# Einführung in Datenbanksysteme

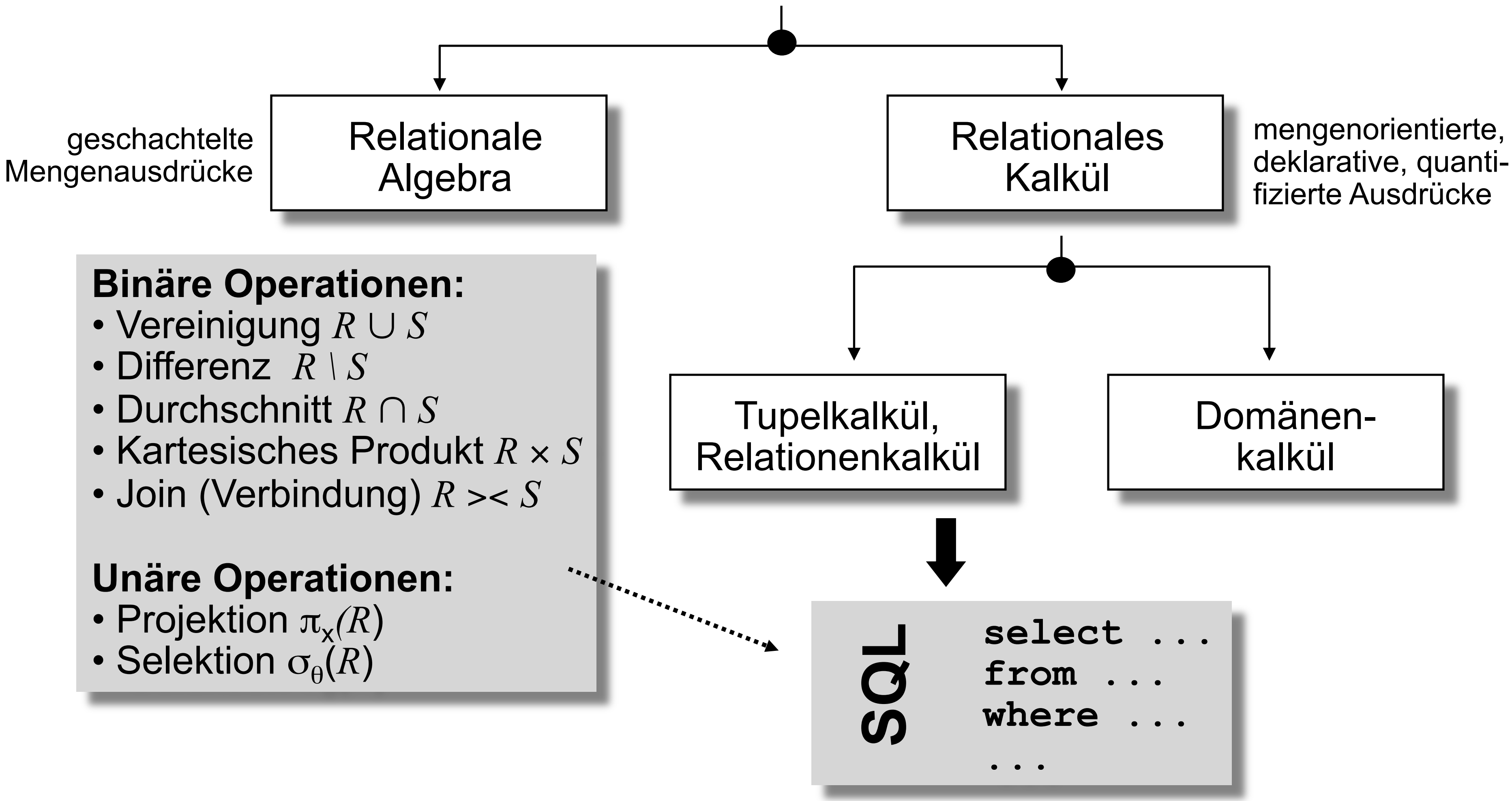
**Prof. Dr. Ralf Möller**

**TUHH**

## Relationale Entwurfstheorie

# RDM: Anfragen

**Relationale Anfragesprachen im Überblick:**



# Wiederholung: „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?

# Wiederholung: Funktionale Abhängigkeiten

Schema

- $\mathcal{R} = \{A, B, C, D\}$

Ausprägung $R$

Seien $\alpha \subseteq \mathcal{R}$, $\beta \subseteq \mathcal{R}$

$\alpha \rightarrow \beta$ genau dann wenn $\forall r, s \in R$ mit $r.\alpha = s.\alpha \Rightarrow r.\beta = s.\beta$

| *R* | | | |
|---|---|---|---|
| *A* | *B* | *C* | *D* |
| a4 | b2 | c4 | d3 |
| a1 | b1 | c1 | d1 |
| a1 | b1 | c1 | d2 |
| a2 | b2 | c3 | d2 |
| a3 | b2 | c4 | d3 |

$\{A\} \rightarrow \{B\}$

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

*Nicht:* $\{B\} \rightarrow \{C\}$

*Notationskonvention:*

$CD \rightarrow B$

# Wiederholung: Schlüssel

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

- $\alpha \rightarrow \mathcal{R}$

Wir nennen $\alpha$ **Super-Schlüssel**, weil noch nichts darüber ausgesagt wird,

daß der Schlüssel $\alpha$ minimal ist.

$\beta$ ist voll funktional abhängig von $\alpha$ genau dann wenn gilt

- $\alpha \rightarrow \beta$ und
- $\alpha$ 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: $\alpha \overset{\bullet}{\rightarrow} \beta$

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

- $\alpha \overset{\bullet}{\rightarrow} \mathcal{R}$

# Schlüsselbestimmung

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

Kandidatenschlüssel von *Städte*:

