Kapitel 3: Das relationale DB-Modell & SQL

	Relationales Datenmodell (RDM)	Objekt-orientierte Datenmodelle (OODM)	Objekt-relationale Datenmodelle	Semi-strukturierte Datenmodelle
Überblick über die Konzepte				
Darstellung von Assoziationen				
Datendefinition	**3.1**	**4.1**	**5.1**	**6.1**
Anfragen				**6.2**
Aktualisierungs-operationen				
Spezifika	**3.2 SQL**	**4.2 ODMG**	**5.2, 5.3**	

RDM: Anfragen

Relationale Anfragesprachen im Überblick:

geschachtelte Mengenausdrücke

Relationale Algebra

Relationales Kalkül

mengenorientierte, deklarative, quantifizierte Ausdrücke

Tupelkalkül, Relationenkalkül

Domänenkalkül

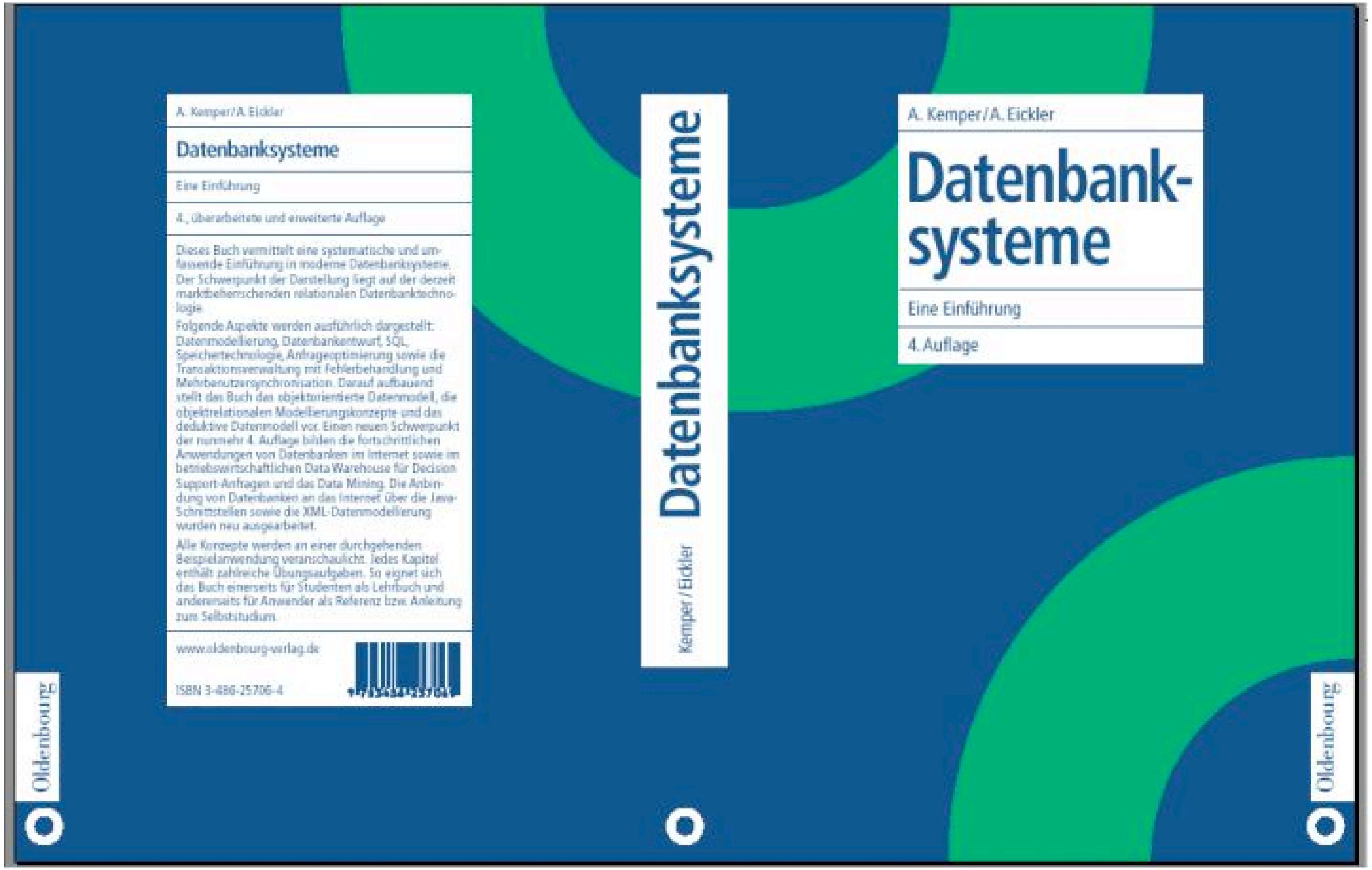
Binäre Operationen:
- Vereinigung $R \cup S$
- Differenz $R \setminus S$
- Durchschnitt $R \cap S$
- Kartesisches Produkt $R \times S$
- Join (Verbindung) $R >< S$

Unäre Operationen:
- Projektion $\pi_x(R)$
- Selektion $\sigma_\varphi(R)$

SQL

```
select ...
from ...
where ...
...
```

Acknowledgments



Bestimmung funktionaler Abhängigkeiten

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

- {PersNr} → {PersNr, Name, Rang, Raum, Ort, Straße, PLZ, Vorwahl, Bland, EW, Landesregierung}
- {Ort,BLand} → {EW, Vorwahl}
- {PLZ} → {Bland, Ort, EW}
- {Bland, Ort, Straße} → {PLZ}
- {Bland} → {Landesregierung}
- {Raum} → {PersNr}

Zusätzliche Abhängigkeiten, die aus obigen abgeleitet werden können:

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

Herleitung funktionaler Abhängigkeiten: Armstrong-Axiome

Reflexivität

- Falls □ eine Teilmenge von □ ist (□ □ □) dann gilt immer □ □ □.
  Insbesondere gilt immer □ □ □.

Verstärkung

- Falls □ □ □ gilt, dann gilt auch □□□ □□. Hierbei stehe z.B. □□ für □ □ □.

Transitivität

- Falls □ □ □ und □ □ □ gilt, dann gilt auch □ □ □.

