6. Relationale Entwurfstheorie

1. Funktionale Abhängigkeiten
2. Armstrong-Kalkül
3. Zerlegung von Relationen
4. Normalformen und Normalisierungen

Informatik 5 - Informationssysteme und Datenbanken | RWTH AACHEN UNIVERSITY

Einführung

Bis jetzt:

- Nutzen- und Anforderungsanalys (Pflichtenheft)
- Entity-Relationship-Entwurf
- relationales Schema

Zu tun:

- Feintuning des erstellten Schemas auf der Basis von intrarelationalen Abhängigkeiten
  - Funktionale Abhängigkeiten
  - Kriterien für gute Schemata, schlechte Schemata
  - Normalformen
  - Algorithmen zur Normalisierung

Was ist faul mit diesem Schema?

ProfVorl						
PersNr	Name	Rang	Raum	VorlNr	Titel	SWS
2125	Sokrates	W3	226	5041	Ethik	4
2125	Sokrates	W3	226	5049	Mäeutik	2
2125	Sokrates	W3	226	4052	Logik	4
...	...	...	...	...	...	...
2132	Popper	W2	52	5259	Der Wiener Kreis	2
2137	Kant	W3	7	4630	Die 3 Kritiken	4

Änderungs-Anomalie

ProfVorl						
PersNr	Name	Rang	Raum	VorlNr	Titel	SWS
2125	Sokrates	W3	226	5041	Ethik	4
2125	Sokrates	W3	226	5049	Mäeutik	2
2125	Sokrates	W3	226	4052	Logik	4
...	...	...	...	...	...	...
2132	Popper	W2	52	5259	Der Wiener Kreis	2
2137	Kant	W3	7	4630	Die 3 Kritiken	4

- Angenommen Sokrates soll von Raum 226 nach Raum 233 umziehen
- Die Information ‘Raum’ existiert in diesem Fall mehrfach
- Lösung: Änderung aller Einträge gleichzeitig
  - hoher Speicherbedarf durch Redundanz
  - erhöhter Zeitbedarf bei Änderungen

Einfüge-Anomalie

ProfVorl						
PersNr	Name	Rang	Raum	VorlNr	Titel	SWS
2125	Sokrates	W3	226	5041	Ethik	4
2125	Sokrates	W3	226	5049	Mäeutik	2
2125	Sokrates	W3	226	4052	Logik	4
…	…	…	…	…	…	…
2132	Popper	W2	52	5259	Der Wiener Kreis	2
2137	Kant	W3	7	4630	Die 3 Kritiken	4

- Schema kombiniert Informationen verschiedener unpassender Entitytypen
  - Hinzufügen eines Professors ohne Vorlesung

    ⇒ NULL-Werte in VorlNr, Titel und SWS
  - Analog: Hinzufügen einer Vorlesung zu der noch kein Dozent festgelegt wurde

    ⇒ NULL-Werte in PersNr, Name, Rang und Raum

Lösch-Anomalie

ProfVorl						
PersNr	Name	Rang	Raum	VorlNr	Titel	SWS
2125	Sokrates	W3	226	5041	Ethik	4
2125	Sokrates	W3	226	5049	Mäeutik	2
2125	Sokrates	W3	226	4052	Logik	4
...	...	...	...	...	...	...
2132	Popper	W2	52	5259	Der Wiener Kreis	2
2137	Kant	W3	7	4630	Die 3 Kritiken	4

- Schema kombiniert Informationen verschiedener unpassender Entitytypen
  - Löschen von Elementen eines Entitytyps kann Verlust eines anderen Entitytyps bewirken
    - Löschen des Eintrags zu „Der Wiener Kreis“ (die einzige Vorlesung von Popper) würde auch Informationen zu Popper löschen
    - Alternative: Prüfen der *gesamten* Datenbank, ob dieser Eintrag die einzige Vorlesung von Popper ist. In diesem Fall durch NULL-Werte ersetzen