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

Kandidatenschlüssel lassen sich nicht aus Beispielen bestimmen!

Funktionale Abhängigkeiten zählen.

Schlüssel sollen FDs umsetzen!

Beachte, dass 2 kleinere Städte dieselbe Vorwahl haben können

# Widerholung: 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}

# 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)

**Def:** $F^+$ ist Hülle von F, d. h. Menge von funktionalen Abhängigkeiten, die durch F impliziert werden (Folge von F sind)

**Erläuterung:** $R(A_1, \ldots, A_n)$ Schema,

$$F = \{X_1 \rightarrow Y_1, X_2 \rightarrow Y_2, \ldots, X_k \rightarrow Y_k\}$$

Seien $R_i$ beliebige Instantiierungen von R, d. h. konkrete Relationen, die F erfüllen.
Die Menge aller solcher Instantiierungen sei $\Im(R, F)$

Eine funktionale Abhängigkeit $X \rightarrow Y$ heißt ***von F impliziert*** (bzw. Folge von F), wenn

$$\forall R_i \in \Im(R, F) \text{ gilt auch } X \rightarrow Y$$

## Vergleiche auch Begriffe aus Logik:

**Interpretation**: beliebige Instantiierungen $R_j$ für R,
ohne daß F gelten muß.

**Modell**: $R_i \in \Im\,(R, F)$
logische Folge: von F implizierte funktionale Abhängigkeit

$\Rightarrow \Im\,(R, F)$ ist Menge der Modelle für F mit Schema R,
$F^+$ ist Menge der logischen Folgerungen von F

# Wie kann man $F^+$ methodisch bestimmen?

- Kalkül: Armstrong-Axiome

# Herleitung funktionaler Abhängigkeiten: Armstrong-Axiome

Reflexivität

- Falls $\beta$ eine Teilmenge von $\alpha$ 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 $\alpha \rightarrow \gamma$.

