Kapitel 3: Das relationale DB-Modell & SQL

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

Schlüssel

$\alpha \subseteq \mathcal{R}$ ist ein Super-Schlüssel, falls folgendes gilt:

- ❑ $\alpha \rightarrow \mathcal{R}$

Wir nennen α Super-Schlüssel, weil noch nichts darüber ausgesagt wird,

daß der Schlüssel α minimal ist.

β ist voll funktional abhängig von α genau dann wenn gilt

- ❑ $\alpha \rightarrow \beta$ und
- ❑ α kann nicht mehr verkleinert werden, d.h.
  - $\forall A \in \alpha$ folgt, dass $(\alpha - \{A\}) \rightarrow \beta$ nicht gilt, oder kürzer
  - $\forall A \in \alpha$: $\neg((\alpha - \{A\}) \rightarrow \beta)$

Notation für volle funktionale Abhängigkeit: $\alpha \stackrel{\bullet}{\rightarrow} \beta$
$\alpha \subseteq \mathcal{R}$ ist ein Kandidaten-Schlüssel, falls folgendes gilt:

- ❑ $\alpha \stackrel{\bullet}{\rightarrow} \mathcal{R}$

Schlüsselbestimmung

Städte			
Name	BLand	Vorwahl	EW
Frankfurt	Hessen	069	650000
Frankfurt	Brandenburg	0335	84000
München	Bayern	089	1200000
Passau	Bayern	0851	50000
...	...	...	...

Kandidaten-schlüssel von *Städte*:

- {Name,BLand}
- {Name,Vorwahl}

Beachte, dass 2 kleinere Städte dieselbe Vorwahl haben können

Bestimmung funktionaler Abhängigkeiten

Professoren: {[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}

RDM: Relationale Algebra - Anfragen (1)

Vereinigung R ∪ S:

- ❑ Alle Tupel zweier Relationen werden in einer Ergebnisrelation zusammengefaßt.
- ❑ Das Ergebnis enthält keine Duplikate.

$$R \cup S := \{\, r \mid r \in R \lor r \in S \,\}$$

Relation R

ANr	*AName*	*Menge*
001	Anlasser	1.000
⋮	⋮	⋮
199	Kolben	5.000

Relation S

ANr	*AName*	*Menge*
237	Ölfilter	1.560
⋮	⋮	⋮
851	Schraube	25.000

Ergebnisrelation $R \cup S$

ANr	*AName*	*Menge*
001	Anlasser	1.000
⋮	⋮	⋮
199	Kolben	5.000
237	Ölfilter	1.560
⋮	⋮	⋮
851	Schraube	25.000

RDM: Relationale Algebra - Anfragen (2)

Differenz R \ S:

- ❏ Die Tupel zweier Relationen werden miteinander verglichen.
- ❏ Die in der ersten, nicht aber in der zweiten Relation befindlichen Tupel werden in die Ergebnisrelation aufgenommen.

$$R \setminus S := \{ r \mid r \in R \wedge r \notin S \}$$

Relation R

ANr	*AName*	*Menge*
001	Anlasser	1.000
237	Ölfilter	1.560
199	Kolben	5.000

Relation S

ANr	*AName*	*Menge*
851	Schraube	25.000
232	Gummiring	2.000
001	Anlasser	1.000

Ergebnisrelation $R \setminus S$

ANr	*AName*	*Menge*
237	Ölfilter	1.560
199	Kolben	5.000

RDM: Relationale Algebra - Anfragen (3)

Durchschnitt R ∩ S:

- Alle Tupel, die sowohl in der Relationen R als auch in der Relation S enthalten sind, werden in der Ergebnisrelation zusammengefaßt.

$$R \cap S := \{ r \mid r \in R \wedge r \in S \}$$

Relation R

ANr	*AName*	*Menge*
001	Anlasser	1.000
007	Zündkerze	1.380
199	Kolben	5.000

Relation S

ANr	*AName*	*Menge*
001	Anlasser	1.000
199	Kolben	5.000
237	Ölfilter	1.560

Ergebnisrelation $R \cap S$

ANr	*AName*	*Menge*
001	Anlasser	1.000
199	Kolben	5.000

RDM: Relationale Algebra - Anfragen (4)

Kartesisches Produkt R × S:

- ❑ Alle Tupel zweier Relationen R und S werden kombinatorisch miteinander verbunden. Wenn die Relation R n Spalten und die Relation S m Spalten umfaßt, dann besitzt R × S (*n*+*m*) Spalten.
- ❑ Wenn die Relation R k Zeilen und die Relation S l Zeilen umfaßt, dann besitzt R × S (*k***l*) Zeilen.
- ❑ Um eindeutige Attributbezeichnungen in der Ergebnisrelation zu gewährleisten, müssen Attribute, die in den Relationen R und S gleich bezeichnet sind, vor der Bildung des kartesischen Produkts umbenannt werden.

$$R \times S := \{ (r_1, ..., r_n, s_1, ..., s_m) \mid (r_1, ..., r_n) \in R, (\mathrm{s}_1, ..., \mathrm{s}_\mathrm{m}) \in S \}$$

- ❑ Beispiel:
  - Projekte × Projektdurchführung (s. nächste Folie)

RDM: Relationale Algebra - Anfragen (5)

Beispiel: Projekte × Projektdurchführung

Projekte

Nr	*Titel*	*Budget*
100	DB Fahrpläne	300.000
200	ADAC Kundenstamm	100.000
300	Telekom Statistik	200.000

Projektdurchführung
(Ausschnitt)

Nr	*Kurz*
100	MFSW
200	PERS
300	MFSW

Ergebnisrelation
Projekte × Projektdurchführung

Nr	*Titel*	*Budget*	*Nr2*	*Kurz*
100	DB Fahrpläne	300.000	100	MFSW
100	DB Fahrpläne	300.000	200	PERS
100	DB Fahrpläne	300.000	300	MFSW
200	ADAC Kundenstamm	100.000	100	MFSW
200	ADAC Kundenstamm	100.000	200	PERS
200	ADAC Kundenstamm	100.000	300	MFSW
300	Telekom Statistik	200.000	100	MFSW
300	Telekom Statistik	200.000	200	PERS
300	Telekom Statistik	200.000	300	MFSW

RDM: Relationale Algebra - Anfragen (6)

Join (Verbindung) R $><_\theta$ S:

- ❏ Eine Verbindung zwischen zwei Relationen wird in einer Kombination von kartesischem Produkt und nachfolgender Selektion (σ) gemäß des Prädikats θ hergestellt.
- ❏ Im allgemeinen Fall (*Theta-Join*) vergleicht ein (beliebiges) Prädikat θ mehrere Attribute aus den Relationen R und S (Spezialfall: Equi-Join).

$$R >\!<_{\theta} S := \sigma_\theta(R \times S)$$

- ❏ Beispiele:
  - *Projekte* $><_{(\text{Nr} \neq \text{Nr})}$ *Projektdurchführung* (s. nächste Folie)
  - *Projekte* $><_{(\text{Budget} > 150000) \wedge (\text{Nr} = \text{Nr})}$ *Projektdurchführung*
- ❏ Die Ergebnisrelation enthält die Zeilen des kartesischen Produkts der Relationen R und S, die θ erfüllen.