6. Relationale Entwurfstheorie

Informatik 5 - Informationssysteme und Datenbanken

RWTH AACHEN UNIVERSITY

Funktionale Abhängigkeiten

- Das zentrale Konzept der relationalen Entwurfstheorie

- Sei X die Attributmenge eines Relationenschemas $\mathcal{R}$. Die funktionalen Abhängigkeiten über X bilden eine zweistellige Relation „→" auf den Attributmengen aus X:

$$\alpha \to \beta, \qquad \text{für } \alpha, \beta \subseteq X.$$

(gesprochen: von Alpha nach Beta)

- In Worten: β ist funktional abhängig von α
  oder die α-Werte bestimmen die β-Werte funktional (d.h. eindeutig)

- Für zwei Attributmengen $\alpha, \beta \subseteq X$ und eine Relation R sagen wir R *erfüllt* die funktionale Abhängigkeit $\alpha \to \beta$, wenn gilt:

$$r.\alpha = t.\alpha \text{ impliziert } r.\beta = t.\beta \text{ für alle } r, t \in R.$$

Funktionale Abhängigkeiten

- Verallgemeinerung der Schlüsseleigenschaft
  - Eindeutigkeitseigenschaft der Schlüssel als funktionale Abhängigkeit:

$$\alpha \rightarrow X$$

- Eine funktionale Abhängigkeit $\alpha \rightarrow \beta$ lässt sich ebenfalls als intrarelationale Abhängigkeit auffassen:

$$\sigma_{\alpha \rightarrow \beta}: \mathrm{Rel}(X) \rightarrow \{true, false\}, \qquad R \longmapsto \begin{cases} true, & \text{falls } \alpha \rightarrow \beta \text{ in } R \text{ gilt} \\ false, & \text{sonst} \end{cases}$$

- Ist $\beta \subseteq \alpha$, so heißt $\alpha \rightarrow \beta$ eine *triviale Abhängigkeit*

- Funktionale Abhängigkeiten werden auch als FDs (functional dependencies) abgekürzt

Funktionale Abhängigkeiten – Beispiel

- Betrachte Schema $\mathcal{R}$ mit Attributmenge $\{A, B, C, D\}$ und FD $\{A\} \rightarrow \{B\}$.

  Die Ausprägung r erfüllt diese FD:

  nur für die Tupel t_2, t_3 gilt $t_2.A = t_3.A \ (= a_1)$

  und für diese gilt ebenfalls $t_2.B = t_3.B \ (= b_1)$

  – diese Ausprägung erfüllt auch die FDs
    - $\{A\} \rightarrow \{C\}$
    - $\{A, B\} \rightarrow \{C\}$
    - $\{C, D\} \rightarrow \{B\}$

    nicht aber die FDs
    - $\{B\} \rightarrow \{C\}$
    - $\{A, B, C\} \rightarrow \{D\}$

	r			
	A	B	C	D
t_1	a_4	b_2	c_4	d_3
t_2	a_1	b_1	c_1	d_1
t_3	a_1	b_1	c_1	d_2
t_4	a_2	b_2	c_3	d_2
t_5	a_3	b_2	c_4	d_3

Funktionale Abhängigkeiten – Beispiel

- Wichtig:
  - funktionale Abhängigkeiten beschreiben die Menge aller gültigen Relationen (wenn nichts anderes gesagt wird)
  - üblicherweise wird gefragt: welche zusätzlichen FDs lassen sich aus den gegebenen FDs ableiten
  - es wird nicht gefragt: welche zusätzlichen FDs erfüllt diese konkrete Ausprägung

	r			
	A	B	C	D
t_1	a_4	b_2	c_4	d_3
t_2	a_1	b_1	c_1	d_1
t_3	a_1	b_1	c_1	d_2
t_4	a_2	b_2	c_3	d_2
t_5	a_3	b_2	c_4	d_3

