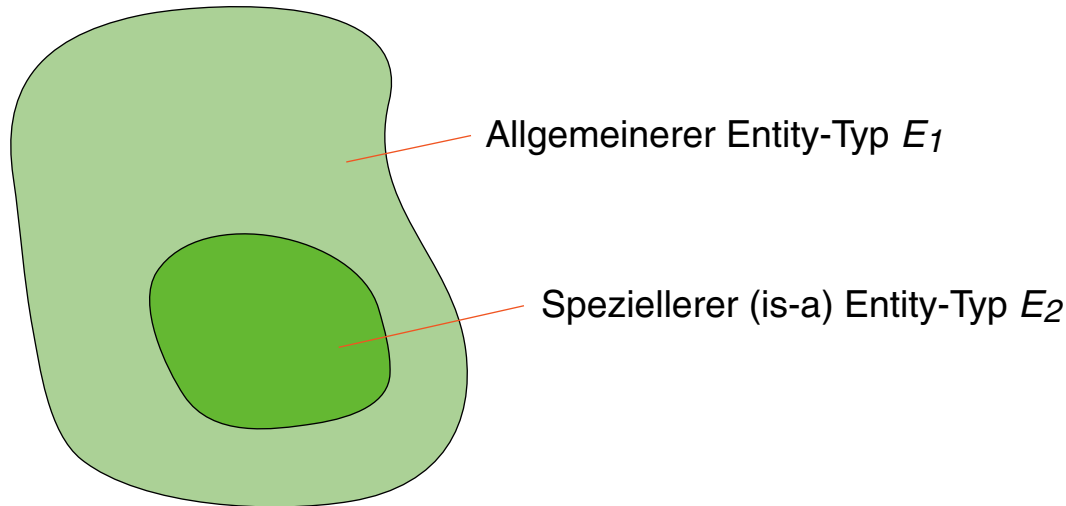
Umsetzung ER-Schema in relationales Schema

IST-Beziehungstypen [Sonderfall 5]



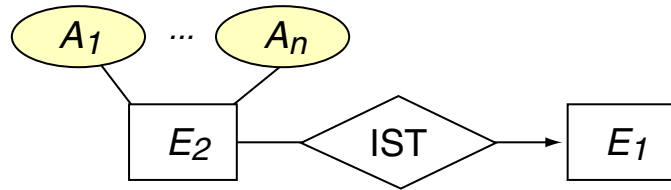
$$\mathcal{R}_{E_2} = \{\underline{ID_1}, A_1, \dots, A_n\}$$

Schlüssel: κ_1 bzw. $\{ID_1\}$



Umsetzung ER-Schema in relationales Schema

IST-Beziehungstypen [Sonderfall 5]



$$\mathcal{R}_{E_2} = \{\underline{ID_1}, A_1, \dots, A_n\}$$

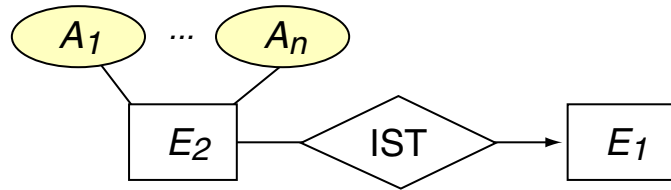
Schlüssel: κ_1 bzw. $\{ID_1\}$

Umsetzung:

1. Dem speziellerem Entity-Typ E_2 wird Relationenschema \mathcal{R}_{E_2} zugeordnet.
Die Attribute A_1, \dots, A_n von E_2 werden Attribute von \mathcal{R}_{E_2} .

Umsetzung ER-Schema in relationales Schema

IST-Beziehungstypen [Sonderfall 5]



$$\mathcal{R}_{E_2} = \{\underline{ID_1}, A_1, \dots, A_n\}$$

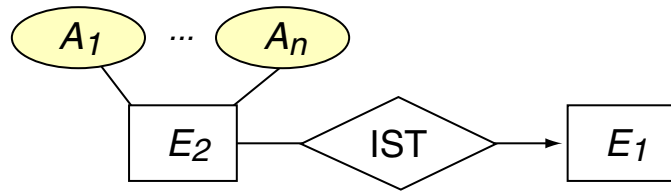
Schlüssel: κ_1 bzw. $\{ID_1\}$

Umsetzung:

1. Dem speziellerem Entity-Typ E_2 wird Relationenschema \mathcal{R}_{E_2} zugeordnet.
Die Attribute A_1, \dots, A_n von E_2 werden Attribute von \mathcal{R}_{E_2} .
2. Die Attribute in κ_1 (bzw. ID_1) von \mathcal{R}_{E_1} werden Attribute von \mathcal{R}_{E_2} und stellen dort eine Art „Fremdschlüssel“ dar.

