

Syntax im Relationenmodell

Konzept	Heuer/Saake 2000	Kemper/Eickler 2004	Vossen 2000	Stein 2004-08
Attribut	A, A_i, B	A, A_i, B	A, A_i, B	A, A_i, B
Domäne von Attribut	$dom(A_i), D_i$	$dom(A_i), D_i$	$dom(A_i)$	$dom(A_i)$
Attributmenge	X, Y	α, β	X	$\alpha, \beta, \{A_{i_1}, \dots, A_{i_k}\}$
Domäne von Attributmenge	$dom(X)$	–	–	–
Relationenschema	R	\mathcal{R}, \mathcal{S}	X	\mathcal{R}
Tupel	$t : R \rightarrow \bigcup_{D_i}$	r, s	$\mu : X \rightarrow dom(X)$	$t : \mathcal{R} \rightarrow \bigcup_{D_i}$
Menge aller Tupel über Attributmenge	–	–	$\text{ Tup}(X)$	–
Teiltupel	$t(X)$	$r.\alpha, s.\kappa$	–	$t(\alpha)$
Relation	$r(R), r$	R, S	r	$r(\mathcal{R}), r$
Menge aller Relationen über Schema	$\mathbf{REL}(R) = \{r \mid r(R)\}$	–	$\mathbf{Rel}(X) = \{r \mid r \subseteq \text{ Tup}(X)\}$	$\{r \mid r(\mathcal{R})\}$
Datenbankschema	$S = \{R_1, \dots, R_p\}$	–	$\mathbf{R} = \{R_1, \dots, R_k\}$	$\mathcal{R} = \{\mathcal{R}_1, \dots, \mathcal{R}_p\}$
Datenbank	$d(S) = \{r_1, \dots, r_p\}, r_i \in \mathbf{REL}(R)$	–	$d(\mathbf{R}) = \{r_1, \dots, r_k\}, r_i \in \mathbf{Rel}(X)$	$d(\mathcal{R}) = \{r_1, \dots, r_p\}$
Menge aller punktweise konsistenten Datenbanken	–	–	$\mathbf{Dat}(\mathbf{R})$	–
Menge aller Datenbanken	–	–	$\mathbf{Sat}(\mathbf{R})$	–
lokale Integritätsbedingung / intrarelationale Abhängigkeit	$b : \{r \mid r(R)\} \rightarrow \{\text{true}, \text{false}\}$	–	$\sigma : \mathbf{Rel}(X) \rightarrow \{0, 1\}$	$b : \{r \mid r(\mathcal{R})\} \rightarrow \{\text{true}, \text{false}\}$
Menge lokaler Integritätsbedingungen	\mathcal{B}	–	Σ_X	–
globale Integritätsbedingung / interrelationale Abhängigkeit	$\gamma : \{d \mid d(S)\} \rightarrow \{\text{true}, \text{false}\}$	–	$\sigma : \mathbf{Dat}(\mathbf{R}) \rightarrow \{0, 1\}$	$b : \{d \mid d(\mathcal{R})\} \rightarrow \{\text{true}, \text{false}\}$
Menge globaler Integritätsbedingungen	Γ	–	$\Sigma_{\mathbf{R}}$	–
erweitertes Relationenschema	$\mathcal{R} = (R, \mathcal{B})$	–	$R = (X, \Sigma_X)$	–
Menge aller Relationen, die lokale Integritätsbedingungen erfüllen	$\mathbf{SAT}_R(\mathcal{B}) = \{r \mid r(\mathcal{R})\}$	–	$\mathbf{Sat}(X, \Sigma_X)$	–
lokal erweitertes Datenbankschema	$S = \{\mathcal{R}_1, \dots, \mathcal{R}_p\}$	–	–	–
global erweitertes Datenbankschema	$\mathcal{S} = (S, \Gamma)$	–	$\mathbf{D} = (\mathbf{R}, \Sigma_{\mathbf{R}})$	–
Menge aller Datenbanken, die globale Integritätsbedingungen erfüllen	$\mathbf{SAT}(\mathcal{S}) = \{d \mid d(\mathcal{S})\}$	–	$\mathbf{Sat}(\Sigma_{\mathbf{R}})$	–
Schlüssel	K	κ	K	κ
Fremdschlüsselbedingung	$X(R_1) \rightarrow Y(R_2)$	–	–	–
Funktionale Abhängigkeit	$X \rightarrow Y$	$\alpha \rightarrow \beta$	$X \rightarrow Y$	$\alpha \rightarrow \beta$
volle funktionale Abhängigkeit	–	$\alpha \xrightarrow{\bullet} \beta$	–	–
Menge funktionaler Abhängigkeiten	F	F	F	F
Hülle funktionaler Abhängigkeiten	F^+	F^+	F^+	F^+

Kapitel DB: IV

- I. Einführung und grundlegende Konzepte von Datenbanken
- II. Datenbankentwurf und Datenbankmodelle
- III. Konzeptueller Datenbankentwurf
- IV. Logischer Datenbankentwurf mit dem relationalen Modell**
 - ❑ Das relationale Modell
 - ❑ Integritätsbedingungen
 - ❑ Umsetzung ER-Schema in relationales Schema
- V. Grundlagen relationaler Anfragesprachen
- VI. Die relationale Datenbanksprache SQL
- VII. Entwurfstheorie relationaler Datenbanken

Das relationale Modell

- Im relationalen Modell werden die Objekttypen der zu modellierenden Anwendungswelt durch *Relationenschemata* beschrieben.
- Ein Relationenschema besteht aus einer Menge von *Attributen*, die Eigenschaften der Objekte eines Objekttyps repräsentieren.
- Attribute haben als *Wertebereiche (Domains)* einfache Mengen, die meist von einem Standard-Datentyp wie `Integer`, `String` oder `Boolean` sind.
- Legt man eine Reihenfolge der Attribute fest, so kann eine *Relation* als Teilmenge des kartesischen Produktes über die Wertebereiche der Attribute eines Relationenschemas aufgefasst werden.
Eine Relation wird auch als *Instanz* eines Relationenschemas bezeichnet.
- Ein *Datenbankschema* besteht aus einer Menge von Relationenschemata.
- Die Menge der zu einem bestimmten Zeitpunkt zu den Relationenschemata eines Datenbankschemas vorhandenen Relationen heißt *Datenbank* oder *Datenbankzustand*.

Das relationale Modell

Interpretation eine Relation als Tabelle:

Buch			
Inv_Nr	Titel	ISBN	Autor
0110	Lesebuch	2-341...	Popper
1201	C++	2-123...	Stroustrup
3309	Längengrad	2-123...	Sobel
4711	Glücksformel	2-679...	Klein
7510	Heuristics	9-212...	Pearl

- } Relationenname
- } Relationenschema mit
 - Spalten = Attribute
 - Zeilen = Tupel
- } Relation = Instanz des Relationenschemas

Bemerkungen:

- ❑ Andere Bezeichnungen für den logischen Datenbankentwurf:
logischer Entwurf, Implementationsentwurf
- ❑ Die Begriffe „relationales Modell“ und „Relationenmodell“ werden synonym verwendet.
- ❑ Die Tabellensicht unterstellt eine (beliebige aber feste) Ordnung der Attribute; das Relationenmodell macht das nicht.

Das relationale Modell

Definition 7 (Relationenschema, Tupel, Relation)

Sei $\mathcal{R} = \{A_1, \dots, A_n\}$ eine endliche Menge von Attributen mit nichtleeren Wertebereichen $dom(A_1), \dots, dom(A_n)$. Dann wird \mathcal{R} als Relationenschema bezeichnet; ein $w \in dom(A_i)$ wird Attributwert für A_i genannt.

- Jedes (einzelne) Tupel über \mathcal{R} ist eine totale Abbildung

$$t : \mathcal{R} \rightarrow \bigcup_{i=1}^n dom(A_i),$$

mit $t(A) \in dom(A)$ für alle $A \in \mathcal{R}$.

- Für eine Teilmenge der Attribute, $\alpha \subseteq \mathcal{R}$, bezeichne $t|_{\alpha}$ die Einschränkung der Abbildung t auf die Menge α , also ein Teiltupel über \mathcal{R} .

In Anlehnung an die Datenbankliteratur schreiben wir für die Abbildung bzw. das Teiltupel $t|_{\alpha}$ vereinfachend auch $t(\alpha)$.

- Eine Relation r über \mathcal{R} , in Zeichen $r(\mathcal{R})$, ist eine endliche Menge von Tupeln über \mathcal{R} .

Bemerkungen:

- Ein Objekt der Anwendungswelt wird hier durch eine Tupelfunktion t repräsentiert. Dieses Objekt lässt sich als eine Zeile in einer Tabelle in der entsprechenden Relation auffassen, wobei die Spalten der Tabelle in beliebiger Reihenfolge angegeben werden dürfen.

$$t = \{(A_1, t(A_1)), (A_2, t(A_2)), \dots, (A_n, t(A_n))\}$$

- Vergleiche hierzu die Mathematik, innerhalb derer ein Tupel eine geordnete Folge fester Länge von Werten bezeichnet:

$$t \in \text{dom}(A_1) \times \text{dom}(A_2) \times \dots \times \text{dom}(A_n)$$

- Im Entity-Relationship-Modell wird ein Attribut A als eine Funktion betrachtet, die jedem Objekt eines Entity-Typs eine Attributausprägung zuordnet; der Datentyp eines Attributs A wird mit T bezeichnet, in Zeichen: $A : T$. [vgl. D:III-19]

Im Relationenmodell wird ein Attribut A lediglich als Bezeichner eines Datentyps verstanden. Somit entspricht der Wertebereich $\text{dom}(A)$ eines Attributes A im Relationenmodell der Menge $\text{dom}(T)$ im Entity-Relationship-Modell.

Das relationale Modell

Beispiel:

- $\mathcal{R} = \text{Pers_Telefon} = \{\text{PANr}, \text{Telefon}\}$
- $\text{dom}(\text{PANr}) = \text{Integer}$
- $\text{dom}(\text{Telefon}) = \text{String}$
- $r(\mathcal{R}) = r(\text{Pers_Telefon}) = \{t_1, t_2, \dots\}$

Pers_Telefon	
PANr	Telefon
4711	038203-12230
4711	0381-498-3427
5588	0391-34677
...	

- $t_i : \{\text{PANr}, \text{Telefon}\} \rightarrow \text{Integer} \cup \text{String}$
- $t_1 = \{(\text{PANr}, 4711), (\text{Telefon}, 038203-12230)\}$
 $t_1(\text{PANr}) = 4711$
 $t_1(\text{Telefon}) = 038203-12230$
- $t_2 = \dots$

Bemerkungen:

- Eine Relation ist als endliche Menge von Abbildungen t_i definiert. Somit können Attributwerte über den Attributnamen eindeutig identifiziert werden.
- Fasst man eine Relation als Teilmenge eines kartesischen Produktes auf, $r \subseteq \text{dom}(\text{PANr}) \times \text{dom}(\text{Telefon})$, so werden Attributwerte durch ihre Position identifiziert. Insbesondere ist $\text{dom}(\text{PANr}) \times \text{dom}(\text{Telefon}) \neq \text{dom}(\text{Telefon}) \times \text{dom}(\text{PANr})$.

Das relationale Modell

Definition 8 (Datenbankschema, Datenbank(zustand), Basisrelation)

- Eine Menge von Relationenschemata $\mathcal{R} = \{\mathcal{R}_1, \dots, \mathcal{R}_p\}$ heißt Datenbankschema.
- Eine Datenbank bzw. ein Datenbankzustand über einem Datenbankschema $\mathcal{R} = \{\mathcal{R}_1, \dots, \mathcal{R}_p\}$ ist eine Menge von Relationen

$$d(\mathcal{R}) = \{r_1, \dots, r_p\}, \quad \text{mit } r_i = r_i(\mathcal{R}_i), \quad i = 1, \dots, p$$

- Eine Relation $r \in d(\mathcal{R})$ heißt Basisrelation.

Bemerkungen:

- ❑ Eine Datenbank d hat zu einem Zeitpunkt i den Zustand $state_i$. [vgl. D:II-24]
- ❑ Der Begriff „Basisrelation“ dient als Abgrenzung zu hieraus abgeleiteten Relationen, die nicht in der Datenbank gespeichert sind.

Das relationale Modell

Beispiel:

- $\mathcal{R} = \{ \text{Personen}, \text{Pers_Telefon} \}$
- $d(\mathcal{R}) = \{ r_1(\text{Personen}), r_2(\text{Pers_Telefon}) \}$

r_1

Personen				
PANr	Vorname	Nachname	PLZ	ORT
4711	Andreas	Heuer	18209	DBR
5588	Gunter	Saake	39106	MD
6834	Michael	Korn	39104	MD
...				

