

Kapitel DB:III

III. Logischer Datenbankentwurf mit dem relationalen Modell

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

Das relationale Modell

1. Im relationalen Modell werden die Objekttypen der zu modellierenden Anwendungswelt durch *Relationenschemata* beschrieben.
2. Ein Relationenschema besteht aus einer Menge von *Attributen*, die Eigenschaften der Objekte eines Objekttyps repräsentieren.
3. Attribute haben als *Wertebereiche (Domains)* einfache Mengen, die meist von einem Standard-Datentyp wie `Integer`, `String` oder `Boolean` sind.
4. Legt man eine Reihenfolge für die 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.

[\[Definition 1\]](#)

Das relationale Modell

1. Im relationalen Modell werden die Objekttypen der zu modellierenden Anwendungswelt durch *Relationenschemata* beschrieben.
2. Ein Relationenschema besteht aus einer Menge von *Attributen*, die Eigenschaften der Objekte eines Objekttyps repräsentieren.
3. Attribute haben als *Wertebereiche (Domains)* einfache Mengen, die meist von einem Standard-Datentyp wie `Integer`, `String` oder `Boolean` sind.
4. Legt man eine Reihenfolge für die 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.

[\[Definition 1\]](#)

5. Ein *Datenbankschema* besteht aus einer Menge von Relationenschemata.
6. Die Menge der zu einem bestimmten Zeitpunkt zu den Relationenschemata eines Datenbankschemas vorhandenen Relationen heißt *Datenbank* oder *Datenbankzustand*. [\[Definition 2\]](#)

Das relationale Modell

Interpretation einer Relation als Tabelle:

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

Bemerkungen:

- ❑ Andere Bezeichnungen für den logischen Datenbankentwurf:
logischer Entwurf, Implementationsentwurf [[DB:II Entwurfsprozess](#)]
- ❑ 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 1 (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 $v \in dom(A_i)$ wird Attributwert für A_i genannt.

Das relationale Modell

Definition 1 (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 $v \in dom(A_i)$ wird Attributwert für A_i genannt.

1. 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 $A \in \mathcal{R}$.

Das relationale Modell

Definition 1 (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 $v \in dom(A_i)$ wird Attributwert für A_i genannt.

1. 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 $A \in \mathcal{R}$.

2. 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 notieren wir die Abbildung bzw. das Teiltupel $t|_{\alpha}$ vereinfachend auch als $t(\alpha)$.

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

Bemerkungen:

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

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

- Vergleiche hierzu die Mathematik; hier wird ein Tupel als eine geordnete Folge fester Länge von Werten bezeichnet:

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

- Zur Bezeichnung einer Relation können r und $r(\mathcal{R})$ gleichermaßen verwendet werden – abhängig davon, ob auf das Relationenschema \mathcal{R} im aktuellen Kontext Bezug genommen werden muss.
- Im Entity-Relationship-Modell wird ein Attribut A als eine Funktion betrachtet, die jeder Instanz eines Entity-Typs eine Eigenschaftsausprägung zuordnet; der Datentyp eines Attributs A wird mit T bezeichnet, in Zeichen: $A : T$. [\[DB:II ER-Konzepte und ihre Semantik\]](#)
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:

□ $\mathcal{R} = \text{Pers_Telefon} = \{\text{PANr}, \text{Telefon}\}$

$\text{dom}(\text{PANr}) = \text{Integer}$

$\text{dom}(\text{Telefon}) = \text{String}$

$r(\mathcal{R}) = r(\text{Pers_Telefon}) = \{t_1, t_2, t_3, \dots\}$

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

Das relationale Modell

Beispiel:

- $\mathcal{R} = \text{Pers_Telefon} = \{\text{PANr}, \text{Telefon}\}$

$\text{dom}(\text{PANr}) = \text{Integer}$

$\text{dom}(\text{Telefon}) = \text{String}$

$r(\mathcal{R}) = r(\text{Pers_Telefon}) = \{t_1, t_2, t_3, \dots\}$

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

- $t : \{\text{PANr}, \text{Telefon}\} \rightarrow \text{Integer} \cup \text{String}$

$t_1 = \{(\text{PANr}, 4711), (\text{Telefon}, 038203-12230)\}$

$t_1(\text{PANr}) = 4711$

$t_1(\text{Telefon}) = 038203-12230$

$t_2 = \{(\text{PANr}, 4711), (\text{Telefon}, 0381-498-3427)\}$

$t_2(\text{PANr}) = 4711$

$t_2(\text{Telefon}) = 0381-498-3427$

Bemerkungen:

- Eine Relation ist als endliche Menge von Abbildungen t_i definiert. Attributwerte können über den Attributnamen eindeutig identifiziert werden.
- Fasst man jedoch 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 2 (Datenbankschema, Datenbankzustand, Basisrelation)

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

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

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

Bemerkungen:

- ❑ Zur Bezeichnung einer Datenbank können d und $d(\mathcal{R})$ gleichermaßen verwendet werden – abhängig davon, ob auf das Datenbankschema \mathcal{R} im aktuellen Kontext Bezug genommen werden muss.
- ❑ Eine Datenbank d hat zu einem Zeitpunkt i den Zustand $state_i$.
[\[DB:II Datenbankmodelle > Datenbankzustand\]](#)
- ❑ Der Begriff der Basisrelation dient als Unterscheidung zu den hieraus *abgeleiteten Relationen*, die nicht in der Datenbank gespeichert sind.

Das relationale Modell

Beispiel:

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

r_1

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

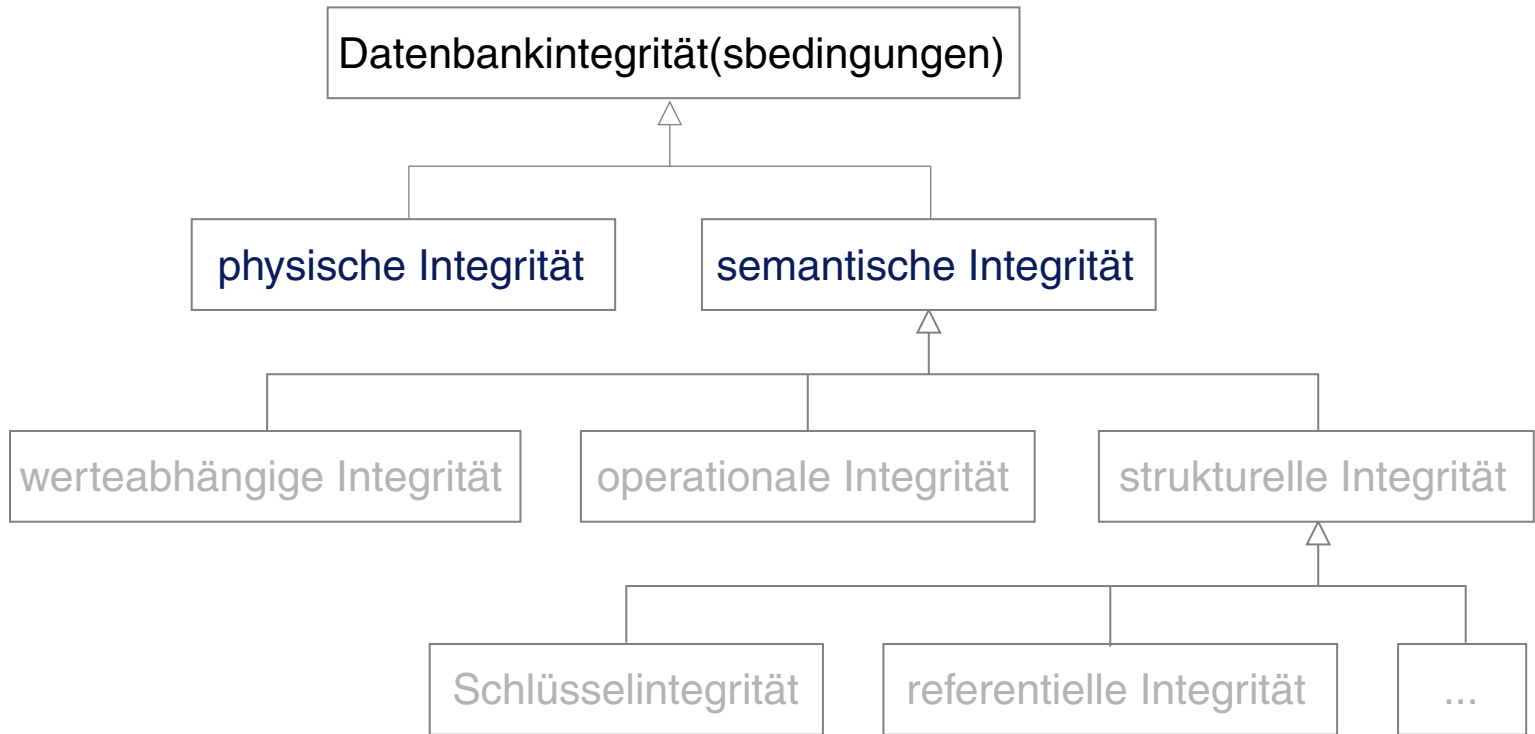
r_2

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

Integritätsbedingungen

Definition 3 (Integrität)