William W. Armstrong
Dependency Structures of Data Base Relationships
1974

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 $\alpha\gamma \rightarrow \delta$

# Armstrong Axiome

- Der Kalkül, der sich durch sättigende Anwendung der Armstrong-Axiome ergibt, ist korrekt und vollständig
- Beweisargumente: siehe Übung

# Schlüsselbestimmung

Manuelle Bestimmung der Kandidatenschlüssel

- ist aufwendig und
- bei vielen FDs fehleranfällig.

Automatisierbar?

# Bestimmung der Hülle einer **Attribut**menge

Bei der Schlüsselbestimmung ist man nicht an der gesamten Hülle einer Menge F von FDs interessiert, sondern nur an der Menge von Attributen, die von $\alpha$ gemäß F funktional bestimmt werden (sog. Attributhülle).

Eingabe: eine Menge F von FDs und eine Menge von Attributen $\alpha$.

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

AttrHülle(F,$\alpha$)

- Erg := $\alpha$
- **While** (Änderungen an Erg) **do**

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

        **If** $\beta \subseteq$ Erg **then** Erg := Erg $\cup\ \gamma$

- Ausgabe $\alpha+$ = Erg

# Nutzen der Attributhülle

- Mit Hilfe der Attributhülle kann man bestimmen, ob eine Menge von Attributen $\kappa$ einen Superschlüssel für ein Relationenschema $\mathcal{R}$ darstellt:
  - Bestimme $\kappa^+$ und prüfe ob $\kappa^+ = \mathcal{R}$

- Mit Hilfe der Attributhülle kann man die Kandidatenschlüssel für ein Relationenschema bestimmen:
  - Bestimme alle bzgl. Mengeninklusion minimalen Mengen $\kappa$, so dass $\kappa^+ = \mathcal{R}$, und damit $\kappa \rightarrow \mathcal{R}$

# Herleitung von Relationenschemata aus FDs

- {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}

Welche Relationenschemata sollen verwendet werden, so dass FDs durch Schlüsselbedingung geprüft werden können?

- Professoren: {[PersNr, Name, Rang, Raum, Ort, Straße, PLZ, Vorwahl, Bland, EW, Landesregierung]} ???
- Für jede FD ein Schema???

# Vermeidung von Redundanz

- In den Daten:
  - Können wir doppelte repräsentierte Daten vermeiden?
- In der Modellierung:
  - Sind einige der aufgeschriebenen FDs überflüssig?

# Redundanzfreie Darstellung von FDs

## Kanonische Überdeckung

- $F_c$ heißt kanonische Überdeckung von F, wenn die folgenden 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:
    - $\forall A \in \alpha: (F_c - (\alpha \rightarrow \beta) \cup ((\alpha - \{A\}) \rightarrow \beta)) \not\equiv F_c$
    - $\forall B \in \beta: (F_c - (\alpha \rightarrow \beta) \cup (\alpha \rightarrow (\beta - \{B\}))) \not\equiv F_c$
  - 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 $\alpha \rightarrow \beta \in$ F die Linksreduktion durch, also:

- ❑ Überprüfe für alle A $\in \alpha$, ob A überflüssig ist, d.h., ob
  - $\beta \subseteq$ AttrHülle(F, $\alpha$ - A)

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

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

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

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

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

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

# Nutzung der kanonischen Überdeckung

## Naiver Ansatz:

- Bilde relationales Schema für jede FD der kanonischen Überdeckung
  - Eventuell immer noch zu viele Relationen
  - Beispiel:
    - FDs = $\{A \rightarrow B, D \rightarrow ABCD\}$
    - Zwei Relationen?
- Mehrere FDs einem relationalem Schema zuordnen?

# Vereinbarungen

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*".

$\mathcal{R}$ steht für ein Relationenschema, $F_{\mathcal{R}}$ ist die zugeordnete Menge von FDs. Wenn $\mathcal{R}$ klar ist, dann wird meist nur F geschrieben.

