

Kapitel DB: VII

- I. Einführung und grundlegende Konzepte von Datenbanken
- II. Datenbankentwurf und Datenbankmodelle
- III. Konzeptueller Datenbankentwurf
- IV. Logischer Datenbankentwurf mit dem relationalen Modell
- V. Grundlagen relationaler Anfragesprachen
- VI. Die relationale Datenbanksprache SQL

VII. Entwurfstheorie relationaler Datenbanken

- Informelle Entwurfsziele für Relationenschemata
- Funktionale Abhängigkeiten
- Normalformen
- Dekompositionseigenschaften von Relationen
- Relationale Dekomposition
- Relationale Synthese
- Mehrwertige Abhängigkeiten

Informelle Entwurfskriterien für Relationenschemata

Ziel I: Semantik von Relationenschemata klar halten

Datenbankschema mit einleuchtender Semantik:

Mitarbeiter

Name PersNr Wohnort AbtNr

Abteilung

AbtNr AbtName ChefPersNr

Standorte

AbtNr AbtOrt

Projekte

ProjektName ProjektNr ProjektOrt AbtNr

ArbeitetInProjekt

PersNr ProjektNr Stunden

Informelle Entwurfskriterien für Relationenschemata

Ziel I: Semantik von Relationenschemata klar halten

Datenbankschema mit einleuchtender Semantik:

Mitarbeiter			
Name	<u>PersNr</u>	Wohnort	AbtNr

Abteilung		
<u>AbtNr</u>	AbtName	ChefPersNr

Standorte	
<u>AbtNr</u>	<u>AbtOrt</u>

Projekte			
ProjektName	<u>ProjektNr</u>	ProjektOrt	AbtNr

ArbeitetInProjekt		
<u>PersNr</u>	<u>ProjektNr</u>	Stunden

Problematisches Datenbankschema:

MitarbeiterAbteilung					
Name	<u>PersNr</u>	Wohnort	AbtNr	AbtName	ChefPersNr

MitarbeiterProjekte					
<u>PersNr</u>	<u>ProjektNr</u>	Stunden	Name	ProjektName	ProjektOrt

Informelle Entwurfskriterien für Relationenschemata

Ziel II: Vermeidung von (Update-)Anomalien

Mitarbeiter			
Name	<u>PersNr</u>	Wohnort	AbtNr
Smith	1234	Weimar	5
Wong	3334	Köln	5
Zelaya	9998	Erfurt	4
Wallace	9876	Berlin	4

⊗

Abteilung		
AbtName	<u>AbtNr</u>	ChefPersNr
Forschung	5	3334
Verwaltung	4	9876
Stab	1	8886

=

MitarbeiterAbteilung					
Name	<u>PersNr</u>	Wohnort	AbtNr	AbtName	ChefPersNr
Smith	1234	Weimar	5	Forschung	3334
Wong	3334	Köln	5	Forschung	3334
Zelaya	9998	Erfurt	4	Verwaltung	9876
Wallace	9876	Berlin	4	Verwaltung	9876

Redundanz

Informelle Entwurfskriterien für Relationenschemata

Ziel II: Vermeidung von (Update-)Anomalien

MitarbeiterAbteilung					
Name	<u>PersNr</u>	Wohnort	AbtNr	AbtName	ChefPersNr
Smith	1234	Weimar	5	Forschung	3334
Wong	3334	Köln	5	Forschung	3334
Zelaya	9998	Erfurt	4	Verwaltung	9876
Wallace	9876	Berlin	4	Verwaltung	9876

Einfüge- bzw. Insert-Anomalie:

1. Beim Einfügen eines neuen Tupels ist sicherzustellen, dass für redundante Attribute (im Beispiel: AbtName, ChefPersNr) die richtigen Werte eingetragen werden.
2. Das Einfügen von Tupeln, für die nicht alle Attribute bekannt sind, ist problematisch.

Im Beispiel: Wie soll eine neue Abteilung ohne Mitarbeiter eingetragen werden? Beachte, dass das Attribut „PersNr“ Schlüssel dieser Relation ist und hier keine Nullwerte eingetragen werden dürfen.

Informelle Entwurfskriterien für Relationenschemata

Ziel II: Vermeidung von (Update-)Anomalien

MitarbeiterAbteilung					
Name	<u>PersNr</u>	Wohnort	AbtNr	AbtName	ChefPersNr
Smith	1234	Weimar	5	Forschung	3334
Wong	3334	Köln	5	Forschung	3334
Zelaya	9998	Erfurt	4	Verwaltung	9876
Wallace	9876	Berlin	4	Verwaltung	9876

Lösch- bzw. Delete-Anomalie:

Das Löschen von Tupeln ist problematisch, weil hierbei Wissen über andere Konzepte verloren gehen kann.

Im Beispiel: Wird der „letzte“ Mitarbeiter einer Abteilung gelöscht, so verschwindet auch das Wissen über die Abteilung.

Informelle Entwurfskriterien für Relationenschemata

Ziel II: Vermeidung von (Update-)Anomalien

MitarbeiterAbteilung					
Name	<u>PersNr</u>	Wohnort	AbtNr	AbtName	ChefPersNr
Smith	1234	Weimar	5	Forschung	3334
Wong	3334	Köln	5	Forschung	3334
Zelaya	9998	Erfurt	4	Verwaltung	9876
Wallace	9876	Berlin	4	Verwaltung	9876

Modifizierungs- bzw. Modify-Anomalie:

Die Änderung eines Attributes muss in allen Tupeln, die den gleichen Attributwert haben, nachvollzogen werden. Ansonsten wird die Datenbank inkonsistent.

Im Beispiel: Änderung des Abteilungsnamens oder der Chefpersonalnummer.

Bemerkungen:

- ❑ Bzgl. der Konsistenzerhaltung müssen Modifizierungsanomalien nicht kritisch sein, weil durch die Verwendung von Triggern, Stored-Procedures oder durch ein einziges SQL-Statement alle Korrekturen erfassbar sind. Das Problem des erhöhten Speicherbedarfs oder von Performanzeinbußen bleibt jedoch.
- ❑ Umgekehrt kann es aus Performanzgründen in manchen Situationen sinnvoll sein, Anomalien in Kauf zu nehmen und *mehrere* Entity-Typen innerhalb *einer* Relation zu modellieren. Beispiel: Eine Anwendung hat extrem viele Anfragen, in denen immer wieder die gleichen Entity-Typen vorkommen. Stichwort: Denormalisierung
Einige professionelle Datenbanksysteme erkennen solche Situationen und richten automatisch – ausgehend von Views – verbundene Relationen ein, um die Anzahl der Join-Operationen klein zu halten.
- ❑ In [Kemper/Eickler 2004] werden Modifizierungsanomalien als Update-Anomalien bezeichnet.

Informelle Entwurfskriterien für Relationenschemata

Ziel III: Möglichst wenig Nullwerte

Nullwerte sind aus folgenden Gründen kritisch:

1. Speicherplatz

Attribute, die nur für sehr wenige Tupel einen Wert besitzen, sollten in einem eigenen Schema untergebracht werden.

2. Eindeutigkeit

Die Semantik von Nullwerten ist nicht eindeutig – u. a.:

- (a) Wert unbekannt (aber könnte irgendwann bekannt sein)
- (b) Wert existiert nicht für das Attribut (und wird auch nie existieren)
- (c) Wert soll/darf nicht gespeichert werden

3. Verrechnung

Nullwerte verhindern die sinnvolle Verrechnung von Attributen mittels Aggregatfunktionen wie `count`, `max` etc.

Informelle Entwurfskriterien für Relationenschemata

Ziel IV: Verhinderung der Erzeugung falscher Tupel

MitarbeiterProjekte					
<u>PersNr</u>	<u>ProjektNr</u>	Stunden	Name	ProjektName	ProjektOrt
1234	1	32.5	Smith	Produkt-X	Bellaire
1234	2	7.5	Smith	Produkt-Y	Sugarland
3334	2	10.0	Wong	Produkt-Y	Sugarland
3334	3	10.0	Wong	Produkt-Z	Houston
3334	10	10.0	Wong	Vernetzung	Stafford
3334	20	10.0	Wong	Neuorganisation	Houston

Nach entsprechender Projektion:

MitarbeiterEinsatzOrte	
Name	ProjektOrt
Smith	Bellaire
Smith	Sugarland
Wong	Sugarland
Wong	Houston
Wong	Stafford

MitarbeiterProjekte2				
<u>PersNr</u>	<u>ProjektNr</u>	Stunden	ProjektName	ProjektOrt
1234	1	32.5	Produkt-X	Bellaire
1234	2	7.5	Produkt-Y	Sugarland
3334	2	10.0	Produkt-Y	Sugarland
3334	3	10.0	Produkt-Z	Houston
3334	10	10.0	Vernetzung	Stafford
3334	20	10.0	Neuorganisation	Houston

Informelle Entwurfskriterien für Relationenschemata

Ziel IV: Verhinderung der Erzeugung falscher Tupel

MitarbeiterEinsatzOrte	
Name	ProjektOrt
Smith	Bellaire
Smith	Sugarland
Wong	Sugarland
Wong	Houston
Wong	Stafford

⋈

MitarbeiterProjekte2					
PersNr	ProjektNr	Stunden	ProjektName	ProjektOrt	
1234	1	32.5	Produkt-X	Bellaire	
1234	2	7.5	Produkt-Y	Sugarland	
3334	2	10.0	Produkt-Y	Sugarland	
3334	3	10.0	Produkt-Z	Houston	
3334	10	10.0	Vernetzung	Stafford	
3334	20	10.0	Neuorganisation	Houston	

=

PersNr	ProjektNr	Stunden	Name	ProjektName	ProjektOrt	
1234	1	32.5	Smith	Produkt-X	Bellaire	
1234	2	7.5	Smith	Produkt-Y	Sugarland	
1234	2	7.5	Wong	Produkt-Y	Sugarland	*
3334	2	10.0	Smith	Produkt-Y	Sugarland	*
3334	2	10.0	Wong	Produkt-Y	Sugarland	
3334	3	10.0	Wong	Produkt-Z	Houston	
3334	10	10.0	Wong	Vernetzung	Stafford	
3334	20	10.0	Wong	Neuorganisation	Houston	

*

*

Bemerkungen:

- Die Zerlegung der Relation „MitarbeiterProjekte“ in „MitarbeiterEinsatzOrte“ und „MitarbeiterProjekte2“ ist nicht sinnvoll, da mittels des natürlichen Verbundes die Originalrelation nicht mehr hergestellt werden kann: Es entstehen falsche (*spurious*) Tupel, hier mit \star gekennzeichnet.

Funktionale Abhängigkeiten

Definition 15 (funktionale Abhängigkeit, FD)

Gegeben sei ein Relationenschema \mathcal{R} . Weiterhin seien α, β Mengen von Attributen mit $\alpha \subseteq \mathcal{R}$ und $\beta \subseteq \mathcal{R}$. Eine funktionale Abhängigkeit

$$\alpha \rightarrow \beta$$

(functional dependency, FD) liegt vor, wenn die möglichen gültigen Ausprägungen $r(\mathcal{R})$ von \mathcal{R} folgende Bedingung erfüllen:

$$\forall t_1, t_2 \in r : t_1(\alpha) = t_2(\alpha) \text{ impliziert } t_1(\beta) = t_2(\beta)$$

Sprechweisen:

- die α -Werte bestimmen die β -Werte funktional
- die β -Werte sind funktional abhängig von den α -Werten
- r erfüllt die funktionale Abhängigkeit $\alpha \rightarrow \beta$
- r genügt der funktionalen Abhängigkeit $\alpha \rightarrow \beta$

Bemerkungen:

- Funktionale Abhängigkeiten stellen eine semantische Konsistenzbedingung dar, die sich aus einem sachlogischen Verständnis der Anwendung und nicht aus einer aktuellen Relationenausprägung ergeben.

Somit sind funktionale Abhängigkeiten Eigenschaften von Relationenschemata und können nicht automatisch von der Ausprägung eines Schemas abgeleitet werden.

- Folgende Konventionen zur Syntax haben sich eingebürgert:
 - große lateinische Buchstaben A, B, C, \dots bezeichnen Attribute
 - kleine griechische Buchstaben $\alpha, \beta, \gamma, \dots$ bezeichnen Attributmengen
 - $AB \rightarrow C$ steht für $\{A, B\} \rightarrow \{C\}$
 - $\alpha\beta$ steht für $\alpha \cup \beta$

Funktionale Abhängigkeiten

Beispiel:

	\mathcal{R}			
	A	B	C	D
t_1	a_4	b_2	c_4	d_3
t_2	a_1	b_1	c_1	d_1
t_3	a_1	b_1	c_1	d_2
t_4	a_2	b_2	c_3	d_2
t_5	a_3	b_2	c_4	d_3

Beobachtung:

Die funktionale Abhängigkeit $\alpha \rightarrow \beta$ ist in einer Relation r erfüllt, falls für jede mögliche Ausprägung $t(\alpha) = c$ gilt, dass die Projektion $\pi_\beta(\sigma_{\alpha=c}(r))$ nur *ein* Element enthält.

