Informationssysteme

Sommersemester 2016

Prof. Dr.-Ing. Sebastian Michel
TU Kaiserslautern

smichel@cs.uni-kl.de

Wiederholung: Normalisierung von Relationen

Um Qualitätsprobleme im ursprünglichen Entwurf 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 $\mathcal{F}_{\mathcal{R}}$ sollten in den $\mathcal{F}_{\mathcal{R}_1}, \mathcal{F}_{\mathcal{R}_2}, ...\mathcal{F}_{\mathcal{R}_n}$ bewahrt bleiben.

Wiederholung: 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 $\mathcal{R}_2$
- $\mathcal{F}_\mathcal{R}$ ist die Menge der FDs in $\mathcal{R}$
- **Zerlegung ist verlustlos, wenn mindestens eine der folgenden FDs herleitbar ist:**
  - $(\mathcal{R}_1 \cap \mathcal{R}_2) \rightarrow \mathcal{R}_1 \in \mathcal{F}_\mathcal{R}^+$ **oder** d.h. Schlüssel bestimmt $\mathcal{R}_1$
  - $(\mathcal{R}_1 \cap \mathcal{R}_2) \rightarrow \mathcal{R}_2 \in \mathcal{F}_\mathcal{R}^+$ 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:
  - $\beta \subseteq \mathit{AttrHülle}(\mathcal{F}_\mathcal{R}, \alpha)$ **oder**
  - $\gamma \subseteq \mathit{AttrHülle}(\mathcal{F}_\mathcal{R}, \alpha)$

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

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

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

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

- Unser 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:
  - $\{Kneipe\} \subseteq \mathit{AttrHülle}(\{Kneipe, Gast\} \rightarrow \{Bier\}, \{Gast\})$ **oder**
  - $\{Bier\} \subseteq \mathit{AttrHülle}(\{Kneipe, Gast\} \rightarrow \{Bier\}, \{Gast\})$
- 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

- $\mathcal{R} = \{Vater, Mutter, Kind\}$
- $\mathcal{F}_{\mathcal{R}} = \{\{Kind\} \rightarrow \{Mutter\}, \{Kind\} \rightarrow \{Vater\}\}$
- $\mathcal{R}_1 = \{Vater, Kind\}$, $\mathcal{R}_2 = \{Mutter, Kind\}, \mathcal{R}_1 \cap \mathcal{R}_2 = \{Kind\}$

- Dann muss **eine** der Bedingungen gelten:
    - $\{Vater\} \subseteq$ *AttrHülle*$(\mathcal{F}_{\mathcal{R}}, \{Kind\})$ **oder**
    - $\{Mutter\} \subseteq$ *AttrHülle*$(\mathcal{F}_{\mathcal{R}}, \{Kind\})$

- 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 jedes mal den Join $R_1 \bowtie ... \bowtie R_n$ berechnen 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:
$\mathcal{F}_\mathcal{R} \equiv (\mathcal{F}_{\mathcal{R}_1} \cup ... \cup \mathcal{F}_{\mathcal{R}_n})$ bzw. $\mathcal{F}_\mathcal{R}^+ = (\mathcal{F}_{\mathcal{R}_1} \cup ... \cup \mathcal{F}_{\mathcal{R}_n})^+$
- 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

$\pi_{PLZ,\textit{Straße}}$

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 {PLZ} → {Orte, Bland} und PLZ das einzige gemeinsame Attribut.
- die FD {Straße, Ort, Bland} → {PLZ} ist im zerlegten Schema **nicht** mehr enthalten. Also: Zerlegung ist **nicht abhängigkeitserhaltend**.

$\pi_{PLZ,\textit{Straße}}$

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

<table>
<tr><th colspan="4">Eltern</th></tr>
<tr><th>Vater</th><th>Mutter</th><th colspan="2">Kinder</th></tr>
<tr><td rowspan="2">Johan</td><td rowspan="2">Martha</td><td>Else</td><td>5</td></tr>
<tr><td>Lucie</td><td>3</td></tr>
<tr><td rowspan="2">Johann</td><td rowspan="2">Maria</td><td>Theo</td><td>3</td></tr>
<tr><td>Josef</td><td>1</td></tr>
<tr><td>Heinz</td><td>Martha</td><td>Cleo</td><td>9</td></tr>
</table>

- Vorteil: keine unnötige Wiederholung (=Redundanz) von Vater und Mutter
- NF^2 ist 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 $\mathcal{F}_{\mathcal{R}}$ ist in 2NF, falls jedes Nichtschlüssel-Attribut $A \in \mathcal{R}$ voll funktional abhängig ist von jedem Kandidatenschlüssel der Relation.