# Vermeidung von Redundanz in den Daten

Beispiel:

$\mathcal{R}$ = {[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: $\alpha$ Superschlüssel oder FD ist trivial

ggf. Dekomposition notwendig (verlustfrei und abhängigkeitsbewahrend)

# 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

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

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

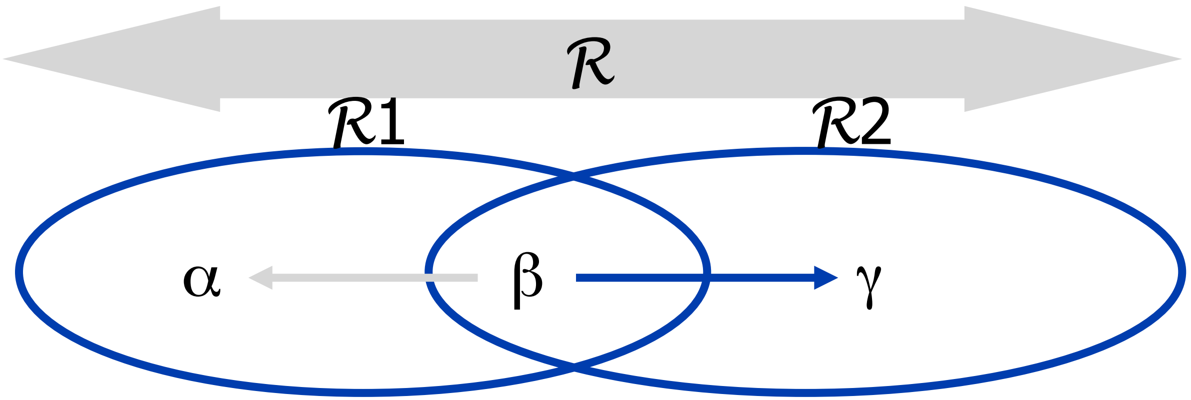
Eine Zerlegung von $\mathcal{R}$ in $\mathcal{R}1$ und $\mathcal{R}2$ ist verlustlos,

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}→{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 ist → Normalform (s.u.)

# Abhängigkeitsbewahrung

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

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

Beispiel für Abhängigkeitsverlust

- Geg. Schema 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} $\rightarrow$ {PLZ} ist im zerlegten Schema nicht mehr enthalten $\Rightarrow$ 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 |

$\Pi_{\text{PLZ,Straße}}$

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

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

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

## 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 |

# Gütekriterien für Relationenschemata

- Redundanzfreiheit in den Daten
- Prüfung der einem Relationenschema zugeordneten FDs möglichst nur durch Schlüsselbedingung

→ Normalformen

# Erste Normalform: nur „einfache“ Domänen

Beispiel:

| *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

<table>
<tr><th colspan="4">Eltern</th></tr>
<tr><th rowspan="2">Vater</th><th rowspan="2">Mutter</th><th colspan="2">Kinder</th></tr>
<tr><th>KName</th><th>KAlter</th></tr>
<tr><td rowspan="2">Johann</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>