Umsetzung ER-Schema in relationales Schema

IST-Beziehungstypen [Sonderfall 5]



$$\mathcal{R}_{E_2} = \{\underline{ID_1}, A_1, \dots, A_n\}$$

Schlüssel: κ_1 bzw. $\{ID_1\}$

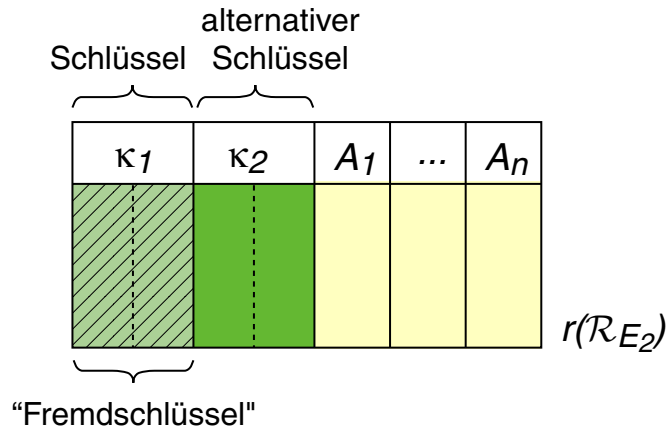
Umsetzung:

1. Dem speziellerem Entity-Typ E_2 wird Relationenschema \mathcal{R}_{E_2} zugeordnet.
Die Attribute A_1, \dots, A_n von E_2 werden Attribute von \mathcal{R}_{E_2} .
2. Die Attribute in κ_1 (bzw. ID_1) von \mathcal{R}_{E_1} werden Attribute von \mathcal{R}_{E_2} und stellen dort eine Art „Fremdschlüssel“ dar.
3. Der Primärschlüssel κ_1 (bzw. $\{ID_1\}$) von E_1 wird Schlüssel für \mathcal{R}_{E_2} .

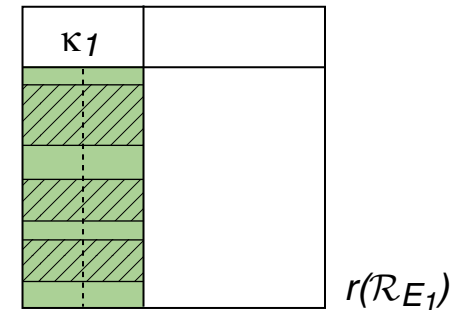
Umsetzung ER-Schema in relationales Schema

IST-Beziehungstypen

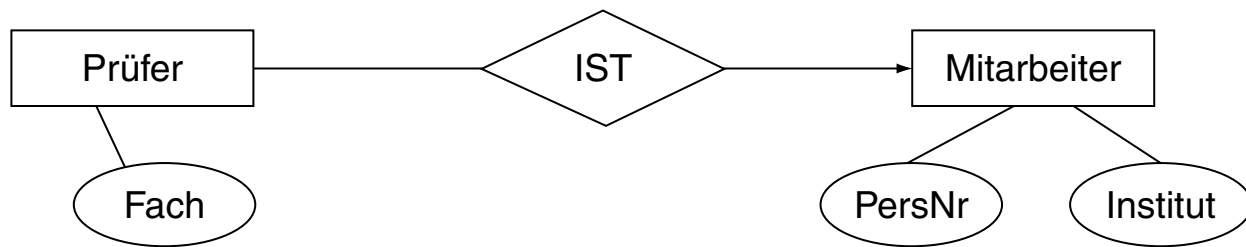
Speziellerer (is-a) Entity-Typ:



Allgemeinerer Entity-Typ:



Beispiel:



Bemerkungen:

- ❑ Es wird die Bezeichnung „Fremdschlüssel“ benutzt, obwohl es sich bei der Spezialisierung nicht um einen Verweis auf einen anderen Entity-Typ handelt, sondern um eine Rollenbeschreibung für ein und denselben Entity-Typ.
- ❑ Ein spezialisierter Entity-Typ E_2 kann bereits einen Schlüssel κ_2 unabhängig von dem Entity-Typ E_1 besitzen, von dem er spezialisiert ist. In diesem Fall hat man für E_2 die Wahl zwischen zwei Schlüsseln, von denen einer als Primärschlüssel festzulegen ist.
- ❑ Bei mehrstufigen IST-Beziehungstypen wird der Primärschlüssel – und damit die Identität – top-down (vom allgemeineren zum spezielleren Entity-Typ) vererbt. Damit ist auch eine Transformationsreihenfolge vorgegeben.

Umsetzung ER-Schema in relationales Schema

Reihenfolge der Regelanwendung [Elmasri/Navathe 2016]

1. Transformation der regulären Entity-Typen.
2. Transformation der abhängigen Entity-Typen.
3. Transformation der 1:1-Beziehungstypen.
4. Transformation der 1:n-Beziehungstypen.
5. Transformation der n:m-Beziehungstypen.
6. Transformation der übrigen Beziehungstypen.
7. Transformation der IST-Beziehungstypen.