Die Integrität einer Datenbank d bezeichnet die korrekte und widerspruchsfreie Speicherung von Daten in d .



[Weber 2003]

Bemerkungen:

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

Integritätsbedingungen

Physische Integrität

- ❑ Regelt alle Aspekte der physikalischen Datenspeicherung.
- ❑ Berücksichtigt Hard- und Softwarefehler (soweit möglich).
- ❑ Behandelt Probleme des gleichzeitigen Zugriffs auf Daten.

Integritätsbedingungen

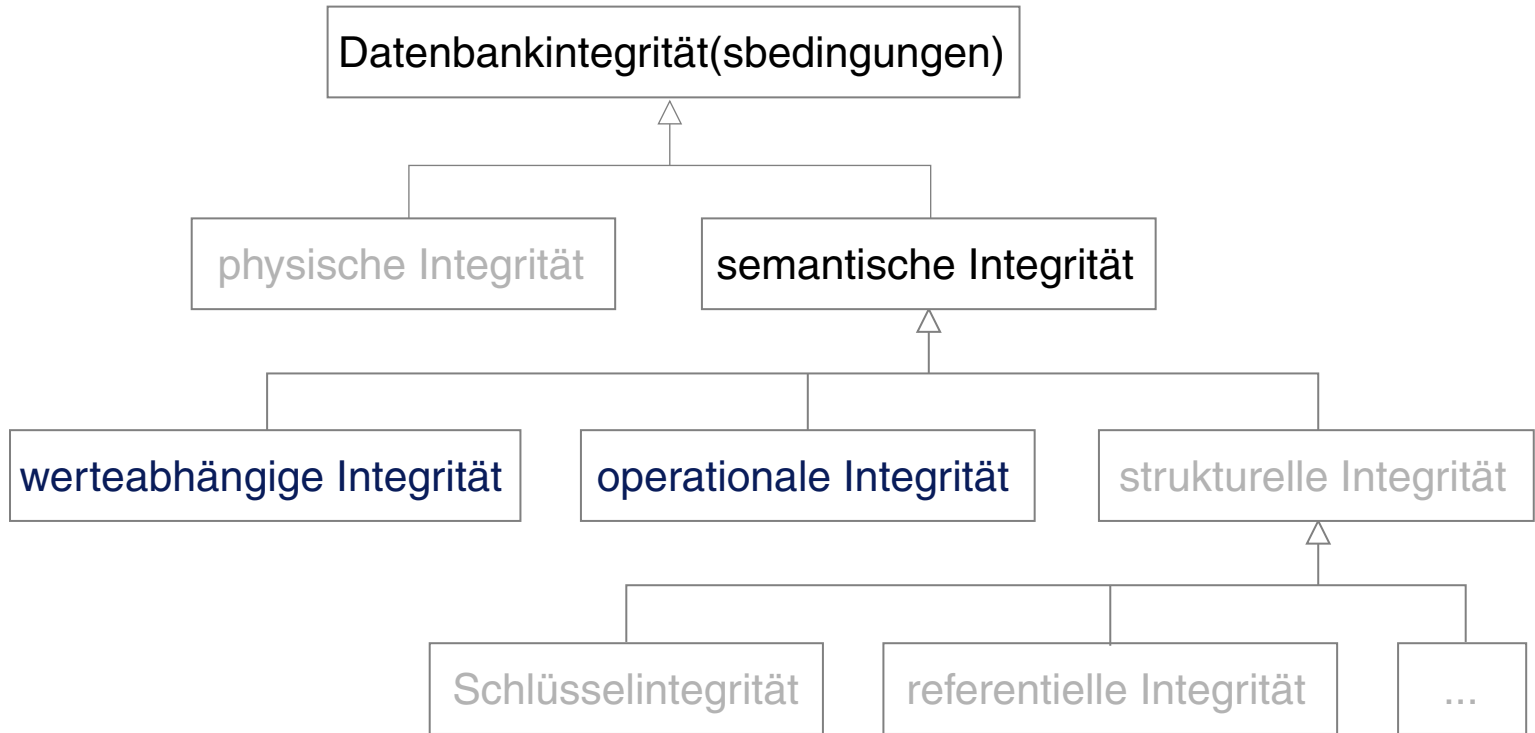
Physische Integrität

- ❑ Regelt alle Aspekte der physikalischen Datenspeicherung.
- ❑ Berücksichtigt Hard- und Softwarefehler (soweit möglich).
- ❑ Behandelt 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. [\[DB:II Charakterisierung von Beziehungstypen\]](#)
- ❑ Das Datenbank-Management-System überprüft mit semantischen Integritätsbedingungen, ob eine gewünschte Änderung der Datenbank zulässig ist. Das betrifft die Ebene 2 im DBMS. [\[DB:I DBMS > Komponenten\]](#)
- ❑ Semantische Integritätsbedingungen werden nach dem Inhalt unterschieden, den sie behandeln – in werteabhängige, operationale und strukturelle Integritätsbedingungen.

Integritätsbedingungen



[Weber 2003]

- ❑ Die Überprüfung der werteabhängigen Integrität basiert immer auf *einem* Zustand der Datenbank, sozusagen auf einer „eingefrorenen“ Datenbank.