# Zweite Normalform

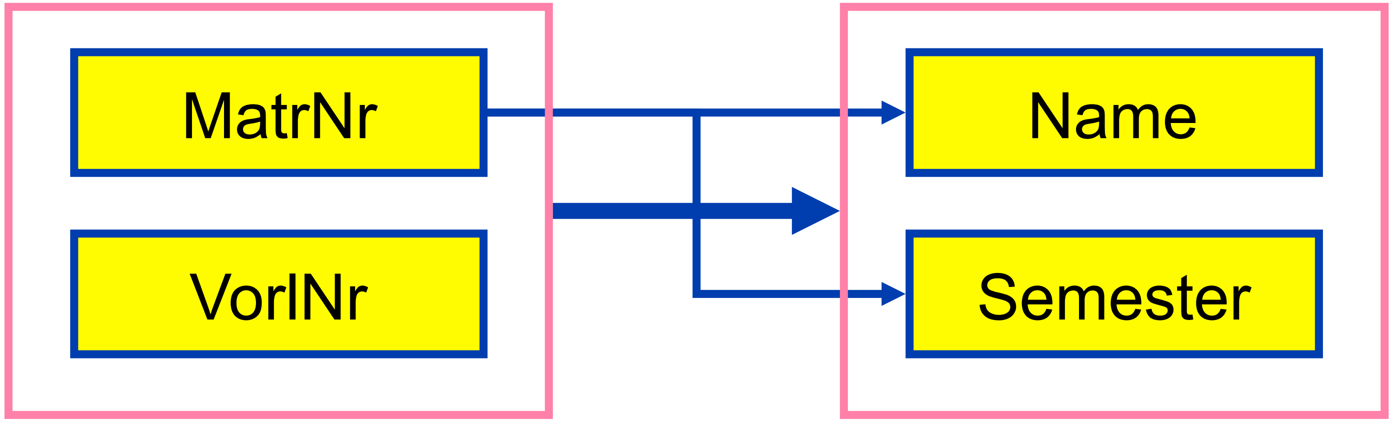
Eine Relation $\mathcal{R}$ mit zugehörigen FDs $F_{\mathcal{R}}$ ist in zweiter Normalform, falls jedes Nichtschlüssel-Attribut $A \in \mathcal{R}$ voll funktional abhängig ist von jedem Kandidatenschlüssel der Relation.

| 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: $\alpha$ Superschlüssel oder FD ist trivial

ggf. Dekomposition notwendig (verlustfrei und abhängigkeitsbewahrend)

# Boyce-Codd-Normalform

Die Boyce-Codd-Normalform (BCNF) stellt nochmals eine Verschärfung der zweiten Normalform 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
- ❑ $\alpha$ ist Superschlüssel von $\mathcal{R}$

Man kann jede Relation verlustlos in BCNF-Relationen zerlegen

Manchmal läßt sich dabei die Abhängigkeiterhaltung aber nicht erzielen

# Beispiel

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

Prüfung der FDs durch Schlüsselbedingung nicht möglich

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 (die zweite FD kann keiner Subrelation zugeordnet werden

# Was, wenn Boyce-Codd-Normalform nicht möglich?

Beispiel:

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

Schlüsselkandidaten: {AC}, {DC}

Codd 71: Schema „ganz gut“, wenn es keine „transitiven Abhängigkeiten“ gibt: 3. Normalform

FD A → D im obigen Schema wäre tolerierbar

# Dritte Normalform (formuliert nach Zaniolo 82)

Ein Relationenschema $\mathcal{R}$ ist in dritter Normalform, wenn für **jede für $\mathcal{R}$ geltende** funktionale Abhängigkeit der Form $\alpha \rightarrow \mathrm{B}$ 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
- ❑ $\alpha$ ist Superschlüssel von $\mathcal{R}$
- ❑ Das Attribut B ist in einem Kandidatenschlüssel von $\mathcal{R}$ enthalten (B ist prim)

Man beachte: Es wird **jede für $\mathcal{R}$** geltende FD betrachtet wird → FD-Hülle!

Warum ist Zaniolos Formulierung interessant?

# Frage

- Können wir relationale Schemata finden, so dass alle FDs einem Schema zugeordnet werden können (Abhängigkeitserhaltung) und wenigstens die dritte Normalform gegeben ist?

# Synthesealgorithmus

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

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

- $\mathcal{R}\alpha \subseteq \mathcal{R}\alpha`$

# 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]}

# Boyce-Codd-Normalform (2)

Wiederholung:

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
- ❑ $\alpha$ 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}

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

# Frage

- Sollten wir nicht wenigstens versuchen, Boyce-Codd-Normalform herzustellen?

