Datenbankanwendung

Wintersemester 2014/15

Prof. Dr.-Ing. Sebastian Michel
TU Kaiserslautern

smichel@cs.uni-kl.de

Wiederholung:

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

Wieso ist dies ein schlechtes Schema?

Normalisierung von Relationen

Um Qualitätsprobleme im ursprünglichen Entwuft zu beheben, wird das bestehende Relationenschema $\mathcal{R}$ wird in mehrere Relationenschemata $\mathcal{R}_1, ..., \mathcal{R}_n$ zerlegt, die dann "besser" sind.

- Die Güte einer Zerlegung wird mit **Normalformen** beschrieben.
- Normalformen: 1NF, 2NF, 3NF, BCNF, 4NF, ...

Korrektheitskriterien für Zerlegung:

- **Verlustlosigkeit:** Die in der ursprünglichen Relationenausprägung R des Schemas $\mathcal{R}$ enthaltenen Daten müssen aus den Ausprägungen $R_1, ..., R_n$ der neuen Relationenschemata $\mathcal{R}_1, ..., \mathcal{R}_n$ rekonstruierbar sein.
- **Abhängigkeitsbewahrung:** Alle FDs in F_R sollten in den $F_{\mathcal{R}_1}, F_{\mathcal{R}_2}, ...F_{\mathcal{R}_n}$ bewahrt bleiben.

Verlustlosigkeit

- Zerlegung ist gültig, wenn: $\mathcal{R} = \mathcal{R}_1 \cup \mathcal{R}_2$
- D.h. alle Attribute aus $\mathcal{R}$ bleiben in der Zerlegung erhalten
- Wir definieren:
  - $R_1 := \pi_{\mathcal{R}_1}(R)$
  - $R_2 := \pi_{\mathcal{R}_2}(R)$
- die Zerlegung von $\mathcal{R}$ in $\mathcal{R}_1$ und $\mathcal{R}_2$ ist **verlustlos**, falls für jede mögliche (gültige) Ausprägung R von $\mathcal{R}$ gilt:

$$R = R_1 \bowtie R_2$$

Die Zerlegung muss also durch einen natürlichen Verbund (Join) rekonstruierbar sein.

D.h. $\mathcal{R}_1 \cap \mathcal{R}_2 \neq \emptyset$ "sinnvoller" Schlüssel existiert

Formale Charakterisierung Verlustloser Zerlegungen

- Gegeben Zerlegung von $\mathcal{R}$ in $\mathcal{R}_1$ und $\mathcal{R}_2$
- $F_\mathcal{R}$ ist die Menge der FDs in $\mathcal{R}$
- Zerlegung ist verlustlos, wenn mindestens **eine** der folgenden FDs herleitbar ist:
  - $(\mathcal{R}_1 \cap \mathcal{R}_2) \rightarrow \mathcal{R}_1 \in F_\mathcal{R}^+$ **oder** d.h. Schlüssel bestimmt $\mathcal{R}_1$
  - $(\mathcal{R}_1 \cap \mathcal{R}_2) \rightarrow \mathcal{R}_2 \in F_\mathcal{R}^+$ d.h. Schlüssel bestimmt $\mathcal{R}_2$

Beispiel:

- Seien α, β und γ paarweise disjunkte Attributmengen
- $\mathcal{R} = \alpha \cup \beta \cup \gamma$, $\mathcal{R}_1 = \alpha \cup \beta$, $\mathcal{R}_2 = \alpha \cup \gamma$, $\mathcal{R}_1 \cap \mathcal{R}_2 = \alpha$
- Dann muss eine der Bedingungen gelten:
  - $\beta \subseteq$ *AttrHülle*$(F_\mathcal{R}, \alpha)$ **oder**
  - $\gamma \subseteq$ *AttrHülle*$(F_\mathcal{R}, \alpha)$

D.h. die gemeinsamen Joinattribute α müssen $\mathcal{R}_1$ oder $\mathcal{R}_2$ bestimmen.
Dies ist eine hinreichende aber nicht notwendige Bedingung!

Beispiel für Verlust

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

wird zerlegt in ...

$\pi_{Kneipe,Gast}$

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

$\pi_{Gast,Bier}$

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

Beispiel für Verlust: Wiederherstellung

$\pi_{Kneipe,Gast}$

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

$\pi_{Gast,Bier}$

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

Mit Join verbunden gibt ..

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

Erläuterung des Biertrinker-Beispiels

- User Biertrinker-Beispiel war **keine verlustlose Zerlegung**.
- Es gibt in Biertrinker nämlich nur die folgende nicht-triviale funktionale Abhängigkeit:

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

- $\mathcal{R} = \{Kneipe\} \cup \{Gast\} \cup \{Bier\}$
- $\mathcal{R}_1 = \{Kneipe\} \cup \{Gast\}$, $\mathcal{R}_2 = \{Gast\} \cup \{Bier\}$, $\mathcal{R}_1 \cap \mathcal{R}_2 = \{Gast\}$.
- Dann muss eine der Bedingungen gelten:
  - $\{Kneipe\} \subseteq$ *AttrHülle*$(\{Kneipe, Gast\} \rightarrow \{Bier\}, \{Gast\})$ **oder**
  - $\{Bier\} \subseteq$ *AttrHülle*$(\{Kneipe, Gast\} \rightarrow \{Bier\}, \{Gast\})$
- Das ist nicht der Fall!
- Keine der zwei möglichen, die Verlustlosigkeit garantierenden FDs gilt:

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

Beispiel für Verlustfreie Zerlegung

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

wird zerlegt in ...