Funktionale Abhängigkeiten

Definition 16 (triviale funktionale Abhängigkeiten)

Funktionale Abhängigkeiten, die von jeder Relationenausprägung erfüllt sind, heißen triviale funktionale Abhängigkeiten.

Funktionale Abhängigkeiten

Definition 16 (triviale funktionale Abhängigkeiten)

Funktionale Abhängigkeiten, die von jeder Relationenausprägung erfüllt sind, heißen triviale funktionale Abhängigkeiten.

Definition 17 (voll funktional abhängig)

Gegeben sei ein Relationenschema \mathcal{R} mit $\alpha \subseteq \mathcal{R}$ und $\beta \subseteq \mathcal{R}$. β ist voll funktional abhängig von α , falls gilt:

1. $\alpha \rightarrow \beta$
2. $\forall A \in \alpha : (\alpha - \{A\}) \not\rightarrow \beta$, α ist also nicht verkleinerbar.

Funktionale Abhängigkeiten

Definition 16 (triviale funktionale Abhängigkeiten)

Funktionale Abhängigkeiten, die von jeder Relationenausprägung erfüllt sind, heißen triviale funktionale Abhängigkeiten.

Definition 17 (voll funktional abhängig)

Gegeben sei ein Relationenschema \mathcal{R} mit $\alpha \subseteq \mathcal{R}$ und $\beta \subseteq \mathcal{R}$. β ist voll funktional abhängig von α , falls gilt:

1. $\alpha \rightarrow \beta$
2. $\forall A \in \alpha : (\alpha - \{A\}) \not\rightarrow \beta$, α ist also nicht verkleinerbar.

Definition 18 (Superschlüssel, Kandidatschlüssel, Primattribute, Primärschlüssel)

Gegeben sei ein Relationenschema \mathcal{R} mit $\alpha \subseteq \mathcal{R}$.

- α ist ein Superschlüssel, falls $\alpha \rightarrow \mathcal{R}$ gilt.
- α ist ein Schlüssel bzw. Kandidatschlüssel von \mathcal{R} , falls \mathcal{R} voll funktional abhängig von α ist. Die Attribute eines Schlüssels heißen Primattribute.
- Ein Primärschlüssel ist ein ausgezeichneteter Kandidatschlüssel.

Bemerkungen:

- Man kann zeigen, dass nur funktionale Abhängigkeiten der Art $\alpha \rightarrow \beta$ mit $\beta \subseteq \alpha$ trivial sind.
- Gilt $\alpha \rightarrow \beta$ und $\exists A \in \alpha : (\alpha - \{A\}) \rightarrow \beta$, so nennt man β auch partiell abhängig von α .
- Das Konzept der vollen funktionalen Abhängigkeit dient dazu, Schlüssel von Superschlüsseln abzugrenzen.
- Die Menge aller Attribute einer Relation \mathcal{R} bildet einen Superschlüssel: $\mathcal{R} \rightarrow \mathcal{R}$.

Funktionale Abhängigkeiten

Ableitung funktionaler Abhängigkeiten

Beispiel:

\mathcal{R}		
A	B	C
a_1	b_1	c_1
a_2	b_1	c_1
a_3	b_2	c_1
a_4	b_1	c_1

Beobachtung:

Genügt \mathcal{R} den funktionalen Abhängigkeiten $A \rightarrow B$ und $B \rightarrow C$, so genügt \mathcal{R} auch der funktionalen Abhängigkeit $A \rightarrow C$.

Funktionale Abhängigkeiten

Ableitung funktionaler Abhängigkeiten

Beispiel:

\mathcal{R}		
A	B	C
a_1	b_1	c_1
a_2	b_1	c_1
a_3	b_2	c_1
a_4	b_1	c_1

Beobachtung:

Genügt \mathcal{R} den funktionalen Abhängigkeiten $A \rightarrow B$ und $B \rightarrow C$, so genügt \mathcal{R} auch der funktionalen Abhängigkeit $A \rightarrow C$.

Definition 19 (Hülle einer Menge von FDs)

Sei F eine Menge funktionaler Abhängigkeiten. Dann bezeichnet F^+ die Hülle von F ; sie enthält die Menge F sowie alle funktionale Abhängigkeiten, die auf der Basis von F ableitbar sind.

Funktionale Abhängigkeiten

Ableitung funktionaler Abhängigkeiten

Sei \mathcal{R} ein Relationenschema und seien α, β, γ und δ Teilmengen von \mathcal{R} .

Armstrong-Inferenzregeln:

1. *Reflexivität.* Falls $\beta \subseteq \alpha$, dann gilt $\alpha \rightarrow \beta$. Insbesondere gilt $\alpha \rightarrow \alpha$.
2. *Verstärkung.* Falls $\alpha \rightarrow \beta$, dann gilt $\alpha\gamma \rightarrow \beta\gamma$.
3. *Transitivität.* Falls $\alpha \rightarrow \beta$ und $\beta \rightarrow \gamma$, dann gilt $\alpha \rightarrow \gamma$.

Funktionale Abhängigkeiten

Ableitung funktionaler Abhängigkeiten

Sei \mathcal{R} ein Relationenschema und seien α, β, γ und δ Teilmengen von \mathcal{R} .

Armstrong-Inferenzregeln:

1. *Reflexivität.* Falls $\beta \subseteq \alpha$, dann gilt $\alpha \rightarrow \beta$. Insbesondere gilt $\alpha \rightarrow \alpha$.
2. *Verstärkung.* Falls $\alpha \rightarrow \beta$, dann gilt $\alpha\gamma \rightarrow \beta\gamma$.
3. *Transitivität.* Falls $\alpha \rightarrow \beta$ und $\beta \rightarrow \gamma$, dann gilt $\alpha \rightarrow \gamma$.

Zusätzliche Ableitungsregeln:

4. *Vereinigung.* Falls $\alpha \rightarrow \beta$ und $\alpha \rightarrow \gamma$, dann gilt $\alpha \rightarrow \beta\gamma$.
5. *Dekomposition.* Falls $\alpha \rightarrow \beta\gamma$, dann gilt $\alpha \rightarrow \beta$ und $\alpha \rightarrow \gamma$.
6. *Pseudotransitivität.* Falls $\alpha \rightarrow \beta$ und $\gamma\beta \rightarrow \delta$, dann gilt $\alpha\gamma \rightarrow \delta$.

Bemerkungen:

- ❑ Konvention zur Syntax (Wiederholung): $\alpha\beta$ steht für $\alpha \cup \beta$
- ❑ Häufig werden die drei Armstrong-Inferenzregeln auch als Armstrong-Axiome bezeichnet, obwohl der Begriff „Axiom“ hier nicht korrekt verwendet ist: Axiome sind voraussetzungslose Forderungen, die als wahr angenommen werden (und die zusammen eine Theorie bilden). Tatsächlich entspricht eine Menge von FDs einer Menge von Axiomen: unter der Voraussetzung ihrer Wahrheit können hieraus weitere wahre FDs abgeleitet werden.
- ❑ Die drei Armstrong-Inferenzregeln sind *korrekt* (sound) und *vollständig* (complete).
 1. Korrektheit. Mit Hilfe der Armstrong-Inferenzregeln lassen sich aus einer Menge F von FDs nur solche FDs ableiten, die von jeder Relationenausprägung erfüllt sind, für die F erfüllt ist.
 2. Vollständigkeit. Mit Hilfe der Armstrong-Inferenzregeln lassen sich alle FDs ableiten, die durch F logisch impliziert sind.

Diese Eigenschaften sind zu beweisen.

- ❑ Aufgrund der Vollständigkeit der Armstrong-Inferenzregeln sind die zusätzlichen Ableitungsregeln nicht nötig; mit ihnen lassen sich Ableitungsprozesse jedoch einfacher gestalten.
- ❑ Die Bestimmung aller FDs kann exponentiellen Aufwand haben. Hinzu kommt, dass man oft nicht an F^+ interessiert ist, sondern nur an der Menge von *Attributen* α^+ , die von einer Attributmeng α gemäß einer Menge F von FDs funktional bestimmt werden. α^+ ist die Hülle der Attributmeng α .

Funktionale Abhängigkeiten

Ableitung funktionaler Abhängigkeiten

Algorithm: AttributeClosure

Input: F . Menge funktionaler Abhängigkeiten.
 α . Menge von Attributen.

Output: α^+ . Hülle der Attributmenge α .

1. $\alpha^+ = \alpha$
2. **REPEAT**
3. $\alpha_{\text{tmp}}^+ = \alpha^+$
4. **FOREACH** $(\beta \rightarrow \gamma) \in F$ **DO**
5. **IF** $\beta \subseteq \alpha^+$ **THEN** $\alpha^+ = \alpha^+ \cup \gamma$
6. **ENDDO**
7. **UNTIL** $(\alpha^+ = \alpha_{\text{tmp}}^+)$
8. **RETURN** (α^+)

Funktionale Abhängigkeiten

Äquivalenz von Mengen funktionaler Abhängigkeiten

Definition 20 (Überdeckung einer Menge von FDs)

Eine Menge funktionaler Abhängigkeiten F_1 überdeckt eine Menge funktionaler Abhängigkeiten F_2 , falls jede FD aus F_2 Element in F_1^+ ist.

Definition 21 (Äquivalenz zweier Mengen von FDs)

Zwei Mengen F_1 und F_2 funktionaler Abhängigkeiten heißen äquivalent, in Zeichen $F_1 \equiv F_2$, wenn sie die gleichen Hüllen besitzen, also wenn $F_1^+ = F_2^+$ gilt.

Bemerkungen:

- Falls F_1 die Menge F_2 überdeckt, so kann jede FD in F_2 auch mittels F_1 abgeleitet werden.
- F_1 ist äquivalent zu F_2 genau dann, wenn F_1 die Menge F_2 überdeckt und wenn F_2 die Menge F_1 überdeckt.

Funktionale Abhängigkeiten

Minimale und kanonische Überdeckungen von FDs

Konsistenzprüfungen bei Datenbankmodifikationen oder beim Datenbankentwurf erfordern die Überprüfung der spezifizierten funktionalen Abhängigkeiten.

→ Hierfür interessiert die kleinstmögliche, noch äquivalente Menge an FDs.

Funktionale Abhängigkeiten

Minimale und kanonische Überdeckungen von FDs

Konsistenzprüfungen bei Datenbankmodifikationen oder beim Datenbankentwurf erfordern die Überprüfung der spezifizierten funktionalen Abhängigkeiten.

→ Hierfür interessiert die kleinstmögliche, noch äquivalente Menge an FDs.

Definition 22 (kanonische Überdeckung [Kemper/Eickler 2004])

Zu einer gegebenen Menge F von FDs nennt man F_c eine kanonische Überdeckung, falls folgende Eigenschaften erfüllt sind:

1. $F_c \equiv F$ bzw. $F_c^+ = F^+$

2. Alle FDs $\alpha \rightarrow \beta$ in F_c sind links- reduziert:

(a) $\forall A \in \alpha : (F_c - \{\alpha \rightarrow \beta\}) \cup \{(\alpha - \{A\}) \rightarrow \beta\} \not\equiv F_c$

3. In F_c gibt es keine zwei FDs, die die gleiche linke Seite besitzen.

Funktionale Abhängigkeiten

Minimale und kanonische Überdeckungen von FDs

Konsistenzprüfungen bei Datenbankmodifikationen oder beim Datenbankentwurf erfordern die Überprüfung der spezifizierten funktionalen Abhängigkeiten.

→ Hierfür interessiert die kleinstmögliche, noch äquivalente Menge an FDs.

Definition 22 (kanonische Überdeckung [Kemper/Eickler 2004])

Zu einer gegebenen Menge F von FDs nennt man F_c eine kanonische Überdeckung, falls folgende Eigenschaften erfüllt sind:

1. $F_c \equiv F$ bzw. $F_c^+ = F^+$
2. Alle FDs $\alpha \rightarrow \beta$ in F_c sind links- und rechtsreduziert:
 - (a) $\forall A \in \alpha : (F_c - \{\alpha \rightarrow \beta\}) \cup \{(\alpha - \{A\}) \rightarrow \beta\} \neq F_c$
 - (b) $\forall B \in \beta : (F_c - \{\alpha \rightarrow \beta\}) \cup \{\alpha \rightarrow (\beta - \{B\})\} \neq F_c$
3. In F_c gibt es keine zwei FDs, die die gleiche linke Seite besitzen.

Funktionale Abhängigkeiten

Minimale und kanonische Überdeckungen von FDs

Aus einer gegebenen Menge F von FDs lässt sich eine kanonische Überdeckung wie folgt bestimmen.