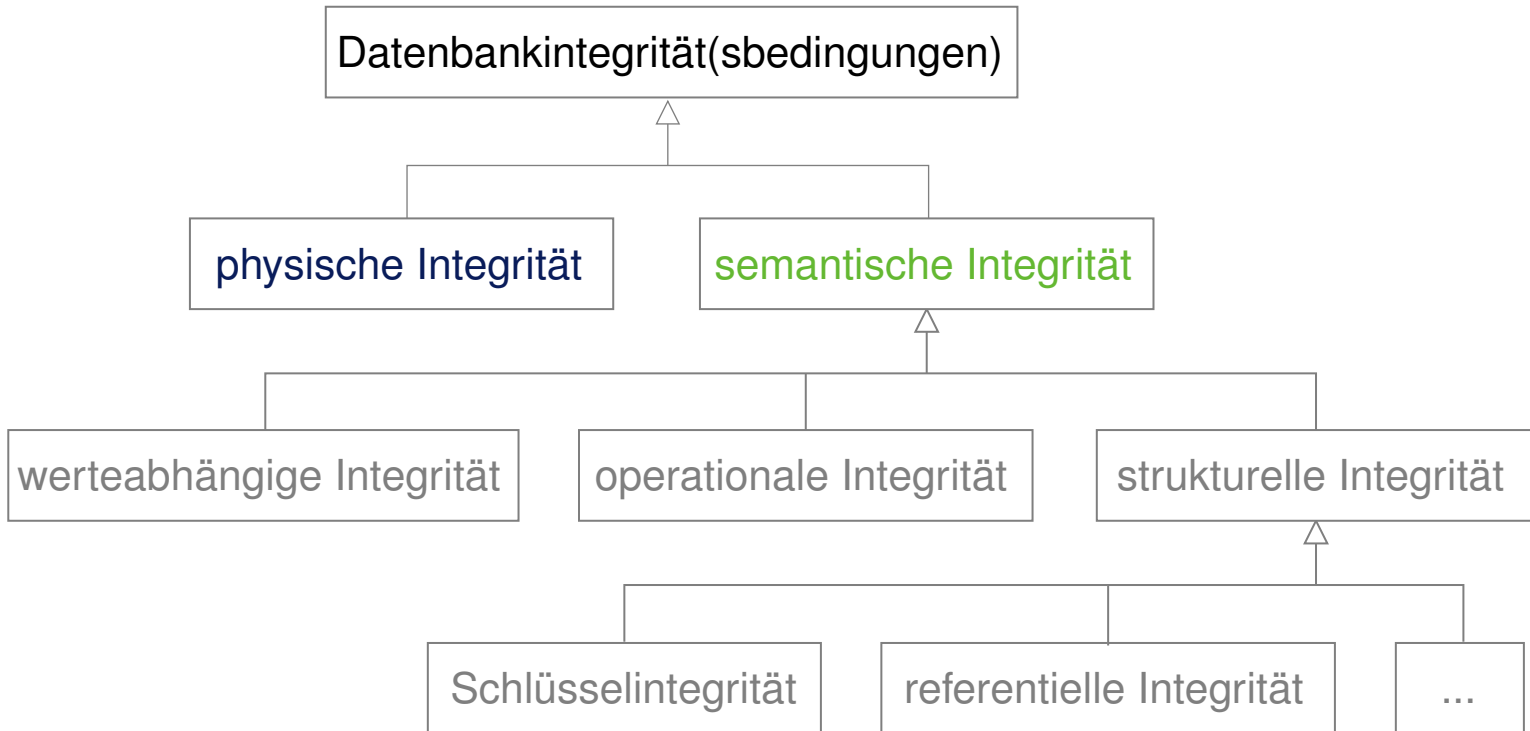
r_2

Pers_Telefon	
PANr	Telefon
4711	038203-12230
4711	0381-498-3427
5588	0391-34677
...	

Integritätsbedingungen

Definition 9 (Integrität)

Mit der Integrität einer Datenbank sei die korrekte und widerspruchsfreie Speicherung von Daten bezeichnet.



[vgl. Weber 2003]

Bemerkungen:

- ❑ Zur Einhaltung bzw. Gewährleistung der Integrität werden Integritätsbedingungen formuliert.
- ❑ Die „Reichweite“ einer Integritätsbedingung sollte möglichst gering gehalten werden.
- ❑ Art und Umfang der realisierten Integritätskonzepte variieren in kommerziellen Datenbankprodukten.
- ❑ Ein flexibler Mechanismus zur Sicherung komplexer Integritätsbedingungen sind sogenannte *Trigger*. Ein Trigger ist eine benutzerdefinierte Funktion, die automatisch aufgerufen wird, sobald eine bestimmte Bedingung erfüllt ist. Ein typischer Einsatz für Trigger sind z. B. Tabellen mit Statistiken, in denen Werte abgeleiteter Attribute bzw. Spalten zu berechnen sind.

Integritätsbedingungen

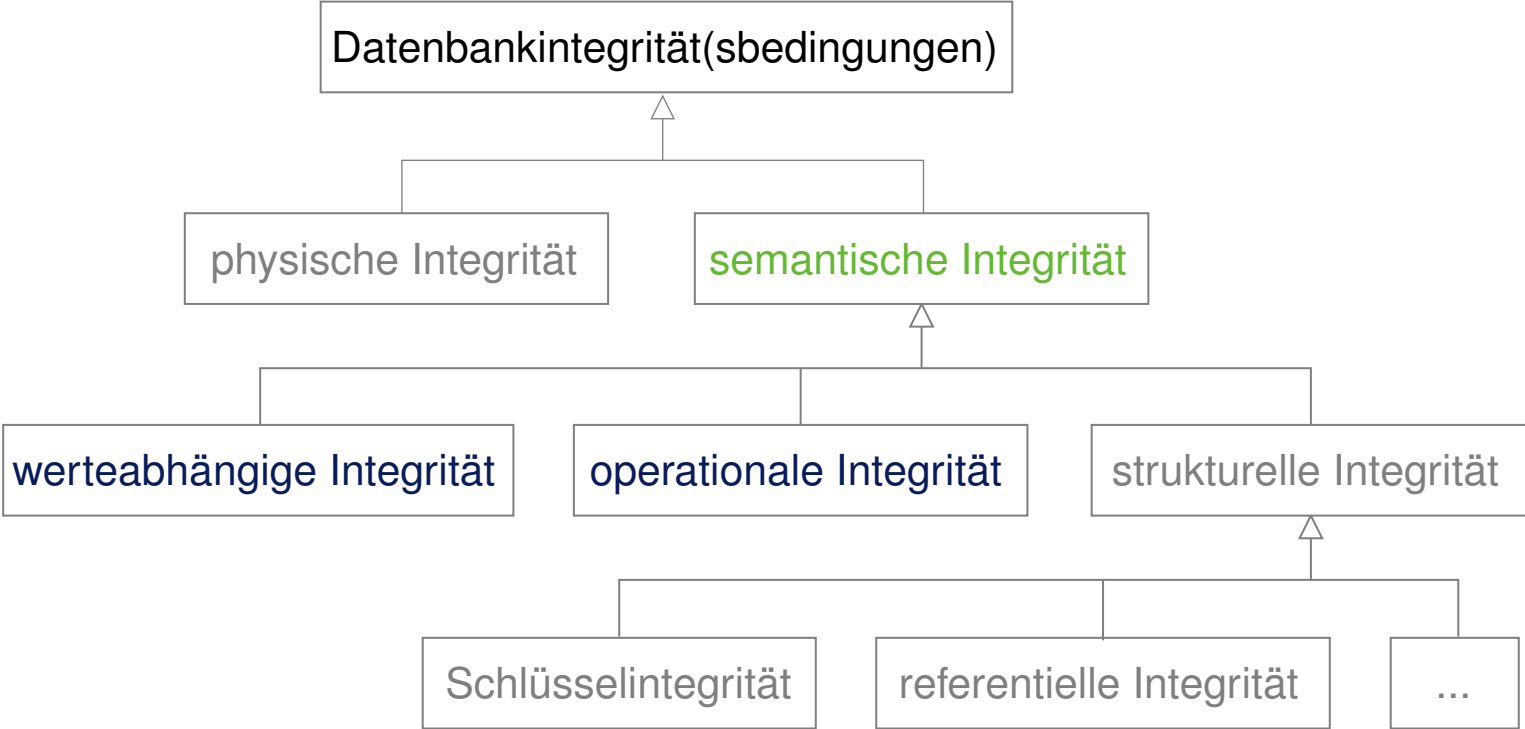
Physische Integrität

- ❑ Behandelt alle Aspekte der physikalischen Speicherung von Daten.
- ❑ Berücksichtigt Hard- und Softwarefehler soweit möglich.
- ❑ Behandelt die Probleme des gleichzeitigen Zugriffs auf Daten.

Semantische Integrität

- ❑ Wird aus den Eigenschaften des zu modellierenden Weltausschnitts abgeleitet und *definiert die zulässigen Zustände der Datenbank*:
Legt fest, welche Werte erlaubt und welche Beziehungen zwischen Datenelementen möglich sind.
- ❑ Das Datenbankmanagementsystem überprüft mit semantischen Integritätsbedingungen, ob eine gewünschte Änderung der Datenbank zulässig ist. Das betrifft die Ebene 2 im DBMS. [vgl. D:I-25/26]
- ❑ Semantische Integritätsbedingungen werden nach dem Inhalt unterschieden, den sie behandeln – in wertabhängige, operationale und strukturelle Integritätsbedingungen.

Integritätsbedingungen



Integritätsbedingungen

Werteabhängige/statische Integrität

Werteabhängige Integrität bezeichnet die Forderung, nicht alle durch Entity-Typen und Beziehungs-Typen definierten Datenbankzustände zuzulassen:

- Werteabhängige Integritätsbedingungen treffen Aussagen über den Zusammenhang von Werten zwischen Datenelementen.
- Sie beschränken die im Anwendungsprogramm möglichen Datenelemente auf Teilmengen.

Integritätsbedingungen

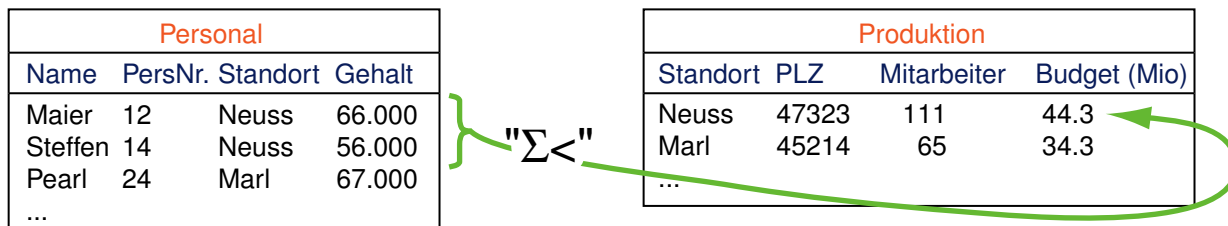
Werteabhängige/statische Integrität

Werteabhängige Integrität bezeichnet die Forderung, nicht alle durch Entity-Typen und Beziehungs-Typen definierten Datenbankzustände zuzulassen:

- ❑ Werteabhängige Integritätsbedingungen treffen Aussagen über den Zusammenhang von Werten zwischen Datenelementen.
- ❑ Sie beschränken die im Anwendungsprogramm möglichen Datenelemente auf Teilmengen.

Beispiele:

- ❑ *Lokal*, innerhalb eines Tupels oder einer Relation: PLZ muss zwischen 1000 und 99999 liegen.
- ❑ *Global*, zwischen Relationen: Die Summe der Gehälter aller Mitarbeiter an einem Standort darf dessen Personal-Budget nicht überschreiten.



Integritätsbedingungen

Operationale/dynamische Integrität

Operationale Integritätsbedingungen gewährleisten korrekte Zustandsübergänge:

- Sie überwachen die Ausführung von Operationen.
- Durch die Angabe von Integritätsbedingungen wird die Ausführung einer Transaktion von der Einhaltung von Vor- und Nachbedingungen abhängig gemacht.

Integritätsbedingungen

Operationale/dynamische Integrität

Operationale Integritätsbedingungen gewährleisten korrekte Zustandsübergänge:

- Sie überwachen die Ausführung von Operationen.
- Durch die Angabe von Integritätsbedingungen wird die Ausführung einer Transaktion von der Einhaltung von Vor- und Nachbedingungen abhängig gemacht.

Beispiele:

- Vom Familienstand „ledig“ ist nur ein Übergang zu verheiratet möglich – und z. B. nicht zu „geschieden“.
- Das Gehalt von Mitarbeitern darf nur steigen.

Zeitpunkt i:

Personal			
Name	PersNr.	Standort	Gehalt
Maier	12	Neuss	66.000
Steffen	14	Neuss	56.000
Pearl	24	Marl	67.000
...			

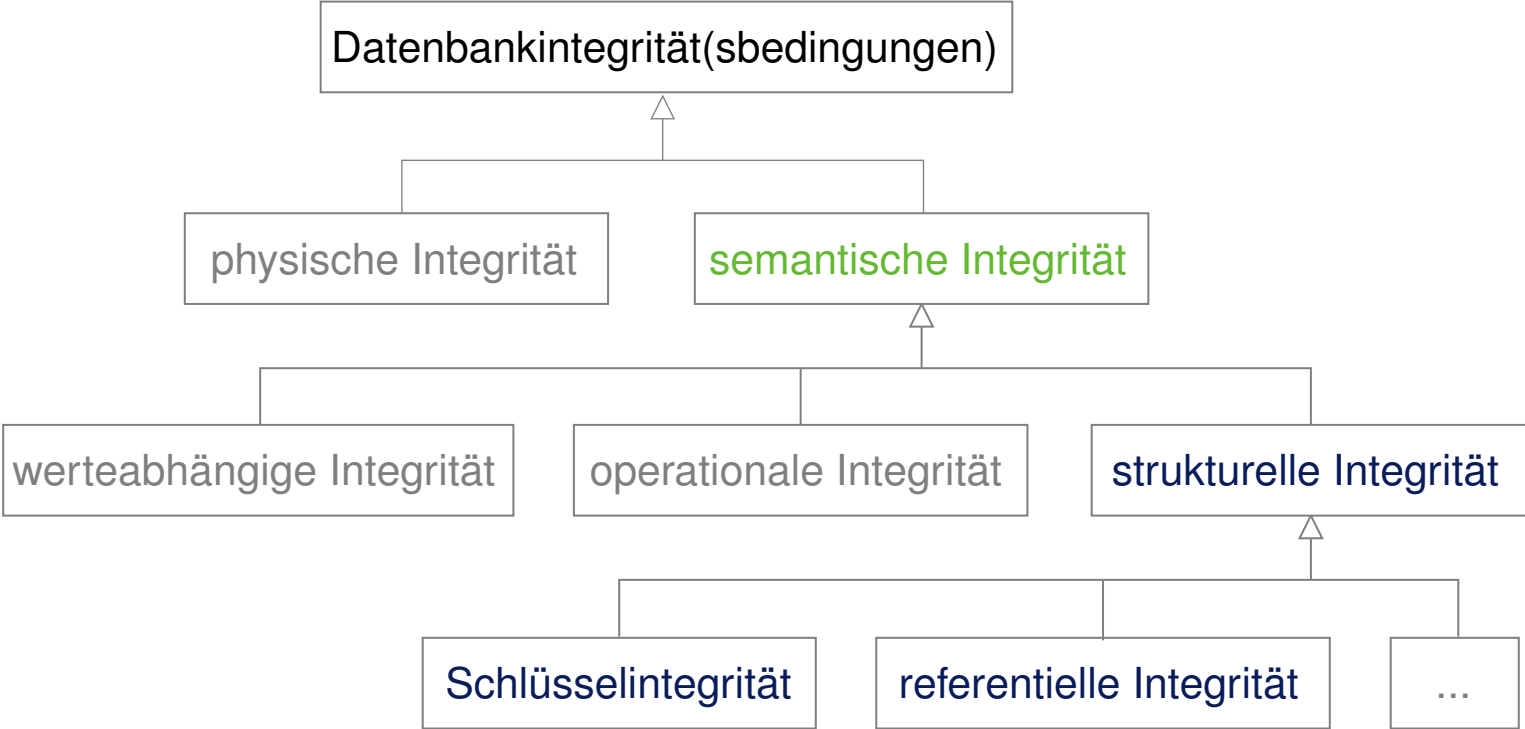
Zeitpunkt i+1:

Personal			
Name	PersNr.	Standort	Gehalt
Maier	12	Neuss	67.000
Steffen	14	Neuss	59.000
Pearl	24	Marl	67.000
...			