$\pi_{Vater,Kind}$

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

$\pi_{Mutter,Kind}$

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

Erläuterung der Zerlegung der Eltern-Relation

- $\mathcal{R} = \{Vater, Mutter, Kind\}$
- $F_{\mathcal{R}} = \{\{Kind\} \rightarrow \{Mutter\}, \{Kind\} \rightarrow \{Vater\}\}$
- $\mathcal{R}_1 = \{Vater, Kind\}$, $\mathcal{R}_2 = \{Mutter, Kind\}, \mathcal{R}_1 \cap \mathcal{R}_2 = \{Kind\}$.
- Dann muss **eine** der Bedingungen gelten:
  - $\{Vater\} \subseteq$ *AttrHülle*$(F_{\mathcal{R}}, \{Kind\})$ **oder**
  - $\{Mutter\} \subseteq$ *AttrHülle*$(F_{\mathcal{R}}, \{Kind\})$
- Hier gelten sogar beide Bedingungen.
- Also ist die Zerlegung verlustlos.

Abhängigkeitsbewahrung:

- ... ist 2. Korrektheitskriterium für eine Zerlegung
- $\mathcal{R}$ ist zerlegt in $\mathcal{R}_1, ..., \mathcal{R}_n$
- Um bei neu eingefügten Daten zu überprüfen, ob es neue Abhängigkeiten gibt, könnte man zur Sicherheit jedesmal den Join $R_1 \bowtie ... \bowtie R_n$ berechne und auf Abhängigkeiten überprüfen
- Das ist allerdings **sehr** teuer!

Idee: die Abhängigkeiten sollten **lokal** überprüfbar sein!

- d.h. Überprüfung kann lokal auf Relationen $\mathcal{R}_1, ..., \mathcal{R}_n$ gemacht werden
- Dafür muss folgende Bedingung gelten:
  $F_{\mathcal{R}} \equiv (F_{\mathcal{R}_1} \cup ... \cup F_{\mathcal{R}_n})$ bzw. $F_{\mathcal{R}}^+ = (F_{\mathcal{R}_1} \cup ... \cup F_{\mathcal{R}_n})^+$
- eine solche Zerlegung heisst auch **hüllengetreue Dekomposition!**

Gegenbeispiel

- **PLZverzeichnis:** {[Straße, Ort, Bland, PLZ]}
- **Annahmen:**
  - Orte werden durch ihren Namen (Ort) und das Bundesland (Bland) eindeutig identifiziert.
  - Innerhalb einer Straße ändert sich die Postleitzahl nicht
  - PLZ Gebiete gehen nicht über Ortsgrenzen und Orte nicht über Bundeslandgrenzen hinweg.

Daraus ergeben sich die folgenden FDs:

- $\{\text{PLZ}\} \rightarrow \{\text{Ort, Bland}\}$
- $\{\text{Straße, Ort, Bland}\} \rightarrow \{\text{PLZ}\}$

Betrachten wir die folgende Zerlegung von PLZverzeichnis:

- Straßen: {[PLZ, Straße]}
- Orte: {[PLZ, Ort, Bland]}

Zerlegung der Relation PLZverzeichnis

PLZVerzeichnis			
Ort	Bland	Straße	PLZ
Frankfurt	Hessen	Goethestraße	60313
Frankfurt	Hessen	Galgenstraße	60437
Frankfurt	Brandenburg	Goethestraße	15234

$\pi_{PLZ,Straße}$

Straßen	
PLZ	Straße
15234	Goethestraße
60313	Goethestraße
60437	Galgenstraße

$\pi_{Ort,Bland,PLZ}$

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

- Diese Zerlegung ist **verlustlos** da $\{\text{PLZ}\} \rightarrow \{\text{Orte, Bland}\}$ und PLZ das einzige gemeinsame Attribut.
- die FD $\{\text{Straße, Ort, Bland}\} \rightarrow \{\text{PLZ}\}$ ist im zerlegten Schema **nicht** mehr enthalten: Also: Zerlegung ist **nicht** abhängigkeitserhaltend.

$\pi_{PLZ,Straße}$

Straßen	
PLZ	Straße
15234	Goethestraße
60313	Goethestraße
60437	Galgenstraße
15235	**Goethestraße**

$\pi_{Ort,Bland,PLZ}$

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

Einfügen der **roten** Tupel in Zerlegung ist OK ..

Zerlegung der Relation PLZverzeichnis (2)

Durch einen Join der Zerlegung erhalten wir dann ...

PLZVerzeichnis			
Ort	Bland	Straße	PLZ
Frankfurt	Hessen	Goethestraße	60313
Frankfurt	Hessen	Galgenstraße	60437
Frankfurt	Brandenburg	Goethestraße	15234
Frankfurt	**Brandenburg**	**Goethestraße**	**15235**

Was fällt auf?

- Join erzeugt zusätzliches rotes Tupel, das die FD {Straße, Ort, Bland} $\rightarrow$ {PLZ} in **PLZverzeichnis** verletzt!
- aber: nur durch die Ausführung des Joins ist diese Verletzung der FD aufzudecken.

Zusammenfassung

- Damit eine Zerlegung **korrekt** ist, muss sie **verlustlos** und **abhängigkeitserhaltend** sein.
- **Verlustlos** = keine Daten gehen verloren oder werden zusätzlich erzeugt bei einem Join der zerlegten Relationen.
- **abhängigkeitserhaltend** = FDs existieren nur lokal aber nicht Relationen-übergreifend

Bisher:

- Korrektheit der Zerlegung

Jetzt:

- Bewertung der Güte von Relationenschemata
- Zerlegung von Relationenschemata, um eine höhere Güte zu erreichen
- Güte = 1NF (NF=Normalform), 2NF, 3NF, BCNF, 4NF, ...

Erste Normalform (1NF)

Intuition: keine mengenwertigen Attributwerte

Eine Relation $\mathcal{R}$ ist in 1NF, wenn sie keine zusammengesetzten, mengenwertigen oder relationenwertigen Domänen hat.

Gegenbeispiel:

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

1NF:

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

Exkurs: NF^2 Relationen

- Non-First Normal-Form-Relationen
- = geschachtelte Relationen

Eltern			
Vater	**Mutter**	**Kinder**	
Johan	Martha	Else	5
		Lucie	3
Johann	Maria	Theo	3
		Josef	1
Heinz	Martha	Cleo	9

- Vorteil: keine unnötige Wiederholung (=Redundanz) von Vater und Mutter
- NF^2 eng verwandt mit hierarchischen Datenmodellen (XML)
- **im Folgenden setzen wir immer 1NF voraus**

Zweite Normalform

- Intuition: nicht mehr als ein Konzept pro Relation modellieren!

