Informationssysteme 1 - Relationale Entwurfstheorie

Inhalt [1]

Inhalt [2]

Allgemeines, Überblick

Dieses Kapitel beschäftigt sich mit der konzeptuellen Feinabstimmung der erstellten relationalen Schemata auf der Grundalge formaler Methoden.

Als Basis dienen die **funktionalen Abhängigkeiten**. Auf sie aufbauend werden **Schlüssel** und **Normalformen** definiert.

Die Normalformen bestimmen sozusagen die Güte eines Relationenschemas.

Weiters werden **Normalisierungsalgorithmen** vorgestellt, um ein Relationenschema in mehrere Schemata zu zerlegen, die dann die entsprechenden Normalform erfüllen.

Am Ende wird noch das Konzept der **mehrwertigen Abhängigkeiten** und die darauf aufbauende **4. Normalform** vorgestellt.

Definition

Gegeben sei ein Relationenschema **sch**(R) und zwei Attributmengen α und β, für die $\alpha \subseteq$ **sch**(R) , bzw. $\beta \subseteq$ **sch**(R) gilt. Dann ist

$$\alpha \rightarrow \beta$$

eine funktionale Abhängigkeit (functional dependency, FD), wenn nur solche Ausprägungen R von **sch**(R) zulässig sind, in denen für alle Paare von Tupeln $r, t \in R$ mit $r.\alpha = t.\alpha$ auch $r.\beta = t.\beta$ gilt.

Dabei ist $r.\alpha = t.\alpha$ eine Kurzform für $\forall\ A \in \alpha : r.A = t.A$

Also bedeutet eine FD $\alpha \rightarrow \beta$, dass wenn zwei Tupel gleiche Werte für alle Attribute in α haben, dann müssen auch die Werte der Attribute in β übereinstimmen. Man sagt auch: „Die α -Werte bestimmen die β -Werte funktional (d.h. eindeutig)“; „Die β -Werte sind von den α -Werten funktional abhängig.“ oder „ α ist die Determinate von β“

Achtung: Eine funktionale Abhängigkeit stellt eine semantische Konsistenz-bedingung dar, die zu allen Zeiten in jedem gültigen Datenbankzustand eingehalten werden muss.

Funktionale Abhängigkeiten

Beispiel 1

Schema: **sch**(R) = {*A*, *B*, *C*, *D*}

Ausprägung: R

	R			
	A	***B***	***C***	***D***
t	a4	b2	c4	d3
p	a1	b1	c1	d1
q	a1	b1	c1	d2
r	a2	b2	c3	d2
s	a3	b2	c4	d3

Es gelten z. B.: folgende funktionale Abhängigkeiten:

$\{A\} \rightarrow \{B\}$
$\{C, D\} \rightarrow \{B\}$

Es gilt jedoch z. B. nicht:

$\{B\} \rightarrow \{C\}$

Konventionen zur Notation

In der Datenbank-Literatur hat sich anstatt der präzisen Notation

$\{C, D\} \rightarrow \{B\}$

folgende Notation eingebürgert:

$CD \rightarrow B$ oder $C, D \rightarrow B$

Weiters wird z.B. $\alpha - A$ statt $\alpha - \{A\}$ (α ist eine Attributmenge, A ein Attribut) geschrieben.

Eine abstrakte Attributmenge {A, B, C, D} wird als ABCD notiert.

Überprüfung einer funktionalen Abhängigkeit

Eine FD $\alpha \to \beta$ ist in R dann erfüllt, wenn für alle möglichen Ausprägungen c von α das Ergebnis der Abfrage

$\pi_\beta(\sigma_{\alpha = c}(R))$ höchstens ein Element enthält.

Algorithmus, der feststellt,
ob eine gegebene Relation R eine FD $\alpha \to \beta$ erfüllt:

Einhaltung (R, $\alpha \to \beta$)

- sortiere R nach α -Werten
- falls alle Gruppen bestehend aus Tupeln mit gleichen α -Werten auch gleiche β -Werte aufweisen: Ausgabe *ja*; sonst: Ausgabe *nein*

Besondere Typen von Funktionalen Abhängigkeiten

Triviale funktionale Abhängigkeiten

$\alpha \rightarrow \beta$ heißt trivial, wenn $\beta \subseteq \alpha$

(Triviale funktionale Abhängigkeiten gelten in jedem Relationenschema)

Volle funktionale Abhängigkeiten

α bestimmt β voll funktional ($\alpha \overset{\bullet}{\longrightarrow} \beta$),
wenn beide nachfolgenden Kriterien gelten:

1. $\alpha \rightarrow \beta$ (α bestimmt funktional β)
2. α kann nicht mehr „verkleinert“ werden, d.h.

   $\forall A \in \alpha : \alpha - \{A\} \not\rightarrow \beta$

Beispiel

Gegeben sei Relationenschema:

ProfessorenAdr: {[PersNr, Name, Rang, Raum, Ort, Straße, PLZ, BLand, Landesregierung]}

Dieses Relationenschema stellt keinen guten Entwurf dar. Es dient zu Demonstrationszwecken für funkt. Abhängigkeiten. Dabei wurden folgende vereinfachende Annahmen gemacht: Orte sind innerhalb von Bundesländern eindeutig benannt. PLZ ändert sich nicht innerhalb einer Straße. Städte und Straßen gehen nicht über Bundeslandgrenzen hinweg.