"<"



Integritätsbedingungen



Integritätsbedingungen

Schlüssel und Schlüsselintegrität

Definition 10 (Schlüssel, Primärschlüssel, Primattribute)

Eine Attributmenge $\kappa \subseteq \mathcal{R}$ eines Relationenschemas \mathcal{R} heißt identifizierende Attributmenge für eine Relation $r(\mathcal{R})$ falls gilt:

$$\forall t_1, t_2 \in r : t_1 \neq t_2 \Rightarrow \exists A \in \kappa : t_1(A) \neq t_2(A)$$

Integritätsbedingungen

Schlüssel und Schlüsselintegrität

Definition 10 (Schlüssel, Primärschlüssel, Primattribute)

Eine Attributmenge $\kappa \subseteq \mathcal{R}$ eines Relationenschemas \mathcal{R} heißt identifizierende Attributmenge für eine Relation $r(\mathcal{R})$ falls gilt:

$$\forall t_1, t_2 \in r : t_1 \neq t_2 \Rightarrow \exists A \in \kappa : t_1(A) \neq t_2(A)$$

- Ein Schlüssel bzw. ein Schlüsselkandidat ist eine minimale, identifizierende Attributmenge. [vgl. D:III-25, Schlüssel im ER-Modell]
- Ein *Primärschlüssel* ist ein unter mehreren Schlüsselkandidaten ausgezeichneter Schlüssel.
- Die Attribute eines Schlüssels nennt man *Primattribute*.
- Die Schlüsselintegrität für eine Relation $r(\mathcal{R})$ fordert die Existenz eines Schlüssels für $r(\mathcal{R})$.

Integritätsbedingungen

Schlüssel und Schlüsselintegrität (Fortsetzung)

Beispiel:

Personen							
<u>PANr</u>	Vorname	Nachname	PLZ	ORT	Straße	HNr.	Geb.Datum
4711	Andreas	Heuer	18209	DBR	BHS	15	31.10.1958
5588	Gunter	Saake	39106	MD	STS	55	05.10.1960
6834	Michael	Korn	39104	MD	BS	41	24.09.1974
7754	Andreas	Möller	18209	DBR	RS	31	25.02.1976
8832	Tamara	Jagellovsk	38106	BS	GS	12	11.11.1973
9912	Antje	Hellhof	18059	HRO	AES	21	04.0.4.1970
9999	Christa	Loeser	69121	HD	TS	38	10.05.1969

- U. a. sind {Vorname, Nachname} und {PANr} Schlüssel für die Relation „Personen“.
- Der Primärschlüssel wird durch Unterstreichen der entsprechenden Attribute gekennzeichnet.

Integritätsbedingungen

Fremdschlüssel und referentielle Integrität

Definition 11 (Fremdschlüssel)

Seien $r_1(\mathcal{R}_1)$ und $r_2(\mathcal{R}_2)$ zwei Relationen mit den Schemata \mathcal{R}_1 bzw. \mathcal{R}_2 . Sei weiterhin κ Primärschlüssel von \mathcal{R}_1 .

Dann ist $\alpha \subset \mathcal{R}_2$ Fremdschlüssel in r_2 bezüglich κ in r_1 , falls für alle Tupel $t_2 \in r_2$ gilt:

$$\exists t_1 \in r_1 \text{ mit } t_1(\kappa) = t_2(\alpha) \quad \text{bzw.} \quad \{t_2(\alpha) \mid t_2 \in r_2\} \subseteq \{t_1(\kappa) \mid t_1 \in r_1\}$$

Die Erfüllung dieser Eigenschaft heißt referentielle Integrität, Fremdschlüsselintegrität oder auch Fremdschlüsselbedingung.

Integritätsbedingungen

Fremdschlüssel und referentielle Integrität (Fortsetzung)

$$\mathcal{R}_2 = \{ B_1, B_2, B_3, \dots, B_m \}$$

Fremd "schlüssel" α

B_1	B_2	B_3	...	B_m

$r_2(\mathcal{R}_2)$

$$\mathcal{R}_1 = \{ \underline{A_1}, \underline{A_2}, \underline{A_3}, \dots, A_n \}$$

Schlüssel κ

A_1	A_2	A_3	...	A_n

$r_1(\mathcal{R}_1)$

Integritätsbedingungen

Fremdschlüssel und referentielle Integrität (Fortsetzung)

$$\mathcal{R}_2 = \{ B_1, B_2, B_3, \dots, B_m \}$$

Fremd "schlüssel" α

B_1	B_2	B_3	...	B_m

$r_2(\mathcal{R}_2)$

$$\mathcal{R}_1 = \{ \underline{A_1}, \underline{A_2}, \underline{A_3}, \dots, A_n \}$$

Schlüssel κ

A_1	A_2	A_3	...	A_n

$r_1(\mathcal{R}_1)$

$$\{ t_2(\alpha) \mid t_2 \in r_2 \} \subseteq \{ t_1(\kappa) \mid t_1 \in r_1 \}$$

Bemerkungen:

- Jedes Teiltupel $t_2(\alpha)$ aus r_2 liegt als Ausprägung des Schlüssels κ in r_1 vor.
- Ein Fremdschlüssel für ein Relationenschema \mathcal{R}_2 ist eine Attributmengung $\alpha \subset \mathcal{R}_2$, die in einem Relationenschema \mathcal{R}_1 Schlüssel ist $\Rightarrow \alpha \subset \mathcal{R}_1$.
- Der Schlüssel κ von Relationenschema \mathcal{R}_1 wird im Relationenschema \mathcal{R}_2 „fremd“ verwendet. Er stellt dort in der Regel keinen Schlüssel dar.
- Die Attribute von Primär- und Fremdschlüssel haben jeweils dieselbe Bedeutung und, falls möglich, auch dieselbe Bezeichnung. [Heuer/Saake 2000] spricht in diesem Zusammenhang von „kompatiblen Attributlisten“.
- Notation nach [Heuer/Saake 2000] dafür, dass $\alpha \subset \mathcal{R}_2$ Fremdschlüssel bezüglich κ ist:
 $\alpha(\mathcal{R}_2) \rightarrow \kappa(\mathcal{R}_1)$
- Eine schärfere Formulierung der referentiellen Integrität fordert, dass $t_2(\alpha)$ entweder nur Nullwerte oder nur Werte ungleich Null besitzen darf. [vgl. Kemper/Eickler 2004].
- Ohne Überprüfung der referentiellen Integrität kann man leicht einen inkonsistenten Zustand der Datenbank erzeugen.

Integritätsbedingungen

Fremdschlüssel und referentielle Integrität (Fortsetzung)

Beispiel:

Personen							
PANr	Vorname	Nachname	PLZ	ORT	Straße	HNr.	Geb.Datum
4711	Andreas	Heuer	18209	DBR	BHS	15	31.10.1958
5588	Gunter	Saake	39106	MD	STS	55	05.10.1960
6834	Michael	Korn	39104	MD	BS	41	24.09.1974
7754	Andreas	Möller	18209	DBR	RS	31	25.02.1976
8832	Tamara	Jagellovsk	38106	BS	GS	12	11.11.1973
9912	Antje	Hellhof	18059	HRO	AES	21	04.0.4.1970
9999	Christa	Loeser	69121	HD	TS	38	10.05.1969

Pers_Telefon	
PANr	Telefon
4711	038203-12230
4711	0381-498-3401
4711	0381-498-3427
5588	0391-34677
5588	0391-5592-3800
9999	06221-400177

Integritätsbedingungen

Fremdschlüssel und referentielle Integrität (Fortsetzung)

Beispiel (Fortsetzung):

- {PANr} ist Schlüssel für die Relation „Personen“;
{PANr, Telefon} ist Schlüssel für die Relation „Pers_Telefon“.
- {PANr} in der Relation „Pers_Telefon“ ist ein Fremdschlüssel und nimmt Bezug auf {PANr} in der Relation „Personen“.
- Fremdschlüsselbedingung:

$$\forall t_2 \in \text{Pers_Telefon} \text{ gilt: } \exists t_1 \in \text{Personen} \text{ mit } t_1(\text{PANr}) = t_2(\text{PANr})$$

alternativ:

$$\{t_2(\text{PANr}) \mid t_2 \in \text{Pers_Telefon}\} \subseteq \{t_1(\text{PANr}) \mid t_1 \in \text{Personen}\}$$

mit den Werten des Datenbankzustands:

$$\{4711, 5588, 9999\} \subseteq \{4711, 5588, 6834, 7754, 8832, 9912, 9999\}$$

Integritätsbedingungen

Fremdschlüssel und referentielle Integrität (Fortsetzung)

Sei $r_1(\mathcal{R}_1)$ eine Relation mit Primärschlüssel κ und sei $r_2(\mathcal{R}_2)$ eine Relation mit Fremdschlüssel α bezüglich κ in r_1 . Gewährleistung der referentiellen Integrität:

1. Das Einfügen eines Tupels in r_2 verlangt, dass der Fremdschlüssel auf ein existierendes Tupel in r_1 verweist.

Das automatische Anlegen eines Tupels mit einer neuen Fremdschlüsselausprägung in r_1 nennt man *Cascading Insert*.

Integritätsbedingungen

Fremdschlüssel und referentielle Integrität (Fortsetzung)

Sei $r_1(\mathcal{R}_1)$ eine Relation mit Primärschlüssel κ und sei $r_2(\mathcal{R}_2)$ eine Relation mit Fremdschlüssel α bezüglich κ in r_1 . Gewährleistung der referentiellen Integrität:

1. Das Einfügen eines Tupels in r_2 verlangt, dass der Fremdschlüssel auf ein existierendes Tupel in r_1 verweist.

Das automatische Anlegen eines Tupels mit einer neuen Fremdschlüsselausprägung in r_2 nennt man *Cascading Insert*.

2. Das Ändern eines Tupels in r_2 verlangt, dass die neue Fremdschlüsselausprägung auf ein existierendes Tupel in r_1 verweist.

Integritätsbedingungen

Fremdschlüssel und referentielle Integrität (Fortsetzung)

Sei $r_1(\mathcal{R}_1)$ eine Relation mit Primärschlüssel κ und sei $r_2(\mathcal{R}_2)$ eine Relation mit Fremdschlüssel α bezüglich κ in r_1 . Gewährleistung der referentiellen Integrität:

1. Das Einfügen eines Tupels in r_2 verlangt, dass der Fremdschlüssel auf ein existierendes Tupel in r_1 verweist.

Das automatische Anlegen eines Tupels mit einer neuen Fremdschlüsselausprägung in r_1 nennt man *Cascading Insert*.

2. Das Ändern eines Tupels in r_2 verlangt, dass die neue Fremdschlüsselausprägung auf ein existierendes Tupel in r_1 verweist.

3. Das Ändern der Primärschlüsselausprägung für ein Tupel in r_1 verlangt, dass kein Tupel aus r_2 auf dieses verwiesen hat.

Das automatische Nachvollziehen solcher Änderungen nennt man *Cascading Update*.

Integritätsbedingungen

Fremdschlüssel und referentielle Integrität (Fortsetzung)

Sei $r_1(\mathcal{R}_1)$ eine Relation mit Primärschlüssel κ und sei $r_2(\mathcal{R}_2)$ eine Relation mit Fremdschlüssel α bezüglich κ in r_1 . Gewährleistung der referentiellen Integrität:

1. Das Einfügen eines Tupels in r_2 verlangt, dass der Fremdschlüssel auf ein existierendes Tupel in r_1 verweist.

Das automatische Anlegen eines Tupels mit einer neuen Fremdschlüsselausprägung in r_1 nennt man *Cascading Insert*.

2. Das Ändern eines Tupels in r_2 verlangt, dass die neue Fremdschlüsselausprägung auf ein existierendes Tupel in r_1 verweist.

3. Das Ändern der Primärschlüsselausprägung für ein Tupel in r_1 verlangt, dass kein Tupel aus r_2 auf dieses verwiesen hat.

Das automatische Nachvollziehen solcher Änderungen nennt man *Cascading Update*.

4. Das Löschen eines Tupels in r_1 verlangt, dass kein Tupel aus r_2 auf dieses verwiesen hat.

Das Verhindern der Löschung von Tupeln aus r_1 solange noch hierauf bezugnehmende Tupel in r_2 vorhanden sind, nennt man *Prohibited Delete*. Die Weitergabe von Löschungen an r_2 nennt man *Cascading Delete*.

Integritätsbedingungen

Definition 12 (Lokale und globale Integritätsbedingung)

Eine lokale Integritätsbedingung b ist eine Bool'sche Funktion, die auf der Menge der Relationen zu einem Relationenschema \mathcal{R} definiert ist:

$$b : \{r \mid r(\mathcal{R})\} \rightarrow \{\text{true}, \text{false}\}$$

Eine globale Integritätsbedingung b ist eine Bool'sche Funktion, die von der Menge der Datenbanken zu einem Datenbankschema \mathcal{R} definiert ist:

$$b : \{d \mid d(\mathcal{R})\} \rightarrow \{\text{true}, \text{false}\}$$

Bemerkungen:

- ❑ Ein Schlüssel bzw. die Schlüsselintegrität ist eine spezielle lokale Integritätsbedingung.
- ❑ Ein Fremdschlüssel bzw. die referentielle Integrität ist eine spezielle globale Integritätsbedingung.

Das relationale Modell

Vergleich mit Entity-Relationship-Modell

Entity-Relationship-Modell:

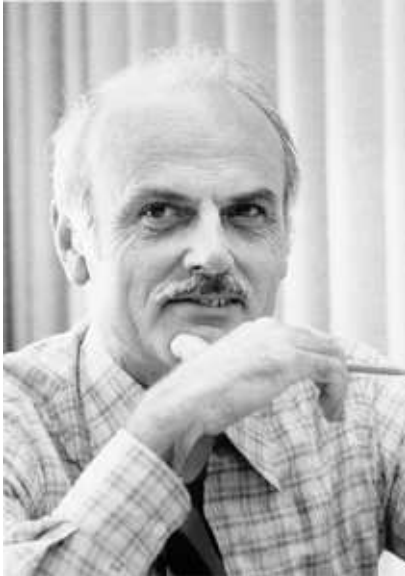
- Attribute haben atomare Werte.
- Beziehungen, die in allen Attributwerten und Entitäten übereinstimmen, sind identisch.
- Die Reihenfolge der Attribute eines Entity- oder Beziehungstyps ist beliebig, aber fest.
- Entitäten haben eine Identität unabhängig von ihren Attributwerten.