1. **Linksreduktion** für jede FD $(\alpha \rightarrow \beta) \in F$.
Falls $\beta \subseteq \text{Attribute-Closure}(F, \alpha - \{A\})$,
dann ersetze $\alpha \rightarrow \beta$ durch $(\alpha - \{A\}) \rightarrow \beta$.

Funktionale Abhängigkeiten

Minimale und kanonische Überdeckungen von FDs

Aus einer gegebenen Menge F von FDs lässt sich eine kanonische Überdeckung wie folgt bestimmen.

1. **Linksreduktion** für jede FD $(\alpha \rightarrow \beta) \in F$.

Falls $\beta \subseteq \text{Attribute-Closure}(F, \alpha - \{A\})$,

dann ersetze $\alpha \rightarrow \beta$ durch $(\alpha - \{A\}) \rightarrow \beta$.

2. **Rechtsreduktion** für jede FD $(\alpha \rightarrow \beta) \in F$.

Falls $B \in \text{Attribute-Closure}((F - \{\alpha \rightarrow \beta\}) \cup \{\alpha \rightarrow (\beta - \{B\})\}, \alpha)$,

dann ersetze $\alpha \rightarrow \beta$ durch $\alpha \rightarrow (\beta - \{B\})$.

Funktionale Abhängigkeiten

Minimale und kanonische Überdeckungen von FDs

Aus einer gegebenen Menge F von FDs lässt sich eine kanonische Überdeckung wie folgt bestimmen.

1. **Linksreduktion** für jede FD $(\alpha \rightarrow \beta) \in F$.
Falls $\beta \subseteq \text{Attribute-Closure}(F, \alpha - \{A\})$,
dann ersetze $\alpha \rightarrow \beta$ durch $(\alpha - \{A\}) \rightarrow \beta$.
2. **Rechtsreduktion** für jede FD $(\alpha \rightarrow \beta) \in F$.
Falls $B \in \text{Attribute-Closure}((F - \{\alpha \rightarrow \beta\}) \cup \{\alpha \rightarrow (\beta - \{B\})\}, \alpha)$,
dann ersetze $\alpha \rightarrow \beta$ durch $\alpha \rightarrow (\beta - \{B\})$.
3. Löschen aller FDs der Form $(\alpha \rightarrow \emptyset) \in F$.

Funktionale Abhängigkeiten

Minimale und kanonische Überdeckungen von FDs

Aus einer gegebenen Menge F von FDs lässt sich eine kanonische Überdeckung wie folgt bestimmen.

1. **Linksreduktion** für jede FD $(\alpha \rightarrow \beta) \in F$.
Falls $\beta \subseteq \text{Attribute-Closure}(F, \alpha - \{A\})$,
dann ersetze $\alpha \rightarrow \beta$ durch $(\alpha - \{A\}) \rightarrow \beta$.
2. **Rechtsreduktion** für jede FD $(\alpha \rightarrow \beta) \in F$.
Falls $B \in \text{Attribute-Closure}((F - \{\alpha \rightarrow \beta\}) \cup \{\alpha \rightarrow (\beta - \{B\})\}, \alpha)$,
dann ersetze $\alpha \rightarrow \beta$ durch $\alpha \rightarrow (\beta - \{B\})$.
3. Löschen aller FDs der Form $(\alpha \rightarrow \emptyset) \in F$.
4. Anwenden der Vereinigungsregel.
Ersetze alle Regeln der Form $\alpha \rightarrow \beta_1, \dots, \alpha \rightarrow \beta_n$ durch $\alpha \rightarrow \beta_1, \dots, \beta_n$.

Bemerkungen:

- In [Elmasri/Navathe 2004] wird das Konzept der *minimalen Überdeckung* definiert und zu dessen Bestimmung ein vergleichbarer Algorithmus vorgestellt. Wesentlicher Unterschied zu der Definition von [Kemper/Eickler 2004] ist, dass statt eindeutiger linker Seiten aller FDs gefordert wird, dass die rechten Seiten aller FDs nur aus einem Attribut bestehen.

Normalformen

Die Normalisierung ist ein formales Werkzeug zur Analyse von Relationenschemata hinsichtlich ihrer funktionalen Abhängigkeiten und Primärschlüssel. Ziele sind:

1. die Minimierung von Redundanzen
2. die Minimierung von Einfüge-, Lösch- und Modifizierungsanomalien

Folgende Normalformen werden zunächst betrachtet:

- ❑ erste Normalform, 1NF
- ❑ zweite Normalform, 2NF
- ❑ dritte Normalform, 3NF
- ❑ Boyce-Codd-Normalform, BCNF

Bemerkungen:

- ❑ Die Normalform eines Schemas \mathcal{R} bezieht sich immer auf die höchste Normalform, die \mathcal{R} einhält.
- ❑ Die Herstellung von einer (hohen) Normalform ist kein hinreichendes Kriterium für ein gutes Datenbankdesign.

Normalformen

Erste Normalform

Ein Relationenschema ist in der ersten Normalform (1NF), wenn der Wertebereich (Domäne) eines Attributes nur atomare Werte enthält. Mengenwertige oder relationenwertige Attributdomänen sind nicht zulässig.

Beispielrelation, die nicht in erster Normalform ist:

Abteilung			
AbtName	<u>AbtNr</u>	ChefPersNr	AbtOrte
Forschung	5	3334	{Bellaire, Sugarland, Houston}
Verwaltung	4	9876	{Stafford}
Stab	1	8886	{Houston}

Normalformen

Erste Normalform

Ein Relationenschema ist in der ersten Normalform (1NF), wenn der Wertebereich (Domäne) eines Attributes nur atomare Werte enthält. Mengenwertige oder relationenwertige Attributdomänen sind nicht zulässig.

Beispielrelation, die nicht in erster Normalform ist:

Abteilung			
AbtName	<u>AbtNr</u>	ChefPersNr	AbtOrte
Forschung	5	3334	{Bellaire, Sugarland, Houston}
Verwaltung	4	9876	{Stafford}
Stab	1	8886	{Houston}

~>

Abteilung			
AbtName	<u>AbtNr</u>	ChefPersNr	<u>AbtOrt</u>
Forschung	5	3334	Bellaire
Forschung	5	3334	Sugarland
Forschung	5	3334	Houston
Verwaltung	4	9876	Stafford
Stab	1	8886	Houston

Normalformen

Erste Normalform

Möglichkeiten zur Herstellung der ersten Normalform:

1. Einführung zusätzlicher Tupel für jeden Wert eines mehrwertigen Attributes.
Nachteil: es wird Redundanz in die ursprüngliche Relation eingebracht
2. Entfernung des mehrwertigen Attributes aus der ursprünglichen Relation und Erzeugung eines weiteren Schemas zusammen mit dem Primärschlüssel der ursprünglichen Relation.

Im Beispiel:

Abteilung		
<u>AbtNr</u>	AbtName	ChefPersNr

Standorte	
<u>AbtNr</u>	<u>AbtOrt</u>

Normalformen

Erste Normalform

Möglichkeiten zur Herstellung der ersten Normalform:

1. Einführung zusätzlicher Tupel für jeden Wert eines mehrwertigen Attributes.
Nachteil: es wird Redundanz in die ursprüngliche Relation eingebracht
2. Entfernung des mehrwertigen Attributes aus der ursprünglichen Relation und Erzeugung eines weiteren Schemas zusammen mit dem Primärschlüssel der ursprünglichen Relation.

Im Beispiel:

Abteilung		
<u>AbtNr</u>	AbtName	ChefPersNr

Standorte	
<u>AbtNr</u>	<u>AbtOrt</u>

3. Falls die Maximalzahl der Werte des mehrwertigen Attributes bekannt sind, Einführung entsprechend vieler atomarer Attribute. Nachteile:
 - ❑ es können viele Nullwerte eingeführt werden
 - ❑ fragwürdige Semantik einer Rangordnung unter diesen Attributen
 - ❑ die Formulierung von Anfragen gestaltet sich schwieriger

Normalformen

Zweite Normalform

Sei \mathcal{R} ein Relationenschema mit FDs F und den Kandidatenschlüsseln $\kappa_1, \dots, \kappa_k$.
 \mathcal{R} ist in zweiter Normalform (2NF), falls jedes Nicht-Schlüsselattribut $A \in \mathcal{R}$, d. h.,
 $A \in (\mathcal{R} - (\kappa_1 \cup \dots \cup \kappa_k))$, voll funktional abhängig von jedem Schlüssel $\kappa_1, \dots, \kappa_k$ ist.

Normalformen

Zweite Normalform

Sei \mathcal{R} ein Relationenschema mit FDs F und den Kandidatenschlüsseln $\kappa_1, \dots, \kappa_k$.
 \mathcal{R} ist in zweiter Normalform (2NF), falls jedes Nicht-Schlüsselattribut $A \in \mathcal{R}$, d. h.,
 $A \in (\mathcal{R} - (\kappa_1 \cup \dots \cup \kappa_k))$, voll funktional abhängig von jedem Schlüssel $\kappa_1, \dots, \kappa_k$ ist.

Alternative Formulierung:

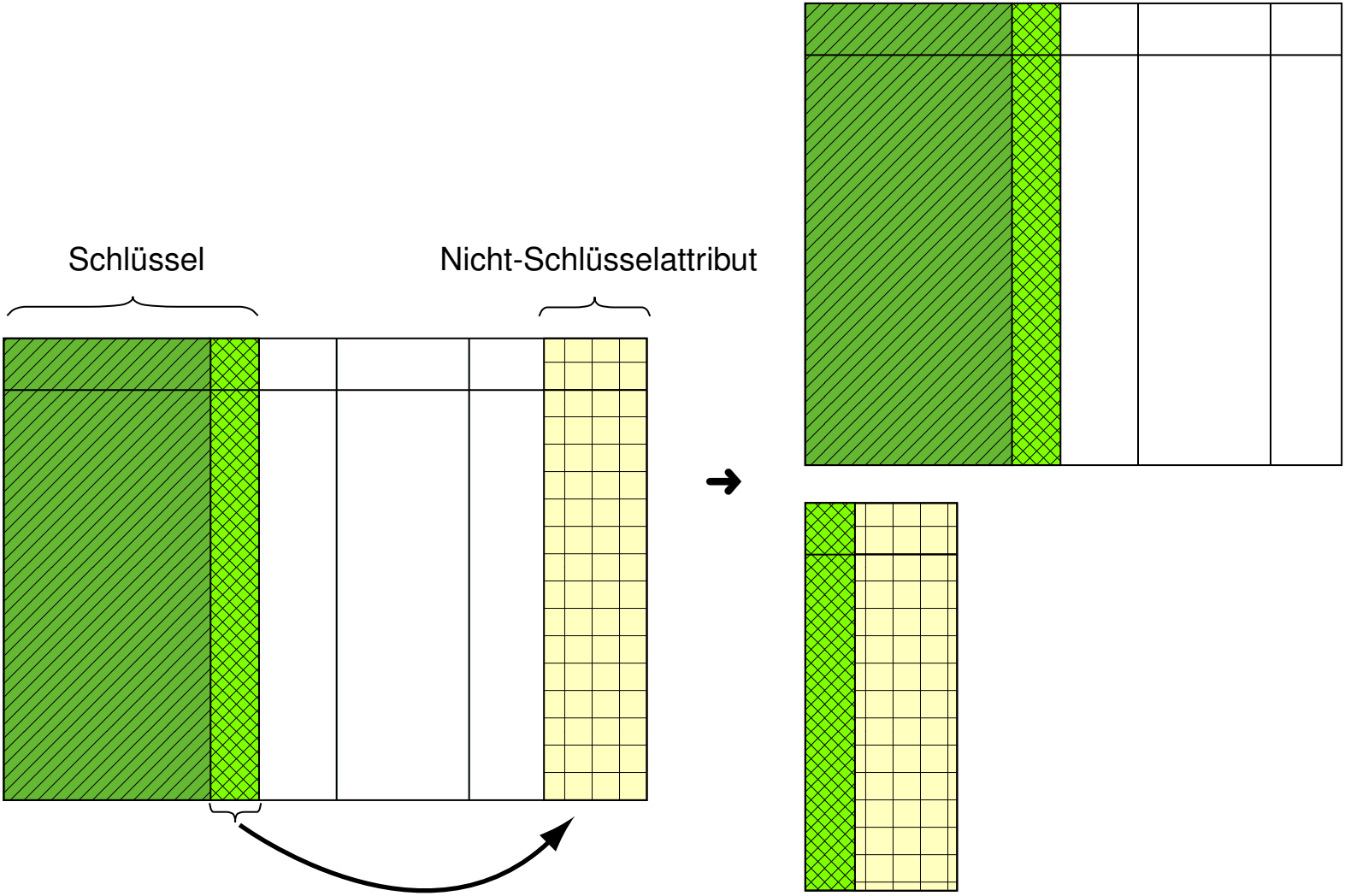
\mathcal{R} ist in zweiter Normalform, falls keine **partiellen** Abhängigkeiten zwischen einem Schlüssel und Nicht-Schlüsselattributen existieren.