Funktionale Abhängigkeiten:

{PersNr} → {PersNr, Name, Rang, Raum, Ort, Straße, Plz, Vorwahl, Bland, EW, Landesregierung}

{Ort, BLand} → {EW, Vorwahl}

{PLZ} → {BLand, Ort EW}

{Ort, BLand, Straße} → {PLZ}

{BLand} → { Landesregierung}

{Raum} → {PersNr}

Hülle von Funktionalen Abhängigkeiten

Aus der Menge der funkt. Abh. aus dem vorigen Beispiel lassen sich auch weitere funktionale Abhängigkeiten ableiten. Z.B.:

{Raum} $\rightarrow$ {PersNr, Name, Rang, Raum, Ort, Straße, Plz, Vorwahl, Bland, EW, Landesregierung}

{PLZ} $\rightarrow$ {Landesregierung}

Dabei ist die Menge aller aus einer gegebenen Menge F von funktionalen Abhängigkeiten herleitbaren funktionalen Abhängigkeiten von besonderem Interesse. Diese Menge bezeichnet man als **Hülle**, und wird als $\mathbf{F^+}$ bezeichnet.

Armstrong-Axiome

Für die Herleitung einer vollständigen Hülle F^+ von funktionalen Abhängigkeiten reichen die folgenden 3 „Armstrong-Axiome“ als Inferenzregeln aus:

Dabei entsprechen α, β, γ und δ Teilmengen der Attribute aus **sch**(R).

- **Reflexivität:** Falls β eine Teilmenge von α ist ($\beta \subseteq \alpha$), dann gilt immer $\alpha \to \beta$. Insbesondere gilt $\alpha \to \alpha$. (siehe auch „triviale f. Abh.)
- **Verstärkung:** Falls $\alpha \to \beta$ gilt, dann gilt auch $\alpha\gamma \to \beta\gamma$. (Dabei stehe $\alpha\gamma$ für $\alpha \cup \gamma$.)
- **Transitivität:** Falls $\alpha \to \beta$ und $\beta \to \gamma$ gilt, dann gilt auch $\alpha \to \gamma$.

Obwohl die Armstrong-Axiome bereits vollständig sind, ist es für die Herleitung der Hülle komfortabel, die folgenden 3 Regeln hinzuzunehmen:

- **Vereinigungsregel:** Wenn $\alpha \to \beta$ und $\alpha \to \gamma$ gelten, dann gilt auch $\alpha \to \beta\gamma$.
- **Dekompositionsregel:** Wenn $\alpha \to \beta\gamma$ gilt, dann gelten auch $\alpha \to \beta$ und $\alpha \to \gamma$.
- **Pseudotransformationsregel:** Wenn $\alpha \to \beta$ und $\beta\gamma \to \delta$ gelten, dann gilt auch $\alpha\gamma \to \delta$.

Attribut-Hülle

Oft ist man nicht an der gesamten Hülle einer Menge F von funktionalen Abhängigkeiten interessiert, sondern nur an der Menge von Attributen α^+, die von α gemäß F funktional bestimmt werden.

Mit dem folgenden Algorithmus kann α^+ gebildet werden:

- **Eingabe:** eine Menge F von FDs und eine Menge von Attributen a.
- **Ausgabe:** die vollständige Menge von Attributen a+, für die gilt a → a+.

- AttrHülle(F,a)

```
Erg := a
While  (Änderungen an Erg) do
    Foreach FD b → g in F do
        If  b ⊆ Erg then Erg := Erg ∪  g
Ausgabe a+ = Erg
```

Äquivalenz von Mengen funktionaler Abhängigkeiten

Zwei Mengen F und G von funktionalen Abhängigkeiten heißen äquivalent ($F \equiv G$), wenn ihre Hüllen gleich sind ($F^+ = G^+$).

Bestimmung der Äquivalenz von F und G:

a) Berechnung der beiden Hüllen und Vergleich.

b) Für jede funktionale Abhängigkeit $\alpha \to \beta$ in F wird geprüft, ob $\beta \subseteq$ AttrHülle(G, α); und umgekehrt.

Kanonische Überdeckung [1]

Zu einer gegebenen Menge F von funktionalen Abhängigkeiten nennt man $\mathbf{F_c}$ eine kanonische Überdeckung, wenn folgende 3 Eigenschaften erfüllt sind:

- $F_c \equiv F$, d.h. $F_c^+ = F^+$
- In F_c existieren keine FDs $\alpha \rightarrow \beta$, bei denen α oder β überflüssige Attribute enthalten. D.h. es muß folgendes gelten:
  (a) $\forall A \in \alpha : (F_c - (\alpha \rightarrow \beta) \cup ((\alpha \vdash \{A\}) \rightarrow b)) \equiv F_c$
  (b) $\forall B \in \beta : (F_c - (\alpha \rightarrow \beta) \cup (\alpha \nrightarrow (\beta - \{B\}))) \equiv F_c$
- Jede linke Seite einer funktionalen Abhängigkeit in F_c ist einzigartig. Dies kann durch sukzessive Anwendung der Vereinigungsregel auf FDs der Art $\alpha \rightarrow \beta$ und $\alpha \rightarrow \gamma$ erzielt werden, so dass die beiden FDs durch $\alpha \rightarrow \beta\gamma$ ersetzt werden.