Diese drei Axiome sind vollständig und korrekt (ohne Beweis).
Zusätzliche Axiome erleichtern die Herleitung:

- Vereinigungsregel:
  - Wenn □ □ □ und □ □ □ gelten, dann gilt auch □ □ □□.
- Dekompositionsregel:
  - Wenn □ □ □□ gilt, dann gelten auch □ □ □ und □ □ □
- Pseudotransitivitätsregel:
  - Wenn □ □ □ und □□ □ □, dann gilt auch □□□ □

Notation

Funktionale Abhängigkeiten (functional dependencies) werden auch als FDs bezeichnet.

Bestimmung der Hülle einer Attributmenge

Eingabe: eine Menge F von FDs und eine Menge von Attributen □.

Ausgabe: die vollständige Menge von Attributen □+, für die gilt □ → □+.

AttrHülle(F,□)

❑ Erg := □

❑ **While** (Änderungen an Erg) **do**

    Foreach FD □□ □**in** F **do**

        If □□ Erg **then** Erg := Erg □ □

❑ Ausgabe □+ = Erg

Kanonische Überdeckung

F_c heißt kanonische Überdeckung von F, wenn die folgenden drei Kriterien erfüllt sind:

1. $F_c \equiv F$, d.h. $F_c^+ = F^+$
2. In F_c existieren keine FDs , die überflüssige Attribute enthalten. D.h. es muß folgendes gelten:
   - $\forall A \in \alpha: (F_c - (\alpha \rightarrow \beta) \cup ((\alpha - \{A\}) \rightarrow \beta)) \not\equiv F_c$
   - $\forall B \in \beta: (F_c - (\alpha \rightarrow \beta) \cup (\alpha \rightarrow (\beta - \{B\}))) \not\equiv F_c$
3. 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.

Berechnung der kanonischen Überdeckung

Führe für jede FD □ □ □ □ F die Linksreduktion durch, also:

- ❑ Überprüfe für alle A □ □, ob A überflüssig ist, d.h., ob

  □ □ □ AttrHülle(F, □ - A)

  gilt. Falls dies der Fall ist, ersetze □ □ □ durch (□ - A) □ □.

Führe für jede (verbliebene) FD die Rechtsreduktion durch, also:

- ❑ Überprüfe für alle B □ □, ob
  - B □ AttrHülle(F – (□ □ □) □ (□ □ (□ □ □)), □)

    gilt. Falls dies der Fall ist, ist B auf der rechten Seite überflüssig und kann eliminiert werden, d.h. ersetze □ $\rightarrow$ □ durch □ $\rightarrow$ (□–B).

Entferne die FDs der Form □ $\rightarrow \emptyset$, die im 2. Schritt möglicherweise entstanden sind.

Fasse mittels der Vereinigungsregel FDs der Form □ $\rightarrow$ □1, ..., □ $\rightarrow$ □n zusammen, so dass □ $\rightarrow$ (□1 □ ... □ □n) verbleibt.

„Schlechte“ Relationenschemata

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

Update-Anomalien

- ❑ Sokrates zieht um, von Raum 226 in R. 338. Was passiert?

Einfüge-Anomalien

- ❑ Neue/r Prof ohne Vorlesungen?

Löschanomalien

- ❑ Letzte Vorlesung einer/s Profs wird gelöscht? Was passiert?

Zerlegung (Dekomposition) von Relationen

Es gibt zwei Korrektheitskriterien für die Zerlegung von Relationenschemata:

1. Verlustlosigkeit
   - Die in der ursprünglichen Relationenausprägung R des Schemas $\mathcal{R}$ enthaltenen Informationen müssen aus den Ausprägungen $R1$, ..., Rn der neuen Relationenschemata $\mathcal{R}1$, .., $\mathcal{R}n$ rekonstruierbar sein.

2. Abhängigkeitserhaltung
   - Die für $\mathcal{R}$ geltenden funktionalen Anhängigkeiten müssen auf die Schemata $\mathcal{R}1$, ..., $\mathcal{R}n$ übertragbar sein.

Kriterien für die Verlustlosigkeit einer Zerlegung

$\mathcal{R} = \mathcal{R}1 \cup \mathcal{R}2$

- ❑ $R1 := \pi_{\mathcal{R}1}(R)$
- ❑ $R2 := \pi_{\mathcal{R}2}(R)$

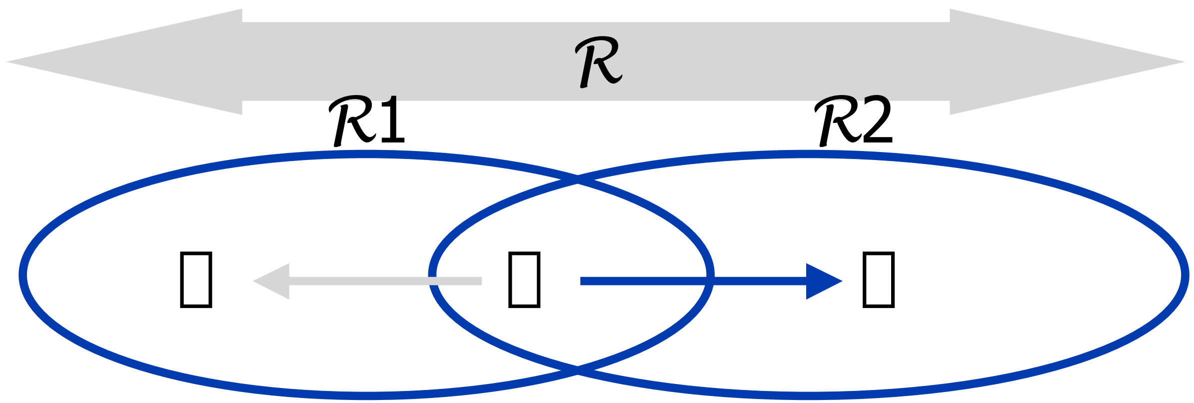
Eine Zerlegung von $\mathcal{R}$ in $\mathcal{R}1$ und $\mathcal{R}2$ ist verlustlos,

falls für jede mögliche (gültige) Ausprägung R von $\mathcal{R}$ gilt:

- ❑ R = R1 |><| R2

Hinreichende Bedingung für die Verlustlosigkeit einer Zerlegung

- ❑ $(\mathcal{R}1 \cap \mathcal{R}2) \rightarrow \mathcal{R}1$ oder
- ❑ $(\mathcal{R}1 \cap \mathcal{R}2) \rightarrow \mathcal{R}2$



Biertrinker-Beispiel

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

„Verlustige“ Zerlegung

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

□ Kneipe, Gast

□ Gast, Bier

Besucht	
Kneipe	*Gast*
Kowalski	Kemper
Kowalski	Eickler
Innsteg	Kemper

Trinkt	
Gast	*Bier*
Kemper	Pils
Eickler	Hefeweizen
Kemper	Hefeweizen

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

Besucht	
Kneipe	*Gast*
Kowalski	Kemper
Kowalski	Eickler
Innsteg	Kemper

 ....