RDM: Relationale Algebra - Anfragen (7)

Beispiel: Projekte >< $_{(Nr \neq Nr)}$ Projektdurchführung

Projekte

Nr	*Titel*	*Budget*
100	DB Fahrpläne	300.000
200	ADAC Kundenstamm	100.000
300	Telekom Statistik	200.000

Projektdurchführung (Ausschnitt)

Nr	*Kurz*
100	MFSW
200	PERS
300	MFSW

Ergebnisrelation

Nr	*Titel*	*Budget*	*Nr2*	*Kurz*
100	DB Fahrpläne	300.000	200	PERS
100	DB Fahrpläne	300.000	300	MFSW
200	ADAC Kundenstamm	100.000	100	MFSW
200	ADAC Kundenstamm	100.000	300	MFSW
300	Telekom Statistik	200.000	100	MFSW
300	Telekom Statistik	200.000	200	PERS

RDM: Relationale Algebra - Anfragen (8)

Join (Verbindung): Fortsetzung

- ❑ Von besonderer Bedeutung im RDM ist der *Natural Join*, da er eine Verknüpfung von Tabellen über ihre Fremdschlüsselwerte erlaubt.
  - Beispiel:
    - *Projekte* >< *Projektdurchführung* := *Projekt* >< $_{\text{Nr = Nr}}$ *Projektdurchführung*
    - In diesem Fall betrachtet θ nur die Gleichheit zwischen Fremdschlüssel und Primärschlüssel, die den gleichen Attributnamen (*Nr*) besitzen.
- ❑ Weitere abgeleitete *Joinoperationen* (*Semi-Join, Outer-Join*, ...) und die Division zweier Relationen sind beschrieben in:
  - S.M. Lang, P.C. Lockemann. Datenbankeinsatz. Springer, Berlin u.a., 1995.

RDM: Relationale Algebra - Anfragen (9)

Natural Join: Projekte >< $_{(Nr = Nr)}$ Projektdurchführung

Projekte

Nr	*Titel*	*Budget*
100	DB Fahrpläne	300.000
200	ADAC Kundenstamm	100.000
300	Telekom Statistik	200.000

Projektdurchführung (Ausschnitt)

Nr	*Kurz*
100	MFSW
200	PERS
300	MFSW

Ergebnisrelation

Nr	*Titel*	*Budget*	*Nr2*	*Kurz*
100	DB Fahrpläne	300.000	100	MFSW
200	ADAC Kundenstamm	100.000	200	PERS
300	Telekom Statistik	200.000	300	MFSW

RDM: Relationale Algebra - Anfragen (10)

Projektion $\pi_{(r_{f_1}, \ldots, r_{f_n})}$ (R):

- n Spalten einer m-stelligen Relation R werden über ihren Namen ausgewählt.
- Dadurch entsteht eine n-stellige Relation *(n ≤ m).*
- Die Reihenfolge der Spalten in der Ergebnisrelation kann definiert werden.
- Duplikatelimination in der Ergebnisrelation.

$$\pi_{(r_{f_1}, \ldots, r_{f_n})}(R) := \{ (r_{f_1}, \ldots, r_{f_n}) \mid (r_1, \ldots, r_m) \in R \}$$

- Beispiel: $\pi_{(\text{Nr, Budget})}$ (Projekte)

Projekte

Nr	*Titel*	*Budget*
100	DB Fahrpläne	300.000
200	ADAC Kundenstamm	100.000
300	Telekom Statistik	200.000

→

Ergebnisrelation $\pi_{(\mathit{Nr},\ \mathit{Budget})}$(Projekte)

Nr	*Budget*
100	300.000
200	100.000
300	200.000

RDM: Relationale Algebra - Anfragen (11)

Selektion $\sigma_\theta(R)$:

- ❑ Bestimmte Tupel einer Relation werden ausgewählt und in der Ergebnisrelation vereinigt.
- ❑ Zur Auswahl der zu übernehmenden Tupel dient das Prädikat $\theta : R \rightarrow \{$ `true, false` $\}$, in dem die Attributbezeichner als Eingabevariablen dienen.
- ❑ Anwendung dieses Prädikats auf jedes Tupel der Ausgangsrelation, indem die Werte des Tupels unter den jeweiligen Attributen für die Variablen eingesetzt werden.
- ❑ In die Ergebnisrelation werden alle Tupel übernommen, für die das Prädikat den Wahrheitswert `true` liefert.