Kanonische Überdeckung [2]

Zu einer gegebenen Menge F kann die kanonische Überdeckung wie folgt bestimmt werden:

1. Führe für jede FD $\alpha \rightarrow \beta \in F$ die Linksreduktion durch, also:
   - Überprüfe für alle $A \in \alpha$, ob A überflüssig ist, d.h., ob

     $\beta \subseteq$ AttrHülle$(F, \alpha - A)$

     gilt. Falls dies der Fall ist, ersetze $\alpha \rightarrow \beta$ durch $(\alpha - A) \rightarrow \beta$.
2. Führe für jede (verbliebene) FD die Rechtsreduktion durch, also:
   - Überprüfe für alle $B \in \beta$, ob

     $B \in$ AttrHülle$(F - (\alpha \rightarrow \beta) \cup (\alpha \rightarrow (\beta - B)), \alpha)$

     gilt. Falls dies der Fall ist, ist B auf der rechten Seite überflüssig und kann eliminiert werden,

     d.h. $\alpha \rightarrow \beta$ wird durch $\alpha \rightarrow (\beta - B)$ ersetzt.
3. Entferne die FDs der Form $\alpha \rightarrow \emptyset$, die im 2. Schritt möglicherweise entstanden sind.
4. Fasse mittels der Vereinigungsregel FDs der Form $\alpha \rightarrow \beta_1, \ldots, \alpha \rightarrow \beta_n$ zusammen, so dass $\alpha \rightarrow (\beta_1 \cup \ldots \cup \beta_n)$ verbleibt.

Schlüssel

Superschlüssel (Superkey, Oberschlüssel):

In einem Relationenschema **sch**(R) ist $\alpha \subseteq$ **sch**(R) ein Superschlüssel, falls gilt:

$$\alpha \rightarrow \textbf{sch}(R)$$

Kandidatenschlüssel (Kandidate Key, Schlüsselkandidat):

In einem Relationenschema **sch**(R) ist $\alpha \subseteq$ **sch**(R) ein Kandidatenschlüssel, falls gilt:

$$\alpha \overset{\bullet}{\longrightarrow} \textbf{sch}(R)$$

Primärschlüssel (Primary Key):

Aus der Menge der Kandidatenschlüssel wird für die Implementierung einer relationalen Datenbank einer als Primärschlüssel bestimmt.

„Schlechte“ Relationenschemata

Anomalien [1]

Schlecht entworfene Relationenschemata können zu sogenannten Anomalien führen. Dazu soll die folgenden Relation *ProfVorl* betrachtet werden:

ProfVorl						
PersNr	**Name**	**Rang**	**Raum**	**VorlNr**	**Titel**	**SWS**
2125	Sokrates	C4	226	5041	Ethik	4
2125	Sokrates	C4	226	5049	Mäeutik	2
2125	Sokrates	C4	226	4052	Logik	4
...	...	...	...	...	...	...
2132	Popper	C3	52	5259	Der Wiener Kreis	2
2137	Kant	C4	7	4630	Die 3 Kritiken	4

„Schlechte“ Relationenschemata

Anomalien [2]

Updateanomalien:

Beim Update von Informationen (z.B. Sokrates übersiedelt in einen anderen Raum) kann es vorkommen, dass Tupel vergessen werden und die Datenbank in einen inkonsistenten Zustand gelangt.
(Selbst wenn man durch entsprechende Programme sicherstellen kann, dass immer alle Tupel gleichzeitig geändert werden, hat man immer noch das Problem des erhöhten Speicherbedarfs wegen unnötiger Redundanz und Performanceeinbußen, da immer mehrere Einträge geändert werden müssen.)

Einfügeanomalien:

Will man z.B. Daten eines neuen Professors einfügen, der noch keine Vorlesung hält, ist dies nur möglich wenn man die Attribute für Vorlesung leer läßt (mit NULL-Werten belegt).

Löschanomalien:

Will man z.B. die Vorlesung „Der Wiener Kreis“ löschen, löscht man auch die Daten von „Popper“ mit, was vielleicht nicht beabsichtigt war.

Allgemeines

Anomalien sind darauf zurückzuführen, dass nicht „zusammenpassende" Informationen in einer Relation abgelegt wurde. Um dies zu korrigieren werden bei der sogenannten „Normalisierung" (, die etwas später genauer behandelt wird,) ein Relationenschema in Teile aufgespalten.

Ein Relationenschema **sch**(R) wird in **sch**(R_1) ... **sch**(R_n) zerlegt.
Es gilt natürlich **sch**(R_i) $\subseteq$ **sch**(R) für $1 \leq i \leq n$.

Es gibt zwei grundlegende Korrektheitskriterien für eine solche Zerlegung:

1) **Verlustlosigkeit:** Die in der ursprünglichen Relationenausprägung R von **sch**(R) enthaltenen Informationen müssen aus den Ausprägungen R_1 ... R_n der neuen Schemata **sch**(R_1) ... **sch**(R_n) rekonstruierbar sein.
2) **Abhängigkeitserhaltung:** Die für **sch**(R) geltenden funktonalen Abhängigkeiten müssen auf die Schemata **sch**(R_1) ... **sch**(R_n) übertragbar sein.