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

$$\kappa_j \overset{\bullet}{\rightarrow} A \in \mathcal{F}^+$$

Also mit anderen Worten:

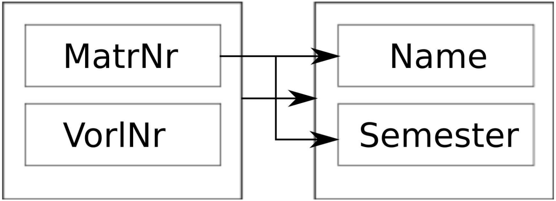
  - A ist **voll funktional abhängig** von **jedem** κ_j
  - κ_j muss bereits **linksreduziert** sein!
  - 2NF **verhindert partielle** Abhängigkeiten

Gegenbeispiel

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}**
- **prim:** {MatrNr, VorlNr}
- **nicht-prim:** {Name, Semester}
- StudentenBelegung ist **nicht in 2NF**. Wieso?
  - {MatrNr} → {Name}, damit **nicht voll** funktional abhängig von {MatrNr, VorlNr}
  - {MatrNr} → {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:

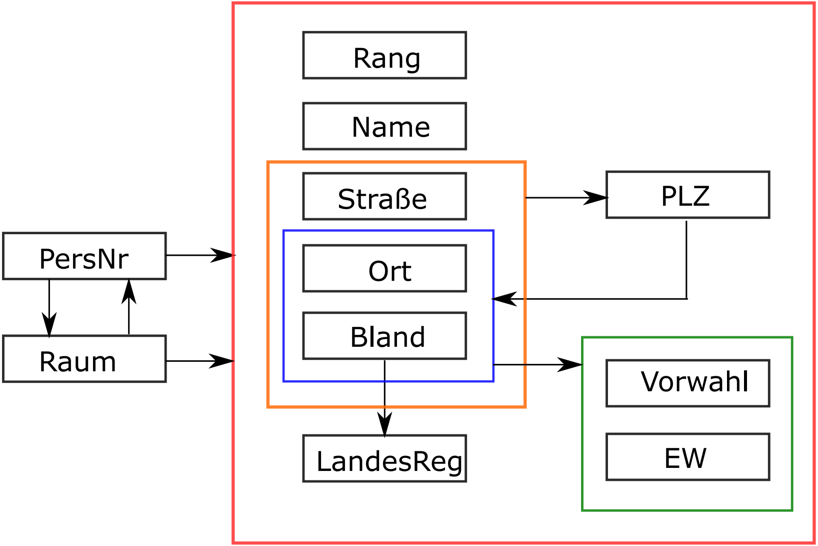
- **3NF verhindert partielle und transitive Abhängigkeiten**
- 3NF $\Rightarrow$ 2NF

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 $\mathcal{F}$ 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 $\mathcal{F}_c$ zu $\mathcal{F}$.** Wiederholung:
   (i) Linksreduktion
   (ii) Rechtsreduktion
   (iii) Entfernung von FDs der Form $\alpha \to \emptyset$
   (iv) Zusammenfassung gleicher linker Seiten
2. **Für jede funktionale Abhängigkeit $\alpha \to \beta \in \mathcal{F}_c$:**
   - Kreiere ein Relationenschema $\mathcal{R}_\alpha := \alpha \cup \beta$
   - Ordne $\mathcal{R}_\alpha$ die FDs $\mathcal{F}_\alpha := \{\alpha' \to \beta' \in \mathcal{F}_c | \alpha' \cup \beta' \subseteq \mathcal{R}_\alpha\}$ zu.
3. **Falls eines der in Schritt 2 erzeugten Schemata einen Kandidatenschlüssel von $\mathcal{R}$ bzgl. $\mathcal{F}_c$ 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} $\rightarrow$ {Name, Rang, Raum, Ort, Straße, Bland}
2. {Raum} $\rightarrow$ {PersNr}
3. {Straße, Bland, Ort} $\rightarrow$ {PLZ}
4. {Ort, Bland} $\rightarrow$ {EW, Vorwahl}
5. {Bland} $\rightarrow$ {Landesregierung}
6. {PLZ} $\rightarrow$ {Bland, Ort}

Anwendung: Schritt 2

Schritt 2: Aus den FDs Relationen erzeugen

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]}

Anwendung: Schritt 3

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

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