Umsetzung ER-Schema in relationales Schema

Zusammenfassung wichtiger Regeln

Konzept im ER-Modell	Konzept im relationalen Modell
Entity-Typ E	Relationenschema \mathcal{R}_E
Attribute A_1, \dots, A_n von E	Attribute A_1, \dots, A_n von \mathcal{R}_E
Primärschlüssel $\kappa \subseteq \{A_1, \dots, A_n\}$ von E	Primärschlüssel κ von \mathcal{R}_E
Beziehungstyp $R(E_1, \dots, E_m; A_1, \dots, A_n)$	Relationenschema \mathcal{R}_R
Attribute A_1, \dots, A_n von R	Attribute A_1, \dots, A_n von \mathcal{R}_R
Attribute in den Primärschlüsseln κ_i der E_i	Attribute von \mathcal{R}_R (als Fremdschlüssel)
1:n-Beziehungstyp zwischen E_1 und E_2	κ_2 wird Primärschlüssel von \mathcal{R}_R
1:1-Beziehungstyp zwischen E_1 und E_2	κ_1 und κ_2 werden jeweils Schlüssel von \mathcal{R}_R , κ_1 oder κ_2 wird Primärschlüssel von \mathcal{R}_R
n:m-Beziehungstyp zwischen E_1 und E_2	$\kappa_1 \cup \kappa_2$ wird Schlüssel von \mathcal{R}_R
E_2 hängt ab von E_1	\mathcal{R}_{E_2} erhält auch alle Attribute in κ_1 , $\kappa_1 \cup \kappa_2$ wird Schlüssel von \mathcal{R}_{E_2}
IST-Beziehungstyp: E_2 IST E_1	\mathcal{R}_{E_2} erhält auch alle Attribute in κ_1 , κ_1 wird Schlüssel von \mathcal{R}_{E_2}

Umsetzung ER-Schema in relationales Schema

Zusammenfassung wichtiger Regeln

Konzept im ER-Modell	Konzept im relationalen Modell
Entity-Typ E	Relationenschema \mathcal{R}_E
Attribute A_1, \dots, A_n von E	Attribute A_1, \dots, A_n von \mathcal{R}_E
Primärschlüssel $\kappa \subseteq \{A_1, \dots, A_n\}$ von E	Primärschlüssel κ von \mathcal{R}_E
Beziehungstyp $R(E_1, \dots, E_m; A_1, \dots, A_n)$	Relationenschema \mathcal{R}_R
Attribute A_1, \dots, A_n von R	Attribute A_1, \dots, A_n von \mathcal{R}_R
Attribute in den Primärschlüsseln κ_i der E_i	Attribute von \mathcal{R}_R (als Fremdschlüssel)
1:n-Beziehungstyp zwischen E_1 und E_2	κ_2 wird Primärschlüssel von \mathcal{R}_R
1:1-Beziehungstyp zwischen E_1 und E_2	κ_1 und κ_2 werden jeweils Schlüssel von \mathcal{R}_R , κ_1 oder κ_2 wird Primärschlüssel von \mathcal{R}_R
n:m-Beziehungstyp zwischen E_1 und E_2	$\kappa_1 \cup \kappa_2$ wird Schlüssel von \mathcal{R}_R
E_2 hängt ab von E_1	\mathcal{R}_{E_2} erhält auch alle Attribute in κ_1 , $\kappa_1 \cup \kappa_2$ wird Schlüssel von \mathcal{R}_{E_2}
IST-Beziehungstyp: E_2 IST E_1	\mathcal{R}_{E_2} erhält auch alle Attribute in κ_1 , κ_1 wird Schlüssel von \mathcal{R}_{E_2}

Umsetzung ER-Schema in relationales Schema

Zusammenfassung wichtiger Regeln

Konzept im ER-Modell	Konzept im relationalen Modell
Entity-Typ E	Relationenschema \mathcal{R}_E
Attribute A_1, \dots, A_n von E	Attribute A_1, \dots, A_n von \mathcal{R}_E
Primärschlüssel $\kappa \subseteq \{A_1, \dots, A_n\}$ von E	Primärschlüssel κ von \mathcal{R}_E
Beziehungstyp $R(E_1, \dots, E_m; A_1, \dots, A_n)$	Relationenschema \mathcal{R}_R
Attribute A_1, \dots, A_n von R	Attribute A_1, \dots, A_n von \mathcal{R}_R
Attribute in den Primärschlüsseln κ_i der E_i	Attribute von \mathcal{R}_R (als Fremdschlüssel)
1:n-Beziehungstyp zwischen E_1 und E_2	κ_2 wird Primärschlüssel von \mathcal{R}_R
1:1-Beziehungstyp zwischen E_1 und E_2	κ_1 und κ_2 werden jeweils Schlüssel von \mathcal{R}_R , κ_1 oder κ_2 wird Primärschlüssel von \mathcal{R}_R
n:m-Beziehungstyp zwischen E_1 und E_2	$\kappa_1 \cup \kappa_2$ wird Schlüssel von \mathcal{R}_R
E_2 hängt ab von E_1	\mathcal{R}_{E_2} erhält auch alle Attribute in κ_1 , $\kappa_1 \cup \kappa_2$ wird Schlüssel von \mathcal{R}_{E_2}
IST-Beziehungstyp: E_2 IST E_1	\mathcal{R}_{E_2} erhält auch alle Attribute in κ_1 , κ_1 wird Schlüssel von \mathcal{R}_{E_2}