Relationales Modell:

- Attribute haben atomare Werte (1NF).
- Zwei Tupel einer Relation besitzen immer unterschiedliche Schlüsselwerte.
- Die Reihenfolge der Attribute eines Relationenschemas ist bedeutungslos.
- Die Reihenfolge der Tupel einer Relation ist bedeutungslos.

Das relationale Modell

Historie



Codd, E.: *A Relational Model of Data for Large Shared Databanks*.
Communications of the ACM, Vol. 13,
No. 6, 377-387, 1970.

[E. F. Codd, *1923, † 2003]

- Das relationale Modell ist von E. F. Codd im Jahre 1970 vorgestellt worden.
- Mittlerweile ist es das am weitesten verbreitete Datenbankmodell.
- 1981 erhielt E. F. Codd den Turing Award für seinen fundamentalen und fortwährenden Beitrag zur Theorie und Praxis von Datenbanksystemen.

Bemerkungen:

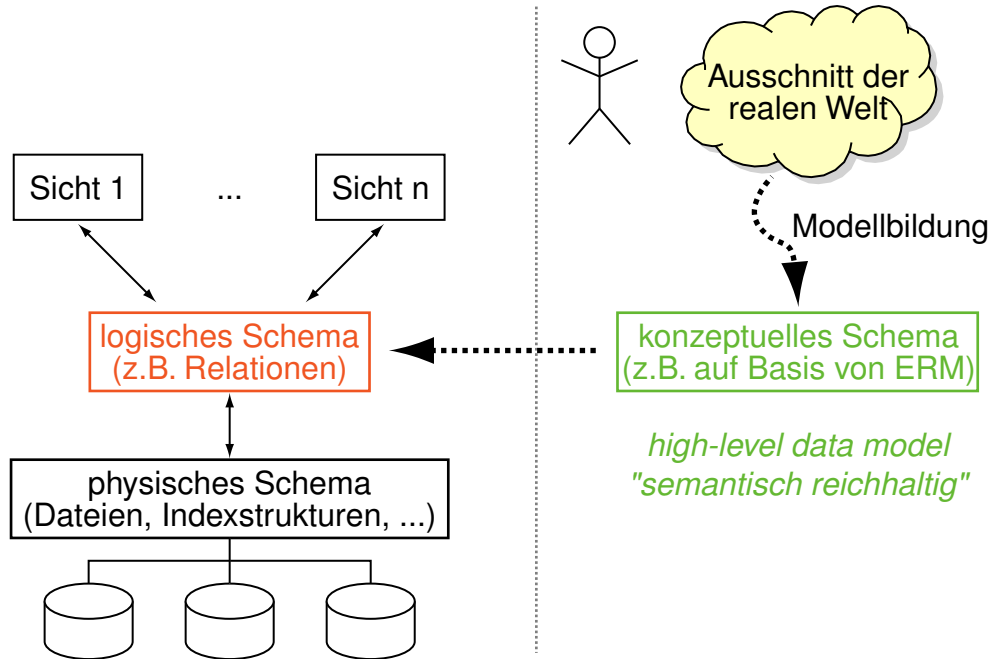
- Dr. E.F. Codd, an IBM researcher, first developed the relational data model in 1970. In 1985, Dr. Codd published a list of 12 rules that concisely define an ideal relational database, which have provided a guideline for the design of all relational database systems ever since.

[www.itworld.com/nl/db_mgr/05072001]

Kapitel DB: IV (Fortsetzung)

- I. Einführung und grundlegende Konzepte von Datenbanken
- II. Datenbankentwurf und Datenbankmodelle
- III. Konzeptueller Datenbankentwurf
- IV. Logischer Datenbankentwurf mit dem relationalen Modell**
 - Das relationale Modell
 - Integritätsbedingungen
 - Umsetzung ER-Schema in relationales Schema
- V. Grundlagen relationaler Anfragesprachen
- VI. Die relationale Datenbanksprache SQL
- VII. Entwurfstheorie relationaler Datenbanken

Umsetzung ER-Schema in relationales Schema



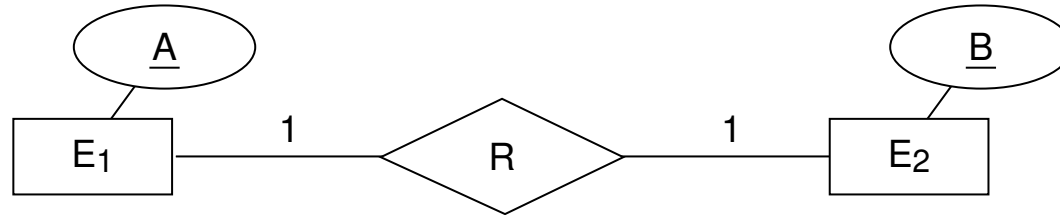
Eine zentrale Forderung bei der Transformation eines Entity-Relationship-Modells in ein relationales Modell ist die *Kapazitätserhaltung*: Alle Instanzen des Entity-Relationship-Modells sind auch Instanzen des relationalen Modells und umgekehrt.

Bemerkungen:

- Eine Transformation ist kapazitätserhaltend, falls es eine bijektive Abbildung zwischen den Instanzen des Entity-Relationship-Modells und den Instanzen des relationalen Modells gibt.

Umsetzung ER-Schema in relationales Schema

Kapazitätserhaltung



Alternative 1. (kapazitätserhöhend)

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

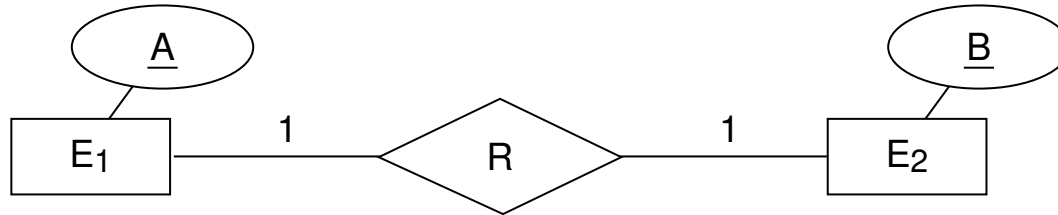
mögliche Relationen:

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

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

Umsetzung ER-Schema in relationales Schema

Kapazitätserhaltung



Alternative 1. (kapazitätserhöhend)

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

mögliche Relationen:

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

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

Alternative 2. (kapazitätserhaltend)

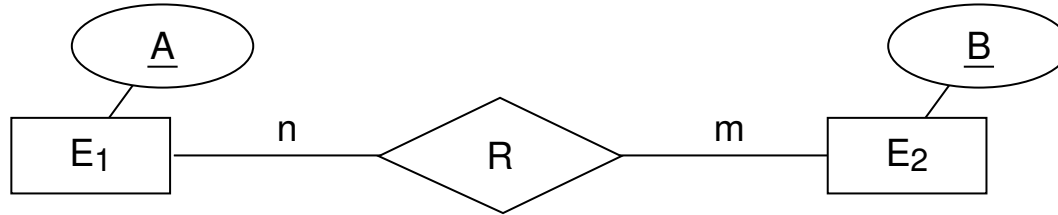
$\mathcal{R} = \{A, B\}$ mit zwei Schlüsseln $\{A\}, \{B\}$

mögliche Relation:

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

Umsetzung ER-Schema in relationales Schema

Kapazitätserhaltung (Fortsetzung)



Alternative 1. (kapazitätsvermindernd)

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

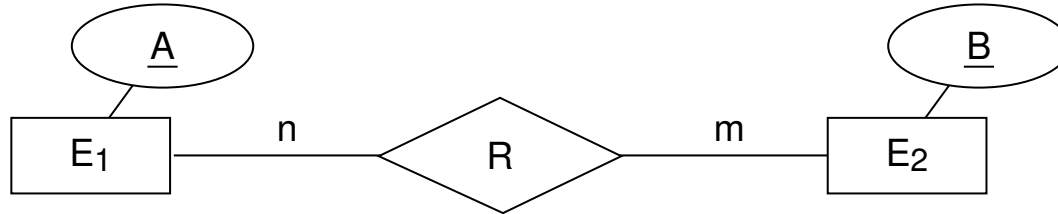
mögliche Relation:

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

⊥

Umsetzung ER-Schema in relationales Schema

Kapazitätserhaltung (Fortsetzung)



Alternative 1. (kapazitätsvermindernd)

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

mögliche Relation:

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

⊥

Alternative 2. (kapazitätserhaltend)

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

mögliche Relationen:

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

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

Umsetzung ER-Schema in relationales Schema

Prinzip

Das Entity-Relationship-Modell besitzt zwei grundlegende Strukturierungskonzepte:

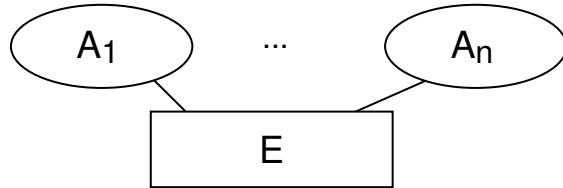
1. Entity-Typen $E(A_1, \dots, A_n)$
2. Beziehungstypen $R(E_1, \dots, E_m; A_1, \dots, A_n)$

Im relationalen Modell werden beide auf das einzige Strukturierungskonzept „Relationenschema“, \mathcal{R} , abgebildet.

Dabei dient das Konzept der Fremdschlüssel zur Nachbildung von Beziehungstypen.

Umsetzung ER-Schema in relationales Schema

Relationale Darstellung von regulären Entity-Typen



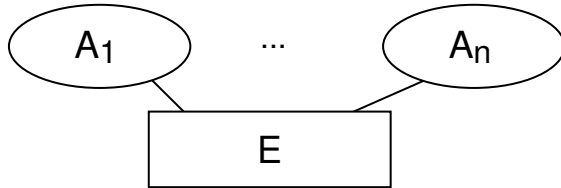
$$\mathcal{R}_E = \{\underline{ID}, A_1, \dots, A_n\}$$

Schlüssel: κ, ID

- 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 .
- Der Schlüssel $\kappa \subseteq \{A_1, \dots, A_n\}$ von E wird Primärschlüssel von \mathcal{R}_E .

Umsetzung ER-Schema in relationales Schema

Relationale Darstellung von regulären Entity-Typen



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

Schlüssel: κ, ID

- 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 .
- Der Schlüssel $\kappa \subseteq \{A_1, \dots, A_n\}$ 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 Schlüssel κ ist dann ein weiterer Schlüssel im Relationenschema \mathcal{R}_E .

- Der Primärschlüssel wird durch Unterstreichen gekennzeichnet.

Bemerkungen:

- Die Bezeichnung „regulärer Entity-Typ“ dient als Abgrenzung zu
 - abhängigen bzw. schwachen Entity-Typen sowie zu den
 - spezialisierten Entity-Typen, die in einer IST-Beziehung stehen.

Umsetzung ER-Schema in relationales Schema

Umsetzung von Beziehungstypen

Zwei Strategien:

1. Kanonische Umwandlung der Beziehungstypen und anschließende Verfeinerung bzw. Zusammenfassung von Relationenschemata.
[Heuer/Saake 2000, Kemper/Eickler 2004]
2. Direkte Abbildung der Beziehungstypen auf ein adäquates Schema.
[Elmasri/Navathe 2004]

Umsetzung ER-Schema in relationales Schema

Umsetzung von Beziehungstypen

Zwei Strategien:

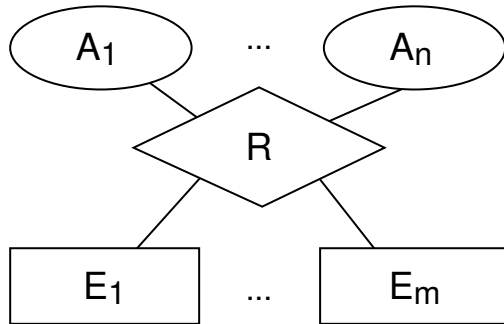
1. Kanonische Umwandlung der Beziehungstypen und anschließende Verfeinerung bzw. Zusammenfassung von Relationenschemata.
[Heuer/Saake 2000, Kemper/Eickler 2004]
2. Direkte Abbildung der Beziehungstypen auf ein adäquates Schema.
[Elmasri/Navathe 2004]

Besondere Behandlung folgender Fälle:

- ❑ m:n-Beziehung (Formalismus I für Kardinalitäten)
- ❑ 1:n-Beziehung (Formalismus I für Kardinalitäten)
- ❑ 1:1-Beziehung (Formalismus I für Kardinalitäten)
- ❑ [0,1] und [1,1]-Beschränkung (Formalismus II für Kardinalitäten)
- ❑ existenzabhängige (schwache) Entity-Typen
- ❑ IST-Beziehungstypen
- ❑ rekursive Beziehungstypen