- Übrigens: die Notation

$$\{A, B, C\} \to \{D\}$$

wird häufig auch abgekürzt, z.B. durch

$$A, B, C \to D.$$

Funktionale Abhängigkeiten – Beispiel

Student			
$\underline{MatrNr}$: *Inte*	Name: *String*	#Semester: *Int*	Status: *Status*
1234	Michael	6	eingeschrieben
5678	Andrea	4	eingeschrieben
4711	Sabine	8	beurlaubt
815	Franz	12	exmatrikuliert

$\sigma_{\alpha \to \beta}$	Wert in der Ausprägung	Wert in allen Ausprägungen
$MatrNr \to Name, Semester, Status$		
$Name \to Semester, Status, MatrNr$		
$Semester \to Status$		
$Status \to Semester$		
$MatrNr, Name \to Semester, Status$		

Überprüfen funktionaler Abhängigkeiten

- Ein einfacher Algorithmus zur Überprüfung einer FD:
  - Eingabe: eine Relation R und eine FD $\alpha \rightarrow \beta$
  - Ausgabe: *ja* genau dann, wenn $\alpha \rightarrow \beta$ in R erfüllt ist
  - Algorithmus:
    - sortiere R nach den α-Werten
    - falls alle Gruppen, bestehend aus Tupeln mit gleichen α-Werten, auch gleiche β-Werte aufweisen: *ja*; sonst: *nein*

- Die Laufzeit dieses Algorithmus wird durch die Sortierung dominiert
  - Komplexität $O\big(n \log(n)\big)$

Schlüssel

- Präzisierung des Schlüsselbegriffs
  - Dazu sei $\mathcal{R}$ ein Relationenschema mit Attributmenge X und funktionalen Abhängigkeiten F
- Superschlüssel (Eindeutigkeit) :
  - $\alpha \subseteq X$ heißt *Superschlüssel*, falls $\alpha \rightarrow X$ gilt
  - α bestimmt also alle anderen Attributwerte
  - X selbst ist stets auch ein Superschlüssel, da trivialerweise $X \rightarrow X$ gilt
- voll funktional abhängig (Minimalität):
  - $\beta \subseteq X$ heißt *voll funktional abhängig* von α, falls
    - $\alpha \rightarrow \beta$ gilt
    - $\alpha - \{A\} \nrightarrow \beta$ für alle $A \in \alpha$ gilt, d.h. α kann nicht „verkleinert" werden

Schlüssel

- Präzisierung des Schlüsselbegriffs
  - Dazu sei $\mathcal{R}$ ein Relationenschema mit Attributmenge X und funktionalen Abhängigkeiten F
- Schlüsselkandidat:
  - Eine Attributmenge $\alpha \subseteq X$ heißt *Schlüsselkandidat*, falls X voll funktional abhängig von α ist
- Primärschlüssel:
  - In einem Relationenschema wird einer der Schlüsselkandidaten als *Primärschlüssel* ausgewählt
  - Fremdschlüssel sollten z.B. immer nur auf den Primärschlüssel verweisen

Schlüssel – Beispiel

Orte			
Name	BLand	Vorwahl	EW
Frankfurt	Hessen	069	690.000
Frankfurt	Brandenburg	0335	60.000
München	Bayern	089	1.378.000.000
Passau	Bayern	0851	50.000

- Annahme: Ortsnamen sind innerhalb eines Bundeslandes eindeutig
- Schlüsselkandidaten:

$$\{\,\mathrm{Name}, \mathrm{BLand}\,\}, \qquad \{\,\mathrm{Name}, \mathrm{Vorwahl}\,\}$$

  beide sind minimal:
  - Städte in unterschiedlichen Bundesländern können denselben Namen besitzen
  - kleine Dörfer können sich dieselbe Vorwahl teilen

6. Relationale Entwurfstheorie

1. Funktionale Abhängigkeiten
2. **Armstrong-Kalkül**
3. Zerlegung von Relationen
4. Normalformen und Normalisierungen

Bestimmung aller funktionalen Abhängigkeiten

- Frage: Ausgehend von einer Menge funktionaler Abhängigkeiten F (beim Datenbank-Entwurf erstellt), welche zusätzlichen funktionalen Abhängigkeiten sind implizit immer erfüllt?