Trinkt	
Gast	*Bier*
Kemper	Pils
Eickler	Hefeweizen
Kemper	Hefeweizen

|><|

≠

Besucht A Trinkt		
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 eine „verlustige“ Zerlegung und dementsprechend war die hinreichende Bedingung verletzt. Es gilt nämlich nur die eine nicht-triviale funktionale Abhängigkeit

- {Kneipe,Gast}→{Bier}

Wohingegen keine der zwei möglichen, die Verlustlosigkeit garantierenden FDs gelten

- {Gast}→{Bier}
- {Gast}→{Kneipe}

Das liegt daran, dass die Leute (insbes. Kemper) in unterschiedlichen Kneipen unterschiedliches Bier trinken. In derselben Kneipe aber immer das gleiche Bier

- (damit sich die KellnerInnen darauf einstellen können?)

Verlustfreie 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

Erläuterung der verlustfreien Zerlegung der Eltern-Relation

Eltern: {[Vater, Mutter, Kind]}

Väter: {[Vater, Kind]}

Mütter: {[Mutter, Kind]}

Verlustlosigkeit ist garantiert

Es gilt nicht nur eine der hinreichenden FDs, sondern gleich beide

- {Kind}→{Mutter}
- {Kind}→{Vater}

Also ist {Kind} natürlich auch der Schlüssel der Relation Eltern

Die Zerlegung von Eltern ist zwar verlustlos, aber auch ziemlich unnötig, da die Relation in sehr gutem Zustand (~Normalform) ist

Abhängigkeitsbewahrung

$\mathcal{R}$ ist zerlegt in $\mathcal{R}1, ..., \mathcal{R}n$

$F_{\mathcal{R}} = (F_{\mathcal{R}1} \cup ... \cup F_{\mathcal{R}n})$ bzw $F_{\mathcal{R}}+ = (F_{\mathcal{R}1} \cup ... \cup F_{\mathcal{R}n})+$

Beispiel für Abhängigkeitsverlust

- 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
- Postleitzahlengebiete gehen nicht über Ortsgrenzen und Orte nicht über Bundeslandgrenzen hinweg

Daraus resultieren die FDs

- {PLZ} → {Ort, BLand}
- {Straße, Ort, BLand} → {PLZ}

Betrachte die Zerlegung

- 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

□ PLZ,Straße

□ Ort,Bland,PLZ

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

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

Die FD {Straße, Ort, BLand} → {PLZ} ist im zerlegten Schema nicht mehr enthalten ➔ Einfügen inkonsistenter Tupel möglich

Einfügen zweier Tupel, die die FD Ort,Bland,Straße→PLZ verletzen

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

$\pi_{\text{PLZ,Straße}}$

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

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

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

Einfügen zweier Tupel, die die FD Ort,Bland,Straße→PLZ verletzen

PLZverzeichnis

Ort	BLand	Straße	PLZ
Frankfurt	Hessen	Goethestraße	60313
Frankfurt	Hessen	Galgenstraße	60437
Frankfurt	Brandenburg	Goethestraße	15234
Frankfurt	Brandenburg	Goethestraße	15235