# 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.

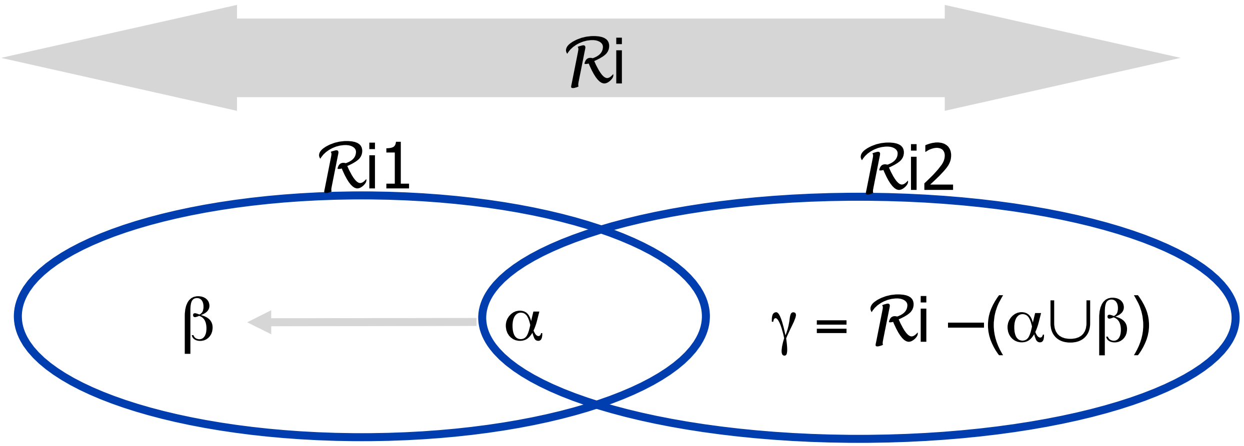
# Dekompositions-Algorithmus

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 $\beta$ alle von $\alpha$ 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 – Glück gehabt!

# 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

# Weitere verwendbare Einschränkungen

Gibt es weitere sinnvolle Einschränkungen (Dependencies)?

- Enthaltensein-Einschränkungen (Inclusion Dependencies)
  - Relationale Kodierung notwendig (z.B. gleiche Primärschlüssel)
- Mehrwertige Abhängigkeiten (Multiple-Value Dependencies)

# 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 | | |
|---|---|---|---|
| | $\underbrace{\text{A1 ... Ai}}_{}$ α | $\underbrace{\text{Ai+1 ... Aj}}_{}$ β | $\underbrace{\text{Aj+1 ... An}}_{}$ γ |
| 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 |

$|\!\!><\!\!|$

| 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**
- $\alpha$ ist Superschlüssel von $\mathcal{R}$