- ❑ Die Überprüfung der operationalen Integrität basiert auf dem Vergleich des Datenbankzustandes zu zwei unterschiedlichen Zeitpunkten.

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 in der Anwendungsoberfläche aller möglichen Datenelemente auf erlaubte 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 in der Anwendungsoberfläche aller möglichen Datenelemente auf erlaubte Teilmengen.

Beispiele:

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

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

Produktion			
Standort	PLZ	Mitarbeiter	Budget (Mio)
Neuss	47323	111	44.3
Marl	45214	65	34.3
...			

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 in der Anwendungsoberfläche aller möglichen Datenelemente auf erlaubte Teilmengen.

Beispiele:

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

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

" $\Sigma <$ "

Produktion			
Standort	PLZ	Mitarbeiter	Budget (Mio)
Neuss	47323	111	44.3
Marl	45214	65	34.3

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 hinsichtlich 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 hinsichtlich 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 t :

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

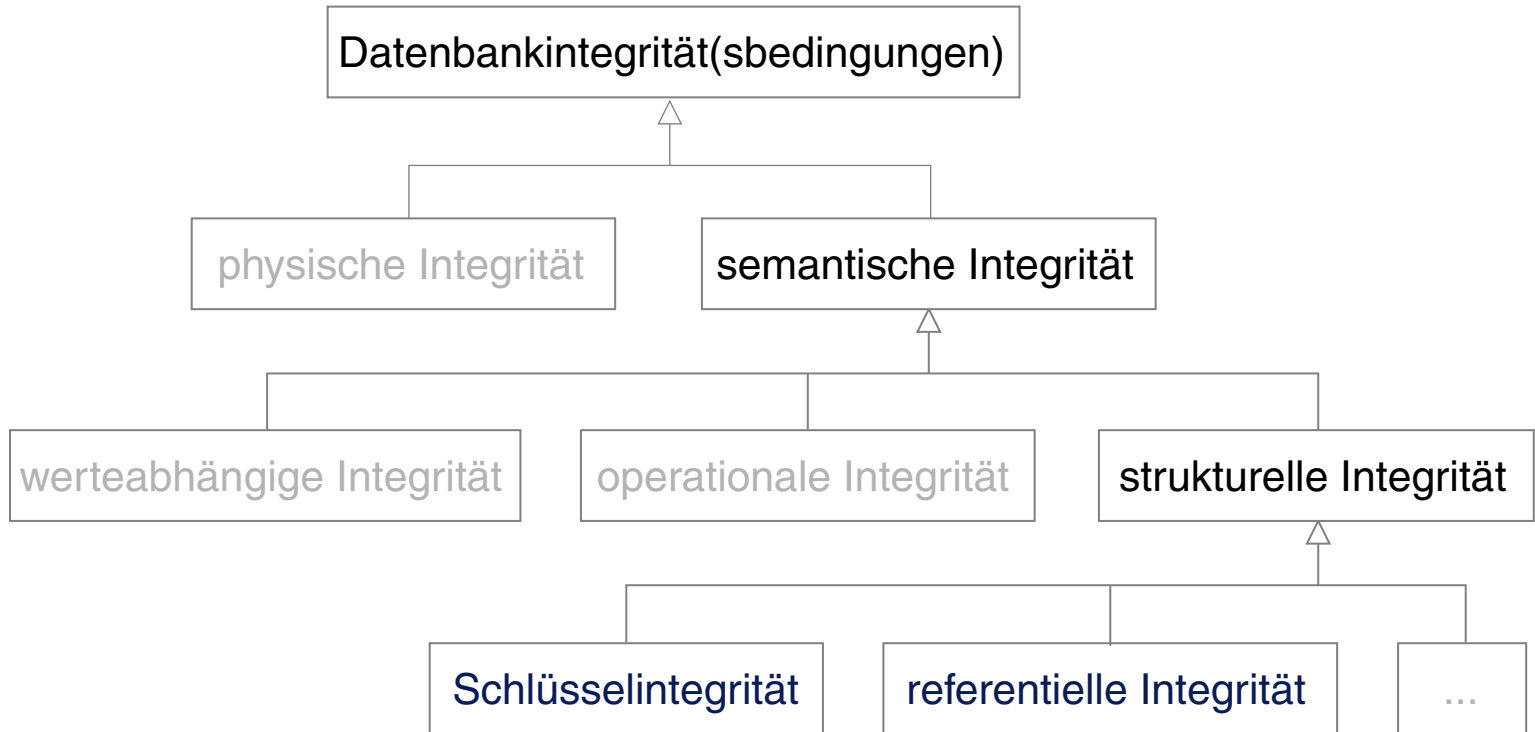
Zeitpunkt $t + 1$:

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

"<"



Integritätsbedingungen



[Weber 2003]

Integritätsbedingungen

Schlüssel und Schlüsselintegrität

Definition 4 (Schlüssel im Relationenmodell, Primärschlüssel, Primattribute)

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

$$\forall t_1, t_2 \in r : t_1 \neq t_2 \quad \text{impliziert} \quad \exists A \in \kappa : t_1(A) \neq t_2(A)$$

Integritätsbedingungen

Schlüssel und Schlüsselintegrität

Definition 4 (Schlüssel im Relationenmodell, Primärschlüssel, Primattribute)

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

$$\forall t_1, t_2 \in r : t_1 \neq t_2 \quad \text{impliziert} \quad \exists A \in \kappa : t_1(A) \neq t_2(A)$$

1. Ein Schlüssel bzw. ein Schlüsselkandidat ist eine minimal identifizierende Attributmenge. [\[DB:II ER-Konzepte und ihre Semantik > Schlüssel\]](#)
2. Ein Primärschlüssel ist ein unter mehreren Schlüsselkandidaten ausgezeichneteter Schlüssel.
3. 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							
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	04.04.1970
9999	Christa	Loeser	69121	HD	TS	38	10.05.1969

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

Integritätsbedingungen

Fremdschlüssel und referentielle Integrität

Definition 5 (Fremdschlüssel, referentielle Integrität)

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:

$$\exists t_1 \in r_1 \text{ mit } t_1(\kappa) = t_2(\alpha)$$

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

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

Integritätsbedingungen

Fremdschlüssel und referentielle Integrität

Definition 5 (Fremdschlüssel, referentielle Integrität)

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:

$$\exists t_1 \in r_1 \text{ mit } t_1(\kappa) = t_2(\alpha)$$

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

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)

$$\mathcal{R}_2 = \{ \underbrace{B_1, B_2, B_3, \dots, B_m}_{\text{Fremd"schlüssel" } \alpha} \}$$

Fremd"schlüssel" α

B_1	B_2	B_3	...	B_m

$r_2(\mathcal{R}_2)$

$$\mathcal{R}_1 = \{ \underbrace{\underline{A_1}, \underline{A_2}, \underline{A_3}, \dots, A_n}_{\text{Schlüssel } \kappa} \}$$