|><|

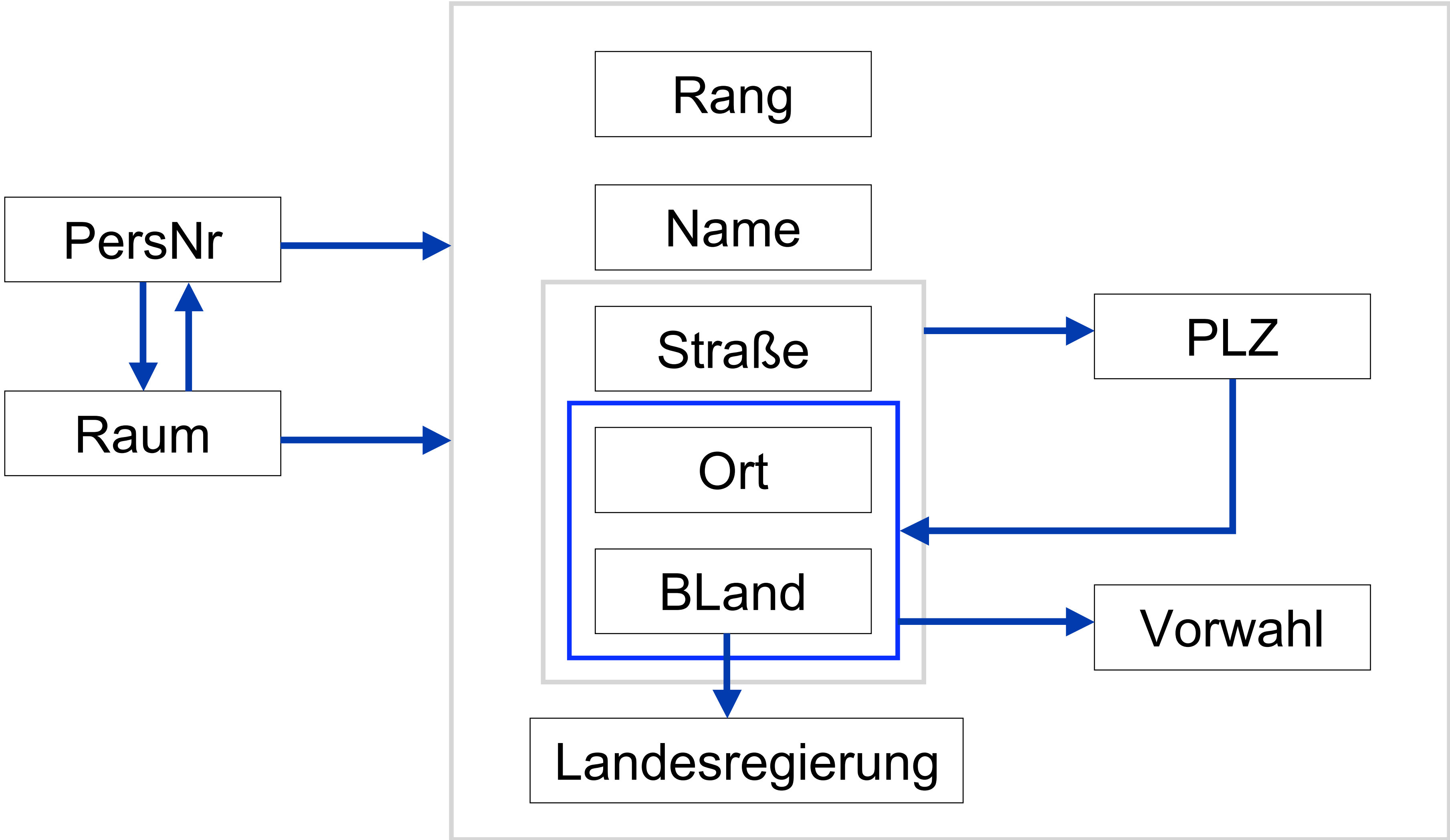
Straßen

PLZ	Straße
15234	Goethestraße
60313	Goethestraße
60437	Galgenstraße
15235	Goethestrasse

Orte

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

Graphische Darstellung der funktionalen Abhängigkeiten



Erste Normalform

Nur atomare Domänen

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

1 NF

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*	
		KName	*KAlter*
Johann	Martha	Else	5
		Lucie	3
Johann	Maria	Theo	3
		Josef	1
Heinz	Martha	Cleo	9

Vereinbarung

FDs, die von jeder Relationenausprägung automatisch immer erfüllt werden, nennen wir *trivial*. Nur FDs der Art □□ □ mit □□□ sind trivial.

Attribute eines Relationenschemas, die Elemente eines Kandidatenschlüssels des Relationenschemas sind, heißen "*prim*". Alle anderen Attribute des Relationenschemas nennen wir "*nicht prim*".

Zweite Normalform

Eine Relation $\mathcal{R}$ mit zugehörigen FDs $F_{\mathcal{R}}$ ist in zweiter Normalform,

falls jedes Nichtschlüssel-Attribut $A \in \mathcal{R}$ voll funktional abhängig ist von

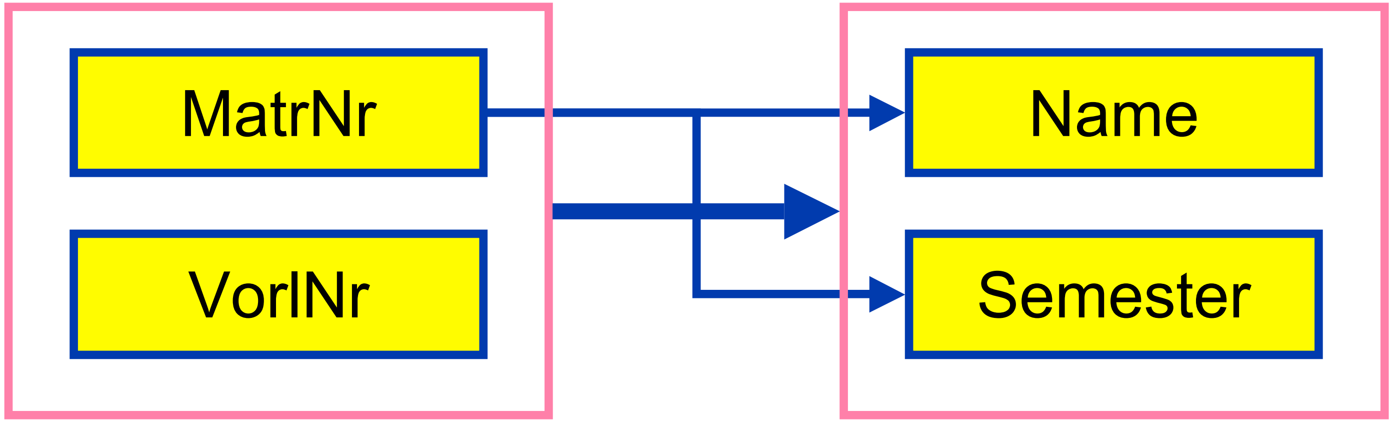
jedem Kandidatenschlüssel der Relation.

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

Studentenbelegung mit Schlüssel {MatrNr, VorlNr} ist nicht in zweiter NF

- {MatrNr} → {Name}
- {MatrNr} → {Semester}

Zweite Normalform



Einfügeanomalie: Was macht man mit Studenten, die keine Vorlesungenen hören?

Updateanomalien: Wenn z.B. Carnap ins vierte Semester kommt, muss man sicherstellen, dass alle vier Tupel geändert werden.

Löschanomalie: Was passiert wenn Fichte ihre einzige Vorlesung absagt?

Zerlegung in zwei Relationen

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

Beide Relationen sind in 2 NF – erfüllen sogar noch „höhere“ Gütekriterien ~ Normalformen.

Weitere Normalisierung: Motivation

Beispiel:

R = {[A, B, C ,D]}, F = {A → B, D → ABCD}, Schlüsselkandidat: {D}

R			
A	B	C	D
3	4	5	1
3	4	6	2

Do not represent the same fact twice

Allgemeiner Fall: □ □ □ □ F, dann: □ Superschlüssel oder FD ist trivial

ggf. Dekomposition notwendig (verlustfrei und abhändigkeitsbewahrend)

Dritte Normalform

Ein Relationenschema $\mathcal{R}$ ist in dritter Normalform, wenn für jede für $\mathcal{R}$ geltende funktionale Abhängigkeit der Form □ □ □ mit □ □ $\mathcal{R}$ und B □ $\mathcal{R}$ mindestens eine von drei Bedingungen gilt:

- B □ □, d.h., die FD ist trivial
- Das Attribut B ist in einem Kandidatenschlüssel von $\mathcal{R}$ enthalten
  (Man sagt: B ist prim)
- □ ist Superschlüssel von $\mathcal{R}$

Bestimmung funktionaler Abhängigkeiten

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

- {PersNr} → {PersNr, Name, Rang, Raum, Ort, Straße, PLZ, Vorwahl, Bland, EW, Landesregierung}
- {Ort,BLand} → {EW, Vorwahl}
- {PLZ} → {Bland, Ort, EW}
- {Bland, Ort, Straße} → {PLZ}
- {Bland} → {Landesregierung}
- {Raum} → {PersNr}

Zusätzliche Abhängigkeiten, die aus obigen abgeleitet werden können:

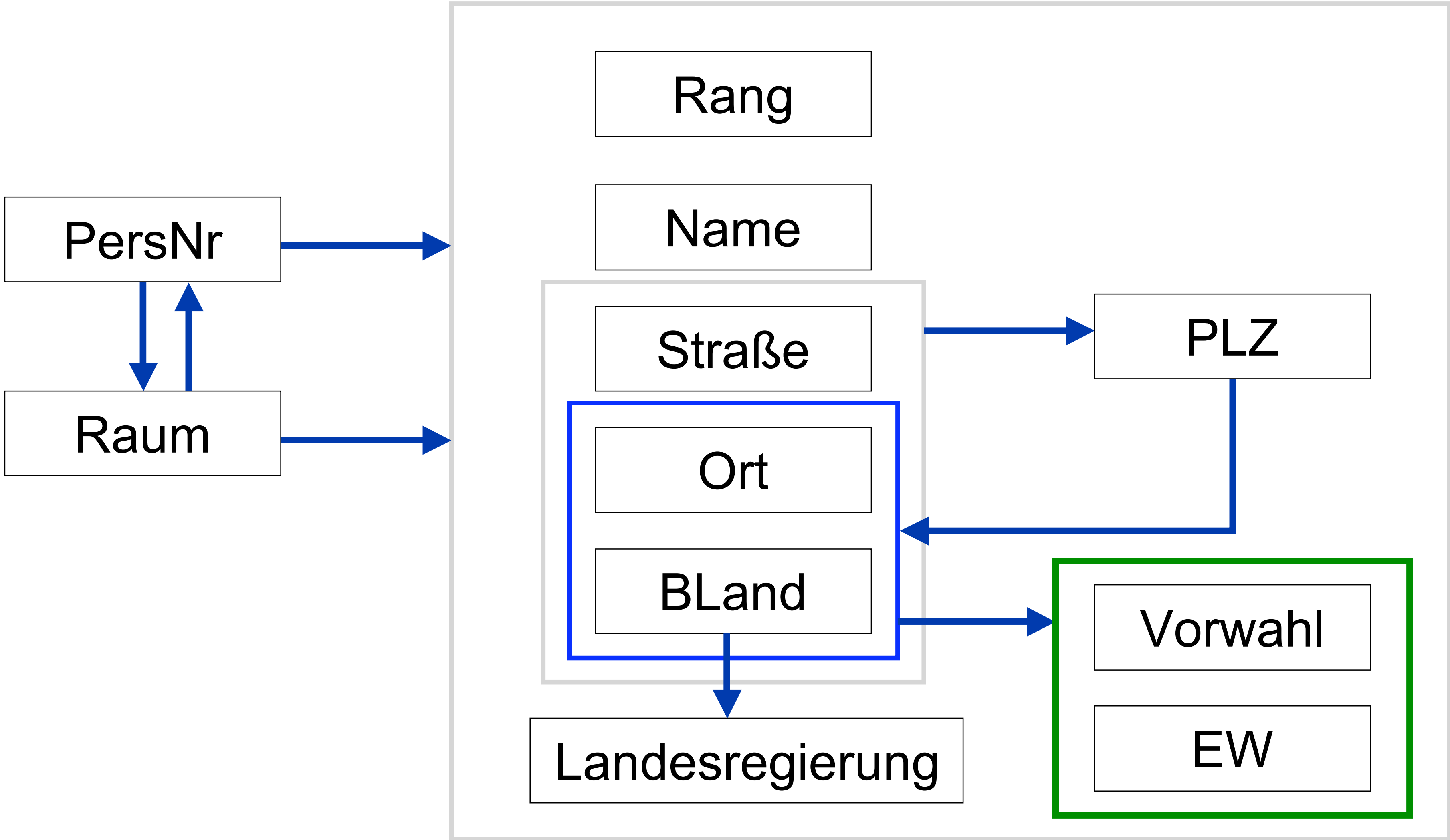
- {Raum} → {PersNr, Name, Rang, Raum, Ort, Straße, PLZ, Vorwahl, Bland, EW, Landesregierung}
- {PLZ} → {Landesregierung}

Bestimmung der Schlüssel

Schlüsselkandidaten: {Raum} und {PLZ}:

Problem: mit diesen Schlüsselkandidaten 3NF nicht gegeben

Graphische Darstellung



Zerlegung mit dem Synthesealgorithmus

Wir geben jetzt einen sogenannten Synthesealgorithmus an, mit dem zu einem gegebenen Relationenschema $\mathcal{R}$ mit funktionalen Anhängigkeiten F eine Zerlegung in $\mathcal{R}$1, ..., $\mathcal{R}$n ermittelt wird, die alle drei folgenden Kriterien erfüllt.

- ❏ $\mathcal{R}$1, ..., $\mathcal{R}$n ist eine verlustlose Zerlegung von $\mathcal{R}$.
- ❏ Die Zerlegung $\mathcal{R}$1, ..., $\mathcal{R}$n ist abhängigkeitserhaltend.
- ❏ Alle $\mathcal{R}$1, ..., $\mathcal{R}$n sind in dritter Normalform.

Synthesealgorithmus

Bestimme die kanonische Überdeckung F_c zu F. Wiederholung:

a. Linksreduktion

b. Rechtsreduktion

c. Entfernung von FDs der Form $\square \rightarrow \emptyset$

d. Zusammenfassung gleicher linker Seiten

Für jede funktionale Abhängigkeit $\square\ \square\ \ \square\ \square$ F_c:

- Kreiere ein Relationenschema $\mathcal{R}\square := \square\ \square\ \square$
- Ordne $\mathcal{R}\square$ die FDs $F\square := \{\square`\square\ \ \square`\ \square\ F_c \mid \square`\square\ \ \square`\square\ \mathcal{R}\square\}$ zu.