Zerlegung von Relationen

Verlustlosigkeit [1]

Definition:

Eine Zerlegung eines Relationenschemas **sch**(R) in zwei Relationenschemata **sch**(R_1) und **sch**(R_e) ist gültig, wenn:

$$\mathbf{sch}(R) = \mathbf{sch}(R_1) \cup \mathbf{sch}(R_e)$$

Für eine Ausprägung R von **sch**(R) sind die Ausprägungen R_1 von **sch**(R_1) und R_2 von **sch**(R_e) wie folgt definiert:

$$R_1 = \pi_{\mathbf{sch}(R_1)}(R) \quad \text{und} \quad R_2 = \pi_{\mathbf{sch}(R_2)}(R)$$

Eine Zerlegung eines Relationenschemas **sch**(R) in zwei Relationenschemata **sch**(R_1) und **sch**(R_e) ist verlustlos, wenn für jede mögliche (gültige) Ausprägung von R gilt:

$$R = R_1 \mid\times\mid R_2$$

Verlustlosigkeit [2]

Kriterien:

Eine Zerlegung eines Relationenschemas **sch**(R) mit zugehörigen funktionalen Abhängigkeiten $F_{\mathbf{sch}(R)}$ in **sch**(R_1) und **sch**(R_e) ist verlustlos, wenn mindestens eine der folgenden funktionalen Abhängigkeiten herleitbar ist:

- $\mathbf{sch}(R_1) \cap \mathbf{sch}(R_e) \rightarrow \mathbf{sch}(R_1) \in F^{+}_{\mathbf{sch}(R)}$
- $\mathbf{sch}(R_1) \cap \mathbf{sch}(R_e) \rightarrow \mathbf{sch}(R_2) \in F^{+}_{\mathbf{sch}(R)}$

In anderen Worten: Es gelte $\mathbf{sch}(R) = \alpha \cup \beta \cup \gamma$, $\mathbf{sch}(R_1) = \alpha \cup \beta$ und $\mathbf{sch}(R_2) = \alpha \cup \gamma$. Dann muss mindesten eine der beiden Bedingungen gelten:

- $\beta \subseteq AttrHülle(F_{\mathbf{sch}(R)}, \alpha)$ oder
- $\gamma \subseteq AttrHülle(F_{\mathbf{sch}(R)}, \alpha)$

Zerlegung von Relationen

Verlustlosigkeit [3]

Beispiel 1:

Eine nicht-verlustlose Zerlegung:

Es gilt nur die nicht-triviale funktionale Abhängigkeit

$\{\text{Kneipe, Gast}\} \rightarrow \{\text{Bier}\}$,

jedoch nicht

$\{\text{Gast}\} \rightarrow \{\text{Bier}\}$

oder

$\{\text{Gast}\} \rightarrow \{\text{Kneipe}\}$.

Biertrinker		
Kneipe	***Gast***	***Bier***
Kowalski	Kemper	Pils
Kowalski	Eickler	Hefeweizen
Innsteg	Kemper	Hefeweizen

$\Pi_{\text{Kneipe, Gast}}$

$\Pi_{\text{Kneipe, Gast}}$

Besucht	
Kneipe	***Gast***
Kowalski	Kemper
Kowalski	Eickler
Innsteg	Kemper

Trinkt	
Gast	***Bier***
Kemper	Pils
Eickler	Hefeweizen
Kemper	Hefeweizen

$\bowtie$

$\neq$

Besucht A Trinkt		
Kneipe	***Gast***	***Bier***
Kowalski	Kemper	Pils
Kowalski	Kemper	Hefeweizen
Kowalski	Eickler	Hefeweizen
Innsteg	Kemper	Pils
Innsteg	Kemper	Hefeweizen

Verlustlosigkeit [4]

Beispiel 2:

Eine verlustlose Zerlegung:

Eltern		
Vater	***Mutter***	***Kind***
Johann	Martha	Else
Johann	Maria	Theo
Heinz	Martha	Cleo

$\Pi_{\text{Vater, Kind}}$

$\Pi_{\text{Mutter, Kind}}$

Väter	
Vater	***Kind***
Johann	Else
Johann	Theo
Heinz	Cleo

Mütter	
Mutter	***Kind***
Martha	Else
Maria	Theo
Martha	Cleo

Es gelten sogar beide funktionalen Abhängigkeiten

$\{\text{Kind}\} \rightarrow \{\text{Mutter}\}$
$\{\text{Kind}\} \rightarrow \{\text{Vater}\}$.

Abhängigkeitsbewahrung [1]

Definition:

$F_{\mathbf{sch}(R_i)}$ sei die Menge jener funktionalen Abhängigkeiten aus $F^+_{\mathbf{sch}(R)}$, deren Attribute alle in **sch**(R_i) enthalten sind.

Eine Zerlegung eines Relationenschemas **sch**(R) in Relationenschemata **sch**(R_1) ... **sch**(R_n) ist abhängigkeitsbewahrend, wenn:

$$F_{\mathbf{sch}(R)} \equiv (F_{\mathbf{sch}(R_1)} \cup \ldots \cup F_{\mathbf{sch}(R_n)}) \quad \text{bzw.}$$

$$F^+_{\mathbf{sch}(R)} = (F_{\mathbf{sch}(R_1)} \cup \ldots \cup F_{\mathbf{sch}(R_n)})^+$$

Zerlegung von Relationen

Abhängigkeitsbewahrung [2]

Beispiel:

Im Beispiel soll gelten:

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

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

Zerlegung:

PLZverzeichnis			
Ort	**BLand**	**Straße**	**PLZ**
Frankfurt	Hessen	Goethestraße	60313
Frankfurt	Hessen	Galgenstraße	60437
Frankfurt	Brandenburg	Goethestraße	15234

$\Pi_{\text{PLZ,Straße}}$

Straßen	
PLZ	Straße
15234	Goethestraße
60313	Goethestraße
60437	Galgenstraße
15235	Goethestrasse

$\Pi_{\text{Stadt,Bland,PLZ}}$

Orte		
Ort	BLand	PLZ
Frankfurt	Hessen	60313
Frankfurt	Hessen	60437
Frankfurt	Brandenburg	15234
Frankfurt	**Brandenburg**	**15235**

Die Zerlegung ist zwar verlustlos, aber nicht abhängigkeitsbewahrend!