$$\sigma_\theta(R) := \{\, r \in R \mid \theta\,(\, r\,)\,\}$$

Relation *R*

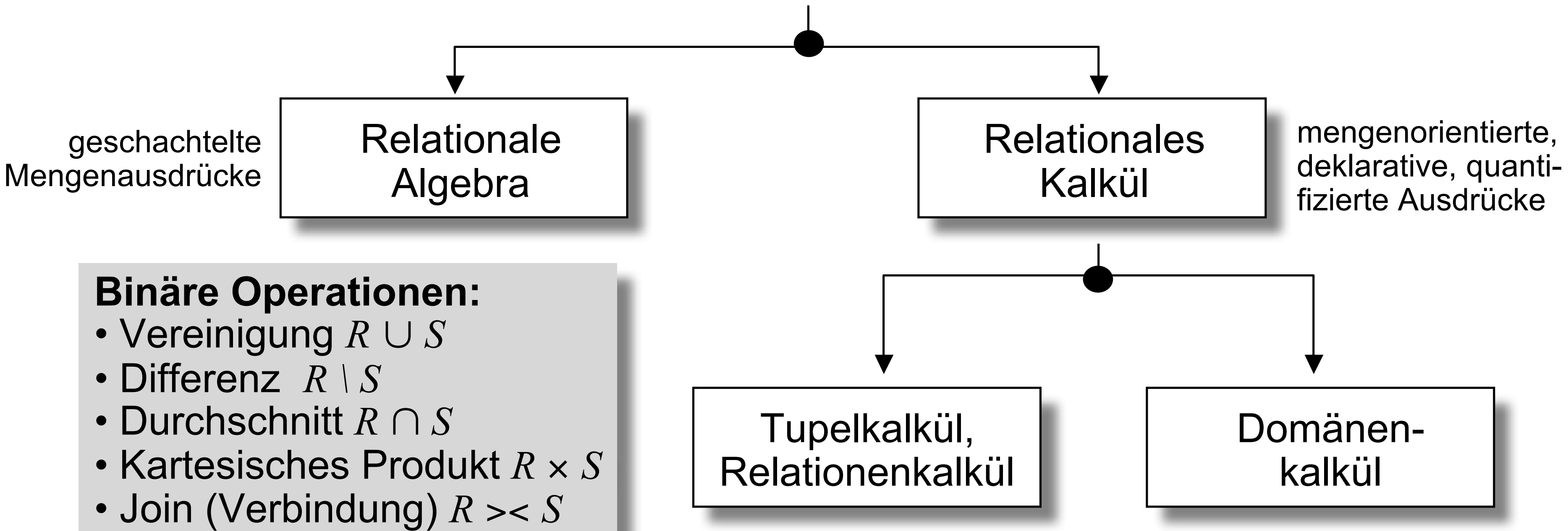
ANr	*AName*
001	Anlasser
007	Zündkerze
199	Kolben

Ergebnisrelation
$\sigma_{ANr\,<\,199}(R)$

ANr	*AName*
001	Anlasser
007	Zündkerze

RDM: Anfragen

Relationale Anfragesprachen im Überblick:



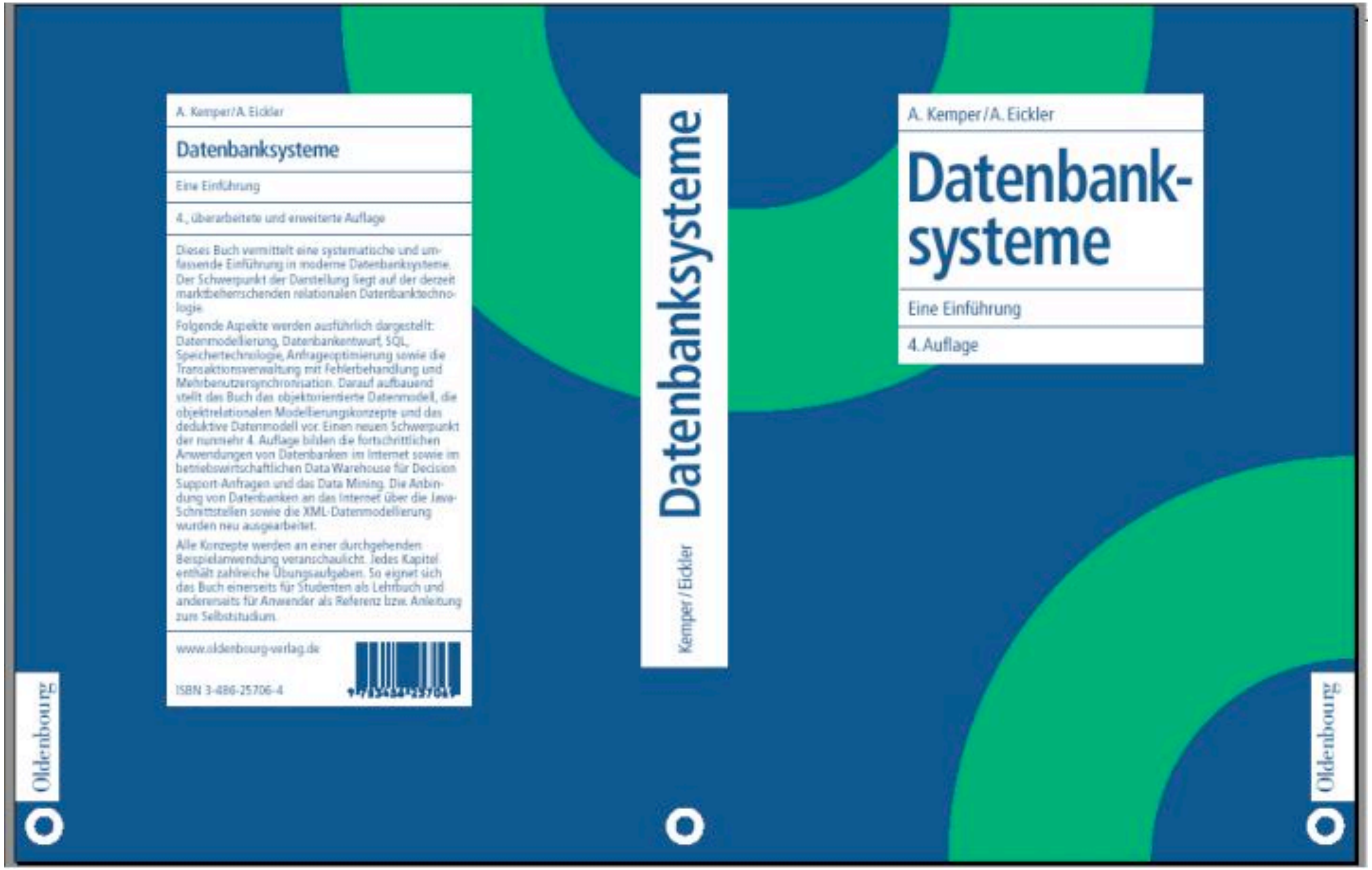
Binäre Operationen:
- Vereinigung $R \cup S$
- Differenz $R \setminus S$
- Durchschnitt $R \cap S$
- Kartesisches Produkt $R \times S$
- Join (Verbindung) $R >< S$

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

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}

Hülle Funktionaler Abhängigkeiten

Sei F eine Menge von Funktionalen Abhängigkeiten (FDs)

F^+ bezeichnet die Menge aller aus F ableitbaren FDs und wird *Hülle* genannt.