Umsetzung ER-Schema in relationales Schema

Relationale Darstellung von regulären Beziehungstypen



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

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

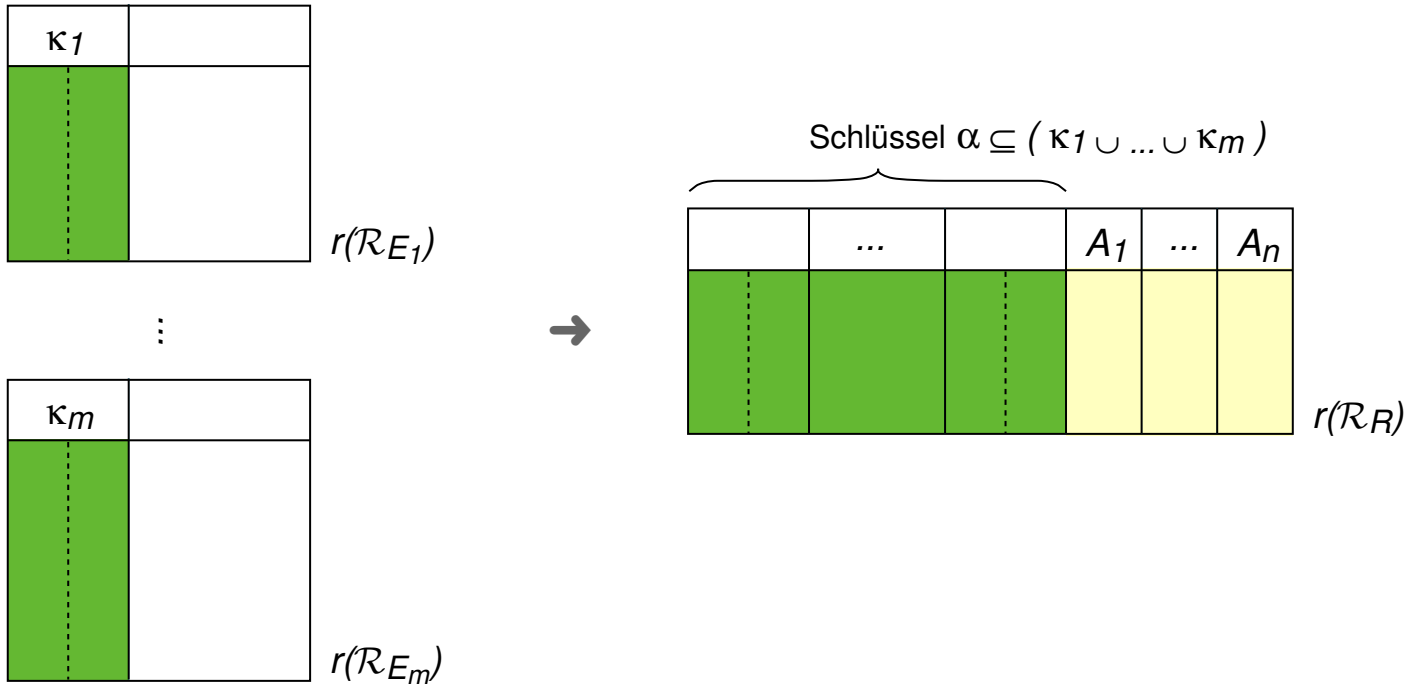
$$\alpha \subseteq \{ID_1, \dots, ID_m\}$$

- Dem Beziehungstyp R wird Relationenschema \mathcal{R}_R zugeordnet.
- Die Attribute A_1, \dots, A_n von R werden Attribute von \mathcal{R}_R .
- Die Attribute in den κ_i (bzw. die ID_i) von \mathcal{R}_{E_i} werden Attribute von \mathcal{R}_R .
- Der Schlüssel von \mathcal{R}_R ist eine **Teilmenge** der Vereinigungsmenge der κ_i (bzw. der Menge aller ID_i).
- Insbesondere sind die $\kappa_i \subset \mathcal{R}_R$ (bzw. die $\{ID_i\} \subset \mathcal{R}_R$) Fremdschlüssel in \mathcal{R}_R bzgl. κ_i (bzw. $\{ID_i\}$) in \mathcal{R}_{E_i} .

Umsetzung ER-Schema in relationales Schema

Relationale Darstellung von regulären Beziehungstypen (Fortsetzung)

Cross-Reference:



Bemerkungen:

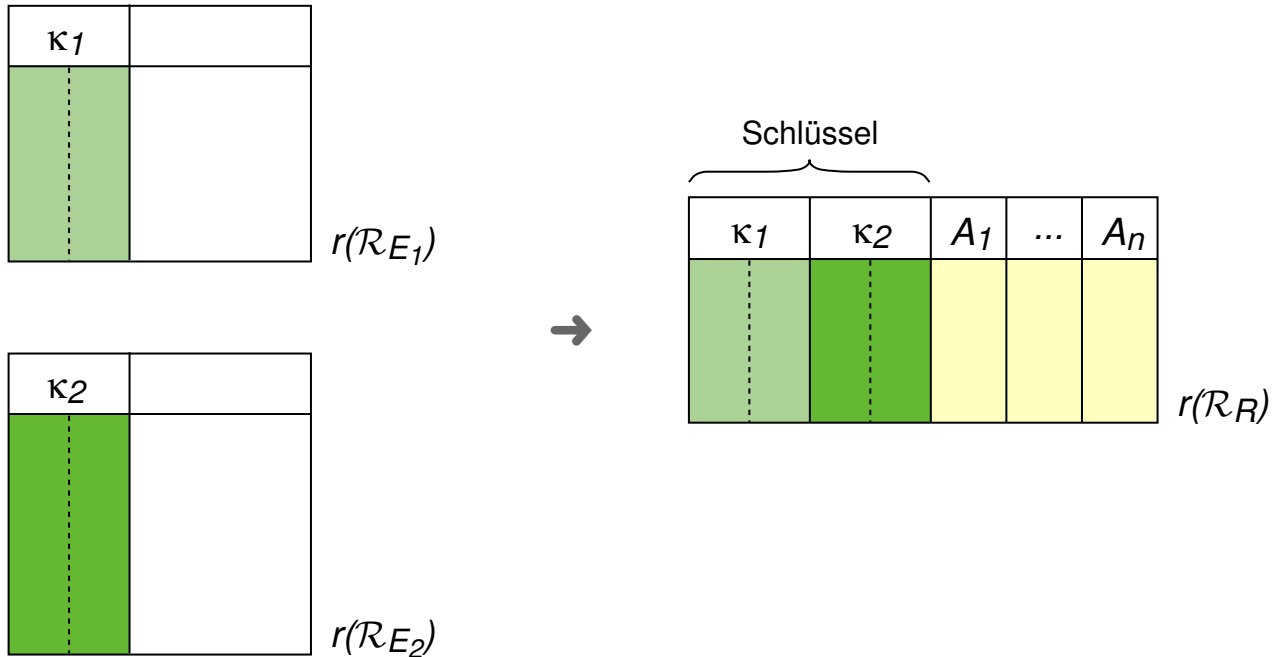
- Die Bezeichnung „regulärer Beziehungstyp“ dient als Abgrenzung zu
 - Beziehungstypen zu abhängigen bzw. schwachen Entity-Typen sowie zu
 - IST-Beziehungstypen.
- Dieses Vorgehen wird in [Elmasri/Navathe 2004] als *Cross-Reference-Ansatz* bezeichnet.
- Eine wichtige Frage ist, welche Teilmenge aus der Vereinigungsmenge der κ_i als Schlüssel ausgewählt wird. Man kann diese Frage nicht in der Allgemeinheit beantworten. Vergleiche hierzu die möglichen funktionalen Beziehungen, die zum Beispiel von einer 1:1:1-Relation impliziert sein können: falls keine funktionale Beziehung gegeben ist, so bildet nur die Menge aller drei Attribute einen Schlüssel; gibt es einen funktionalen Zusammenhang, so bildet die Menge der beiden Attribute des Urbildbereiches einen Schlüssel.

Umsetzung ER-Schema in relationales Schema

Relationale Darstellung von m:n-Beziehungstypen

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

Cross-Reference:



Umsetzung ER-Schema in relationales Schema

Relationale Darstellung von 1:n-Beziehungstypen

Als Verfeinerung des Cross-Reference-Ansatzes 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:

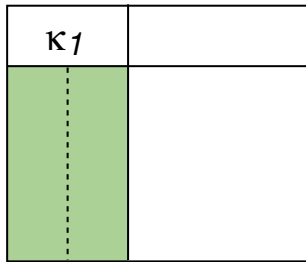
- Die Attribute des Primärschlüssels in \mathcal{R}_{E_1} werden Attribute in \mathcal{R}_{E_2} und stellen dort einen entsprechenden Fremdschlüssel dar.
- Die Attribute des 1:n-Beziehungstyps werden Attribute in \mathcal{R}_{E_2} .
- Der Primärschlüssel des n-Entity-Typs wird Schlüssel im zusammengefassten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

Umsetzung ER-Schema in relationales Schema

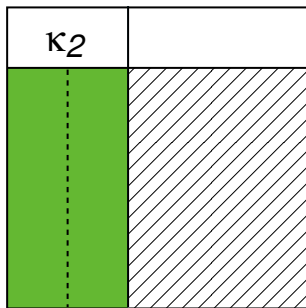
Relationale Darstellung von 1:n-Beziehungstypen (Fortsetzung)

Funktionaler Zusammenhang $E_2 \rightarrow E_1$:

1-Entity-Typ

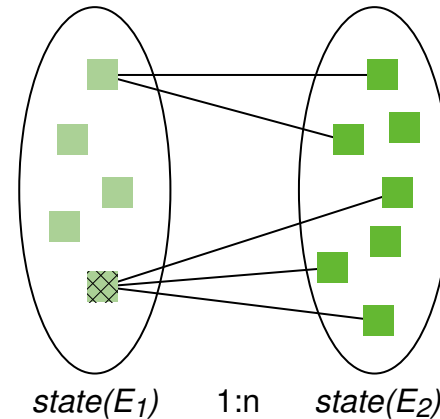


$r(\mathcal{R}_{E_1})$



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

n-Entity-Typ



Umsetzung ER-Schema in relationales Schema

Relationale Darstellung von 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

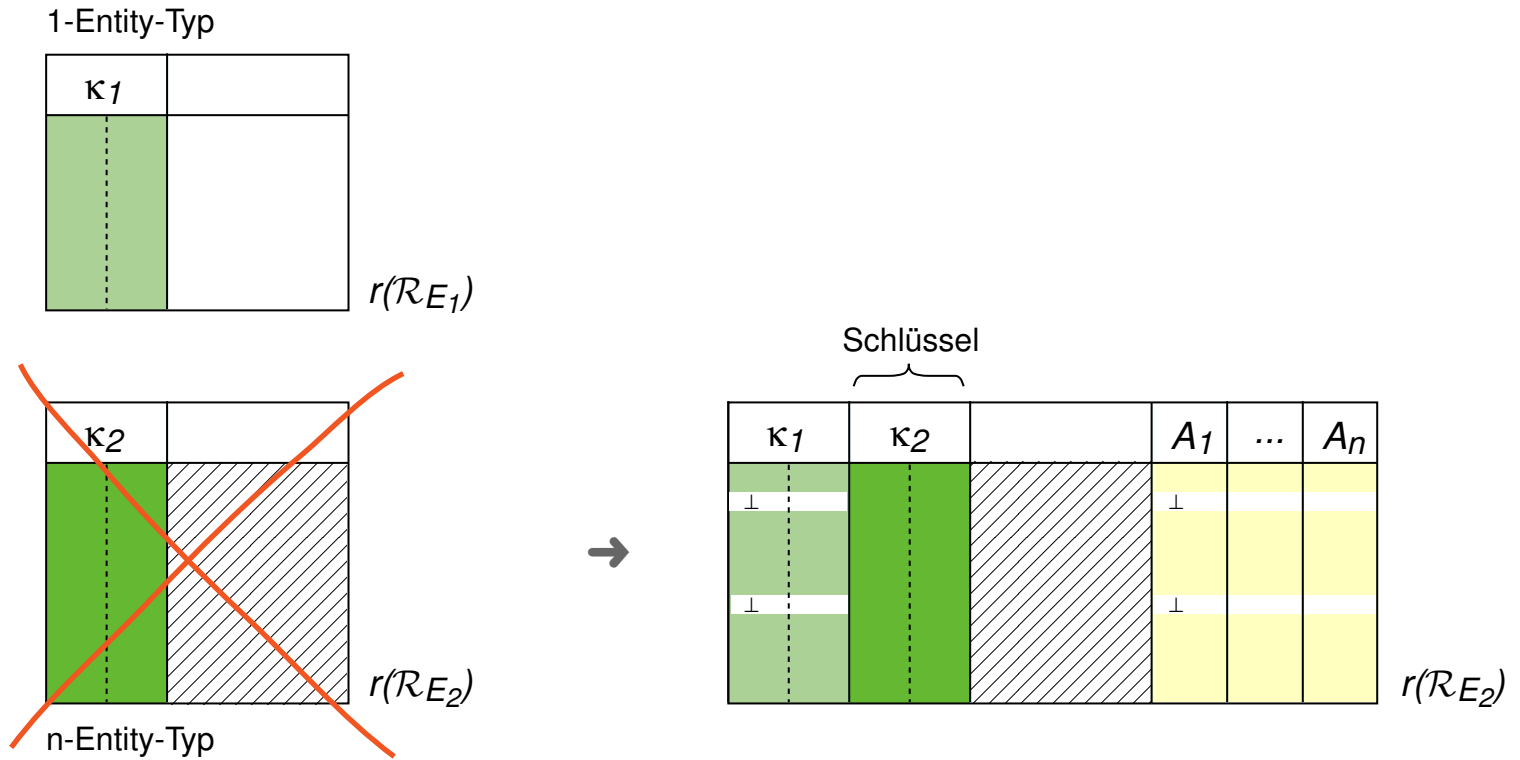
κ_1	κ_2	A_1	...	A_n

$r(\mathcal{R}_R)$

Umsetzung ER-Schema in relationales Schema

Relationale Darstellung von 1:n-Beziehungstypen (Fortsetzung)