Die funktionale Abhängigkeit

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

ging verloren.

Es könnte nun ein Tupel [15235, Goethestraße] in Straßen und [Frankfurt, Brandenburg, 15235] in Orte eingefügt werden, was jedoch die globale funkt. Abh. {Straße, Ort, Bland $\rightarrow$ {PLZ} verletzen würde.

Erste Normalform, 1NF

Definition

Alle Attribute müssen atomare Wertebereiche haben. Mengenwertige oder andere komplexe Attributdomänen sind nicht zulässig.

Beispiel

Nicht in 1NF:

Eltern		
Vater	***Mutter***	***Kinder***
Johann	Martha	{Else, Lucie}
Johann	Maria	{Theo, Josef}
Heinz	Martha	{Cleo}

In 1NF:

Eltern		
Vater	***Mutter***	***Kind***
Johann	Martha	Else
Johann	Martha	Lucie
Johann	Maria	Theo
Johann	Maria	Josef
Heinz	Martha	Cleo

Zweite Normalform, 2NF

Definition

Ein Relationenschema **sch**(R) mit den zugehörigen funktionalen Abhängigkeiten F ist in zweiter Normalform, wenn jedes Nicht-Schlüssel-Attribut $A \in R$ voll funktional von jedem Schlüssel-kandidaten abhängig ist.

Seien also $\kappa_1 \ldots \kappa_i$ Kandidatenschlüssel von **sch**(R) und sei $A \in$ **sch**(R) - ($\kappa_1 \cup \ldots \cup \kappa_i$),
(ein solches Attribut A wird auch als *nicht-prim* bezeichnet)

dann muss also für alle κ_j ($1 \leq j \leq i$) gelten:

$$\kappa_j \xrightarrow{\bullet} A \in F^+$$

Zweite Normalform, 2NF

Beispiel

StudentenBelegung			
MatrNr	**VorlNr**	**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
...	...	...	...

Dieses Relationenschema ist nicht in 2NF, weil:

Kandidatenschlüssel:
MatrNr, VorlNr

Funktionale Abhängigkeiten:
$\{MatrNr\} \rightarrow \{Name, Semester\}$

Name und Semester sind somit nicht voll funktional von {MatrNr, VorlNr} abhängig.

Das Relationenschema zeigt schwerwiegende Anomalien (siehe Folie 16).
Lösung: Man zerlegt die Relation in mehrere Teilrelationen (verlustlos und abhängigkeitstreu), die dann der 2NF genügen:

hören: {[MatrNr, VorlNr]}
Studenten:{[MatrNr, Name, Semester]}

Dritte Normalform, 3NF

Definition

Ein Relationenschema **sch**(R) ist in dritter Normalform, wenn für jede in **sch**(R) geltende funktionale Abhängigkeit $\alpha \rightarrow B$ mit $\alpha \subseteq$ **sch**(R) und $B \in$ **sch**(R) mindestens eine der drei folgenden Bedingungen gilt:

- $B \in \alpha$ (d.h. die funkt. Abh. ist trivial)
- Das Attribut B ist in einem Kandidatenschlüssel von **sch**(R) enthalten

  (d. h. B ist prim)
- α ist Superschlüssel von **sch**(R)

Dritte Normalform, 3NF

Beispiel

ProfessorenAdr: {[PersNr, Name, Rang, Raum, Ort, Straße, PLZ, BLand, Landesregierung]}

Annahmen: Orte sind innerhalb von Bundesländern eindeutig benannt. PLZ ändert sich nicht innerhalb einer Straße. Städte und Straßen gehen nicht über Bundeslandgrenzen hinweg.

Kandidatenschlüssel: {PersNr} und {Raum}

Das Relationenschema erfüllt die 2NF. Es sind alle Nichtschlüsselattribute von jedem Kandidatenschlüssel voll funktional abhängig.

Es verletzt jedoch die 3NF, weil z.B die in dem Relationenschema enthaltene funktionale Abhängigkeit {Ort, Bland} $\rightarrow$ {Vorwahl} nicht den Bedingungen für 3NF genügt.

Synthesealgorithmus für 3NF

Allgemeines

Der im folgenden vorgestellte Synthesealgorithmus zerlegt ein gegebenes Relationenschema **sch**(R) mit der Menge von funktionalen Abhängigkeiten F in **sch**(R_1) ... **sch**(R_n), alle drei folgenden Kriterien erfüllt sind:

- **sch**(R_1) ... **sch**(R_n) ist eine verlustlose Zerlegung von **sch**(R)
- Die Zerlegung ist abhängigkeitsbewahrend
- alle **sch**(R_i) ($1 \leq i \leq n$) sind in dritter Normalform