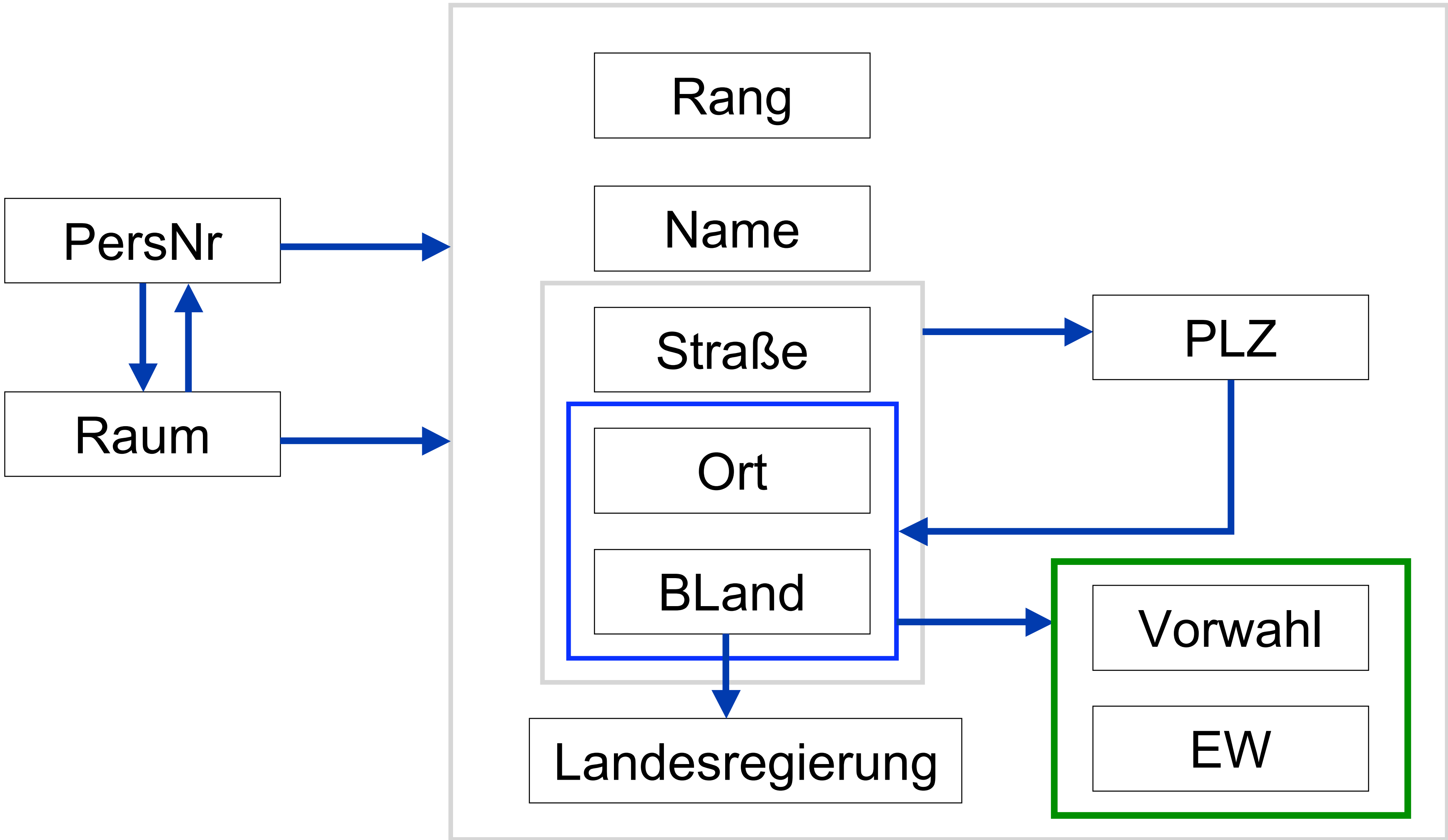
Falls eines der in Schritt 2. erzeugten Schemata einen Kandidatenschlüssel von $\mathcal{R}$ bzgl. F_c enthält, sind wir fertig. Sonst wähle einen Kandidatenschlüssel $\square\ \square\ \mathcal{R}$ aus und definiere folgendes Schema:

- $\mathcal{R}\square := \square$
- $F\square := \emptyset$

Eliminiere diejenigen Schemata $\mathcal{R}\square$, die in einem anderen Relationenschema $\mathcal{R}\square`$ enthalten sind, d.h.,

- $\mathcal{R}\square\ \square\ \mathcal{R}\square`$

Anwendung des Synthesealgorithmus



Anwendung des Synthesealgorithmus

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

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}

Professoren: {[PersNr, Name, Rang, Raum, Ort, Straße, BLand]}

PLZverzeichnis: {[Straße, BLand, Ort, PLZ]}

OrteVerzeichnis: {[Ort, BLand, EW, Vorwahl]}

Regierungen: {[Bland, Landesregierung]}

Beispiel

Wir nehmen an, aus der Analyse der Anwendung haben sich die folgenden funktionalen Abhängigkeiten ergeben:

Matrikelnr → Student-Name Student-PLZ Student-Strasse
Vorlesungsnr → Vorlesungsdozent
Die unten dargestellte Relation (Tabelle) befindet sich in der zweiten Normalform. Schlüsselattribute sind unterstrichen. Zur besseren Nachvollziehbarkeit sind Beispieldaten in Form von drei Tupeln angegeben.

<u>Matrikelnr.</u>	Student-Name	Student-PLZ	Vorlesungsnr.	Vorlesungsdozent	Student-Strasse
94-647-889	Schmid	3007	W3488	Jung	Schwarztorstr. 4
95-667-103	Moser	8052	W3988	Kühn	Rennweg 12
94-504-112	Huber	3007	W3988	Kühn	Zwyssigstr. 41

Transformieren Sie die Relation unter Berücksichtigung der oben genannten Dependenzen in die dritte Normalform und tragen Sie die Beispieldaten auch in den neuen Relationen ein.

Boyce-Codd-Normalform

Die Boyce-Codd-Normalform (BCNF) stellt nochmals eine Verschärfung dar.

Ein Relationenschema $\mathcal{R}$ mit FDs F ist in BCNF, wenn für jede für $\mathcal{R}$ geltende funktionale Abhängigkeit der Form $\alpha \to \beta \in$ F mindestens eine der folgenden zwei Bedingungen gilt:

- ❑ $\beta \subseteq \alpha$, d.h., die Abhängigkeit ist trivial oder
- ❑ α ist Superschlüssel von $\mathcal{R}$

Man kann jede Relation verlustlos in BCNF-Relationen zerlegen

Manchmal läßt sich dabei die Abhängigkeiterhaltung aber nicht erzielen

Städte ist in 3NF, aber nicht in BCNF

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

Geltende FDs:

- ❏ {Ort, BLand} → {EW}
- ❏ {BLand} → {Ministerpräsident/in}
- ❏ {Ministerpräsident/in} → {BLand}

Schlüsselkandidaten:

- ❏ {Ort, BLand}
- ❏ {Ort, Ministerpräsident/in}

Dekomposition

Man kann grundsätzlich jedes Relationenschema $\mathcal{R}$ mit funktionalen Anhängigkeiten 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 leider nicht immer erreicht werden, dass die Zerlegung $\mathcal{R}1$, ..., $\mathcal{R}n$ abhängigkeitserhaltend ist.