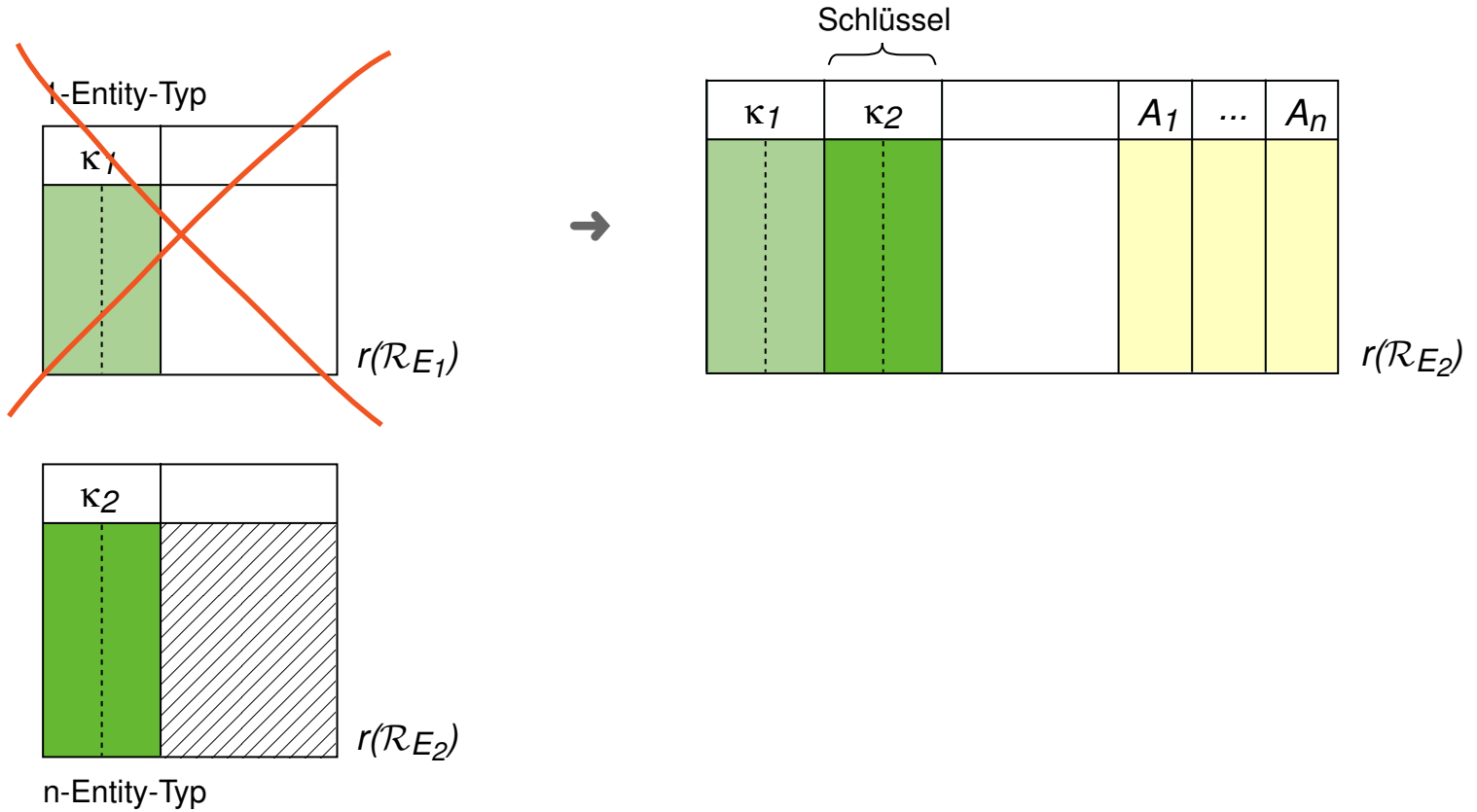
Erlaubte Zusammenfassung von \mathcal{R}_R und \mathcal{R}_{E_2} :



Umsetzung ER-Schema in relationales Schema

Relationale Darstellung von 1:n-Beziehungstypen (Fortsetzung)

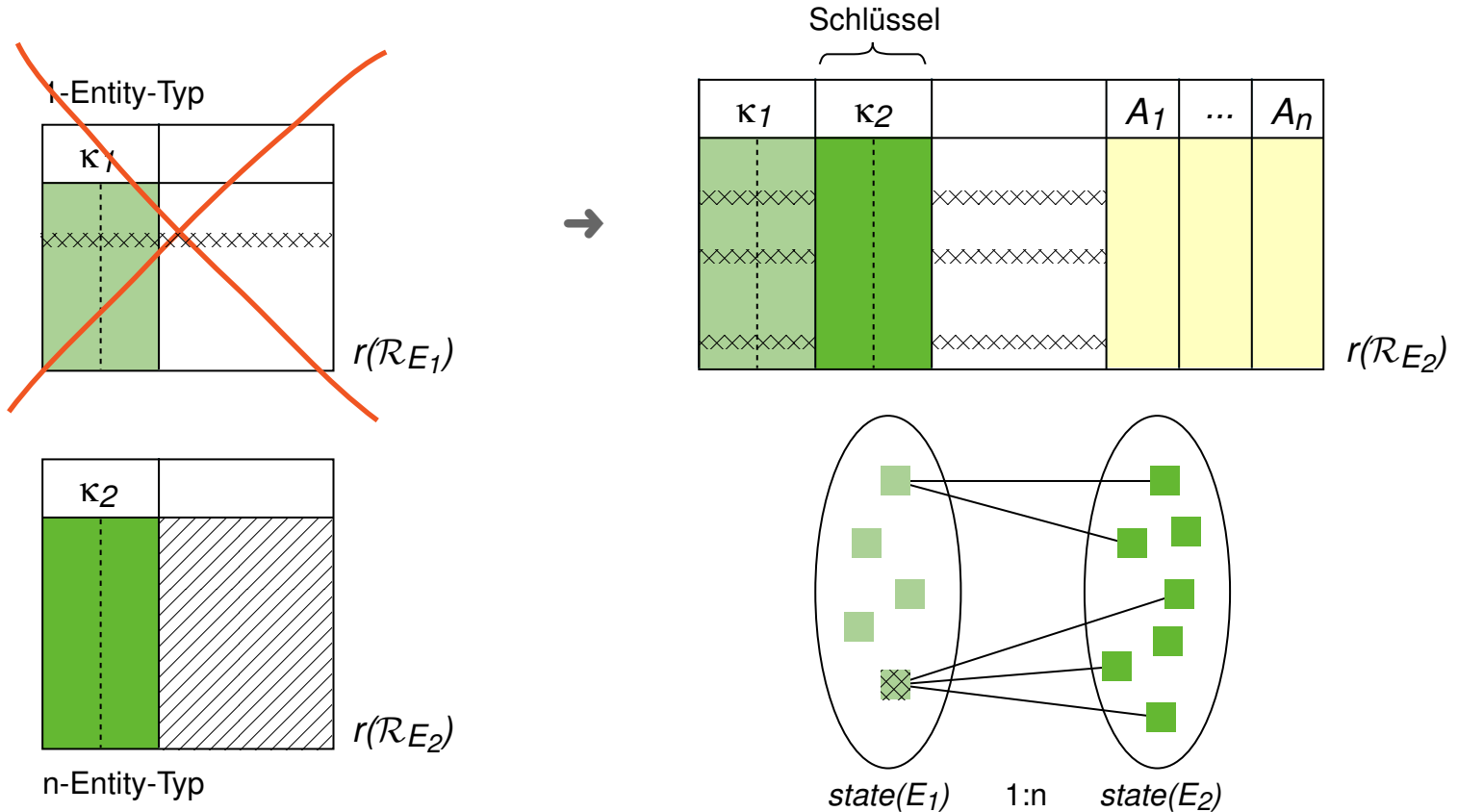
Unerlaubte Zusammenfassung von \mathcal{R}_R und \mathcal{R}_{E_1} :



Umsetzung ER-Schema in relationales Schema

Relationale Darstellung von 1:n-Beziehungstypen (Fortsetzung)

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



Bemerkungen:

- In diesem Zusammenhang gibt [Kemper/Eickler 2004] folgende Regel als Hilfe bei der Zusammenfassung von Relationen an: „Nur Relationen mit gleichem Schlüssel zusammenfassen.“

Umsetzung ER-Schema in relationales Schema

Relationale Darstellung von 1:1-Beziehungstypen

Als Verfeinerung des Cross-Reference-Ansatzes kann man bei 1:1-Beziehungstypen das Relationenschema \mathcal{R}_R mit einem der Relationenschemata der beteiligten Beziehungstypen zusammenfassen.

- (a) 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.
 - Die Attribute des 1:1-Beziehungstyps werden Attribute in \mathcal{R}_{E_2} (\mathcal{R}_{E_1}).
 - Der Primärschlüssel von \mathcal{R}_{E_2} (\mathcal{R}_{E_1}) wird Schlüssel im zusammengefassten Relationenschema.

Umsetzung ER-Schema in relationales Schema

Relationale Darstellung von 1:1-Beziehungstypen

Als Verfeinerung des Cross-Reference-Ansatzes kann man bei 1:1-Beziehungstypen das Relationenschema \mathcal{R}_R mit einem der Relationenschemata der beteiligten Beziehungstypen zusammenfassen.

- (a) 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.
 - Die Attribute des 1:1-Beziehungstyps werden Attribute in \mathcal{R}_{E_2} (\mathcal{R}_{E_1}).
 - Der Primärschlüssel von \mathcal{R}_{E_2} (\mathcal{R}_{E_1}) wird Schlüssel im zusammengefassten Relationenschema.

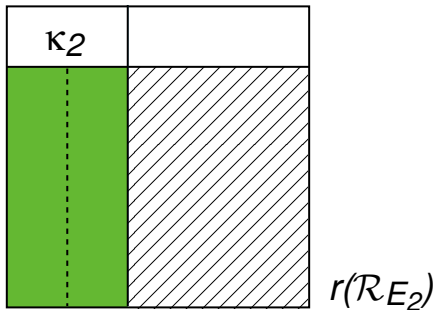
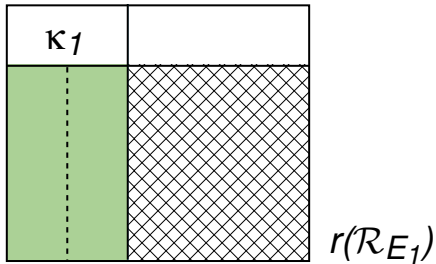
- (b) Ist die Teilnahme von beiden Entity-Typen am Beziehungstyp total – lässt sich also eine bijektive Abbildung zwischen E_1 und E_2 angeben – so können \mathcal{R}_{E_1} und \mathcal{R}_{E_2} auch in einem einzigen Relationenschema zusammengefasst werden.
 - Die Primärschlüssel **beider** Entity-Typen sind Schlüssel im zusammengefassten Relationenschema; von ihnen wird **einer** als Primärschlüssel gewählt. Stichwort: Kapazitätserhaltung

Umsetzung ER-Schema in relationales Schema

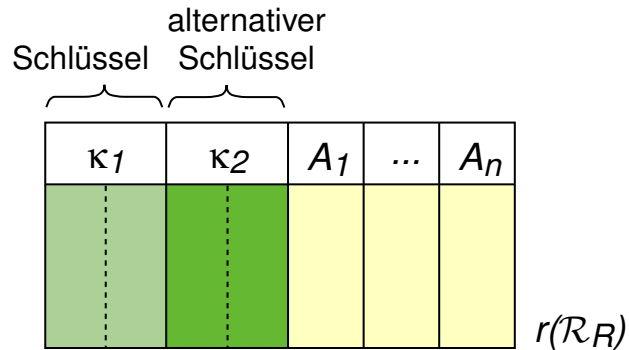
Relationale Darstellung von 1:1-Beziehungstypen (Fortsetzung)

Cross-Reference:

1-Entity-Typ



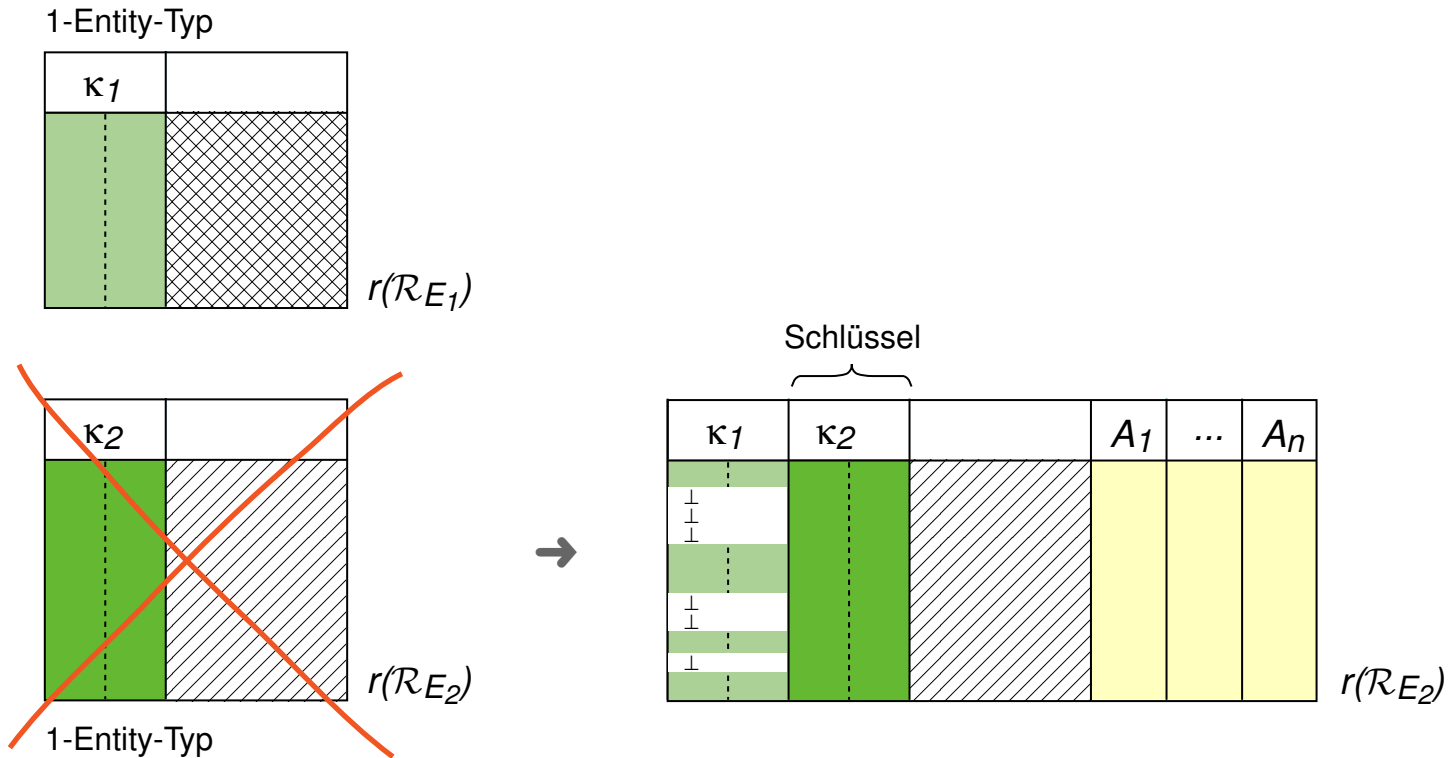
1-Entity-Typ



Umsetzung ER-Schema in relationales Schema

Relationale Darstellung von 1:1-Beziehungstypen (Fortsetzung)