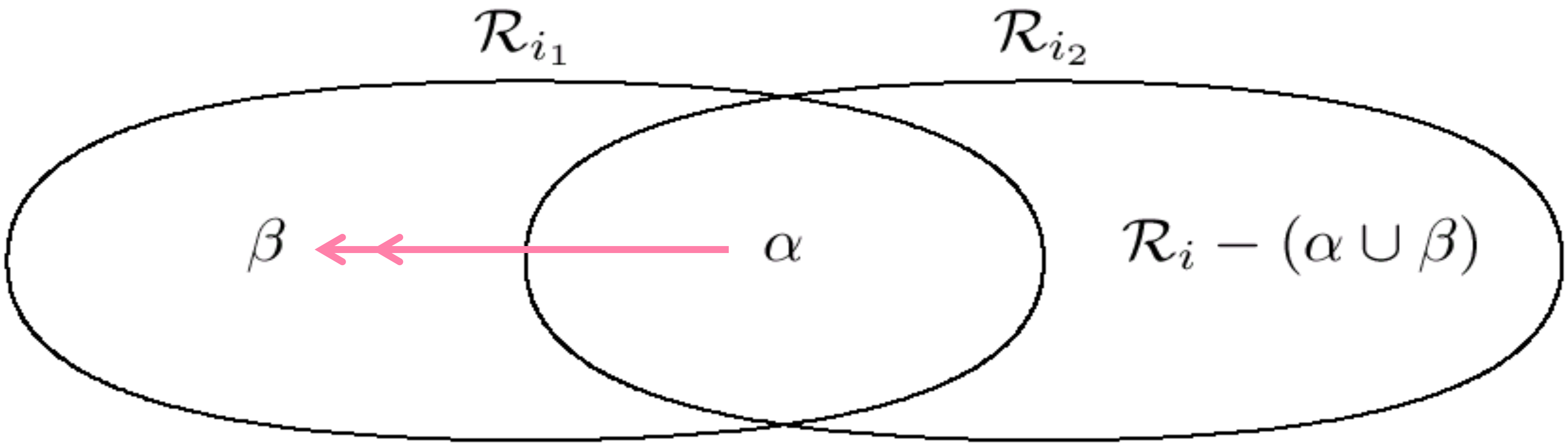
# Dekomposition in 4 NF

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:

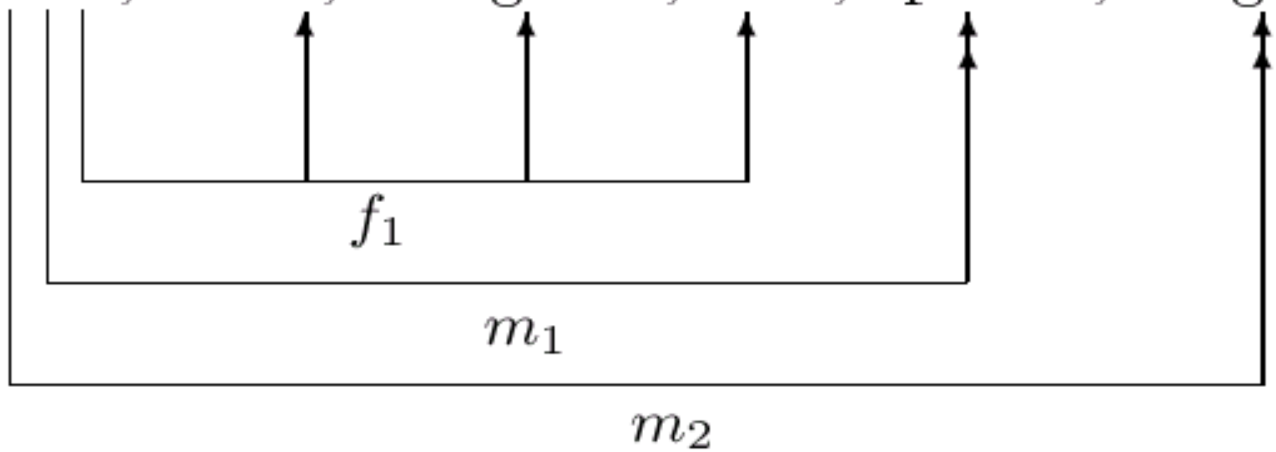
- ❑ Es gibt also eine für $\mathcal{R}$i geltende nicht-triviale MVD ($\alpha \twoheadrightarrow \beta$), für die gilt:
  - $\alpha \cap \beta = \emptyset$
  - $\neg(\alpha \rightarrow \mathcal{R}\mathrm{i})$
- ❑ Finde eine solche MVD
- ❑ Zerlege $\mathcal{R}$i in $\mathcal{R}$i1 := $\alpha \cup \beta$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