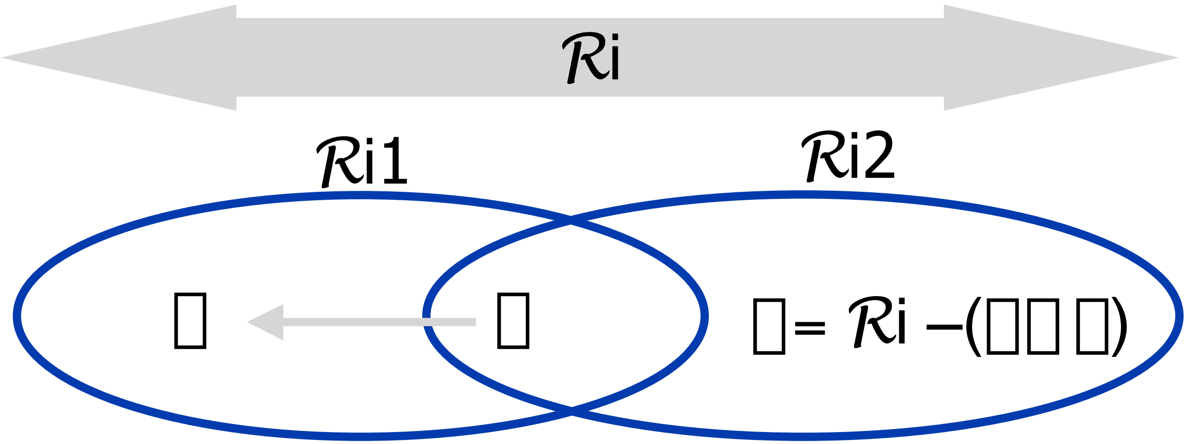
Dekompositions-Algorithmus

Starte mit $Z = \{\mathcal{R}\}$

Solange es noch ein Relationenschema $\mathcal{R}i$ in Z gibt, das nicht in BCNF ist, mache folgendes:

- ❑ Es gibt also eine für $\mathcal{R}i$ geltende nicht-triviale funktionale Abhängigkeit ($\alpha \to \beta$) mit
  - $\alpha \cap \beta = \emptyset$
  - $\neg(\alpha \to \mathcal{R}i)$
- ❑ Finde eine solche FD
  - Man sollte sie so wählen, dass β alle von α funktional abhängigen Attribute $B \in (\mathcal{R}i - \alpha)$ enthält, damit der Dekompositionsalgorithmus möglichst schnell terminiert.
- ❑ 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
  - $Z := (Z - \{\mathcal{R}i\}) \cup \{\mathcal{R}i1\} \cup \{\mathcal{R}i2\}$

Veranschaulichung der Dekomposition



Dekomposition der Relation Städte in BCNF-Relationen

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

Geltende FDs:

- ❑ {BLand} → {Ministerpräsident/in}
- ❑ {Ort, BLand} → {EW}
- ❑ {Ministerpräsident/in} → {BLand}

$\mathcal{R}$i1:

- ❑ Regierungen: {[BLand, Ministerpräsident/in]}

$\mathcal{R}$i2:

- ❑ Städte: {[Ort, BLand, EW]}

Zerlegung ist verlustlos und auch abhängigkeitserhaltend

Dekomposition des PLZverzeichnis in BCNF-Relationen

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

Funktionale Abhängigkeiten:

- ❑ {PLZ} → {Ort, BLand}
- ❑ {Straße, Ort, BLand} → {PLZ}

Betrachte die Zerlegung

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

Diese Zerlegung

- ❑ ist verlustlos   aber
- ❑ Nicht abhängigkeitserhaltend
- ❑ Siehe oben

Mehrwertige Abhängigkeiten: ein Beispiel

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

Mehrwertige Abhängigkeiten dieser Relation:

- ❑ {PersNr}→→{Sprache} und
- ❑ {PersNr}→→{ProgSprache}

MVDs führen zu Redundanz und Anomalien

Mehrwertige Abhängigkeiten

	R		
	$\underbrace{\text{A1 ... Ai}}_{}$ □	$\underbrace{\text{Ai+1 ... Aj}}_{}$ □	$\underbrace{\text{Aj+1 ... An}}_{}$ □
	A1 ... Ai	Ai+1 ... Aj	Aj+1 ... An
t1	a1 ... ai	ai+1 ... aj	aj+1 ... an
t2	a1 ... ai	bi+1 ... bj	bj+1 ... bn
t3	a1 ... ai	bi+1 ... bj	aj+1 ... an
t4	a1 ... ai	ai+1 ... aj	bj+1 ... bn

□ →→ □ gilt genau dann wenn

- es zu zwei Tupel t1 und t2 mit gleichen □–Werten
- auch zwei Tupel t3 und t4 gibt mit
  - t3.□ = t4.□ = t1.□ = t2.□
  - t3.□ = t1.□, t4.□ = t2.□
  - t3.□= t2.□, t4.□= t1.□

"Zu zwei Tupeln mit gleichem □ -Wert kann man die □ -Werte vertauschen, und die Tupel müssen auch in der Relation sein"

MVDs

Tuple-generating dependencies

- ❑ Man kann eine Relation MVD-konform machen, indem man zusätzliche Tupel einfügt
- ❑ Bei FDs geht das nicht!!

Mehrwertige Abhängigkeiten

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

A →→ B

A →→ C

Mehrwertige Abhängigkeiten: ein Beispiel

Fähigkeiten

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

□ PersNr, Sprache

□ PersNr, ProgSprache

Sprachen

PersNr	Sprache
3002	griechsich
3002	lateinisch
3005	deutsch

Sprachen

PersNr	ProgSprache
3002	C
3002	Pascal
3005	Ada

Mehrwertige Abhängigkeiten: ein Beispiel

Fähigkeiten

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

$\bowtie$

Sprachen

PersNr	Sprache
3002	griechsich
3002	lateinisch
3005	deutsch

Sprachen

PersNr	ProgSprache
3002	C
3002	Pascal
3005	Ada

Zusatzinformation

Die nachfolgenden Inhalte dieses Dokumentes wurden im WS04/05 nicht behandelt.