Formal:

$$\neg(\exists \kappa \exists \alpha \exists A : \kappa \in \{\kappa_1, \dots, \kappa_k\} \wedge \alpha \subset \kappa \wedge A \in (\mathcal{R} - (\kappa_1 \cup \dots \cup \kappa_k)) : \alpha \rightarrow A)$$

Normalformen

Zweite Normalform



Normalformen

Zweite Normalform

Für ein Relationenschema \mathcal{R} kann die zweite Normalform durch Zerlegung hergestellt werden: Elimination der rechten Seite der partiellen Abhängigkeit in \mathcal{R} . Erstellung eines weiteren Schemas bestehend aus den Attributen der partiellen Abhängigkeit.

Beispielrelation, die nicht in zweiter Normalform ist:

Studentenbelegung			
<u>MatrNr</u>	<u>VorlNr</u>	Name	Semester
26120	5001	Fichte	10
27550	5001	Schopenhauer	6
27550	4052	Schopenhauer	6
28106	5041	Carnap	3
28106	5052	Carnap	3
28106	5216	Carnap	3
28106	5259	Carnap	3



Bemerkungen:

- Im Beispiel gibt es neben den Schlüsselabhängigkeiten die FDs „MatrNr \rightarrow Name“ und „MatrNr \rightarrow Semester“. Folgende Anomalien resultieren:
 1. Einfügeanomalie: Was macht man mit Studenten, die noch keine Vorlesung hören?
 2. Modifizierungsanomalie: Wenn Carnap ins vierte Semester kommt, müssen alle vier Tupel geändert werden.
 3. Löschanomalie: Was passiert, wenn Fichte ihre einzige Vorlesung absagt?
- NF2 ist nicht mit 2NF zu verwechseln. NF2 steht für Non First (NF) Normal Form (NF).

Normalformen

Dritte Normalform

Ein Relationenschema \mathcal{R} ist in dritter Normalform (3NF), wenn für jede nicht-triviale funktionale Abhängigkeit $\alpha \rightarrow A$, die für \mathcal{R} gilt, mindestens eine der folgenden Bedingungen erfüllt ist:

1. α ist Superschlüssel von \mathcal{R}
2. A ist ein Primattribut von \mathcal{R}

Normalformen

Dritte Normalform

Alternative Formulierung 1:

\mathcal{R} ist in dritter Normalform, falls \mathcal{R} in zweiter Normalform ist und keine transitiven Abhängigkeiten zwischen einem Schlüssel und Nicht-Schlüsselattributen existieren.

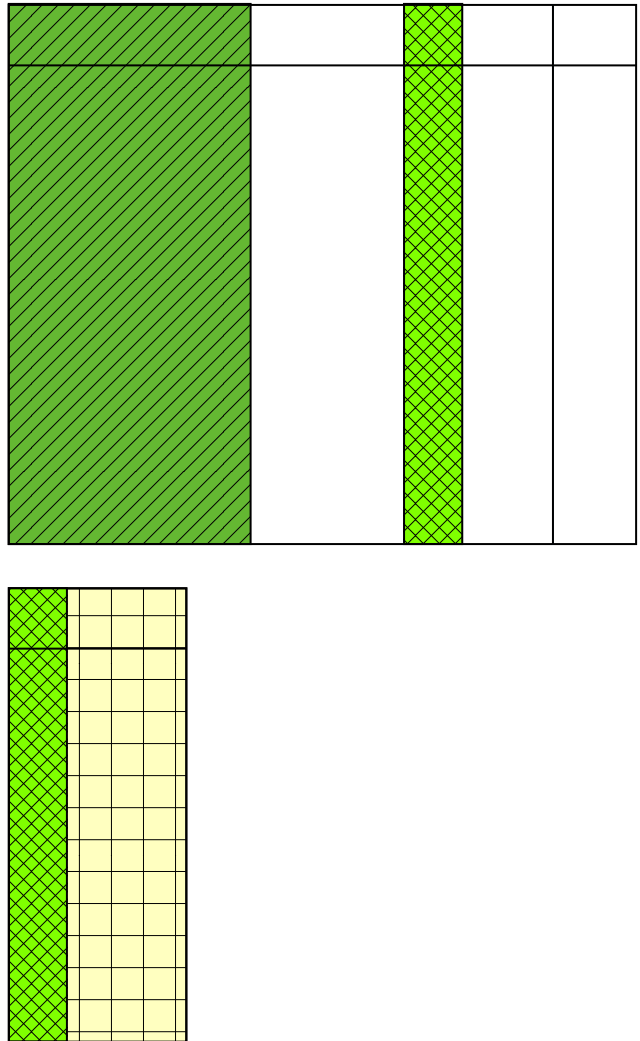
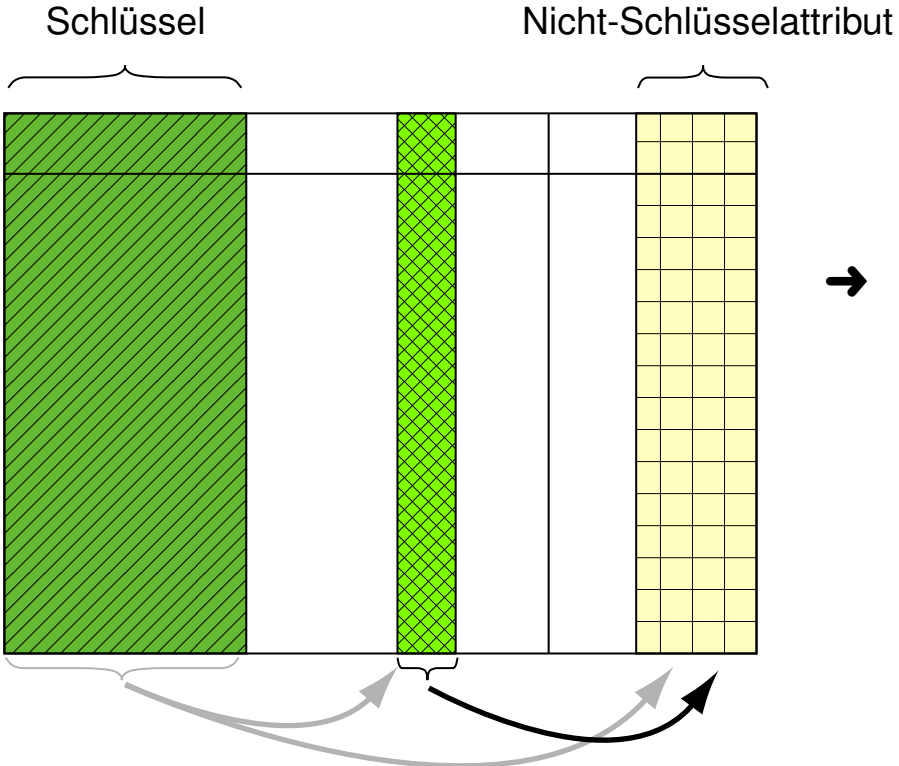
Eine funktionale Abhängigkeit $\alpha \rightarrow \beta$ ist eine transitive Abhängigkeit, falls es eine Menge γ von Attributen gibt, die **weder Schlüssel noch Teilmenge eines Schlüssels** in \mathcal{R} ist, und für die $\alpha \rightarrow \gamma$ und $\gamma \rightarrow \beta$ gilt.

Alternative Formulierung 2:

\mathcal{R} ist in dritter Normalform, falls \mathcal{R} in zweiter Normalform ist und kein Nicht-Schlüsselattribut funktional von einer Menge anderer Nicht-Schlüsselattribute abhängt.

Normalformen

Dritte Normalform



Normalformen

Dritte Normalform

Für ein Relationenschema \mathcal{R} kann die dritte Normalform durch Zerlegung hergestellt werden: Elimination der transitiv abhängigen Attributmenge in \mathcal{R} . Erstellung eines weiteren Schemas bestehend aus der zwischengeschalteten Attributmenge und der transitiv abhängigen Attributmenge.

Beispielrelation, die nicht in dritter Normalform ist:

MitarbeiterAbteilung					
Name	<u>PersNr</u>	Wohnort	AbtNr	AbtName	ChefPersNr
Smith	1234	Weimar	5	Forschung	3334
Wong	3334	Köln	5	Forschung	3334
Zelaya	9998	Erfurt	4	Verwaltung	9876
Wallace	9876	Berlin	4	Verwaltung	9876



Bemerkungen:

- Im Beispiel gibt es die FDs $\text{PersNr} \rightarrow \{\text{Name}, \text{Wohnort}, \text{AbtNr}\}$ und $\text{AbtNr} \rightarrow \{\text{AbtName}, \text{ChefPersNr}\}$. „AbtName“ und „ChefPersNr“ hängen transitiv von „PersNr“ ab. Folge: Der Name und die Chefpersonalnummer für eine Abteilung sind redundant gespeichert.
- Die dritte Normalform wird verletzt, wenn ein Nicht-Schlüsselattribut einen Fakt zu einer Attributmenge darstellt, die keinen Schlüssel bildet. [Bill Kent 1983]
- Beachte, dass die vorgestellten alternativen Formulierungen der dritten Normalform verlangen, dass das Relationenschema \mathcal{R} schon in zweiter Normalform vorliegt.

Normalformen

Boyce-Codd-Normalform [vgl. dritte Normalform]

Ein Relationenschema \mathcal{R} ist in Boyce-Codd-Normalform (BCNF), wenn für jede nicht-triviale funktionale Abhängigkeit $\alpha \rightarrow A$, die für \mathcal{R} gilt, folgende Bedingung erfüllt ist:

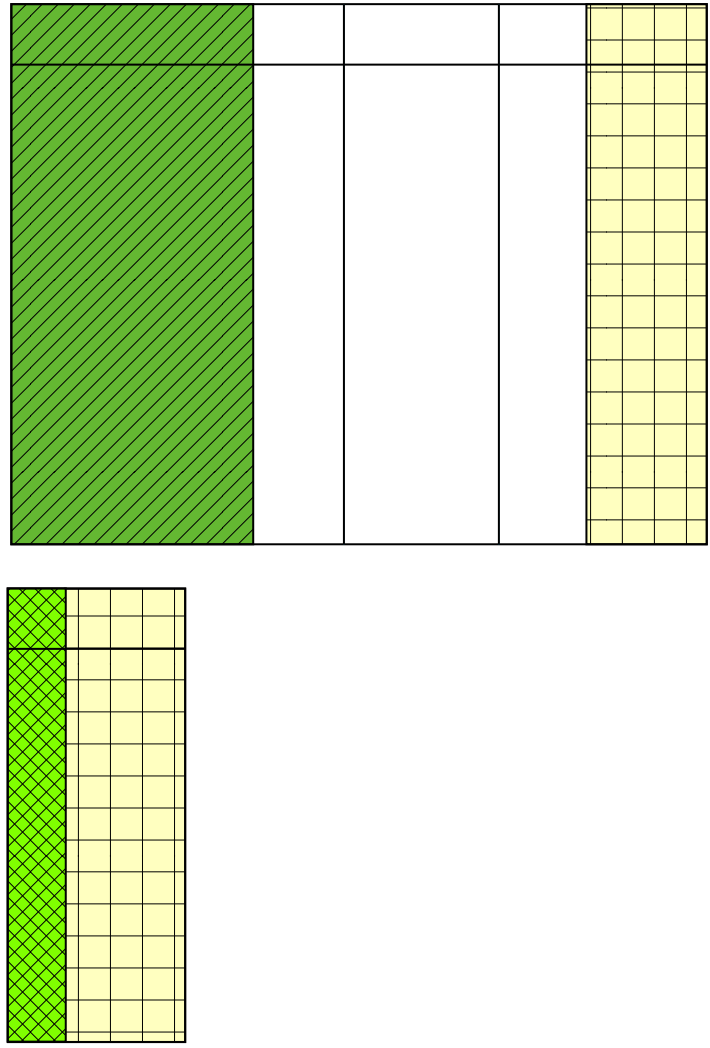
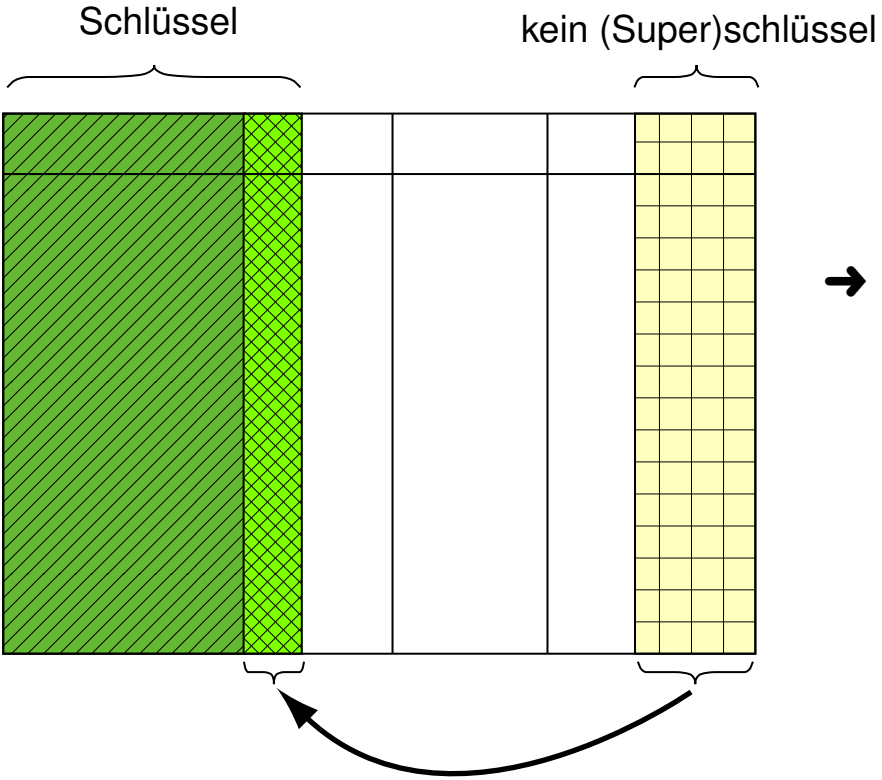
- α ist Superschlüssel von \mathcal{R} .