Im allgemeinen gibt es unterschiedliche Mengen von FDs, deren Hülle gleich sind. In diesem Fall schreiben wir: $F_1 \equiv F_2$ (F_1 und F_2 sind äquivalent)

Wie kann man F^+ methodisch bestimmen?

Herleitung funktionaler Abhängigkeiten: Armstrong-Axiome

Reflexivität

- Falls β eine Teilmenge von α ist ($\beta \subseteq \alpha$) dann gilt immer $\alpha \rightarrow \beta$.
  Insbesondere gilt immer $\alpha \rightarrow \alpha$.

Verstärkung

- Falls $\alpha \rightarrow \beta$ gilt, dann gilt auch $\alpha\gamma \rightarrow \beta\gamma$. Hierbei stehe z.B. $\alpha\gamma$ für $\alpha \cup \gamma$.

Transitivität

- Falls $\alpha \rightarrow \beta$ und $\beta \rightarrow \gamma$ gilt, dann gilt auch $\alpha \rightarrow \gamma$.

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

- Vereinigungsregel:
  - Wenn $\alpha \rightarrow \beta$ und $\alpha \rightarrow \gamma$ gelten, dann gilt auch $\alpha \rightarrow \beta\gamma$
- Dekompositionsregel:
  - Wenn $\alpha \rightarrow \beta\gamma$ gilt, dann gelten auch $\alpha \rightarrow \beta$ und $\alpha \rightarrow \gamma$
- Pseudotransitivitätsregel:
  - Wenn $\alpha \rightarrow \beta$ und $\gamma\beta \rightarrow \delta$, dann gilt auch $\alpha\gamma \rightarrow \delta$

Bestimmung der Hülle einer Attributmenge

Oftmals ist man nicht an der gesamten Hülle einer Menge F von FDs interessiert,

Sondern nur an der Menge von Attributen, die von α gemäß F funktional bestimmt werden.

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

Ausgabe: die vollständige Menge von Attributen $\alpha+$, für die gilt $\alpha \rightarrow \alpha+$.

AttrHülle(F,α)

- Erg := α
- **While** (Änderungen an Erg) **do**

  Foreach FD $\beta \rightarrow \gamma$ **in** F **do**

  If $\beta \subseteq$ Erg **then** Erg := Erg $\cup$ γ
- Ausgabe $\alpha+$ = Erg

Nutzen der Attributhülle

Mit Hilfe der Attributhülle kann man bestimmen, ob eine Menge von Attributen κ einen Superschlüssel für ein Relationenschema $\mathcal{R}$ darstellt

Bestimme κ^+ und prüfe ob $\kappa^+ = \mathcal{R}$

Kanonische Überdeckung

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

- $F_c \equiv F$, d.h. $F_c^+ = F^+$
- In F_c existieren keine FDs , die überflüssige Attribute enthalten. D.h. es muß folgendes gelten:
  - $\forall A \in \alpha$: $(F_c - (\alpha \to \beta) \cup ((\alpha - \{A\}) \to \beta)) \not\equiv F_c$
  - $\forall B \in \beta$: $(F_c - (\alpha \to \beta) \cup (\alpha \to (\beta - \{B\}))) \not\equiv F_c$
- Jede linke Seite einer funktionalen Abhängigkeit in F_c ist einzigartig. Dies kann durch sukzessive Anwendung der Vereinigungsregel auf FDs der Art $\alpha \to \beta$ und $\alpha \to \gamma$ erzielt werden, so dass die beiden FDs durch $\alpha \to \beta\gamma$ ersetzt werden.

Berechnung der kanonischen Überdeckung

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, α - A)

  gilt. Falls dies der Fall ist, ersetze $\alpha \rightarrow \beta$ durch $(\alpha - A) \rightarrow \beta$.

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

- ❑ Überprüfe für alle $B \in \beta$, ob
  - $B \in$ AttrHülle(F – $(\alpha \rightarrow \beta) \cup (\alpha \rightarrow (\beta - B))$, α)

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

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

Fasse mittels der Vereinigungsregel FDs der Form $\alpha \rightarrow \beta 1, ..., \alpha \rightarrow \beta n$ zusammen, so dass $\alpha \rightarrow (\beta 1 \cup ... \cup \beta n)$ 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:

- 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.
- 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

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

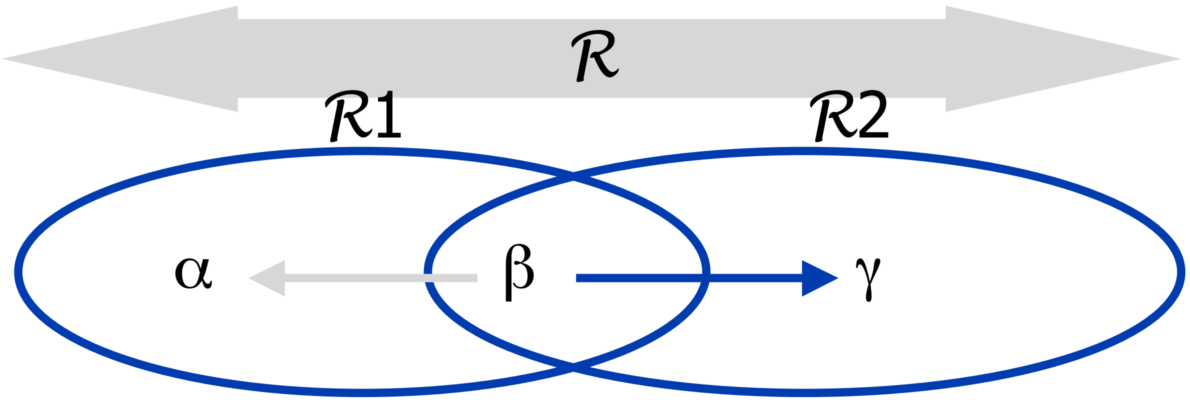
- ❑ R1 := $\Pi_{\mathcal{R}1}$ (R)
- ❑ R2 := $\Pi_{\mathcal{R}2}$ (R)