Synthesealgorithmus für 3NF

Algorithmus

1) Bestimme die kanonische Überdeckung F_c zu F
   (a) Linksreduktion der funktionalen Abhängigkeiten
   (b) Rechtsreduktion der funktionalen Abhängigkeiten
   (c) Entfernung der funkt. Abh. $\alpha \to \{\,\}$
   (d) Zusammenfassung der funkt. Abh. mit gleichen linken Seiten
2) Für jede funktionale Abhängigkeit $\alpha \to \beta \in F_c$:
   - Kreiere ein Relationenschema **sch**(R_α) := $\alpha \cup \beta$
   - Ordne **sch**(R_α) die funkt. Abh. F := $\{\alpha' \to \beta' \mid \alpha' \cup \beta' \subseteq$ **sch**(R_α) $\}$ zu
3) Falls eines der in Schritt 2 erzeugten Schemata **sch**(R_α) einen Kandidatenschlüssel enthält, ist der Algorithmus beendet. Ansonsten wähle einen Kandidatenschlüssel $\kappa \in$ **sch**(R) aus und definiere folgendes zusätzliches Relationenschema:
   - **sch**(R_κ) := κ
   - $F_\kappa := \{\,\}$
4) Eliminiere diejenigen Schemata, **sch**(R_α) die in einem anderen Relationenschema **sch**($R_{\alpha'}$) enthalten sind, d.h. **sch**(R_α) $\subseteq$ **sch**($R_{\alpha'}$)

Synthesealgorithmus für 3NF

Beispiel

- ProfessorenAdr: {[PersNr, Name, Rang, Raum, Ort, Straße, PLZ, Vorwahl, BLand, EW, Landesregierung]}

  fd_1: {PersNr} → {Name, Rang, Raum, Ort, Straße, BLand}
  fd_2: {Raum} → {PersNr}
  fd_3: {Straße, BLand, Ort} → {PLZ}
  fd_4: {Ort,BLand} → {EW, Vorwahl}
  fd_5: {BLand} → {Landesregierung}
  fd_6: {PLZ} → {BLand, Ort}

- Professoren: {[PersNr, Name, Rang, Raum, Ort, Straße, BLand]}
- PLZverzeichnis: {[Straße, BLand, Ort, PLZ]}
- OrteVerzeichnis: {[Ort, BLand, EW, Vorwahl]}
- Regierungen: {[Bland, Landesregierung]}

Boyce-Codd Normalform, BCNF

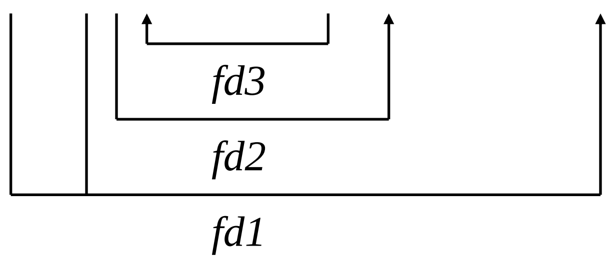
Definition

Ein Relationenschema **sch**(R) ist in BCNF, wenn für jede in **sch**(R) geltende funktionale Abhängigkeit $\alpha \rightarrow B$ mit $\alpha \subseteq$ **sch**(R) und $B \in$ **sch**(R) mindestens eine der zwei folgenden Bedingungen gilt:

- $B \in \alpha$ (d.h. die funkt. Abh. ist trivial)
- α ist Superschlüssel von **sch**(R)

Beispiel

Städte: {[Ort, BLand, Ministerpräsident/in, EW]}



Kandidatenschlüssel:
{Ort, BLand},
{Ort, Ministerpräsident/in}

Das Relationenschema ist in 3NF;
fd2 und *fd3* verletzen jedoch die BCNF

Zerlegung in BCNF

Allgemeines

Man kann grundsätzlich jedes Relationenschema **sch**(R) mit zugeordneten funktionalen Abhängigkeiten F in **sch**(R_1) ... **sch**(R_n), dass gilt:

- **sch**(R_1) ... **sch**(R_n) ist eine verlustlose Zerlegung von **sch**(R)
- alle **sch**(R_i) ($1 \leq i \leq n$) sind in BCNF

Leider kann man nicht immer eine BCNF-Zerlegung finden, die auch abhängigkeitsbewahrend ist.

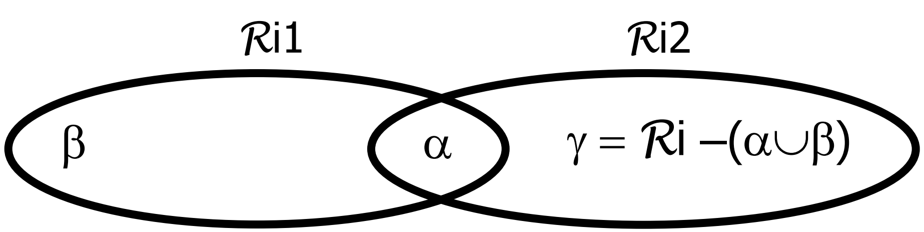
Zerlegung in BCNF

Algorithmus

1) Starte mit Z = **sch**(R)
2) Solange es noch ein Relationenschema **sch**(R_i) $\in$ Z gibt, das noch nicht in BCNF ist, mache folgendes:
   - Es gibt also eine für **sch**(R_i) geltende nicht-triviale funkt. Abh. $\alpha \rightarrow \beta$ mit $\alpha \cap \beta = \{\,\}$ und $\alpha \not\rightarrow$ **sch**(R_i)
   - Find eine solche funktionale Abhängigkeit. (Man sollte sie so wählen, dass β alle von α funktional abhängigen Attribute B $\in$ (**sch**(R_i) - α) enthält, damit der Algorithmus möglichst schnell determiniert.)
   - Zerlege **sch**(R_i) in **sch**(R_{i1}) := $\alpha \cup \beta$ und **sch**(R_{i2}) := **sch**(R_i) - β
   - Entferne **sch**(R_i) aus Z und füge **sch**(R_{i1}) und **sch**(R_{i2}) ein, also