Alternative Formulierung:

\mathcal{R} ist in Boyce-Codd-Normalform, falls \mathcal{R} in zweiter Normalform ist und keine transitiven Abhängigkeiten zwischen einem Schlüssel und anderen Attributen existieren.

Normalformen

Boyce-Codd-Normalform [vgl. dritte Normalform]



Normalformen

Boyce-Codd-Normalform

Für ein Relationenschema \mathcal{R} kann die Boyce-Codd-Normalform durch Zerlegung hergestellt werden: Elimination der abhängigen Attributmengende in \mathcal{R} . Erstellung eines weiteren Schemas bestehend aus den Attributen der funktionalen Abhängigkeit.

Beispielrelation, die nicht in der Boyce-Codd-Normalform ist:

Grundstuecke			
<u>SteuerNr</u>	Landkreis	GrundstNr	GrundstGroesse

↪

Grundstuecke1		
<u>SteuerNr</u>	GrundstGroesse	GrundstNr

Grundstuecke2	
<u>GrundstGroesse</u>	Landkreis

Bemerkungen:

- Im Beispiel gibt es die beiden Schlüssel $\{\text{SteuerNr}\}$ und $\{\text{Landkreis, GrundstNr}\}$ und die FD $\text{GrundstGroesse} \rightarrow \text{Landkreis}$.
Die Relation „Grundstuecke“ ist nicht in BCNF: GrundstGroesse ist kein Superschlüssel bzw. Landkreis ist transitiv abhängig von SteuerNr. Bei der Zerlegung in die Schemata „Grundstuecke1“ und „Grundstuecke2“ geht die FD $\{\text{Landkreis, GrundstNr}\} \rightarrow \{\text{SteuerNr, GrundstGroesse}\}$ verloren.
- Die Definition der Boyce-Codd-Normalform ist eine Vereinfachung der Definition der dritten Normalform, stellt aber eine strengere Forderung dar. D. h., ein Relationenschema in BCNF ist zwangsläufig in 3NF, aber nicht jedes Relationenschema in 3NF ist auch in BCNF.
- “Each field must represent a fact about the key, the whole key and nothing but the key.”
[Bill Kent 1983]
- Stellt man die BCNF für ein Relationenschema her, das bislang nicht in BCNF ist, so gehen zwangsläufig Abhängigkeiten verloren.

Normalformen

Minimalität

Normalformen vermeiden Redundanz innerhalb einer Relation \mathcal{R} .

Redundanz kann aber auch *global* in einem Datenbankschema $\mathcal{R} = \{\mathcal{R}_1, \dots, \mathcal{R}_p\}$ entstehen, obwohl alle Relationenschemata \mathcal{R}_i eine hohe Normalform erfüllen.

Beispiel:

- $\mathcal{R}_1 = \{\{\underline{A}, B\}, \{\underline{B}, C\}\}$
- $\mathcal{R}_2 = \{\{\underline{A}, B\}, \{\underline{B}, C\}, \{\underline{A}, C\}\}$
- \mathcal{R}_1 und \mathcal{R}_2 sind in dritter Normalform. Die Redundanz in \mathcal{R}_2 kann jedoch zu Inkonsistenzen durch Update-Anomalien führen.

Die Forderung nach Minimalität bedeutet, dass von einer Menge von Datenbankschemata $\mathcal{R}_1, \mathcal{R}_2, \dots, \mathcal{R}_p$, deren Relationen in der gewünschten Normalform sind, dasjenige Schema \mathcal{R}^* gewählt wird, für das gilt:

$$|\mathcal{R}^*| \leq |\mathcal{R}_i|, \quad i = 1, \dots, p$$

Kapitel DB: VII (Fortsetzung)

- I. Einführung und grundlegende Konzepte von Datenbanken
- II. Datenbankentwurf und Datenbankmodelle
- III. Konzeptueller Datenbankentwurf
- IV. Logischer Datenbankentwurf mit dem relationalen Modell
- V. Grundlagen relationaler Anfragesprachen
- VI. Die relationale Datenbanksprache SQL

VII. Entwurfstheorie relationaler Datenbanken

- Informelle Entwurfsziele für Relationenschemata
- Funktionale Abhängigkeiten
- Normalformen
- Dekompositionseigenschaften von Relationen
- Relationale Dekomposition
- Relationale Synthese
- Mehrwertige Abhängigkeiten und höhere Normalformen

Herangehensweisen im Datenbankentwurf

Top-Down versus Bottom-Up

Top-Down:

- Anforderungsanalyse
- Erstellung eines konzeptuellen Schemas/Modells in einem „High-Level-Modell“, z. B. im EER-Modell.
- Abbildung des konzeptuellen Modells auf eine Menge von Relationen.
- Verfeinerung des relationalen Modells.

Bottom-Up:

- Festlegung einer Menge funktionaler Abhängigkeiten
- Synthetisierung von Relationenschemata, für die bestimmte formale Eigenschaften garantiert werden können.

Bemerkungen:

- ❑ Die Top-Down-Herangehensweise wird auch als „Design by Analysis“ bezeichnet.
- ❑ Die Bottom-Up-Herangehensweise als „Design by Synthesis“ bezeichnet.

Dekompositionseigenschaften von Relationen

Unterscheidung des formalen Instrumentariums

1. Eigenschaften einzelner Relationen – insbesondere:
 - (a) Einhaltung einer bestimmten Normalform

2. Dekompositionseigenschaften – insbesondere:
 - (a) Abhängigkeitserhaltung (*dependency preservation*)
 - (b) verlustlose Zerlegung bzw. Verlustlosigkeit (*lossless join property*)

Bemerkungen:

- ❑ Die Theorie der Normalformen ist im Zusammenhang mit beiden Entwurfparadigmen (Top-Down, Bottom-Up) nützlich.
- ❑ Beachte: Die ausschließliche Betrachtung von Normalformen garantiert kein gutes Datenbankdesign.
- ❑ In [Heuer/Saake 2000] werden die Dekompositionseigenschaften als Transformationseigenschaften, die Abhängigkeitserhaltung als Abhängigkeitstreue und die verlustlose Zerlegung als Verbundtreue bezeichnet.

Dekompositionseigenschaften von Relationen

Universalrelation und Dekomposition

Definition 23 (Universalrelation)

Die Universalrelation (genauer: das Universalrelationenschema) zu einer Datenbank entsteht durch Zusammenfassung aller in der Datenbank vorhandenen Attribute in einem einzigen relationalen Schema \mathcal{R} .

Definition 24 (Dekomposition eines Relationenschemas)

Sei \mathcal{R} ein Relationenschema. Die Aufteilung von \mathcal{R} in eine Menge $\mathcal{R} = \{\mathcal{R}_1, \dots, \mathcal{R}_m\}$ von Relationenschemata heißt Dekomposition oder Zerlegung von \mathcal{R} .

Attributerhaltung:

Die Eigenschaft der Attributerhaltung einer Dekomposition \mathcal{R} von \mathcal{R} fordert, dass jedes Attribut von \mathcal{R} in mindestens einem Relationenschemata \mathcal{R}_i , $\mathcal{R}_i \in \mathcal{R}$, auftaucht:

$$\bigcup_{i=1}^m \mathcal{R}_i = \mathcal{R}$$

Dekompositionseigenschaften von Relationen

Abhängigkeitserhaltung

Eine Zerlegung von \mathcal{R} mit zugehörigen funktionalen Abhängigkeiten F in die Relationenschemata $\mathcal{R}_1, \dots, \mathcal{R}_m$ sollte so erfolgen, dass die Überprüfung aller funktionaler Abhängigkeiten *lokal* auf den \mathcal{R}_i erfolgen kann – also ohne dass Join-Operationen notwendig sind. Das heißt, jede FD $(\alpha \rightarrow \beta) \in F$

- kommt entweder direkt in einem \mathcal{R}_i vor oder
- kann mit Hilfe der Inferenzregeln abgeleitet werden.

Dekompositionseigenschaften von Relationen

Abhängigkeitserhaltung

Eine Zerlegung von \mathcal{R} mit zugehörigen funktionalen Abhängigkeiten F in die Relationenschemata $\mathcal{R}_1, \dots, \mathcal{R}_m$ sollte so erfolgen, dass die Überprüfung aller funktionaler Abhängigkeiten *lokal* auf den \mathcal{R}_i erfolgen kann – also ohne dass Join-Operationen notwendig sind. Das heißt, jede FD $(\alpha \rightarrow \beta) \in F$

- kommt entweder direkt in einem \mathcal{R}_i vor oder
- kann mit Hilfe der Inferenzregeln abgeleitet werden.

Definition 25 (Einschränkung von FDs, Abhängigkeitserhaltung)

Sei \mathcal{R} ein Relationenschema mit zugehörigen funktionalen Abhängigkeiten F und $\mathcal{R}_i \subseteq \mathcal{R}$. Die Einschränkung von F auf \mathcal{R}_i , in Zeichen: $F_{\mathcal{R}_i}$, ist die Menge von Abhängigkeiten $(\alpha \rightarrow \beta) \in F^+$ für die $(\alpha \cup \beta) \subseteq \mathcal{R}_i$ gilt.

Eine Dekomposition $\mathcal{R} = \{\mathcal{R}_1, \dots, \mathcal{R}_m\}$ von \mathcal{R} ist abhängigkeiterhaltend hinsichtlich F , wenn gilt:

$$F \equiv F_{\mathcal{R}_1} \cup \dots \cup F_{\mathcal{R}_m} \quad \text{bzw.} \quad F^+ = (F_{\mathcal{R}_1} \cup \dots \cup F_{\mathcal{R}_m})^+$$

Bemerkungen:

- Abhängigkeitserhaltung garantiert die effiziente Überprüfung von funktionalen Abhängigkeiten.
- Eine abhängigkeitserhaltende Dekomposition wird auch *hüllentreue* Dekomposition genannt.
- Für ein Relationenschema \mathcal{R} kann immer eine abhängigkeitserhaltende Dekomposition $\mathcal{R} = \{\mathcal{R}_1, \dots, \mathcal{R}_m\}$ gefunden werden, so dass jedes $\mathcal{R}_i, \mathcal{R}_i \in \mathcal{R}$, in 3NF ist. Es kann nicht immer eine Dekomposition gefunden werden, die alle Abhängigkeiten erhält und bei der jedes \mathcal{R}_i in BCNF ist.