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

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

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

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

$\Pi_{....}$

Besucht	
Kneipe	*Gast*
Kowalski	Kemper
Kowalski	Eickler
Innsteg	Kemper

Trinkt	
Gast	*Bier*
Kemper	Pils
Eickler	Hefeweizen
Kemper	Hefeweizen

$\neq$

|><|

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

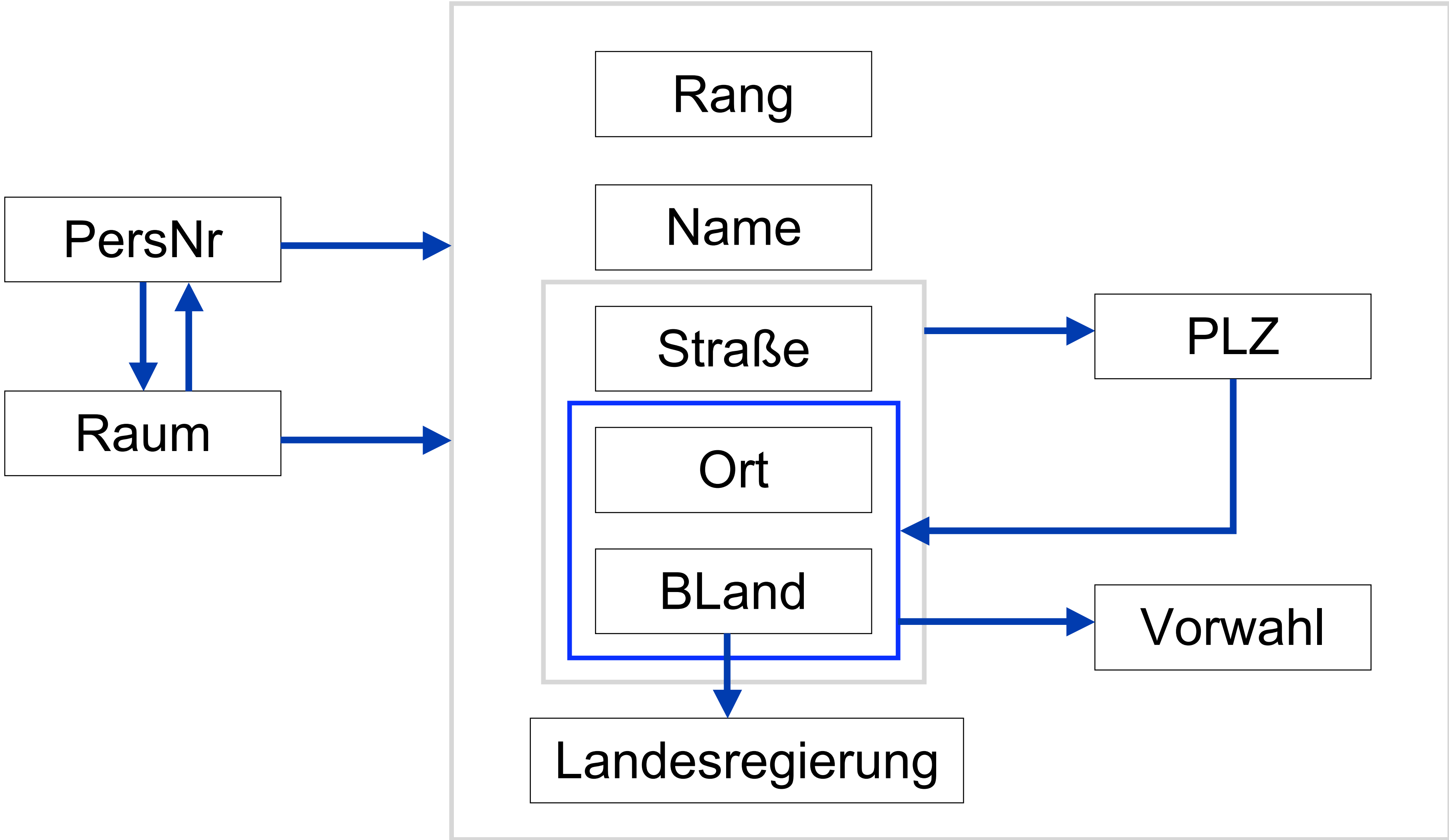
Es gilt nicht nur eine der hinreichenden FDs, sondern gleich beide

- {Kind}$\rightarrow${Mutter}
- {Kind}$\rightarrow${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

Graphische Darstellung der funktionalen Abhängigkeiten



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

$\Pi_{\text{PLZ,Straße}}$

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

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

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 $\alpha \rightarrow \beta$ mit $\beta \subseteq \alpha$ 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

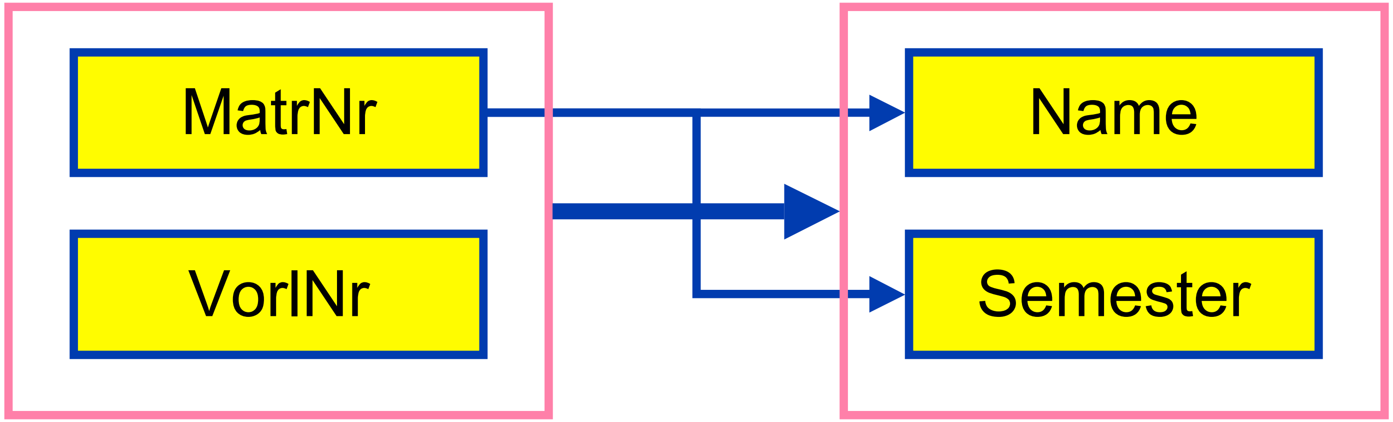
Eine Relation $\mathcal{R}$ mit zugehörigen FDs $F_{\mathcal{R}}$ ist in zweiter Normalform, falls jedes Nichtschlüssel-Attribut $A \in \mathcal{R}$ voll funktional abhängig ist von jedem Kandidatenschlüssel der Relation.