Eine Relation $\mathcal{R}$ mit zugehörigen FDs $F_{\mathcal{R}}$ ist in 2NF, falls jedes Nichtschlüssel-Attribut $A \in \mathcal{R}$ voll funktional abhängig ist von jedem Kandidatenschlüssel der Relation.

- Seien $\kappa_1, ..., \kappa_i$ die Kandidatenschlüssel von $\mathcal{R}$
- Seien $A \in \mathcal{R} - (\kappa_1 \cup ... \cup \kappa_i)$
- Ein solches Attribut A wird als **nicht-prim** (Nichtschlüssel-Attribut) bezeichnet.
- Gegensatz: Schlüsselattribute bezeichnet man **prim**.
- Dann muss für **alle** Kandidatenschlüssel κ_j $(i \leq j \leq i)$ gelten:

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

Also mit anderen Worten:

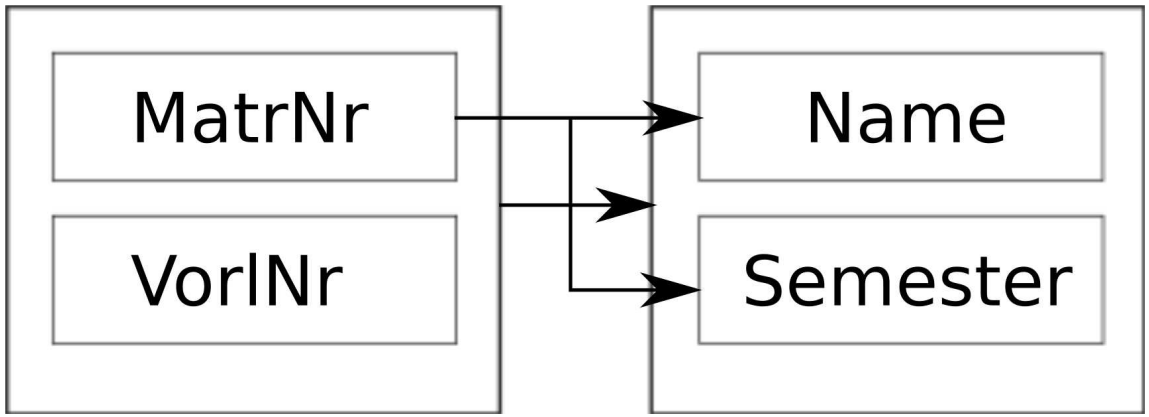
- A ist **voll funktional abhängig** von **jedem** κ_j
- κ_j muss bereits **linksreduziert** sein!
- 2NF verhindert **partielle** Abhängigkeiten

Gegenbeispiel

StudentenBelelgung			
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
...	...	...	...

- Kandidatenschlüssel {MatrNr, VorlNr}
- **prim:** {MatrNr, VorlNr}
- **nicht-prim:** {Name, Semester}
- StudentenBelegung ist **nicht** in 2NF:
  - {MatrNr} $\rightarrow$ {Name}, damit nicht **voll** funktional abhängig von {MatrNr, VorlNr}
  - {MatrNr} $\rightarrow$ {Semester}, damit nicht **voll** funktional abhängig von {MatrNr, VorlNr}
- Abhängigkeit zu **Teilschlüssel!**

Änderungsanomalien im Gegenbeispiel



- **Einfügeanomalie:** Was macht man mit Studenten, die keine Vorlesungen hören?
- **Updateanomalien:** Wenn z.B. Carnap ins vierte Semester kommt, muss man sicherstellen, dass alle vier Tupel geändert werden.
- **Löschanomalie:** Was passiert wenn Fichte seine einzige Vorlesung absagt?
- Zerlegung in zwei Relationen:
  - hoeren: {[MatrNr, VorlNr]}
  - Studenten: {[MatrNr, Name, Semester]}
- Beide Relationen sind in **2NF**.

Dritte Normalform

- Intuition: Nicht-Schlüssel Attribut darf kein anderes Nicht-Schlüssel Attribut bestimmen.

Ein Relationenschema $\mathcal{R}$ ist in dritter Normalform, wenn für jede für $\mathcal{R}$ geltende FD der Form $\alpha \rightarrow B$ mit Attribut $B \in \mathcal{R}$ mindestens eine von drei Bedingungen gilt:

1. $B \in \alpha$, d.h. die FD ist **trivial**
2. α is **Superschlüssel** von R
3. B ist **prim**

Eigenschaften:

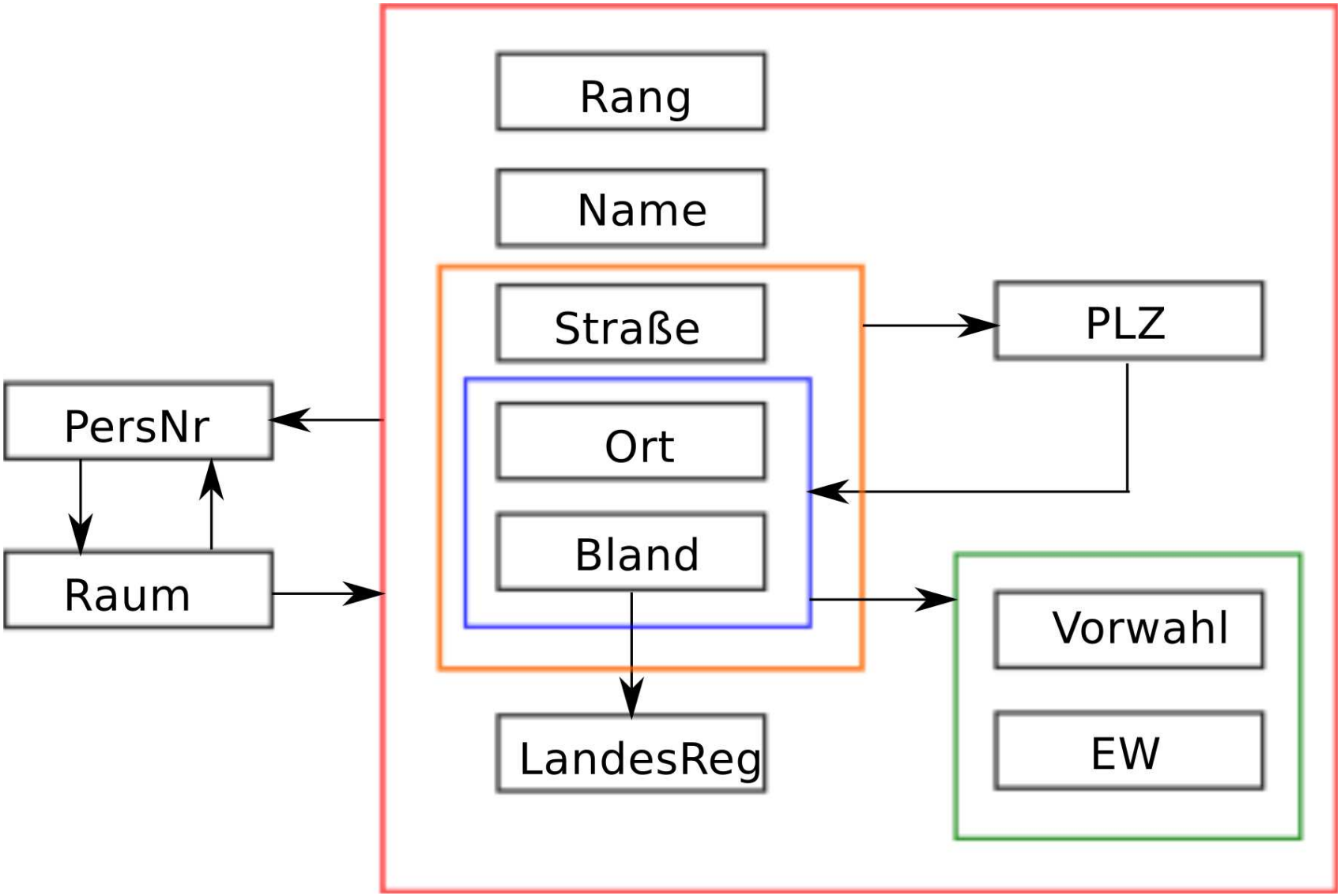
- 3NF verhindert **partielle und transitive** Abhängigkeiten
- 3NF $\Rightarrow$ 2NF

Gegenbeispiel: Nicht 2NF

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
...	...	...	...

- Kandidatenschlüssel {MatrNr, VorlNr}
- {MatrNr} $\rightarrow$ {Name}
  - nicht trivial
  - MatrNr kein Superschlüssel
  - Name ist nicht prim
  - Also: **nicht in 3NF**
- analog für {MatrNr} $\rightarrow$ {Semester}
- Abhängigkeit zu **Teilschlüssel durch Superschlüssel-Bedingung in 3NF-Definition abgefragt!**

Gegenbeispiel: 2NF aber nicht 3NF



Gegenbeispiel: 2NF aber nicht 3NF

- ProfessorenAdr: {[PersNr, Name, Rang, Raum, Ort, Straße, PLZ, Vorwahl, Bland, EW, Landesregierung]}
- Kandidatenschlüssel: PersNr und Raum
- FDs:
  1. {PersNr} → {Name, Rang, Raum, Ort, Straße, Bland}
  2. {Raum} → {PersNr}
  3. {Straße, Bland, Ort} → {PLZ}
  4. {Ort, Bland} → {EW, Vorwahl}
  5. {Bland} → {Landesregierung}
  6. {PLZ} → {Bland, Ort}

Warum ist diese Relation in 2NF?

Warum ist diese Relation nicht in 3NF?

Gegenbeispiel

- ProfessorenAdr: {[PersNr, Name, Rang, Raum, Ort, Straße, PLZ, Vorwahl, Bland, EW, Landesregierung]}
- Kandidatenschlüssel: PersNr und Raum
- FDs:
  1. {PersNr} → {Name, Rang, Raum, Ort, Straße, Bland}
  2. {Raum} → {PersNr}
  3. {Straße, Bland, Ort} → {PLZ}
  4. {Ort, Bland} → {EW, Vorwahl}
  5. {Bland} → {Landesregierung}
  6. {PLZ} → {Bland, Ort}

- FD1 ist OK: PersNr ist Superschlüssel, Raum prim
- FD2 ist OK: Raum ist Superschlüssel, PersNr prim
- FD3-6 nicht OK: weder trivial, noch Superschlüssel, noch prim.

Synthesealgorithmus

Algorithmus ermittelt zu einem gegebenen Relationenschema $\mathcal{R}$ mit funktionalen Abhängigkeiten F eine Zerlegung in $\mathcal{R}_1, ..., \mathcal{R}_n$ die alle drei folgenden Kriterien erfüllt:

1. $\mathcal{R}_1, ..., \mathcal{R}_n$ ist eine verlustlose Zerlegung von $\mathcal{R}$.
2. Die Zerlegung $\mathcal{R}_1, ..., \mathcal{R}_n$ ist abhängigkeitserhaltend.
3. Alle $\mathcal{R}_1, ..., \mathcal{R}_n$ sind in 3NF.