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 Attributmenge $\alpha \subset \mathcal{R}_2$, die in einem Relationenschema \mathcal{R}_1 Schlüssel ist. Daraus folgt $\alpha \subseteq \mathcal{R}_1$.
- ❑ Der Schlüssel κ des Relationenschemas \mathcal{R}_1 wird im Relationenschema \mathcal{R}_2 „fremd“ verwendet. Er stellt in \mathcal{R}_2 in der Regel *keinen Schlüssel* dar.
- ❑ Die Attribute von Primär- und Fremdschlüssel haben jeweils dieselbe Bedeutung und, falls möglich, auch dieselbe Bezeichnung. [Heuer/Saake 2013] spricht in diesem Zusammenhang von „kompatiblen Attributlisten“.
- ❑ Notation nach [Heuer/Saake 2013] für den Sachverhalt, dass $\alpha \subset \mathcal{R}_2$ Fremdschlüssel bezüglich κ ist: $\alpha(\mathcal{R}_2) \rightarrow \kappa(\mathcal{R}_1)$
- ❑ Eine Verschärfung der Definition der referentiellen Integrität entsteht durch die zusätzliche Forderung, dass $t_2(\alpha)$ entweder nur Nullwerte oder nur Werte ungleich Null besitzen darf. [Kemper/Eickler 2011]
- ❑ Ohne Überprüfung der referentiellen Integrität kann ein inkonsistenter Zustand der Datenbank entstehen.

Integritätsbedingungen

Fremdschlüssel und referentielle Integrität (Fortsetzung) [\[Illustration\]](#)

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	04.04.1970
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) [\[Illustration\]](#)

Beispiel (Fortsetzung):

- $\{\text{PANr}\}$ ist Schlüssel für die Relation „Personen“
 $\{\text{PANr}, \text{Telefon}\}$ ist Schlüssel für die Relation „Pers_Telefon“
- $\{\text{PANr}\}$ in der Relation „Pers_Telefon“ ist ein Fremdschlüssel und nimmt Bezug auf $\{\text{PANr}\}$ 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) [\[Illustration\]](#)

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:

Integritätsbedingungen

Fremdschlüssel und referentielle Integrität (Fortsetzung) [\[Illustration\]](#)

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 in r_1 mit der entsprechenden Fremdschlüsselausprägung wird *Cascading Insert* genannt.

Integritätsbedingungen

Fremdschlüssel und referentielle Integrität (Fortsetzung) [\[Illustration\]](#)

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 in r_1 mit der entsprechenden Fremdschlüsselausprägung wird *Cascading Insert* genannt.

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) [\[Illustration\]](#)

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 in r_1 mit der entsprechenden Fremdschlüsselausprägung wird *Cascading Insert* genannt.

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 wird *Cascading Update* genannt.

Integritätsbedingungen

Fremdschlüssel und referentielle Integrität (Fortsetzung) [\[Illustration\]](#)

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 in r_1 mit der entsprechenden Fremdschlüsselausprägung wird *Cascading Insert* genannt.

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 wird *Cascading Update* genannt.

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 beziehende Tupel in r_2 vorhanden sind, wird *Prohibited Delete* genannt. Die Propagierung von Löschungen in r_1 weiter zu r_2 wird als *Cascading Delete* bezeichnet.

Integritätsbedingungen

Definition 6 (Lokale und globale Integritätsbedingung)

Eine lokale Integritätsbedingung b ist eine Bool'sche Funktion, die auf der Menge der Relationen eines Relationenschemas \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 auf der Menge der Datenbanken eines Datenbankschemas \mathcal{R} definiert ist:

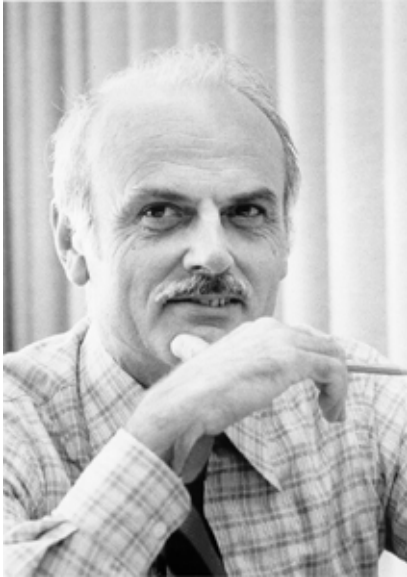
$$b : \{d \mid d(\mathcal{R})\} \rightarrow \{\text{true}, \text{false}\}$$

Bemerkungen:

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

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.computerworld.com 2002]