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: $\alpha \rightarrow \beta \in F$, dann: α Superschlüssel oder FD ist trivial

ggf. Dekomposition notwendig (verlustfrei und abhängigkeitsbewahrend)

Dritte Normalform

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

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

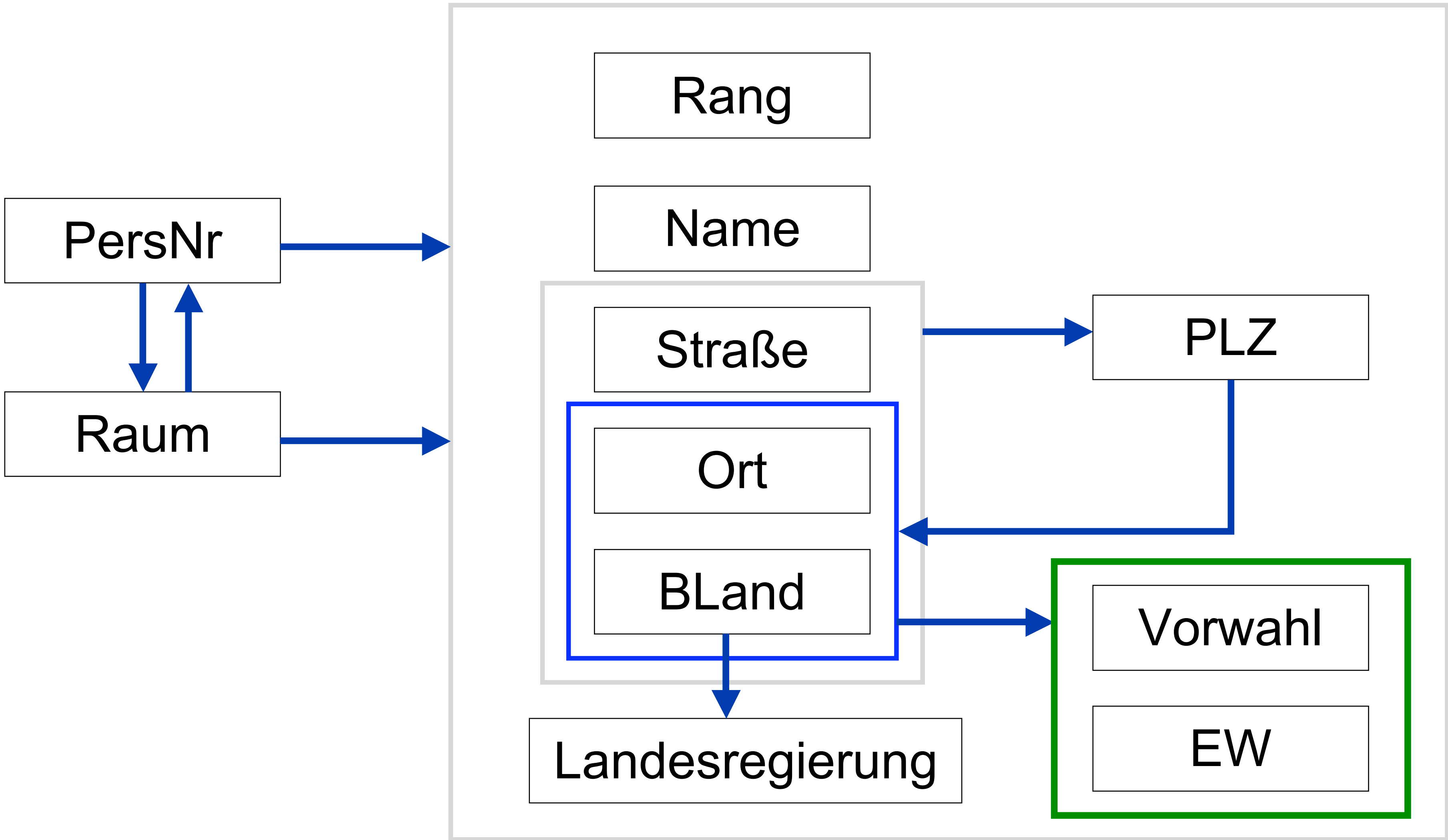
- {Raum} → {PersNr, Name, Rang, Raum, Ort, Straße, PLZ, Vorwahl, Bland, EW, Landesregierung}
- {PLZ} → {Landesregierung}

Bestimmung der Schlüssel

Schlüsselkandidaten: {Raum} und {PersNr}:

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:

- Linksreduktion
- Rechtsreduktion
- Entfernung von FDs der Form $\alpha \rightarrow \emptyset$
- Zusammenfassung gleicher linker Seiten

Für jede funktionale Abhängigkeit $\alpha \rightarrow \beta \in F_c$:

- Kreiere ein Relationenschema $\mathcal{R}\alpha := \alpha \cup \beta$
- Ordne $\mathcal{R}\alpha$ die FDs $F\alpha := \{\alpha` \rightarrow \beta` \in F_c \mid \alpha` \cup \beta` \subseteq \mathcal{R}\alpha\}$ zu.