Synthesealgorithmus

1. Bestimme die kanonische Überdeckung F_c zu F. Wiederholung:
   (i) Linksreduktion
   (ii) Rechtsreduktion
   (iii) Entfernung von FDs der Form $\alpha \to \emptyset$
   (iv) Zusammenfassung gleicher linker Seiten
2. Für jede funktionale Abhängigkeit $\alpha \to \beta \in F_c$:
   - Kreiere ein Relationenschema $\mathcal{R}_\alpha := \alpha \cup \beta$
   - Ordne $\mathcal{R}_\alpha$ die FDs $F_\alpha := \{\alpha' \to \beta' \in F_c | \alpha' \cup \beta' \subseteq \mathcal{R}_\alpha\}$ zu.
3. Falls eines der in Schritt 2 erzeugten Schemata einen Kandidatenschlüssel von $\mathcal{R}$ bzgl. F_c enthält, sind wir fertig. Sonst wähle einen Kandidatenschlüssel $\kappa \subseteq \mathcal{R}$ aus und definiere folgendes Schema:
   - $\mathcal{R}_\kappa := \kappa$
   - $F_\kappa := \emptyset$
4. Eliminiere diejenigen Schemata $\mathcal{R}_\alpha$, die in einem anderen Relationenschema $\mathcal{R}_{\alpha'}$ enthalten sind, d.h., $\mathcal{R}_\alpha \subseteq \mathcal{R}_{\alpha'}$

Anwendung: Schritt 1

ProfessorenAdr: {[PersNr, Name, Rang, Raum, ort, Straße, PLZ, Vorwahl, Bland, EW, Landesregierung]}

Schritt 1 (kanonische Überdeckung) enthält die FDs:

1. {PersNr} → {Name, Rang, Raum, Ort, Straße, Bland}
2. {Raum} → {PersNr}
3. {Straße, Bland, Ort} → {PLZ}
4. {Ort, Bland} → {EW, Vorwahl}
5. {Bland} → {Landesregierung}
6. {PLZ} → {Bland, Ort}

Anwendung: Schritt 2

Schritt 2: Aus den FDs Relationen erzeugen

1. $\{\text{PersNr}\} \rightarrow \{\text{Name, Rang, Raum, Ort, Straße, Bland}\}$
   Professoren: $\{[\text{PersNr, Name, Rang, Raum, Ort, Straße, Bland}]\}$
2. $\{\text{Raum}\} \rightarrow \{\text{PersNr}\}$
   ProfessorenRäume: $\{[\text{Raum, PersNr}]\}$
3. $\{\text{Straße, Bland, Ort}\} \rightarrow \{\text{PLZ}\}$
   PLZverzeichnis: $\{[\text{Straße, Bland, Ort, PLZ}]\}$
4. $\{\text{Ort, Bland}\} \rightarrow \{\text{EW, Vorwahl}\}$
   StädteVerzeichnis: $\{[\text{Ort, Bland, EW, Vorwahl}]\}$
5. $\{\text{Bland}\} \rightarrow \{\text{Landesregierung}\}$
   Regierung: $\{[\text{Bland, Landesregierung}]\}$
6. $\{\text{PLZ}\} \rightarrow \{\text{Bland, Ort}\}$
   PLZverzeichnis2: $\{[\text{PLZ, Bland, Ort}]\}$

Anwendung: Schritt 3

Schritt 3: Neue Relation falls keine Rel. mit Kand.-Schlüssel:

1. $\{\text{PersNr}\} \rightarrow \{\text{Name, Rang, Raum, Ort, Straße, Bland}\}$
   Professoren: {[**PersNr**, Name, Rang, **Raum**, Ort, Straße, Bland]}
2. $\{\text{Raum}\} \rightarrow \{\text{PersNr}\}$
   ProfessorenRäume: {[**Raum**, **PersNr**]}
3. $\{\text{Straße, Bland, Ort}\} \rightarrow \{\text{PLZ}\}$
   PLZverzeichnis: {[Straße, Bland, Ort, PLZ]}
4. $\{\text{Ort, Bland}\} \rightarrow \{\text{EW, Vorwahl}\}$
   StädteVerzeichnis: {[Ort, Bland, EW, Vorwahl]}
5. $\{\text{Bland}\} \rightarrow \{\text{Landesregierung}\}$
   Regierung: {[Bland, Landesregierung]}
6. $\{\text{PLZ}\} \rightarrow \{\text{Bland, Ort}\}$
   PLZverzeichnis2: {[PLZ, Bland, Ort]}

- **Raum** und **PersNr** sind beide Kandidatenschlüssel.
- Schritt 3 erzeugt für dieses Beispiel keine neue Relation.

Anwendung: Schritt 4

Schritt 2: Aus den FDs Relationen erzeugen

1. Professoren: {[**PersNr**, Name, Rang, **Raum**, Ort, Straße, Bland]}
2. ~~ProfessorenRäume: {[**Raum**, **PersNr**]}~~ $\subseteq$ **Professoren**
3. PLZverzeichnis: {[Straße, Bland, Ort, PLZ]}
4. StädteVerzeichnis: {[Ort, Bland, EW, Vorwahl]}
5. Regierung: {[Bland, Landesregierung]}
6. ~~PLZverzeichnis2: {[PLZ, Bland, Ort]}~~ $\subseteq$ **PLZverzeichnis**

Fertig!

Anderes Beispiel für Schritt 3

- StudentenBelegung(**MatrNr**, **VorlNr**, Name, Semester)
- kanonische Überdeckung hierfür:

$$\{\text{MatrNr}\} \rightarrow \{\text{Name, Semester}\}$$

- daraus folgt die Relation:

Student(MatrNr , Name, Semester)

- **Student** enthält keinen Kandidatenschlüssel.
- deswegen Schritt 3: **hören(MatrNr, VorlNr)**
- Ansonsten würde die Information wer welche VL hört wegfallen.

Wiederholung: Dritte Normalform

- Intuition: Nicht-Schlüssel Attribut darf kein anderes Nicht-Schlüssel Attribut bestimmen.

Ein Relationenschema $\mathcal{R}$ ist in dritter Normalform, wenn für jede für $\mathcal{R}$ geltende FD der Form $\alpha \rightarrow B$ mit Attribut $B \in \mathcal{R}$ mindestens eine von drei Bedingungen gilt:

1. $B \in \alpha$, d.h. die FD ist **trivial**
2. α is **Superschlüssel** von R
3. B ist **prim**

Eigenschaften:

- 3NF verhindert **partielle und transitive** Abhängigkeiten
- 3NF $\Rightarrow$ 2NF

Boyce-Codd-Normalform

Die Boyce-Codd-Normalform (BCNF) ist nochmals eine Verschärfung der 3NF. **Intuition:** Jedes Attribut darf **nur** den gesamten Schlüssel beschreiben und nichts anderes.