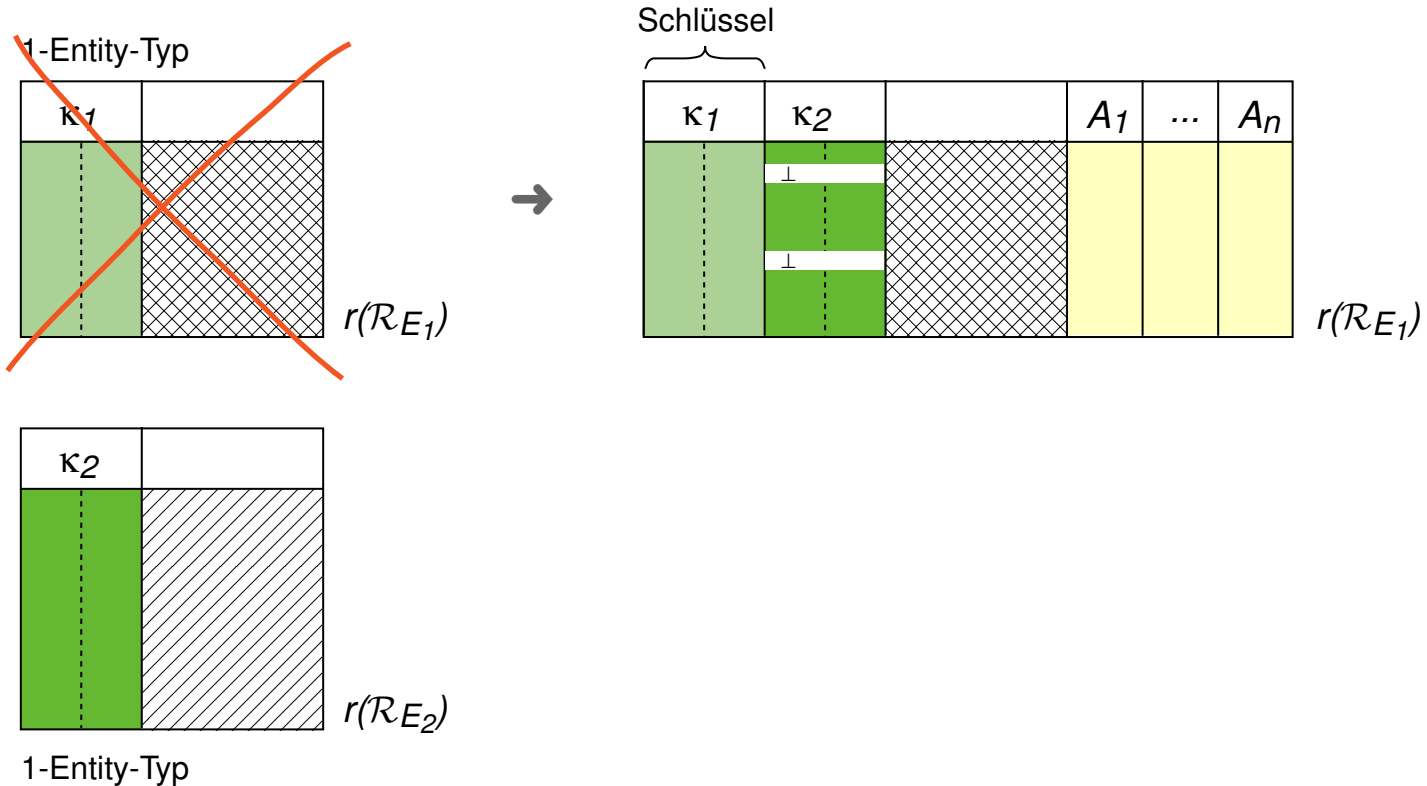
zu (a) Zusammenfassung von \mathcal{R}_R und \mathcal{R}_{E_2} :



Umsetzung ER-Schema in relationales Schema

Relationale Darstellung von 1:1-Beziehungstypen (Fortsetzung)

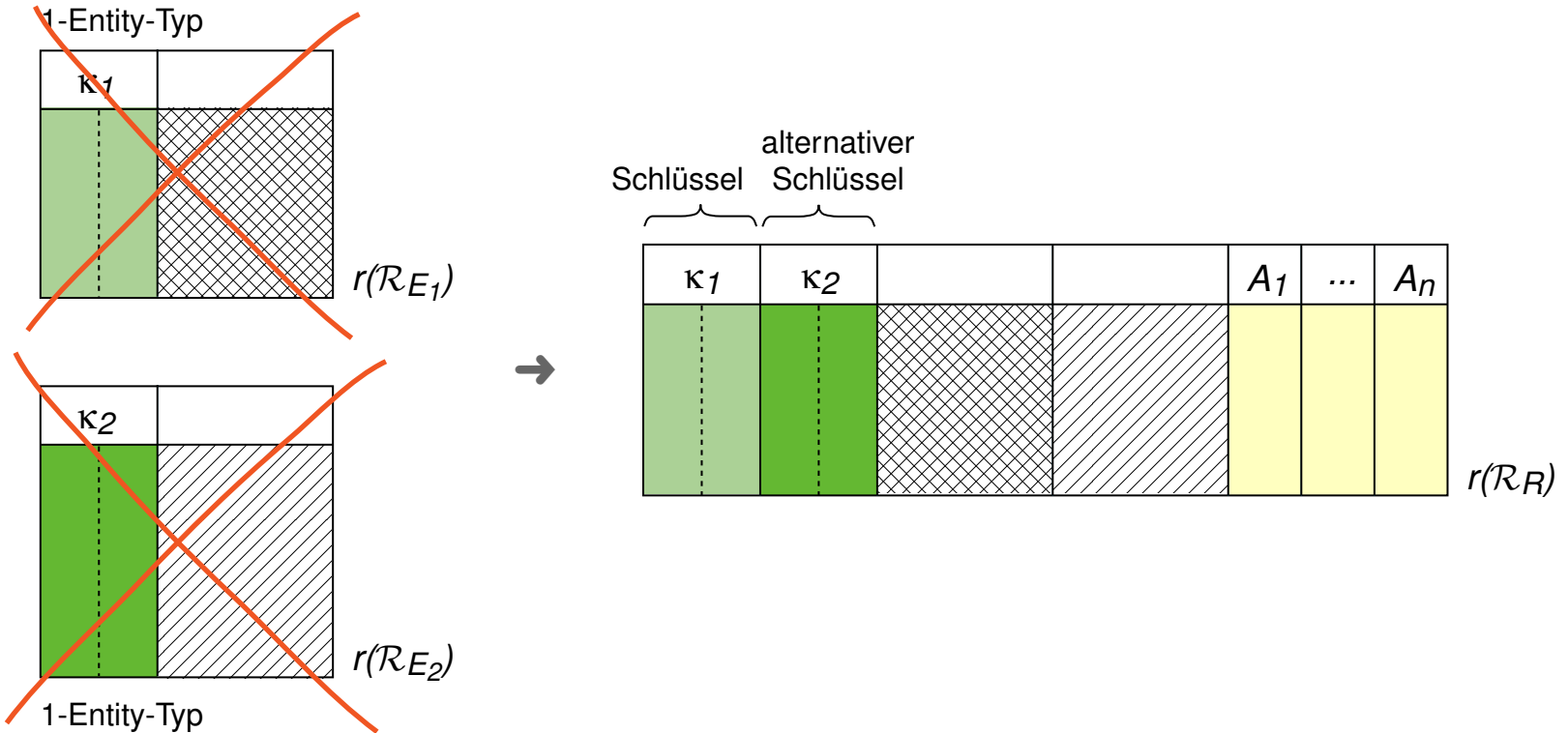
zu (a) Zusammenfassung von \mathcal{R}_R und \mathcal{R}_{E_1} :



Umsetzung ER-Schema in relationales Schema

Relationale Darstellung von 1:1-Beziehungstypen (Fortsetzung)

zu (b) Merged-Relation:



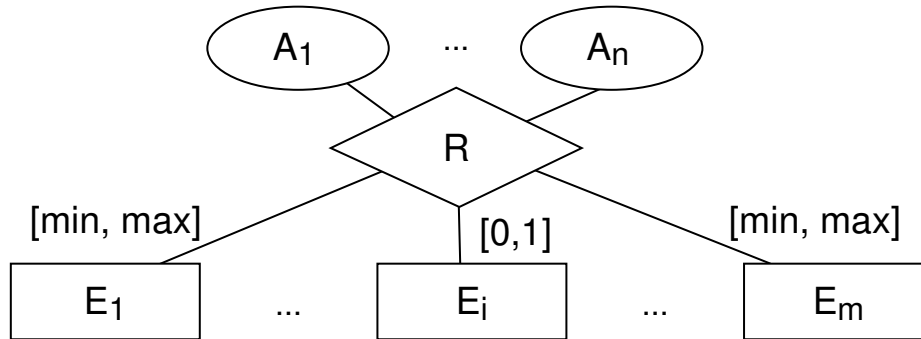
Bemerkungen:

- Nehmen nur wenige Instanzen der beteiligten Entity-Typen an der Beziehung teil, sollte auf eine Zusammenfassung verzichtet werden.

- zu (a) Ob man den Primärschlüssel von \mathcal{R}_{E_1} in \mathcal{R}_{E_2} integriert – oder umgekehrt, sollte man davon abhängig machen, wieviele Nullwerte in den Fremdschlüsselattributen entstehen. Ist z. B. die Teilnahme von E_1 an der 1:1-Beziehung total, ist es sinnvoll, den Primärschlüssel von \mathcal{R}_{E_2} in \mathcal{R}_{E_1} zu integrieren.
- zu (b) Dieses Vorgehen wird in [Elmasri/Navathe 2004] als *Merged-Relation-Ansatz* bezeichnet.

Umsetzung ER-Schema in relationales Schema

Relationale Darstellung von Beziehungstypen mit $[min, max]$ -Beschränkung



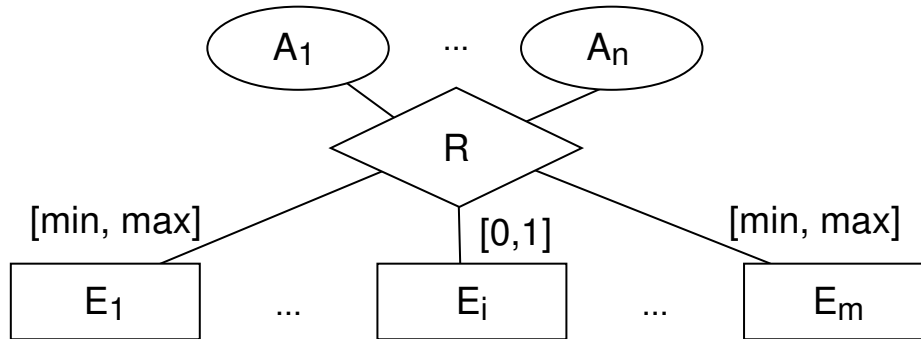
(a) m -äre Beziehungstypen mit $[0, 1]$ -Beschränkung für Entity-Typ E_i :

$$R(E_1[min_1, max_1], \dots, E_i[0, 1], \dots, E_m[min_m, max_m])$$

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

Umsetzung ER-Schema in relationales Schema

Relationale Darstellung von Beziehungstypen mit $[min, max]$ -Beschränkung



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

□ 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 :
 $R(E_1[min_1, max_1], \dots, E_i[1, 1], \dots, E_n[min_m, max_m])$

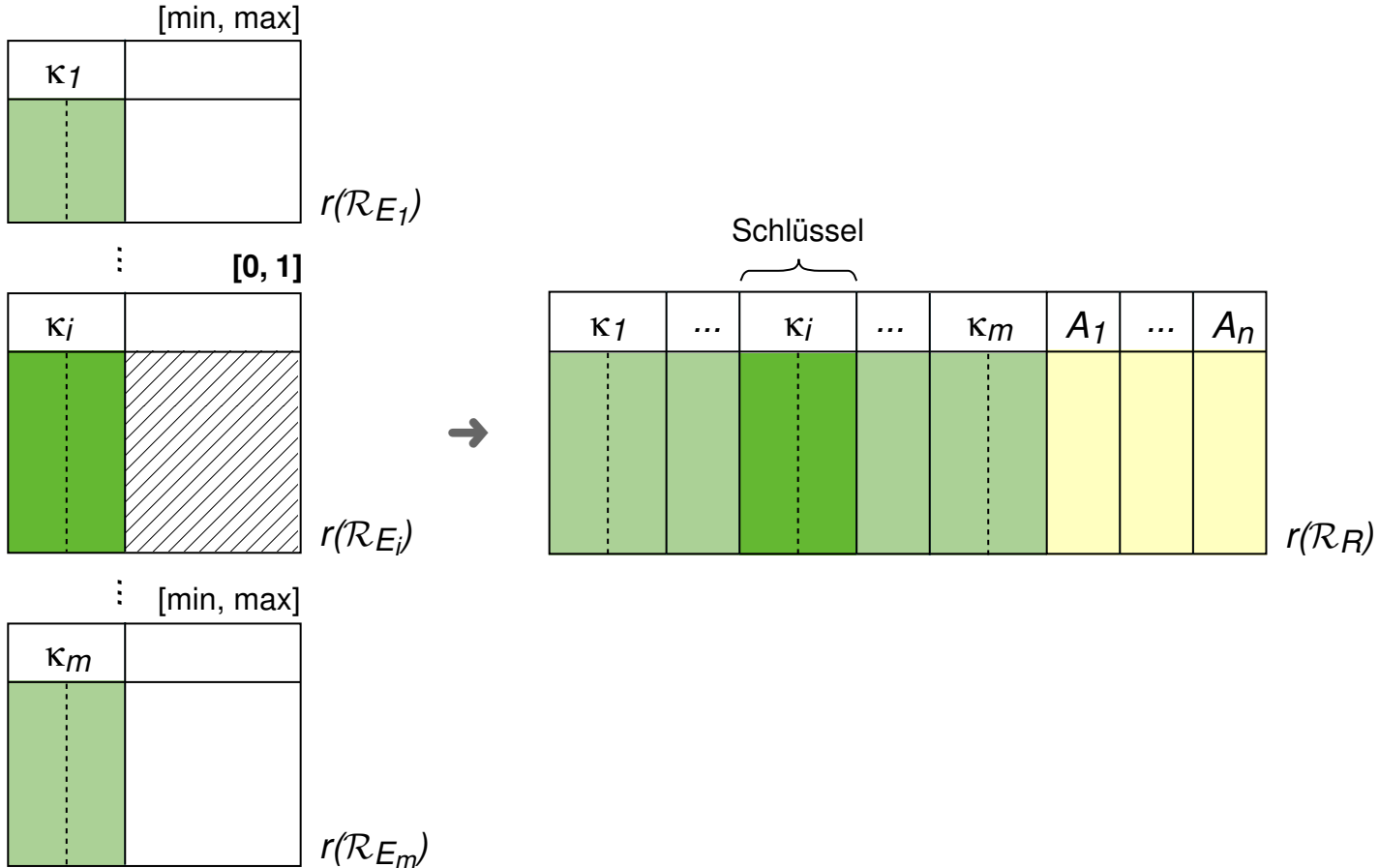
□ Der Primärschlüssel von \mathcal{R}_{E_i} wird ein Schlüssel von \mathcal{R}_R .