Dekompositionseigenschaften von Relationen

Abhängigkeitserhaltung: Beispiel

PLZverzeichnis

<u>Ort</u>	<u>BLand</u>	<u>Strasse</u>	<u>PLZ</u>
Frankfurt	Hessen	Goethestrasse	60313
Frankfurt	Hessen	Schillerstrasse	60437
Frankfurt	Brandenburg	Goethestrasse	15234

Abhängigkeiten:

$\{\text{PLZ}\} \rightarrow \{\text{Ort, BLand}\}$

$\{\text{Ort, BLand, Strasse}\} \rightarrow \{\text{PLZ}\}$

Dekompositionseigenschaften von Relationen

Abhängigkeitserhaltung: Beispiel

PLZverzeichnis			
<u>Ort</u>	<u>BLand</u>	<u>Strasse</u>	<u>PLZ</u>
Frankfurt	Hessen	Goethestrasse	60313
Frankfurt	Hessen	Schillerstrasse	60437
Frankfurt	Brandenburg	Goethestrasse	15234

Abhängigkeiten:

$\{PLZ\} \rightarrow \{Ort, BLand\}$

$\{Ort, BLand, Strasse\} \rightarrow \{PLZ\}$

Durch die Zerlegung geht die FD $\{Ort, BLand, Strasse\} \rightarrow \{PLZ\}$ verloren:

Strassen	
<u>PLZ</u>	<u>Strasse</u>
60313	Goethestrasse
60473	Schillerstrasse
15234	Goethestrasse

Orte		
<u>Ort</u>	<u>BLand</u>	<u>PLZ</u>
Frankfurt	Hessen	60313
Frankfurt	Hessen	60437
Frankfurt	Brandenburg	15234

Dekompositionseigenschaften von Relationen

Abhängigkeitserhaltung: Beispiel

PLZverzeichnis			
<u>Ort</u>	<u>BLand</u>	<u>Strasse</u>	<u>PLZ</u>
Frankfurt	Hessen	Goethestrasse	60313
Frankfurt	Hessen	Schillerstrasse	60437
Frankfurt	Brandenburg	Goethestrasse	15234

Abhängigkeiten:

$\{PLZ\} \rightarrow \{Ort, BLand\}$

$\{Ort, BLand, Strasse\} \rightarrow \{PLZ\}$

Durch die Zerlegung geht die FD $\{Ort, BLand, Strasse\} \rightarrow \{PLZ\}$ verloren:

Strassen	
<u>PLZ</u>	<u>Strasse</u>
60313	Goethestrasse
60473	Schillerstrasse
15234	Goethestrasse

Orte		
<u>Ort</u>	<u>BLand</u>	<u>PLZ</u>
Frankfurt	Hessen	60313
Frankfurt	Hessen	60437
Frankfurt	Brandenburg	15234

Lokal konsistentes Update der Relationen:

Strassen	
<u>PLZ</u>	<u>Strasse</u>
60313	Goethestrasse
60473	Schillerstrasse
15234	Goethestrasse
15235	Goethestrasse

Orte		
<u>Ort</u>	<u>BLand</u>	<u>PLZ</u>
Frankfurt	Hessen	60313
Frankfurt	Hessen	60437
Frankfurt	Brandenburg	15234
Frankfurt	Brandenburg	15235

Bemerkungen:

- Die Zerlegung ist nicht abhängigkeiterhaltend: Die Verletzung der FD $\{\text{Ort, BLand, Strasse}\} \rightarrow \{\text{PLZ}\}$ ist erst nach einem Join erkennbar.
- Weil $\{\text{PLZ}\}$ einen Schlüssel in einer der beiden neuen Relationen darstellt, ist die Zerlegung verlustlos.
- Das Relationenschema „Strassen“ enthält nur noch triviale Abhängigkeiten; der Schlüssel von „Strassen“ besteht deshalb aus der Menge aller Attribute der Relation.

Dekompositionseigenschaften von Relationen

Verlustlose Zerlegung

Eine Zerlegung von \mathcal{R} mit zugehörigen funktionalen Abhängigkeiten F in die Relationenschemata $\mathcal{R}_1, \dots, \mathcal{R}_m$ sollte so erfolgen, dass die in einer Ausprägung r des Relationenschemas \mathcal{R} enthaltene Information aus den Ausprägungen r_1, \dots, r_m der Relationenschemata $\mathcal{R}_1, \dots, \mathcal{R}_m$ konstruierbar ist.

Definition 26 (verlustlose Zerlegung)

Sei \mathcal{R} ein Relationenschema mit zugehörigen funktionalen Abhängigkeiten F . Eine Zerlegung $\mathcal{R} = \{\mathcal{R}_1, \dots, \mathcal{R}_m\}$ von \mathcal{R} ist verlustlos hinsichtlich F , wenn für jede Ausprägung r des Relationenschemas \mathcal{R} , die F erfüllt, gilt:

$$r = \pi_{\mathcal{R}_1}(r) \bowtie \dots \bowtie \pi_{\mathcal{R}_m}(r)$$

Bemerkungen:

- Eine Zerlegung ist dann nicht verlustlos, wenn nach dem Join zusätzliche Tupel entstanden sind. Zusätzliche Tupel bedeuten einen Informationsverlust, weil durch sie eindeutige Zuordnungen verloren gegangen sind.
- Oft ist es ausreichend, sich bei der Dekomposition von \mathcal{R} auf den binären Fall zu beschränken. Es gilt nämlich: Ist $\{\mathcal{R}_1, \dots, \mathcal{R}_m\}$ eine verlustlose Zerlegung von \mathcal{R} hinsichtlich F und ist $\{\mathcal{R}_{i_1}, \dots, \mathcal{R}_{i_k}\}$ eine verlustlose Zerlegung von \mathcal{R}_i hinsichtlich $F_{\mathcal{R}_i}$, so ist auch $\{\mathcal{R}_1, \dots, \mathcal{R}_{i-1}, \mathcal{R}_{i_1}, \dots, \mathcal{R}_{i_k}, \mathcal{R}_{i+1}, \dots, \mathcal{R}_m\}$ eine verlustlose Zerlegung von \mathcal{R} hinsichtlich F .

Dekompositionseigenschaften von Relationen

Verlustlose Zerlegung: Beispiel

Biertrinker		
<u>Kneipe</u>	<u>Gast</u>	Bier
Kowalski	Kemper	Pils
Kowalski	Eickler	Hefeweizen
Innsteg	Kemper	Hefeweizen

Zerlegung in die Relationenschemata „Besucht“ und „Trinkt“:

Besucht	
<u>Kneipe</u>	<u>Gast</u>
Kowalski	Kemper
Kowalski	Eickler
Innsteg	Kemper

Trinkt	
<u>Gast</u>	<u>Bier</u>
Kemper	Pils
Eickler	Hefeweizen
Kemper	Hefeweizen

Dekompositionseigenschaften von Relationen

Verlustlose Zerlegung: Beispiel

Biertrinker		
<u>Kneipe</u>	<u>Gast</u>	Bier
Kowalski	Kemper	Pils
Kowalski	Eickler	Hefeweizen
Innsteg	Kemper	Hefeweizen

Zerlegung in die Relationenschemata „Besucht“ und „Trinkt“:

Besucht	
<u>Kneipe</u>	<u>Gast</u>
Kowalski	Kemper
Kowalski	Eickler
Innsteg	Kemper

Trinkt	
<u>Gast</u>	<u>Bier</u>
Kemper	Pils
Eickler	Hefeweizen
Kemper	Hefeweizen

Die Bildung des natürlichen Verbundes zeigt, dass die Assoziation von Biersorten und Gästen relativ zur Kneipe verloren gegangen ist:

Besucht	
<u>Kneipe</u>	<u>Gast</u>
Kowalski	Kemper
Kowalski	Eickler
Innsteg	Kemper

⋈

Trinkt	
<u>Gast</u>	<u>Bier</u>
Kemper	Pils
Eickler	Hefeweizen
Kemper	Hefeweizen

≈

Besucht ⋈ Trinkt		
<u>Kneipe</u>	<u>Gast</u>	<u>Bier</u>
Kowalski	Kemper	Pils
Kowalski	Kemper	Hefeweizen
Kowalski	Eickler	Hefeweizen
Innsteg	Kemper	Pils
Innsteg	Kemper	Hefeweizen

*

*

Bemerkungen:

- Im Beispiel gilt nur die FD $\{\text{Kneipe, Gast}\} \rightarrow \{\text{Bier}\}$.
- Die Existenz von mindestens einer der folgenden FDs würde Verlustlosigkeit garantieren:
 $\{\text{Gast}\} \rightarrow \{\text{Bier}\}$, $\{\text{Gast}\} \rightarrow \{\text{Kneipe}\}$

Dekompositionseigenschaften von Relationen

Verlustlose Zerlegung (Fortsetzung)

Eine Zerlegung von \mathcal{R} mit zugehörigen funktionalen Abhängigkeiten F in die Relationenschemata \mathcal{R}_1 und \mathcal{R}_2 ist verlustlos, wenn mindestens eine der folgenden Bedingungen gilt:

- $((\mathcal{R}_1 \cap \mathcal{R}_2) \rightarrow \mathcal{R}_1) \in F^+$
- $((\mathcal{R}_1 \cap \mathcal{R}_2) \rightarrow \mathcal{R}_2) \in F^+$

Dekompositionseigenschaften von Relationen

Verlustlose Zerlegung (Fortsetzung)

Eine Zerlegung von \mathcal{R} mit zugehörigen funktionalen Abhängigkeiten F in die Relationenschemata \mathcal{R}_1 und \mathcal{R}_2 ist verlustlos, wenn mindestens eine der folgenden Bedingungen gilt:

- $((\mathcal{R}_1 \cap \mathcal{R}_2) \rightarrow \mathcal{R}_1) \in F^+$
- $((\mathcal{R}_1 \cap \mathcal{R}_2) \rightarrow \mathcal{R}_2) \in F^+$

Alternative Formulierung:

Sei $\mathcal{R} = \alpha \cup \beta \cup \gamma$, $\mathcal{R}_1 = \alpha \cup \beta$, und $\mathcal{R}_2 = \alpha \cup \gamma$ mit paarweisen disjunkten Attributmengen α , β und γ . Dann muss mindestens eine der folgenden Bedingungen gelten:

- $\beta \subseteq \text{Attribute-Closure}(F, \alpha)$
- $\gamma \subseteq \text{Attribute-Closure}(F, \alpha)$

Relationale Dekomposition

Algorithm: Rel-Decomposition

Input: \mathcal{R} . Universalrelation.

F . Menge funktionaler Abhängigkeiten für \mathcal{R} .

Output: \mathcal{R} . Verlustlose Zerlegung von \mathcal{R} mit Schemata in BCNF.

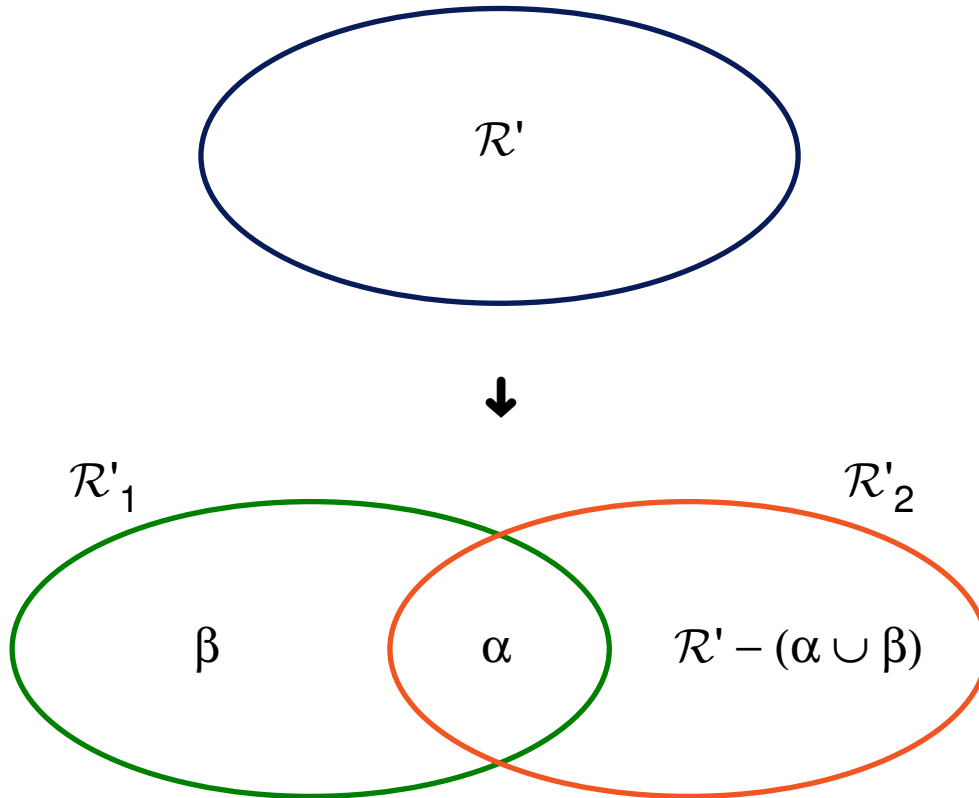
1. $\mathcal{R} = \{\mathcal{R}\}$
2. **WHILE** $\exists \mathcal{R}' : (\mathcal{R}' \in \mathcal{R} \wedge \mathcal{R}' \text{ not in BCNF})$ **DO**
3. Find FD $(\alpha \rightarrow \beta) \in F_{\mathcal{R}'}$ that violates BCNF
4. Decompose \mathcal{R}' into $\mathcal{R}'_1 = \alpha \cup \beta$ and $\mathcal{R}'_2 = \mathcal{R}' - \beta$
5. $\mathcal{R} = (\mathcal{R} - \{\mathcal{R}'\}) \cup \{\mathcal{R}'_1\} \cup \{\mathcal{R}'_2\}$
6. **ENDDO**
7. **RETURN**(\mathcal{R})

Bemerkungen:

- ❑ Wie man eine BCNF-verletzende FD in der While-Schleife findet: Ist ein Relationenschema \mathcal{R} nicht in BCNF, so existiert in \mathcal{R} eine FD $\alpha \rightarrow \beta$ mit $\alpha \cap \beta = \emptyset$ und $\alpha \not\rightarrow \mathcal{R}$.
- ❑ Die Dekomposition in Schritt 4 garantiert Verlustlosigkeit.
- ❑ Die durch den Algorithmus Rel-Decomposition erzeugte Dekomposition ist nicht notwendigerweise abhängigkeiterhaltend.