Verlustlose Zerlegung bei MVDs: hinreichende + notwendige Bedingung

$\mathcal{R} = \mathcal{R}1 \cup \mathcal{R}2$

- $R1 := \pi_{\mathcal{R}1}(R)$
- $R2 := \pi_{\mathcal{R}2}(R)$

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:

- ❑ $R = R1 \bowtie R2$

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:

- ❑ $(\mathcal{R}1 \cap \mathcal{R}2) \rightarrow\rightarrow \mathcal{R}1$ oder
- ❑ $(\mathcal{R}1 \cap \mathcal{R}2) \rightarrow\rightarrow \mathcal{R}2$

Triviale MVDs …

… sind solche, die von jeder Relationenausprägung erfüllt werden

Eine MVD □ →→ □ ist trivial genau dann wenn

- □ □ □ oder
- □ = R - □

Vierte Normalform

Eine Relation $\mathcal{R}$ ist in 4 NF wenn für jede MVD □ →→ □ eine der folgenden Bedingungen gilt:

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

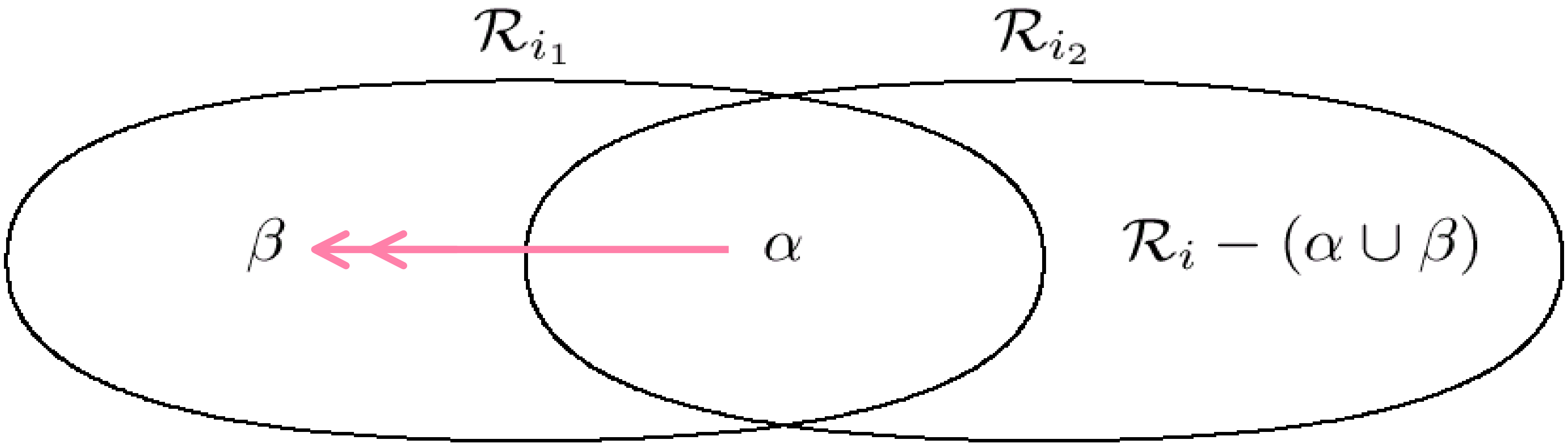
Dekomposition in 4 NF

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

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

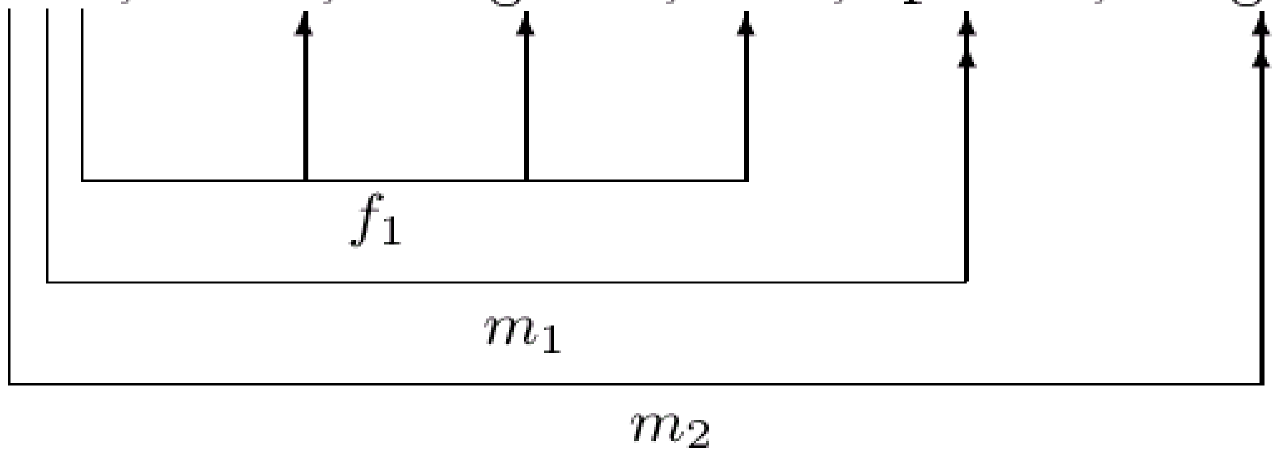
- Es gibt also eine für $\mathcal{R}$i geltende nicht-triviale MVD ($\alpha \twoheadrightarrow \beta$), für die gilt:
  - $\alpha \cap \beta = \emptyset$
  - $\neg(\alpha \rightarrow \mathcal{R}i)$
- Finde eine solche MVD
- Zerlege $\mathcal{R}$i in $\mathcal{R}$i1 := $\alpha \cup \beta$ und $\mathcal{R}$i2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

Dekomposition in 4 NF



Beispiel-Zerlegung

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



- Assistenten: {[PersNr, Name, Fachgebiet, Boss]}
- Fähigkeiten: {[PersNr, Sprache, ProgrSprache]}
- Sprachen: {[PersNr, Sprache]}
- ProgrSprachen: {[PersNr, ProgrSprache]}

Zusammenfassung

Die Verlustlosigkeit ist für alle Zerlegungsalgorithmen in alle Normalformen garantiert

Die Abhängigkeitserhaltung kann nur bis zur dritten Normalform garantiert werden