     Z := (Z - **sch**(R_i)) $\cup$ **sch**(R_{i1}) $\cup$ **sch**(R_{i2})

Die nachfolgende Graphik illustriert abstrakt die Zerlegung eine Relationenschmas **sch**(R_i) in **sch**(R_{i1}) und **sch**(R_{i2}) entlang der funkt. Abh. $\alpha \rightarrow \beta$:



Mehrwertige Abhängigkeiten

Definition [1]

Sei **sch**(R) ein Relationenschema und α, β, γ drei Attributmengen, für die $\alpha \subseteq$ **sch**(R) , $\beta \subseteq$ **sch**(R) bzw. $\gamma \subseteq$ **sch**(R) und $\alpha \cup \beta \cup \gamma =$ **sch**(R) gilt. Dann ist

$$\alpha \to \to \beta$$

eine mehrwertige Abhängigkeit (mulivalued Dependencies, MVD), wenn in jeder gültigen Ausprägungen R von **sch**(R) gilt: Für jedes Paar von Tupeln $t_1, t_2 \in R$ mit $t_1.\alpha = t_2.\alpha$ existieren zwei weitere Tupel t_3, und t_4 mit folgenden Eigenschaften.

$$t_1.\alpha = t_2.\alpha = t_3.\alpha = t_4.\alpha$$
$$t_3.\beta = t_1.\beta$$
$$t_3.\gamma = t_2.\gamma$$
$$t_4.\beta = t_2.\beta$$
$$t_4.\gamma = t_1.\gamma$$

In anderen Worten: Bei 2 Tupeln mit gleichem α - Wert kann man die β - Werte vertauschen, und die resultierenden Tupel müssen ebenfalls in der Relation sein.

Mehrwertige Abhängigkeiten

Definition [2]

Graphisch kann die Definition von mehrwertigen Abhängigkeiten $\alpha \rightarrow\rightarrow \beta$ folgenderweise veranschaulicht werden:

	R		
	α **A1 ... Ai**	**β** **Ai+1 ... Aj**	**γ** **Aj+1 ... An**
t_1	a1 ... ai	ai+1 ... aj	aj+1 ... an
t_2	a1 ... ai	bi+1 ... bj	bj+1 ... bn
t_3	a1 ... ai	bi+1 ... bj	aj+1 ... an
t_4	a1 ... ai	ai+1 ... aj	bj+1 ... bn

Mehrwertige Abhängigkeiten sind eine Verallgemeinerung der funktionalen Abhängigkeiten. D.h. jede funktionale Abhängigkeit ist auch eine mehrwertige (aber nicht umgekehrt).

Mehrwertige Abhängigkeiten

Beispiel

Fähigkeiten		
PersNr	**Sprache**	**ProgSprache**
3002	griechisch	C
3002	lateinisch	Pascal
3002	griechisch	Pascal
3002	lateinisch	C
3005	deutsch	Ada

Es gelten die mehrwertigen Abhängigkeiten:

$$\{PersNr\} \to\to \{Sprache\}$$

$$\{PersNr\} \to\to \{ProgrSprache\}$$

Man erkennt, dass das es sich hier um kein ‚gutes' Relationenschema handelt. Es enthält insert- update- und delete-Anomalien, jedoch nicht auf Attributwert-Ebene, sondern auf Tupel-Ebene.

Wesentlich besser wäre folgende verlustlose Zerlegung:

Sprachen	
PersNr	**Sprache**
3002	griechsich
3002	lateinisch
30005	deutsch

ProgSprachen	
PersNr	**ProgSprache**
3002	C
3002	Pascal
30005	Ada

Weitere Eigenschaften

Gilt in einem Relationenschema **sch**(R) $\alpha \twoheadrightarrow \beta$,

dann immer auch $\alpha \twoheadrightarrow$ **sch**(R) - α - β ,

Ein Relationenschema **sch**(R) mit einer Menge D von zugeordneten funktionalen und mehrwertigen Abhängigkeiten kann dann genau verlustlos in die Schemata **sch**(R_1) und **sch**(R_e) zerlegt werden, wenn

- **sch**(R) = **sch**(R_1) $\cup$ **sch**(R_e)

Und eine der beiden folgenden Bedingungen gilt (Vergleiche Folie 21):

- **sch**(R_1) $\cap$ **sch**(R_e) $\twoheadrightarrow$ **sch**(R_1) $\in$ D^+
- **sch**(R_1) $\cap$ **sch**(R_e) $\twoheadrightarrow$ **sch**(R_2) $\in$ D^+

Es gibt also auch für mehrwertige Abhängigkeiten eine Hülle und eine kanonische Überdeckung. Die entsprechenden Ableitungsregeln finden sie z.B. im Buch von Kemper.

Vierte Normalform, 4NF

Definition

Ein Relationenschema **sch**(R) mit zugeordneter Menge D von funktionalen und mehrwertigen Abhängigkeiten ist in vierter Normalform, wenn für jede mehrwertige Abhängigkeit $\alpha \rightarrow\rightarrow \beta \in D^+$ eine der zwei folgenden Bedingungen gilt:

- die mehrwertige Abhängigkeit ist trivial
- α ist Superschlüssel von **sch**(R)

Eine mehrwertige Abhängigkeit $\alpha \rightarrow\rightarrow \beta$ ist trivial, wenn gilt:

- $\beta \subseteq \alpha$ oder
- β = **sch**(R) - α

Algorithmus

Das Verfahren arbeitet analog der Zerlegung in BCNF und zerlegt somit ein Relationenschema **sch**(R) verlustlos in 4NF-Schemata.