Relationale Dekomposition

Illustration der Zerlegung eines Relationenschemas \mathcal{R}' in die Schemata \mathcal{R}'_1 und \mathcal{R}'_2 entlang der funktionalen Abhängigkeit $\alpha \rightarrow \beta$:



[Kemper/Eickler 2004]

Relationale Dekomposition

Beispiel:

- $\mathcal{R}_{\text{Grundstuecke}} = \{\underline{\text{SteuerNr}}, \text{Landkreis}, \text{GrundstNr}, \text{GrundstGroesse}, \text{Preis}, \text{Steuersatz}\}$
- FD_1 ergibt sich aus dem Schlüssel.
- $FD_2: \{\text{Landkreis}, \text{GrundstNr}\} \rightarrow \{\text{SteuerNr}, \text{GrundstGroesse}, \text{Preis}, \text{Steuersatz}\}$
- $FD_3: \{\text{Landkreis}\} \rightarrow \{\text{Steuersatz}\}$
- $FD_4: \{\text{GrundstGroesse}\} \rightarrow \{\text{Preis}\}$

\rightsquigarrow TAFEL

Relationale Synthese

Algorithm: Rel-Synthesis

Input: \mathcal{R} . Universalrelation.

F . Menge funktionaler Abhängigkeiten für \mathcal{R} .

Output: \mathcal{R} . Verlustlose **und** abhängigkeiterhaltende Zerlegung von \mathcal{R} mit Schemata in 3NF.

1. $\mathcal{R} = \emptyset$
2. Determine a canonical cover F_c of F
- 3.
- 4.
- 5.
- 6.
- 7.
- 8.
- 9.
- 10.
- 11.
12. **RETURN**(\mathcal{R})

Relationale Synthese

Algorithm: Rel-Synthesis

Input: \mathcal{R} . Universalrelation.

F . Menge funktionaler Abhängigkeiten für \mathcal{R} .

Output: \mathcal{R} . Verlustlose **und** abhängigkeiterhaltende Zerlegung von \mathcal{R} mit Schemata in 3NF.

1. $\mathcal{R} = \emptyset$
2. Determine a canonical cover F_c of F
3. **FOREACH** $(\alpha \rightarrow \beta) \in F_c$ **DO**
4. Synthesize $\mathcal{R}_\alpha = \alpha \cup \beta$, $\mathcal{R} = \mathcal{R} \cup \{\mathcal{R}_\alpha\}$
5. $F_\alpha = \{(\alpha' \rightarrow \beta') \in F_c \mid (\alpha' \cup \beta') \subseteq \mathcal{R}_\alpha\}$
6. **ENDDO**
- 7.
- 8.
- 9.
- 10.
- 11.
12. **RETURN**(\mathcal{R})

Relationale Synthese

Algorithm: Rel-Synthesis

Input: \mathcal{R} . Universalrelation.

F . Menge funktionaler Abhängigkeiten für \mathcal{R} .

Output: \mathcal{R} . Verlustlose **und** abhängigkeiterhaltende Zerlegung von \mathcal{R} mit Schemata in 3NF.

1. $\mathcal{R} = \emptyset$
2. Determine a canonical cover F_c of F
3. **FOREACH** $(\alpha \rightarrow \beta) \in F_c$ **DO**
4. Synthesize $\mathcal{R}_\alpha = \alpha \cup \beta$, $\mathcal{R} = \mathcal{R} \cup \{\mathcal{R}_\alpha\}$
5. $F_\alpha = \{(\alpha' \rightarrow \beta') \in F_c \mid (\alpha' \cup \beta') \subseteq \mathcal{R}_\alpha\}$
6. **ENDDO**
7. **IF** $\nexists \alpha : ((\alpha \rightarrow \beta) \in F_c \text{ with } \alpha \rightarrow \mathcal{R})$ **THEN**
8. Determine a candidate key $\kappa \subseteq \mathcal{R}$
9. $\mathcal{R}_\kappa = \kappa$, $\mathcal{R} = \mathcal{R} \cup \{\mathcal{R}_\kappa\}$, $F_\kappa = \emptyset$
10. **ENDIF**
- 11.
12. **RETURN**(\mathcal{R})

Relationale Synthese

Algorithm: Rel-Synthesis

Input: \mathcal{R} . Universalrelation.

F . Menge funktionaler Abhängigkeiten für \mathcal{R} .

Output: \mathcal{R} . Verlustlose **und** abhängigkeiterhaltende Zerlegung von \mathcal{R} mit Schemata in 3NF.

1. $\mathcal{R} = \emptyset$
2. Determine a canonical cover F_c of F
3. **FOREACH** $(\alpha \rightarrow \beta) \in F_c$ **DO**
4. Synthesize $\mathcal{R}_\alpha = \alpha \cup \beta$, $\mathcal{R} = \mathcal{R} \cup \{\mathcal{R}_\alpha\}$
5. $F_\alpha = \{(\alpha' \rightarrow \beta') \in F_c \mid (\alpha' \cup \beta') \subseteq \mathcal{R}_\alpha\}$
6. **ENDDO**
7. **IF** $\nexists \alpha : ((\alpha \rightarrow \beta) \in F_c \text{ with } \alpha \rightarrow \mathcal{R})$ **THEN**
8. Determine a candidate key $\kappa \subseteq \mathcal{R}$
9. $\mathcal{R}_\kappa = \kappa$, $\mathcal{R} = \mathcal{R} \cup \{\mathcal{R}_\kappa\}$, $F_\kappa = \emptyset$
10. **ENDIF**
11. **FOREACH** $\mathcal{R}_\alpha, \mathcal{R}_{\alpha'} \in \mathcal{R}$ **DO** **IF** $\mathcal{R}_\alpha \subseteq \mathcal{R}_{\alpha'}$ **THEN** $\mathcal{R} = \mathcal{R} - \{\mathcal{R}_\alpha\}$
12. **RETURN**(\mathcal{R})

Bemerkungen:

- ❑ Schritt 2: Bestimmung einer kanonischen Überdeckung für die Menge der funktionalen Abhängigkeiten.

Schritt 3-6: Synthetisierung der Relationenschemata, die sich in 3NF befinden.

Schritt 7-10: Überprüfung, ob eines der generierten Relationenschemata einen Schlüssel für die Universalrelation enthält. Falls nicht, ist ein solcher Schlüssel zu bestimmen und ein aus den Schlüsselattributen bestehendes Relationenschema hinzuzunehmen.

Schritt 11: Eliminierung der subsummierten Relationenschemata.

- ❑ Es stellt sich u. a. die Frage, wie wertvoll die Eigenschaft der Verlustlosigkeit ist [vgl. Heuer/Saake 2000]:
 - Muss sich die Universalrelation rekonstruieren lassen bzw. hat sie eine aus einer Anwendung stammende zwingende Semantik?
 - Ist der natürliche Verbund als Rekonstruktionsoperator das Maß aller Dinge?

Relationale Synthese

Beispiel:

- $\mathcal{R}_{\text{MitarbeiterAdr}} = \{\underline{\text{PersNr}}, \text{Name}, \text{Gehaltsstufe}, \text{Raum}, \text{Ort}, \text{Strasse}, \text{PLZ}, \text{Vorwahl}, \text{BLand}\}$
- FD_1 ergibt sich aus dem Schlüssel.
- $FD_2: \{\text{Ort}, \text{BLand}\} \rightarrow \{\text{Vorwahl}\}$
- $FD_3: \{\text{PLZ}\} \rightarrow \{\text{BLand}, \text{Ort}\}$
- $FD_4: \{\text{Ort}, \text{BLand}, \text{Strasse}\} \rightarrow \{\text{PLZ}\}$
- $FD_5: \{\text{Raum}\} \rightarrow \{\text{PersNr}\}$

\leadsto *TAFEL*

Mehrwertige Abhängigkeiten

Definition 27 (mehrwertig abhängig)

Sei \mathcal{R} ein relationales Schema und gelte $\alpha, \beta, \gamma \subseteq \mathcal{R}$ mit $\alpha \cup \beta \cup \gamma = \mathcal{R}$. Dann ist β mehrwertig abhängig von α , in Zeichen: $\alpha \twoheadrightarrow \beta$, wenn in jeder gültigen Ausprägung von \mathcal{R} gilt: Für jedes Paar von Tupeln, t_1, t_2 , mit $t_1(\alpha) = t_2(\alpha)$ existieren zwei weitere Tupel t_3 und t_4 mit folgenden Eigenschaften:

$$t_1(\alpha) = t_2(\alpha) = t_3(\alpha) = t_4(\alpha)$$

$$t_1(\beta) = t_3(\beta)$$

$$t_2(\beta) = t_4(\beta)$$

$$t_1(\gamma) = t_4(\gamma)$$

$$t_2(\gamma) = t_3(\gamma)$$

Mehrwertige Abhängigkeiten

Definition 27 (mehrwertig abhängig)

Sei \mathcal{R} ein relationales Schema und gelte $\alpha, \beta, \gamma \subseteq \mathcal{R}$ mit $\alpha \cup \beta \cup \gamma = \mathcal{R}$. Dann ist β mehrwertig abhängig von α , in Zeichen: $\alpha \twoheadrightarrow \beta$, wenn in jeder gültigen Ausprägung von \mathcal{R} gilt: Für jedes Paar von Tupeln, t_1, t_2 , mit $t_1(\alpha) = t_2(\alpha)$ existieren zwei weitere Tupel t_3 und t_4 mit folgenden Eigenschaften:

$$t_1(\alpha) = t_2(\alpha) = t_3(\alpha) = t_4(\alpha)$$

$$t_1(\beta) = t_3(\beta)$$

$$t_2(\beta) = t_4(\beta)$$

$$t_1(\gamma) = t_4(\gamma)$$

$$t_2(\gamma) = t_3(\gamma)$$

Alternative Darstellung (gleiche Farbe impliziert gleichen Wert):

$$t_1(\alpha) \quad t_1(\beta) \quad t_1(\gamma)$$

$$t_3(\alpha) \quad t_3(\beta) \quad t_3(\gamma)$$

$$t_4(\alpha) \quad t_4(\beta) \quad t_4(\gamma)$$

$$t_2(\alpha) \quad t_2(\beta) \quad t_2(\gamma)$$

Bemerkungen:

- Ist β mehrwertig abhängig von α , so kann man in der zugehörigen Relation r bei je zwei Tupeln, die den gleichen α -Wert haben, die β -Werte vertauschen und es gilt, dass die resultierenden Tupel auch in r enthalten sind.
- Eine MVD kann immer dann entstehen, wenn zwei unabhängige 1:N-Beziehungen in *einer* Relation kombiniert werden. Die Beziehungen können als orthogonal zueinander (= unabhängig voneinander) verstanden werden.
- Mehrwertige Abhängigkeiten (*multivalued dependencies, MVDs*) stellen eine Verallgemeinerung funktionaler Abhängigkeiten (FDs) dar. Es gilt also:

$$\alpha \rightarrow \beta \Rightarrow \alpha \twoheadrightarrow \beta$$

Eine FD ist eine MVD, bei der höchstens eine Ausprägung von β mit je einer Ausprägung von α verknüpft ist.

- Aus der Symmetrie der Definition folgt auch die *Komplementregel*:

$$\alpha \twoheadrightarrow \beta \Rightarrow \alpha \twoheadrightarrow \gamma$$

Mehrwertige Abhängigkeiten

Beispiel:

Faehigkeiten		
PersNr	Sprache	ProgSprache
3002	griechisch	C
3002	lateinisch	Pascal
3002	griechisch	Pascal
3002	lateinisch	C
3005	deutsch	Java

- In dieser Relation gelten die MVDs $\{\text{PersNr}\} \twoheadrightarrow \{\text{Sprache}\}$ und $\{\text{PersNr}\} \twoheadrightarrow \{\text{ProgSprache}\}$.
- Die Attributmengende $\{\text{PersNr}, \text{Sprache}, \text{ProgSprache}\}$ bildet den Schlüssel; die Relation ist also in BCNF.

Mehrwertige Abhängigkeiten

Sei \mathcal{R} ein relationales Schema und gelte $\alpha, \beta, \gamma \subseteq \mathcal{R}$ mit $\alpha \cup \beta \cup \gamma = \mathcal{R}$ und sei r eine Auprägung von \mathcal{R} .