# Dekomposition in 4 NF



# Beispiel-Zerlegung

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

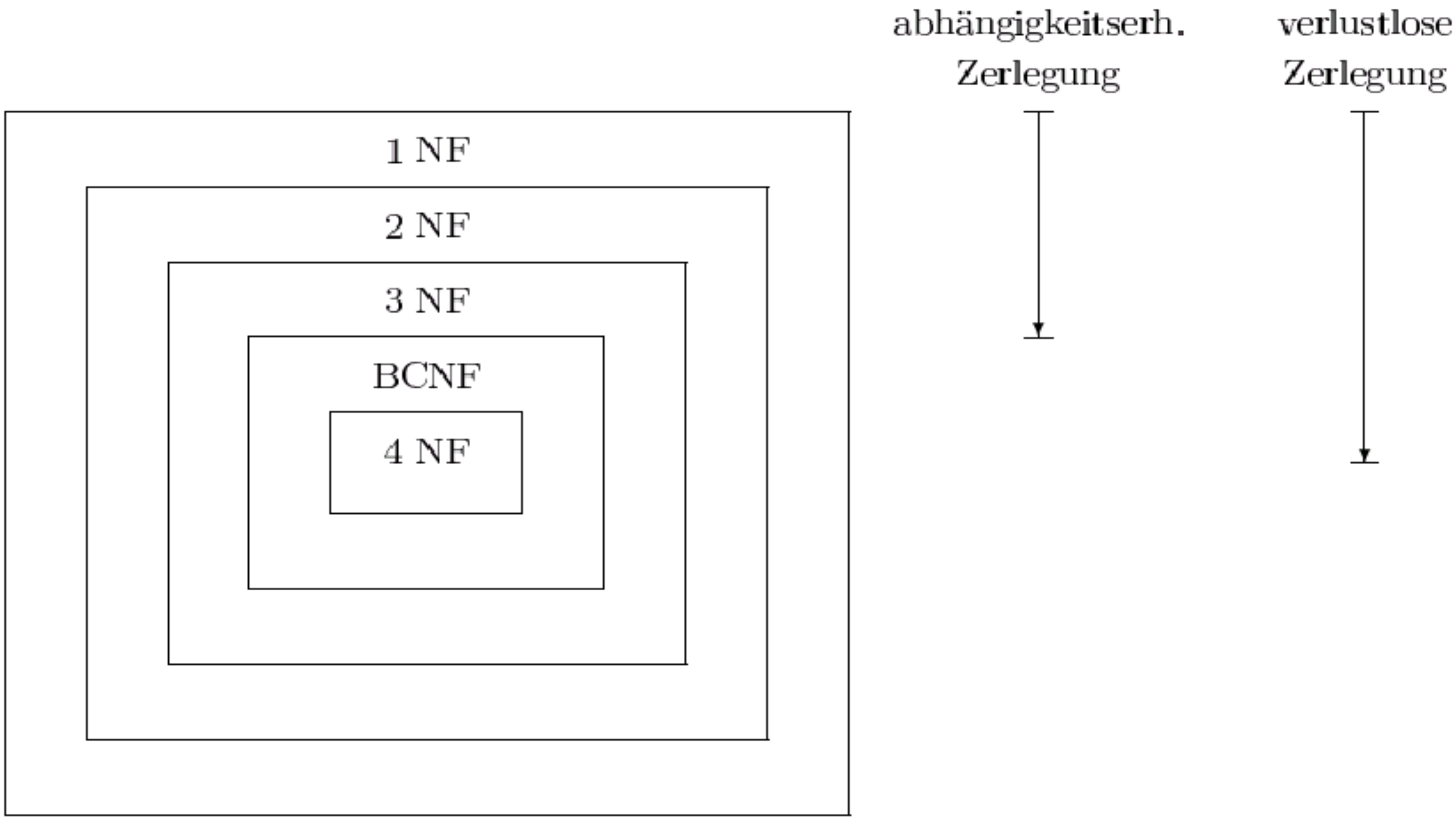



- Assistenten: {[PersNr, Name, Fachgebiet, Boss]}
- Fähigkeiten: {[PersNr, Sprache, ProgrSprache]}
- Sprachen: {[PersNr, Sprache]}
- ProgrSprachen: {[PersNr, ProgrSprache]}

# Zusammenfassung

Die Verlustlosigkeit ist für alle Zerlegungsalgorithmen in alle Normalformen garantiert

Die Abhängigkeitserhaltung kann nur bis zur dritten Normalform garantiert werden



# Acknowledgments