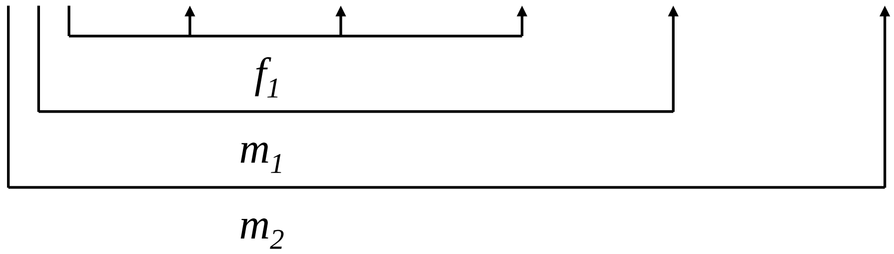
1) Starte mit Z = **sch**(R)
2) Solange es noch ein Relationenschema **sch**(R_i) $\in$ Z gibt, das noch nicht in 4NF ist, mache folgendes:

- Finde eine in **sch**(R_i) geltende nicht-triviale mehrw. Abh. $\alpha \rightarrow\rightarrow \beta$, für die gilt
  - $\alpha \cap \beta = \{\ \}$
  - $\alpha \not\rightarrow$ **sch**(R_i)
- Zerlege **sch**(R_i) in **sch**(R_{i1}) := $\alpha \cup \beta$ und **sch**(R_{i2}) := **sch**(R_i) - β
- Entferne **sch**(R_i) aus Z und füge **sch**(R_{i1}) und **sch**(R_{i2}) ein, also
  Z := (Z - **sch**(R_i)) $\cup$ **sch**(R_{i1}) $\cup$ **sch**(R_{i2}))

Zerlegung in 4NF

Beispiel

Assistenten: {[PersNr, Name, Fachgebiet, Boss, Sprache, ProgSprache]}



Schritt 1, „ f_1 wird herausgelöst“:

Assistenten:{[PersNr, Name, Fachgebiet, Boss]}
Fähigkeiten:{[PersNr, Sprache, ProgSprache]}

‚Assistenten‘ ist in 4NF, ‚Fähigkeiten‘ noch nicht (wegen m_1 und m_2).

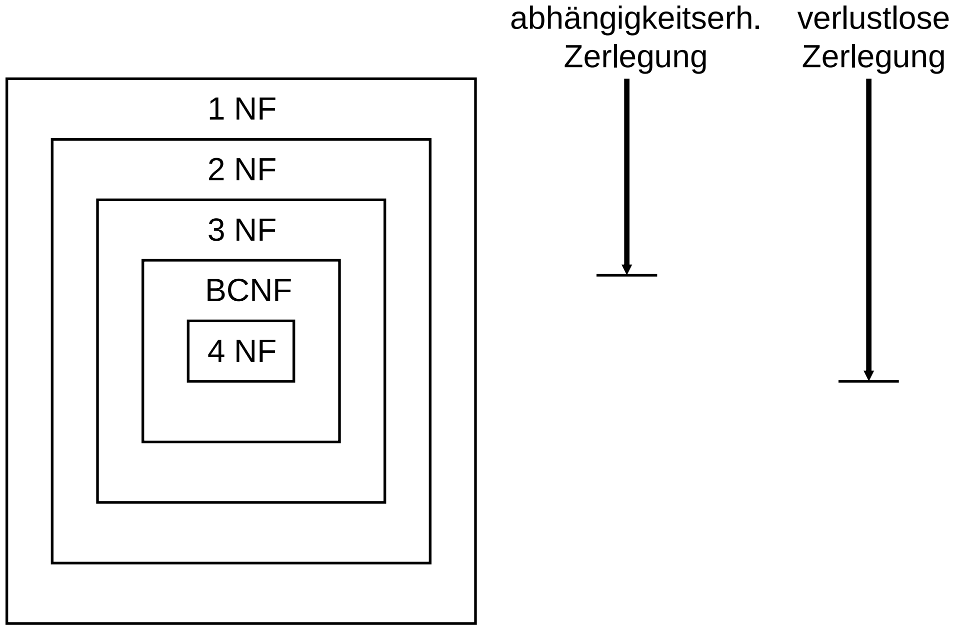
Schritt 2, „m_1 wird aus ‚Fähigkeiten‘ herausgelöst“:

Sprachen:{[PersNr, Sprache]}
ProgSprachen:{[PersNr, ProgSprache]}

Beide Relationenschemata sind in 4NF. Das Gesamtergebnis: Eine Zerlegung in ‚Assistenten‘, ‚Sprachen‘ und ‚ProgSprachen‘.

Zusammenfassung Normalformen

Es gelten folgende Beziehungen zwischen den Normalformen:



Verlustlose Zerlegung in jede Normalform ist möglich.

Abhängigkeitsbewahrung kann nur bis 3NF garantiert werden.

Man sollte die Normalisierung als Feinabstimmung im konzeptuellen Entwurf verwenden. Es sollte nicht mit der Begründung „das kann man bei der Normalisierung sowieso noch richten“ die im Entwurf vorangehende Erstellung des ER-Modells sehr nachlässig durchgeführt werden.

Im Gegenteil. Es sollten aus dem ER-Modell nahezu ausschließlich Relationen in 3NF als Ergebnis der Transformation ins relationale Modell entstehen.