Semantisch drückt eine mehrwertige Abhängigkeit die **Unabhängigkeit** der Attributmengen β und γ voneinander in der Relation r aus: Pro α -Wert bildet das kartesische Produkt der β - und γ -Werte die Menge der $\beta\gamma$ -Werte für α :

$$\begin{aligned} \alpha \twoheadrightarrow \beta \\ \Leftrightarrow \\ \forall a \in \{t(\alpha) \mid t \in r\} : \pi_{\beta\gamma} &= \pi_{\beta} \quad \times \pi_{\gamma} \end{aligned}$$

Mehrwertige Abhängigkeiten

Sei \mathcal{R} ein relationales Schema und gelte $\alpha, \beta, \gamma \subseteq \mathcal{R}$ mit $\alpha \cup \beta \cup \gamma = \mathcal{R}$ und sei r eine Auprägung von \mathcal{R} .

Semantisch drückt eine mehrwertige Abhängigkeit die **Unabhängigkeit** der Attributmengen β und γ voneinander in der Relation r aus: Pro α -Wert bildet das kartesische Produkt der β - und γ -Werte die Menge der $\beta\gamma$ -Werte für α :

$$\alpha \twoheadrightarrow \beta$$

$$\Leftrightarrow$$

$$\forall a \in \{t(\alpha) \mid t \in r\} : \pi_{\beta\gamma}(\sigma_{\alpha=a}(r)) = \pi_{\beta}(\sigma_{\alpha=a}(r)) \times \pi_{\gamma}(\sigma_{\alpha=a}(r))$$

Mehrwertige Abhängigkeiten

Sei \mathcal{R} ein relationales Schema und gelte $\alpha, \beta, \gamma \subseteq \mathcal{R}$ mit $\alpha \cup \beta \cup \gamma = \mathcal{R}$ und sei r eine Auprägung von \mathcal{R} .

Semantisch drückt eine mehrwertige Abhängigkeit die **Unabhängigkeit** der Attributmengen β und γ voneinander in der Relation r aus: Pro α -Wert bildet das kartesische Produkt der β - und γ -Werte die Menge der $\beta\gamma$ -Werte für α :

$$\alpha \twoheadrightarrow \beta$$
$$\Leftrightarrow$$

$$\forall a \in \{t(\alpha) \mid t \in r\} : \pi_{\beta\gamma}(\sigma_{\alpha=a}(r)) = \pi_{\beta}(\sigma_{\alpha=a}(r)) \times \pi_{\gamma}(\sigma_{\alpha=a}(r))$$

Alternative Formulierung für den Spezialfall $|\alpha| = |\beta| = |\gamma| = 1$:

Wenn $\{b_1, \dots, b_k\}$ und $\{c_1, \dots, c_l\}$ die β - bzw. γ -Werte für einen bestimmten α -Wert a in einer Relation r sind, so muss r auch die folgenden $k \cdot l$ Tripel enthalten:

$$\{a\} \times \{b_1, \dots, b_k\} \times \{c_1, \dots, c_l\}$$

Bemerkungen:

- Immer wenn zwei Tupel mit gleichem α -Wert und verschiedenem β -Wert existieren, so müssen auch Tupel existieren, die für diesen α -Wert und jeden β -Wert alle Kombinationen von γ -Werten beinhalten.

Aus diesem Grund nennt man mehrwertige Abhängigkeiten auch „Tupel-generierende“ Abhängigkeiten: Eine Relationenausprägung kann bei Verletzung einer MVD durch das Einfügen zusätzlicher Tupel in einen gültigen Zustand überführt werden.

Mehrwertige Abhängigkeiten

Vierte Normalform

Definition 28 (triviale MVD)

Sei \mathcal{R} ein relationales Schema, $\alpha \cup \beta \subseteq \mathcal{R}$. Eine MVD $\alpha \twoheadrightarrow \beta$ ist trivial hinsichtlich \mathcal{R} , falls jede mögliche Ausprägung r von \mathcal{R} diese MVD erfüllt.

Mehrwertige Abhängigkeiten

Vierte Normalform

Definition 28 (triviale MVD)

Sei \mathcal{R} ein relationales Schema, $\alpha \cup \beta \subseteq \mathcal{R}$. Eine MVD $\alpha \twoheadrightarrow \beta$ ist trivial hinsichtlich \mathcal{R} , falls jede mögliche Ausprägung r von \mathcal{R} diese MVD erfüllt.

Ein Relationenschema \mathcal{R} mit zugehöriger Menge F von funktionalen und mehrwertigen Abhängigkeiten ist in vierter Normalform (4NF), wenn für jede nicht-triviale MVD $(\alpha \twoheadrightarrow \beta) \in F^+$ gilt:

- α ist Superschlüssel von \mathcal{R} .

Bemerkungen:

- ❑ Man kann zeigen, dass $\alpha \twoheadrightarrow \beta$ genau dann trivial ist, wenn $\beta \subseteq \alpha$ oder wenn $\beta = \mathcal{R} - \alpha$ gilt.
- ❑ Bei Relationen in der vierten Normalform wird die durch mehrwertige Abhängigkeiten verursachte Redundanz ausgeschlossen: Relationen in 4NF enthalten keine zwei voneinander unabhängigen mehrwertigen Fakten.
- ❑ Die vierte Normalform erreicht man durch Elimination der rechten Seite einer der beiden mehrwertigen Abhängigkeiten. Der eliminierte Teil bildet zusammen mit der linken Seite der MVD eine neue Relation.
- ❑ Die vierte Normalform ist eine Verschärfung der Boyce-Codd-Normalform.

Mehrwertige Abhängigkeiten

Verlustlose Zerlegung

Eine Zerlegung von \mathcal{R} mit zugehörigen funktionalen und mehrwertigen Abhängigkeiten F in die Relationenschemata \mathcal{R}_1 und \mathcal{R}_2 ist genau dann verlustlos, wenn mindestens eine der folgenden Bedingungen gilt:

- $((\mathcal{R}_1 \cap \mathcal{R}_2) \twoheadrightarrow \mathcal{R}_1) \in F^+$
- $((\mathcal{R}_1 \cap \mathcal{R}_2) \twoheadrightarrow \mathcal{R}_2) \in F^+$

Mehrwertige Abhängigkeiten

Verlustlose Zerlegung

Eine Zerlegung von \mathcal{R} mit zugehörigen funktionalen und mehrwertigen Abhängigkeiten F in die Relationenschemata \mathcal{R}_1 und \mathcal{R}_2 ist genau dann verlustlos, wenn mindestens eine der folgenden Bedingungen gilt:

- $((\mathcal{R}_1 \cap \mathcal{R}_2) \twoheadrightarrow \mathcal{R}_1) \in F^+$
- $((\mathcal{R}_1 \cap \mathcal{R}_2) \twoheadrightarrow \mathcal{R}_2) \in F^+$

Beispiel (Fortsetzung):

Faehigkeiten		
PersNr	Sprache	ProgSprache
3002	griechisch	C
3002	lateinisch	Pascal
3002	griechisch	Pascal
3002	lateinisch	C
3005	deutsch	Java

~>

Sprachen	
PersNr	Sprache
3002	griechisch
3002	lateinisch
3005	deutsch

ProgSprachen	
PersNr	ProgSprache
3002	C
3002	Pascal
3005	Java

Die Zerlegung in die Relationenschemata „Sprachen“ und „ProgSprachen“ ist verlustlos:

$$\text{Faehigkeiten} = \pi_{\text{PersNr, Sprache}}(\text{Faehigkeiten}) \bowtie \pi_{\text{PersNr, ProgSprache}}(\text{Faehigkeiten})$$

Relationale Dekomposition

Algorithm: Rel-Decomposition-MVD

Input: \mathcal{R} . Universalrelation.
 F . Menge funktionaler Abhängigkeiten für \mathcal{R} .

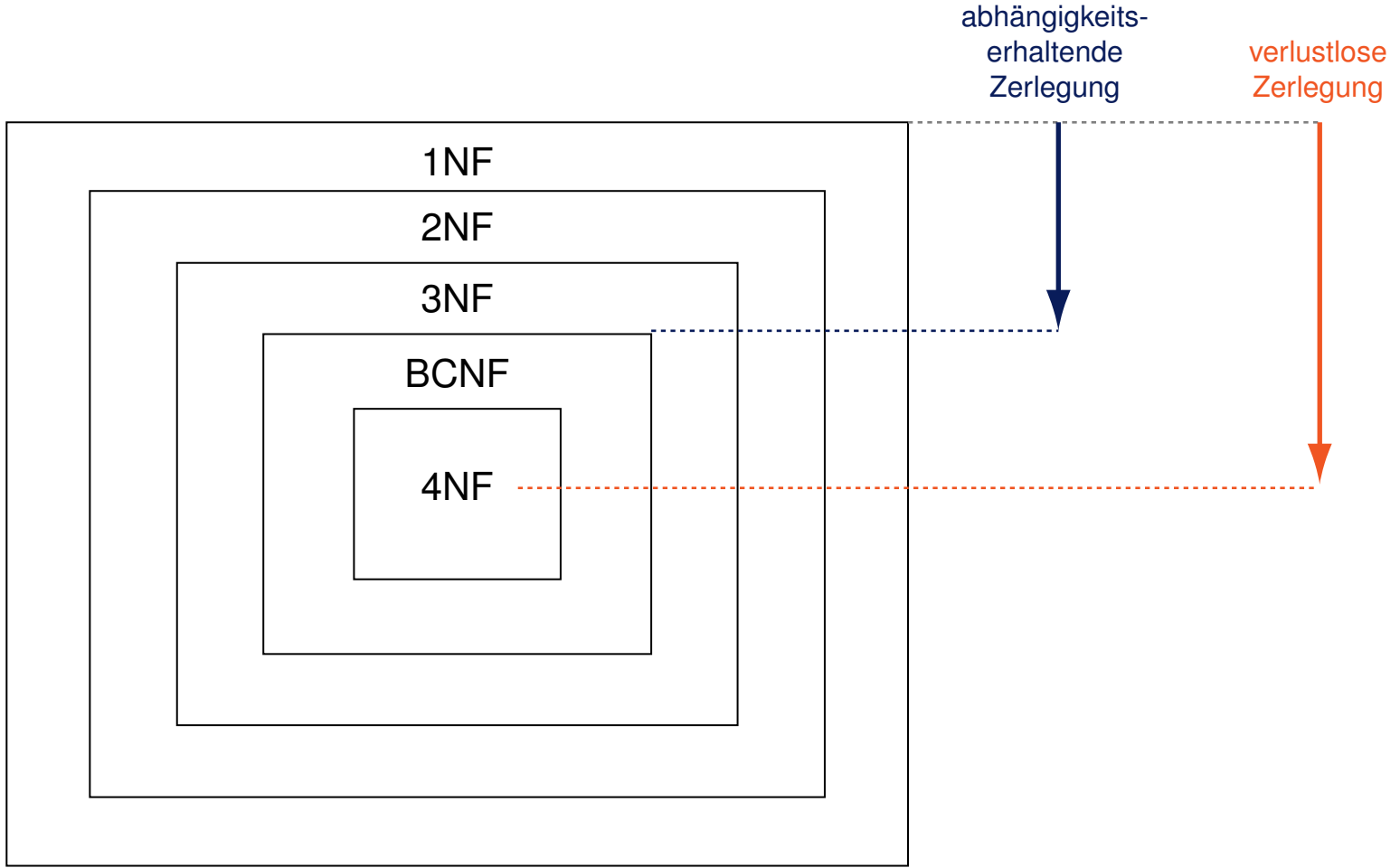
Output: \mathcal{R} . Verlustlose Zerlegung von \mathcal{R} mit Schemata in 4NF.

1. $\mathcal{R} = \{\mathcal{R}\}$
2. **WHILE** $\exists \mathcal{R}' : (\mathcal{R}' \in \mathcal{R} \wedge \mathcal{R}' \text{ not in 4NF})$ **DO**
3. Find nontrivial MVD $(\alpha \twoheadrightarrow \beta) \in F_{\mathcal{R}'}$ that violates 4NF
4. Decompose \mathcal{R}' into $\mathcal{R}'_1 = \alpha \cup \beta$ and $\mathcal{R}'_2 = \mathcal{R}' - \beta$
5. $\mathcal{R} = (\mathcal{R} - \{\mathcal{R}'\}) \cup \{\mathcal{R}'_1\} \cup \{\mathcal{R}'_2\}$
6. **ENDDO**
7. **RETURN**(\mathcal{R})

Bemerkungen:

- Der Algorithmus Rel-Decomposition-MVD erzeugt nicht notwendigerweise eine Zerlegung, die abhängigkeiterhaltend ist.

Beziehungen der Normalformen



[Kemper/Eickler 2004]