- *Beispiel*:
  - Erweiterung von Universitäts-Beispiel um Adressen
  - erster Entwurf für Professoren und Adressen:

$$\text{ProfessorenAdressen}(\ \text{PersNr}, \text{Name}, \text{Rang}, \text{Raum}, \text{Ort}, \text{Straße}, \text{Hausnummer}, \text{PLZ}, \text{Vorwahl}, \text{Bundesland}\ )$$

Bestimmung aller funktionalen Abhängigkeiten – Beispiel

$$\text{ProfessorenAdressen}(\text{PersNr}, \text{Name}, \text{Rang}, \text{Raum}, \quad \text{Ort}, \text{Straße}, \text{Hausnummer}, \text{PLZ}, \text{Vorwahl}, \text{Bundesland})$$

- Funktionale Abhängigkeiten F:
  - $\text{PersNr} \rightarrow \text{PersNr}, \text{Name}, \text{Rang}, \dots, \text{Vorwahl}, \text{Bundesland}$
  - $\text{Raum} \rightarrow \text{PersNr}$
  - $\text{Ort}, \text{Bundesland} \rightarrow \text{Vorwahl}$
  - $\text{Ort}, \text{Bundesland}, \text{Straße}, \text{Hausnummer} \rightarrow \text{PLZ}$
  - $\text{PLZ} \rightarrow \text{Ort}, \text{Bundesland}$
- implizierte funktionale Abhängigkeiten:
  - $\text{Raum} \rightarrow \text{PersNr}, \text{Name}, \text{Rang}, \dots, \text{Vorwahl}, \text{Bundesland}$
  - $\text{PLZ} \rightarrow \text{Vorwahl}$
  - …

Implizierte funktionale Abhängigkeiten (Formalisierung)

- Seien X eine Attributmenge und F eine Menge von funktionalen Abhängigkeiten über X.
- Semantische Grundlagen:
  - Eine Relationsausprägung R *erfüllt* F, falls R jede funktionale Abhängigkeit $f \in F$ erfüllt (man sagt auch: R ist ein Modell von F)
  - Schreibweise:

$$R \vDash F \text{ oder } R \vDash f$$

  - Die Menge aller *gültigen Ausprägungen* des Schemas $\mathcal{R}$ ist

$$\mathrm{Sat}(F) = \{\, R \in \mathrm{Rel}(X) \mid R \vDash F \,\}$$

  - Zwei Mengen F, G von funktionalen Abhängigkeiten über X heißen *äquivalent*, falls sie die gleichen gültigen Relationen definieren:

$$\mathrm{Sat}(F) = \mathrm{Sat}(G)$$

Implizierte funktionale Abhängigkeiten (Definition)

- Seien X eine Attributmenge und F eine Menge von funktionalen Abhängigkeiten über X.
- *Implikation*:
  - Wir sagen, dass die Menge funktionaler Abhängigkeiten F die funktionale Abhängigkeit f impliziert, falls alle F erfüllenden Relationenausprägungen $R \in \mathrm{Sat}(F)$ auch f erfüllen.
  - Schreibweise:

$$F \vDash f$$

$F \vDash f$ ist eine semantische Beziehung

- nicht praktikabel: Überprüfe jede gültige Ausprägung $R \in \mathrm{Sat}(F)$, ob f gilt
- stattdessen: Armstrong-Kalkül

Armstrong-Kalkül

- Algorithmische Bestimmung aller implizierten funktionalen Abhängigkeiten

- Hilfsmittel: Armstrong-Axiome

  Seien $\alpha, \beta, \gamma \subseteq X$ Attributmengen.

  - Reflexivität (A_1): Ist $\beta \subseteq \alpha$ eine Teilmenge von α, so gilt auch $\alpha \rightarrow \beta$.
  - Verstärkung (A_2): Falls $\alpha \rightarrow \beta$ gilt, so gilt auch $\alpha\gamma \rightarrow \beta\gamma$. Wobei hier $\alpha\gamma \coloneqq \alpha \cup \gamma$.
  - Transitivität (A_3): Falls $\alpha \rightarrow \beta$ und $\beta \rightarrow \gamma$ gilt, so gilt auch $\alpha \rightarrow \gamma$.