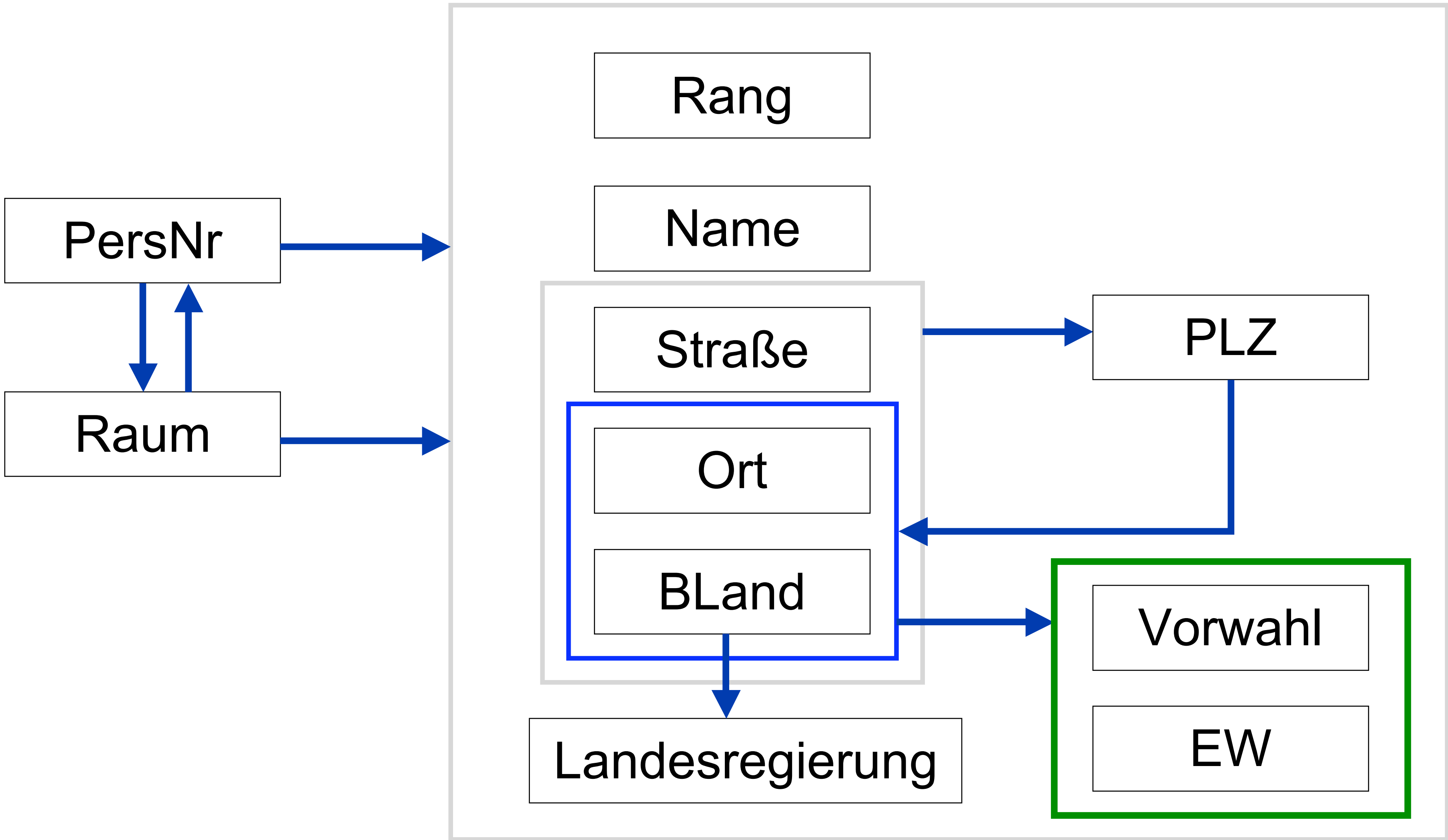
Falls eines der in Schritt 2. erzeugten Schemata einen Kandidatenschlüssel von $\mathcal{R}$ bzgl. F_c enthält, sind wir fertig. Sonst wähle einen Kandidatenschlüssel $\kappa \subseteq \mathcal{R}$ aus und definiere folgendes Schema:

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

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 des Synthesealgorithmus



Beispiel 1: Anwendung des Synthesealgorithmus

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

- {PersNr} → {Name, Rang, Raum, Ort, Straße, BLand}
- {Raum} → {PersNr}
- {Straße, BLand, Ort} → {PLZ}
- {Ort,BLand} → {EW, Vorwahl}
- {BLand} → {Landesregierung}
- {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 2:

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 $\alpha \rightarrow \beta \in F$ 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} $\rightarrow$ {EW}
- {BLand} $\rightarrow$ {Ministerpräsident/in}
- {Ministerpräsident/in} $\rightarrow$ {BLand}

Schlüsselkandidaten:

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

3NF:

- {Ort, BLand} ist Superschlüssel
- BLand und Ministerpräsident/in sind prim

Nicht BCNF:

- {BLand} kein Superschlüssel
- {Ministerpräsident/in} kein Superschlüssel

Information, wer welches Bundesland regiert, wird doppelt abgespeichert

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.

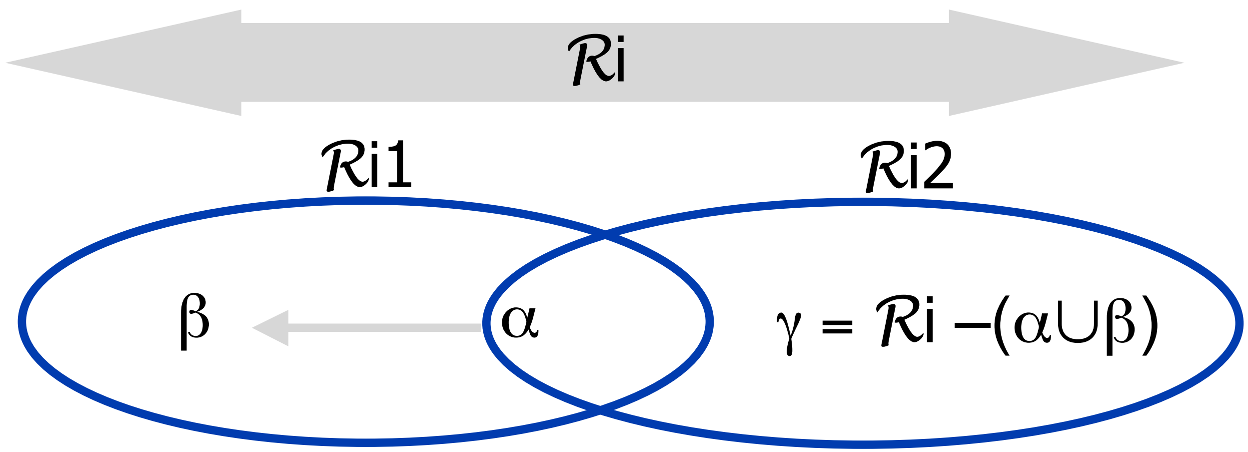
Dekompositions-Algorithmus

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 $(\alpha \rightarrow \beta)$ mit
  - $\alpha \cap \beta = \emptyset$
  - $\neg(\alpha \rightarrow \mathcal{R}i)$
- ❑ 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		
	$\overbrace{\text{A1 ... Ai}}^{\alpha}$	$\overbrace{\text{Ai+1 ... Aj}}^{\beta}$	$\overbrace{\text{Aj+1 ... An}}^{\gamma}$
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