□ Die Relationenschemata \mathcal{R}_R und \mathcal{R}_{E_i} können zusammengefasst werden. In diesem Fall werden alle Schlüssel von \mathcal{R}_{E_i} auch Schlüssel von \mathcal{R}_R .

Umsetzung ER-Schema in relationales Schema

Relationale Darstellung von Beziehungstypen mit $[min, max]$ -Beschränkung

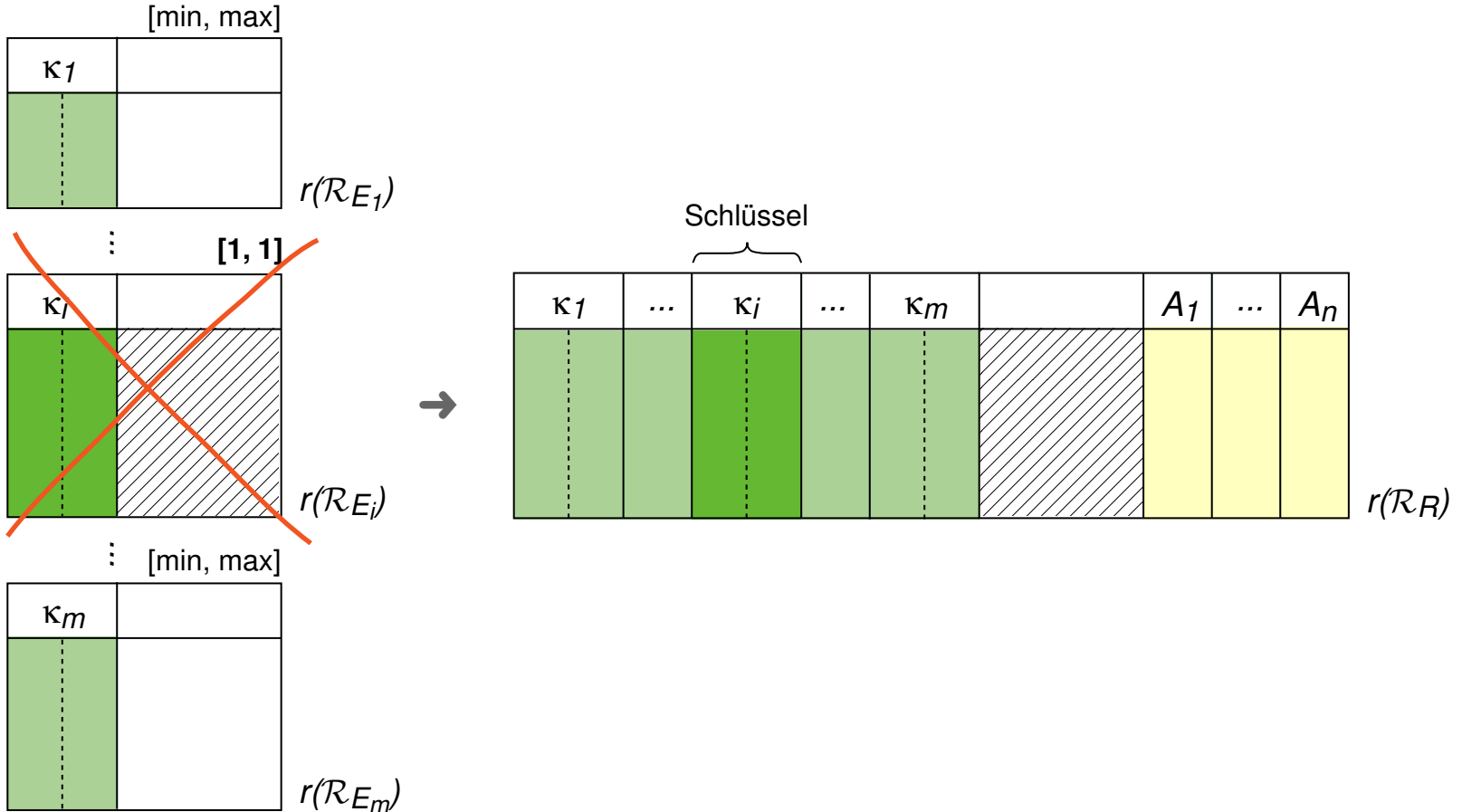
zu (a) Cross-Reference:



Umsetzung ER-Schema in relationales Schema

Relationale Darstellung von Beziehungstypen mit $[min, max]$ -Beschränkung

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

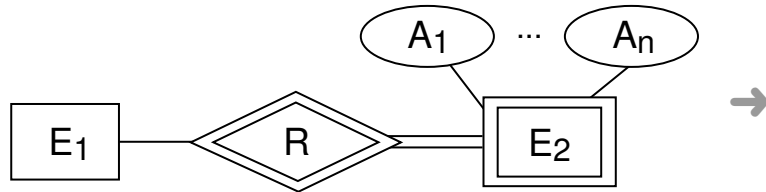


Bemerkungen:

- Eine $[0, 1]$ - bzw. $[1, 1]$ -Beschränkung qualifiziert den Schlüssel des zugehörigen Entity-Typs E_i offensichtlich als Schlüssel für den Beziehungstyp R , denn jede Ausprägung von R ist höchstens bzw. genau mit einer Instanz von E_i assoziiert.
- Für $m = 2$ und Vorliegen einer $[0, 1]$ -Beschränkung entspricht die Umsetzung dem Cross-Reference-Ansatz für binäre 1:n-Beziehungen.
- Für $m = 2$ und Vorliegen einer $[1, 1]$ -Beschränkung entspricht die Umsetzung der Zusammenfassung für binäre 1:n-Beziehungen.
- Für $m = 2$ und Vorliegen einer $[1, 1]$ -Beschränkung bei beiden Entities ist eine Umsetzung als Merged-Relation wie bei binären 1:1-Beziehungen möglich.

Umsetzung ER-Schema in relationales Schema

Relationale Darstellung von existenzabhängigen Entity-Typen



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

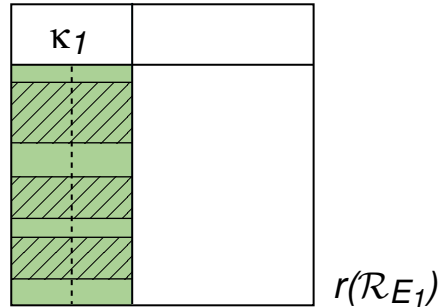
Schlüssel: $\kappa_1 \cup \kappa_2$ bzw. $\{ID_1\} \cup \kappa_2$

- 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} .
- 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.
- 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} .

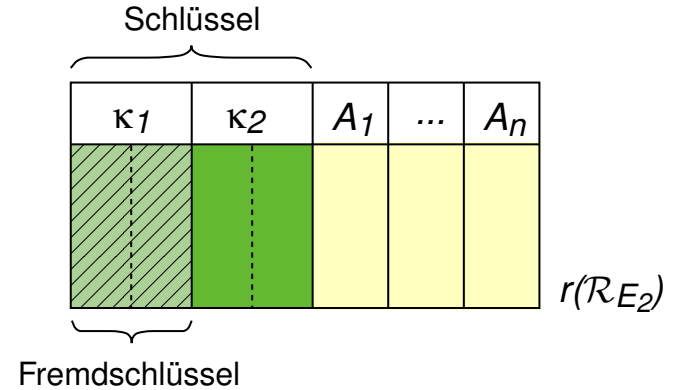
Umsetzung ER-Schema in relationales Schema

Relationale Darstellung von existenzabhängigen Entity-Typen

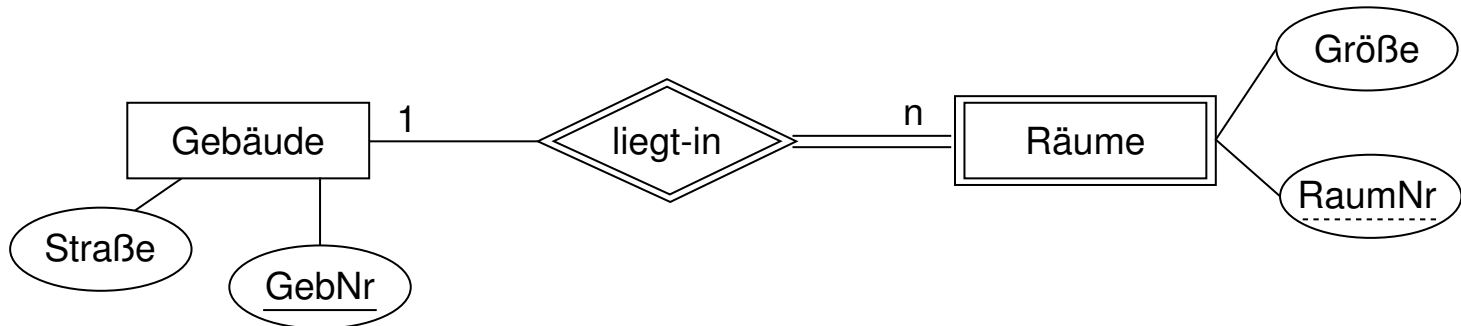
Regulärer Entity-Typ:



Abhängiger Entity-Typ:

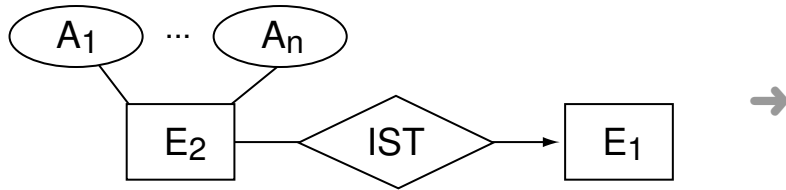


Beispiel:



Umsetzung ER-Schema in relationales Schema

Relationale Darstellung von IST-Beziehungstypen



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

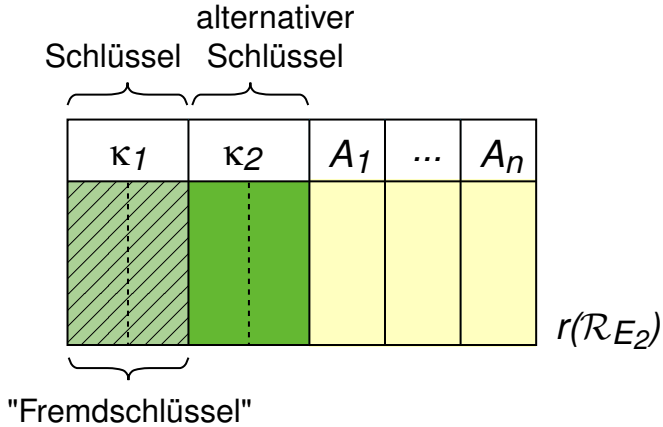
Schlüssel: κ_1 bzw. $\{ID_1\}$

- 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} .
- 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.
- Der Primärschlüssel κ_1 (bzw. $\{ID_1\}$) von E_1 wird Schlüssel für \mathcal{R}_{E_2} .

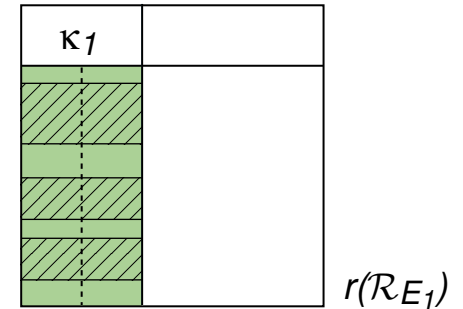
Umsetzung ER-Schema in relationales Schema

Relationale Darstellung von IST-Beziehungen

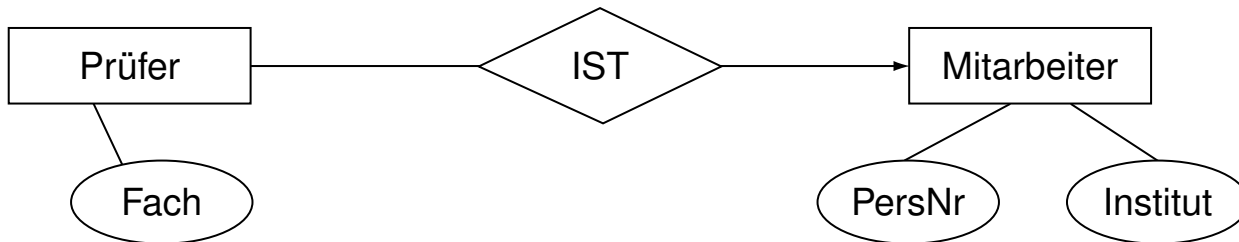
Speziellerer (is-a) Entity-Typ:



Allgemeinerer Entity-Typ:



Beispiel:



Bemerkungen:

- ❑ Es wird zwar die Bezeichnung „Fremdschlüssel“ benutzt, obwohl es sich bei der Spezialisierung nicht um einen Verweis auf einen anderen Entity-Typ handelt, sondern lediglich um eine Rollenangabe 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 vererbt. Damit ist auch eine Transformationsreihenfolge vorgegeben.

Umsetzung ER-Schema in relationales Schema

Reihenfolge der Regelanwendung [Elmasri/Navathe 2004]

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 m:n-Beziehungstypen.
6. Transformation der übrigen Beziehungstypen.
7. Transformation der IST-Beziehungstypen.

Bemerkungen:

- ❑ Beachte, dass bei der Zusammenfassung von Relationenschemata sich Namen und Attributmengen ändern und dass die dargestellten Regeln entsprechend anzupassen sind.

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

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 Schlüssel von \mathcal{R}_R , κ_1 oder κ_2 wird Primärschlüssel von \mathcal{R}_R
m:n-Beziehungstyp zwischen E_1 und E_2	$\kappa_1 \cup \kappa_2$ wird Schlüssel von \mathcal{R}_R

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 Schlüssel von \mathcal{R}_R , κ_1 oder κ_2 wird Primärschlüssel von \mathcal{R}_R
m:n-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 Attribute aus 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 Attribute aus κ_1 , κ_1 wird Schlüssel von \mathcal{R}_{E_2}