Ein Relationenschema $\mathcal{R}$ mit FDs F ist in BCNF, wenn für jede für $\mathcal{R}$ geltende funktionale Abhängigkeit der Form $\alpha \rightarrow B$ mit Attribut $B \in \mathcal{R}$ mindestens **eine** von zwei Bedingungen gilt:

1. $B \in \alpha$, d.h. die FD ist **trivial**
2. α ist Superschlüssel von $\mathcal{R}$

- Unterschied zu 3NF: 3. Bedingung fällt weg (B ist prim).
- Man kann jede Relation **verlustlos** in BCNF-Relationen zerlegen
- **Aber:** manchmal lässt sich dabei die Abhängigkeitserhaltung nicht erziehlen!

Städte ist in 3NF, aber nicht in BCNF

- Städte: {[Ort, Bland, Ministerpräsident/in, EW]}
- Geltende FDs:
  1. {Ort, Bland} → {EW}
  2. {Bland} → {Ministerpräsident/in}
  3. {Ministerpräsident/in} → {Bland}
- Kandidatenschlüssel:
  - {Ort, Bland}
  - {Ort, Ministerpräsident/in}
- linke Seite von FD1 ist Superschlüssel (Bedingung 2)
- rechten Seiten von FD2 und FD3 sind prim (Bedingung 3)
- Also: **3NF**
- linke Seiten von FD2 und FD3 sind aber **kein Superschlüssel**
- Also: **nicht in BCNF**
- **D.h.: BCNF verhindert zusätzlich zu 3NF transitive Abhängigkeiten zu Schlüssel-Attributen**
- wer welches Bundesland regiert wird hier also mehrfach abgespeichert

Dekomposition

Man kann grundsätzlich jedes Relationenschema $\mathcal{R}$ mit funktionalen Abhängigkeiten F so in $\mathcal{R}_1, ..., \mathcal{R}_n$ zerlegen, dass gilt

- $\mathcal{R}_1, ..., \mathcal{R}_n$ ist eine verlustlose Zerlegung von $\mathcal{R}$
- Alle $\mathcal{R}_1, ..., \mathcal{R}_n$ sind in BCNF.

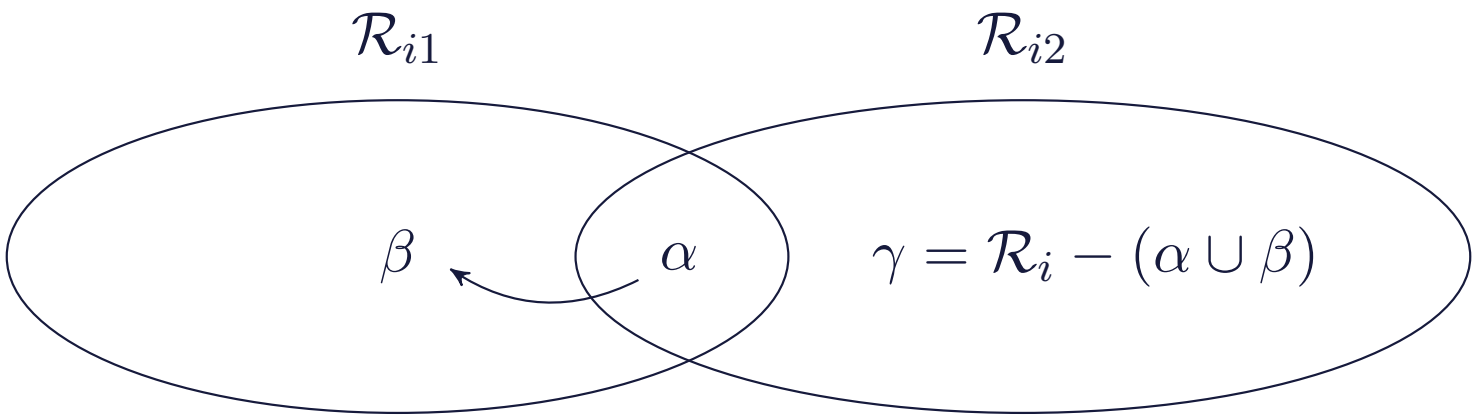
Es kann allerdings nicht immer erreicht werden, dass die Zerlegung $\mathcal{R}_1, ..., \mathcal{R}_n$ **abhängigkeitserhaltend** ist

Dies ist in der Praxis selten.

Dekompositions-Algorithmus für BCNF

- Starte mit $Z = \{\mathcal{R}\}$
- Solange es noch ein Relationenschame $\mathcal{R}_i \in Z$ gibt, das nicht in BCNF ist, mache folgendes:
  1. Es gibt also eine für $\mathcal{R}_i$ geltende nicht-triviale funktionale Abhängigkeit $\alpha \to \beta$ mit:
     - $\alpha \cap \beta = \emptyset$ (α und β sind disjunkt)
     - $\neg(\alpha \to \mathcal{R}_i)$ (α kein Superschlüssel von $\mathcal{R}_i$)
  2. Finde eine solche FD:
     Man sollte sie so wählen, dass β alle von α funktionalen abhängigen Attribute $B \in (\mathcal{R}_i - \alpha)$ enthält, damit der Dekompositionsalgorithmus möglichst schnell terminiert.
  3. Zerlege $\mathcal{R}_i$ in $\mathcal{R}_{i1} := \alpha \cup \beta$ und $\mathcal{R}_{i2} := \mathcal{R}_i - \beta$
  4. Entferne $\mathcal{R}_i$ aus Z und füge $\mathcal{R}_{i1}$ und $\mathcal{R}_{i2}$ ein:
     $Z := (Z - \mathcal{R}_i) \cup \mathcal{R}_{i1} \cup \mathcal{R}_{i2}$

Veranschaulichung eines Dekompositionsschrittes



Dekomposition der Relation Städte in BCNF-Relationen

- Städte: {[Ort, Bland, Ministerpräsident/in, EW]}
- Geltende FDs:
  1. {Ort, Bland} → {EW}
  2. {Bland} → {Ministerpräsident/in}
  3. {Ministerpräsident/in} → {Bland}