$\alpha \rightarrow\rightarrow \beta$ gilt genau dann wenn

- es zu zwei Tupel t1 und t2 mit gleichen α–Werten
- auch zwei Tupel t3 und t4 gibt mit
  - $t3.\alpha = t4.\alpha = t1.\alpha = t2.\alpha$
  - $t3.\beta = t2.\beta$, $t4.\beta = t1.\beta$
  - $t4.\gamma = t2.\gamma$, $t3.\gamma = t1.\gamma$

"Zu zwei Tupeln mit gleichem α -Wert kann man die β -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

$A \rightarrow\rightarrow B$

$A \rightarrow\rightarrow C$

Mehrwertige Abhängigkeiten: ein Beispiel

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

$\Pi_{\text{PersNr, Sprache}}$

$\Pi_{\text{PersNr, ProgSprache}}$

Sprachen	
PersNr	Sprache
3002	griechsich
3002	lateinisch
3005	deutsch

ProgSprachen	
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

ProgSprachen	
PersNr	ProgSprache
3002	C
3002	Pascal
3005	Ada

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 >< 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 $\alpha \to\to \beta$ ist trivial genau dann wenn

- $\beta \subseteq \alpha$ oder
- $\beta = R - \alpha$

Vierte Normalform

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

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

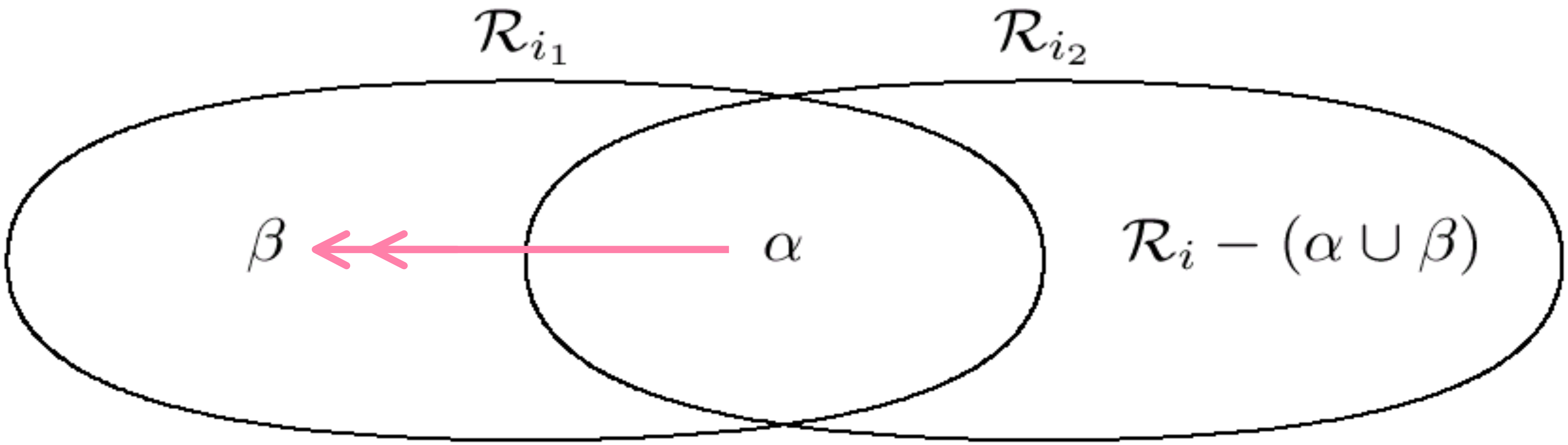
Dekomposition in 4 NF

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:

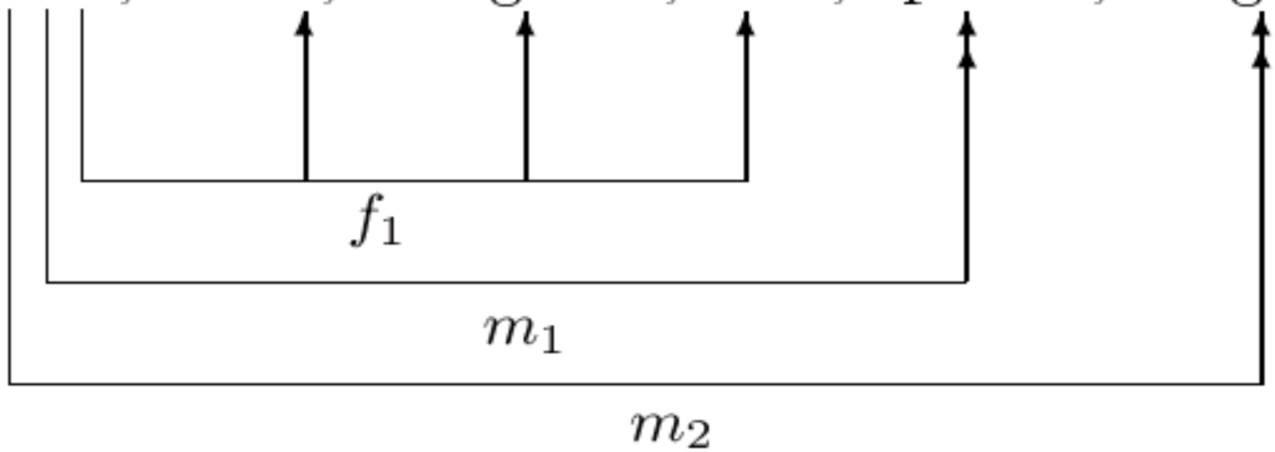
- ❑ Es gibt also eine für $\mathcal{R}$i geltende nicht-triviale MVD ($\alpha \rightarrow\rightarrow \beta$), für die gilt:
  - $\alpha \cap \beta = \emptyset$
  - $\neg(\alpha \rightarrow \mathcal{R}i)$
- ❑ Finde eine solche MVD
- ❑ Zerlege $\mathcal{R}$i in $\mathcal{R}$i1 := $\alpha \cup \beta$ und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