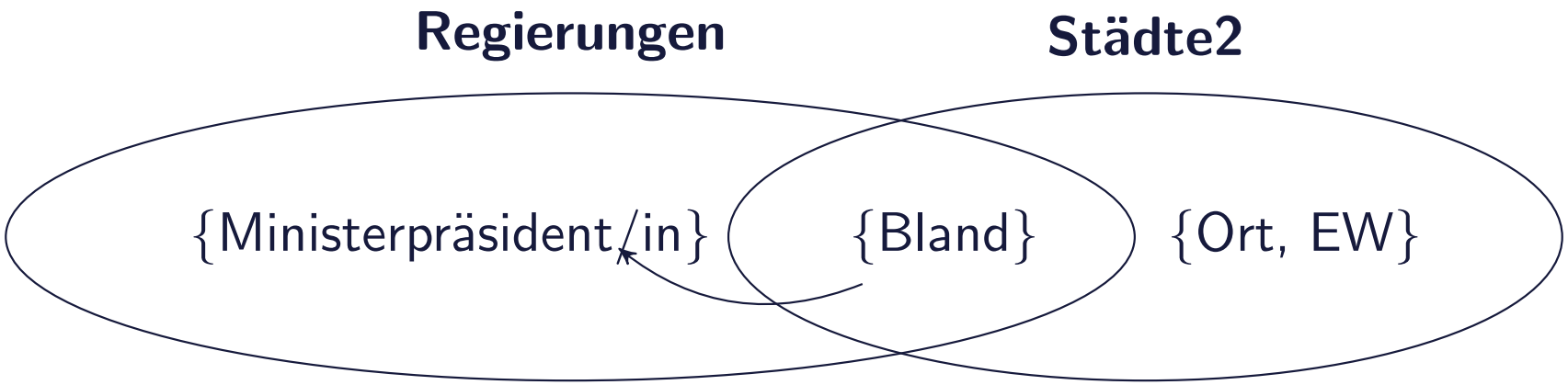
$\mathcal{R}_{i1}$

- **Regierungen:** {[Bland, Ministerpräsident/in]}

$\mathcal{R}_{i2}$

- **Städte:** {[Ort, Bland, EW]}

Veranschaulichung



- Zerlegung ist verlustlos und auch abhängigkeitserhaltend.
- **warum?**

Weiteres Beispiel: Dekomposition des PLZVerzeichnis in BCNF-Relationen

PLZVerzeichnis: {[Straße, Ort, Bland, PLZ]}

1. {Straße, Ort, Bland} $\rightarrow$ {PLZ} OK
2. {PLZ} $\rightarrow$ {Ort, Bland} verletzt BCNF **warum?**

Betrachte nun die Zerlegung:

- Orte: {[PLZ, Ort, Bland]}
- Straßen: {[PLZ, Straße]}

Diese Zerlegung

- ist verlustlos, **aber**
- **nicht abhängigkeitserhaltend, warum?**

Mehrwertige Abhängigkeiten

Bislang:

Funktionale Abhängigkeiten der Form

- Ausprägung R
- Seien $\alpha \subseteq \mathcal{R}$ und $\beta \subseteq \mathcal{R}$
- $\alpha \rightarrow \beta$ genau dann, wenn $\forall r, s \in R$ gilt: $r.\alpha = s.\alpha \Rightarrow r.\beta = s.\beta$
- D.h. die α-Werte bestimmen die β-Werte funktional (=eindeutig)

Nun: Mehrwertige Abhängigkeiten (multivalued dependencies)

Notation: $\alpha \twoheadrightarrow \beta$
Falls einem Attributwert α eine **Menge** von β-Werten zugeordnet werden.

Genaue Defintion folgt (gleich).

Mehrwertige Abhängigkeiten: Beispiel

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

Mehrwertige Abhängigkeiten dieser Relation:

- $\{\text{PersNr}\} \twoheadrightarrow \{\text{Sprache}\}$ und
- $\{\text{PersNr}\} \twoheadrightarrow \{\text{ProgSprache}\}$

MVDs führen zu Redundanz und Anomalien

Mehrwertige Abhängigkeiten

R		
A	B	C
a	b	c
a	bb	cc
a	b	cc
a	bb	c

- $A \twoheadrightarrow B$
- $A \twoheadrightarrow C$

Bei zwei Tupeln mit gleichen α-Werten kann man die β-Werte vertauschen und die resultierenden Tupel müssen auch in der Relation sein.

Mehrwertige Abhängigkeiten: Definition 1

	R		
	$\overbrace{A_1, ..., A_i}^{\alpha}$	$\overbrace{A_{i+1}, ..., A_j}^{\beta}$	$\overbrace{A_{j+1}, ..., A_n}^{\gamma}$
t_1	$a_1, ..., a_i$	$a_{i+1}, ..., a_j$	$a_{j+1}, ..., a_n$
t_2	$a_1, ..., a_i$	$b_{i+1}, ..., b_j$	$b_{j+1}, ..., b_n$
t_3	$a_1, ..., a_i$	$a_{i+1}, ..., a_j$	$b_{j+1}, ..., b_n$
t_4	$a_1, ..., a_i$	$b_{i+1}, ..., b_j$	$a_{j+1}, ..., a_n$

$\alpha \twoheadrightarrow \beta$ gilt genau dann, wenn für jede Ausprägung von R gilt:

- wenn es zwei Tupel t_1 und t_2 mit gleichen α-Werten gibt, dann muss es auch zwei Tupel t_3 und t_4 geben mit
  - $t_1.\alpha = t_2.\alpha = t_3.\alpha = t_4.\alpha$ (alle α-Werte gleich)
  - $t_3.\beta = t_1.\beta$,$t_4.\beta = t_2.\beta$ (β-Paare gleich)
  - $t_3.\gamma = t_2.\gamma$, $t_4.\gamma = t_1.\gamma$ (γ-Paare **vertauscht**)

Veranschaulichung: Spezialfall

Veranschaulichung für MVD $\alpha \twoheadrightarrow \beta$:

Wenn α, β, γ jeweils nur aus einem Attribut A, B, und C bestehen:

Wenn $\{b_1, ..., b_i\}$ und $\{c_1, ..., c_j\}$ die B bzw. C-Werte für einen bestimmen A-Wert a sind, dann muss die Relation auch die folgenden $(i * j)$ Tupel enhalten:

$$\{a\} \times \{b_1, ..., b_i\} \times \{c_1, ..., c_j\}$$

Mehrwertige Abhängigkeiten: Definition 2

- Eine mehrwertige Abhängigkeit (multivalued dependency, MVD) $\alpha \twoheadrightarrow \beta$ besagt, dass einem Attribut α in $\mathcal{R}$ eine **Menge** von β-Werten zugeordnet werden.
- Wenn die MVD $\alpha \twoheadrightarrow \beta$ in R erfüllt, dann kann es als Erweiterung zu FDs α, β, c geben mit

$$|\pi_\beta(\sigma_{\alpha=c}(R))| > 1$$

- Diese Zuordnung ist **unabhängig** von den restlichen Attributen in $\mathcal{R}$

Mit anderen Worten:

- α bestimmt **nicht nur** einen **einzelnen** Wert (ein singleton)
- genau das ist ja bei einer normalen FD $\alpha \to \beta$ der Fall!
- **sondern** eine Menge von Werten
- diese Wertemenge ist unabhängig von den anderen Attributen in $\gamma = \mathcal{R} - \alpha - \beta$

Natürlich: Jede FD ist auch eine MVD (aber nicht umgekehrt)

Beispiel

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

$\pi_{PersNr,Sprache}$

Sprachen	
PersNr	Sprache
3002	griechisch
3002	lateinisch
3005	deutsch

$\pi_{PersNr,ProgSprache}$

ProgSprachen	
PersNr	ProgSprache
3002	C
3002	Pascal
3005	Ada

Verlustlose Zerlegung bei MVDs: hinreichende + notwendige Bedingung

Die Zerlegung von $\mathcal{R}$ in $\mathcal{R}_1$ und $\mathcal{R}_2$ ist verlustlos **genau dann, wenn**

- $\mathcal{R} = \mathcal{R}_1 \cup \mathcal{R}_2$
- **und** mindestens eine von zwei MVDs gilt:
  - $\mathcal{R}_1 \cap \mathcal{R}_2 \twoheadrightarrow \mathcal{R}_1$ **oder**
  - $\mathcal{R}_1 \cap \mathcal{R}_2 \twoheadrightarrow \mathcal{R}_2$

Für unser Beispiel gilt:

- {PersNr, Sprache, ProgrSprache} = {PersNr, Sprache} ∪ {PersNr, ProgSprache}
- {PersNr} $\twoheadrightarrow$ {Sprache}
- {PersNr} $\twoheadrightarrow$ {ProgSprache}

D.h. es gelten sogar beide MVDs!

MVDs in Paaren

- es gilt zusätzlich: wenn $\alpha \twoheadrightarrow \beta$, dann gilt immer
  - $\alpha \twoheadrightarrow \gamma$
  - mit $\gamma = \mathcal{R} - \alpha - \beta$
- D.h. MVDs treten immer als Paare auf
- wir könnten MVDs deshalb auch so notieren: $\alpha \twoheadrightarrow \beta | \gamma$

Triviale MVDs ...

- .... sind solche, die von jeder Relationenausprägung R von $\mathcal{R}$ erfüllt werden.
- $\alpha \cup \beta \subseteq \mathcal{R}$
- Eine MVD $\alpha \twoheadrightarrow \beta$ ist trivial genau dann, wenn
  1. $\beta \subseteq \alpha$ oder
  2. $\beta = \mathcal{R} - \alpha$
- **Nur** die Bedigung 1 galt auch für normale FDs.
- Beispiel für Bedingung 2:
  - $\mathcal{R} = \{PersNr, Sprache\}$
  - $\alpha = \{PersNr\}$
  - $\beta = \{Sprache\}$
  - $\mathcal{R} - \alpha = \{PersNr, Sprache\} - \{PersNr\} = \{Sprache\} = \beta \Rightarrow$ MVD ist trivial!

Vierte Normalform

Eine Relation $\mathcal{R}$ ist in 4NF, wenn für jede MVD $\alpha \twoheadrightarrow \beta$ eine der folgenden Bedingungen gilt:

- Die MVD ist trivial **oder**
- α ist Superschlüssel von $\mathcal{R}$

- D.h. 4NF ist sehr ähnlich zu BCNF!
- Unterschied:
  - MVDs statt FDs
  - Definition von "trivial" wurde erweitert.
- 4NF erfüllt $\Rightarrow$ BCNF erfüllt, da jede FD eine MVD ist.

Dekomposition in 4NF

Starte mit der Menge $Z := \{\mathcal{R}\}$

Solange es noch ein Relationenschema $\mathcal{R}_i$ in Z gibt, dass nicht in 4NF ist, mache folgendes:

- Es gibt also eine für $\mathcal{R}_i$ geltende nicht-triviale MVD $(\alpha \twoheadrightarrow \beta)$, für die gilt:
  - $\alpha \cap \beta = \emptyset$
  - $\neg(\alpha \rightarrow \mathcal{R}_i)$
- Finde eine solche MVD
- Zerlege $\mathcal{R}_i$ in $\mathcal{R}_{i1} := \alpha \cup \beta$ und $\mathcal{R}_{i2} := \mathcal{R}_i - \beta$
- Entferne $\mathcal{R}_i$ aus Z und füge $\mathcal{R}_{i1}$ und $\mathcal{R}_{i2}$ ein, also $Z := (Z - \mathcal{R}_i) \cup \{\mathcal{R}_{i1}\} \cup \{\mathcal{R}_{i2}\}$

Zusammenfassung

- Die Verlustlosigkeit ist für alle Zerlegungsalgorithmen in alle Normalformen garantiert.
- Die Abhängigkeitserhaltung kann nur bis zur dritten Normalform garantiert werden.