Armstrong-Kalkül

- Algorithmische Bestimmung aller implizierten funktionalen Abhängigkeiten

- *Ableitbar* :
  - Wir sagen, dass die funktionale Abhängigkeit f aus der Menge der funktionalen Abhängigkeiten F *ableitbar* ist, falls:

    Es gibt eine endliche Folge $f_1, \dots, f_{n-1}, f_n = f$, sodass für jedes $1 \leq i \leq n$ gilt:

    f_i erhält man aus $F \cup \{f_1, \dots, f_{i-1}\}$ durch Anwendung der Axiome A_1, A_2 oder A_3

  - Schreibweise

$$F \vdash f$$

$F \vdash f$ ist eine syntaktische Beziehung

Armstrong-Kalkül – Beispiel

- *Ableitbar* :

  $\text{ProfessorenAdressen(PersNr, Name, Rang, Raum,} \quad \text{Ort, Straße, Hausnummer, PLZ, Vorwahl, Bundesland)}$

- Funktionale Abhängigkeiten F:
  - $\text{Ort, Bundesland} \rightarrow \text{Vorwahl}$
  - $\text{PLZ} \rightarrow \text{Ort, Bundesland}$
  - ...
- Mithilfe der Transitivitätsregel (A_3) :Falls $\alpha \rightarrow \beta$ und $\beta \rightarrow \gamma$ gilt, so gilt auch $\alpha \rightarrow \gamma$ erhält man aus

$$\text{PLZ} \rightarrow \text{Ort, Bundesland,}$$

$$\text{Ort, Bundesland} \rightarrow \text{Vorwahl}$$

  die funktionale Abhängigkeit $\text{PLZ} \rightarrow \text{Vorwahl.}$

$$F \vdash (\text{PLZ} \rightarrow \text{Vorwahl})$$

Armstrong-Kalkül – Korrektheit und Vollständigkeit

- Liefert das Armstrong-Kalkül alle „gültigen" bzw. von F implizierten FDs?

  Sei X eine Attributmenge und F eine Menge von FDs über X

  – Zu F nennen wir $F^+ = \{ f \mid F \vdash f \}$ die *(geschlossene) Hülle* von F

  – Gilt also

$$F^+ = \{ f \mid F \vDash f \}\,?$$

- Ja, der Armstrong-Kalkül ist *korrekt* und *vollständig*.

  – *korrekt*: Für jede funktionale Abhängigkeit f mit $F \vdash f$ gilt auch $F \vDash f$

  (es lassen sich nur „gültige" funktionale Abhängigkeiten ableiten, „$\subseteq$")

  – *vollständig*: Jede von F implizierte funktionale Abhängigkeit f (also $F \vDash f$) lässt sich mithilfe des Armstrong-Kalküls ableiten, d.h. $F \vdash f$ („$\supseteq$")

  (Beweis der Korrektheit jetzt, Beweis der Vollständigkeit später)

Armstrong-Kalkül – Korrektheit

- Korrektheit der Reflexivitätsregel (A_1): Für $\beta \subseteq \alpha$ gilt $\alpha \rightarrow \beta$.
  - Seien $R \in \mathrm{Sat}(F)$ eine gültige Relationsausprägung, $\alpha \subseteq X$ und $\beta \subseteq \alpha$. Außerdem seien $t_1, t_2 \in R$ zwei beliebige Tupel mit $t_1[\alpha] = t_2[\alpha]$. Dann gilt auch $t_1[\beta] = t_2[\beta]$.

    Insgesamt: $\alpha \rightarrow \beta$ gilt auch in R.

Armstrong-Kalkül – Korrektheit

- Korrektheit der Verstärkungsregel (A_2): Für $\alpha \rightarrow \beta$ gilt auch $\alpha\gamma \rightarrow \beta\gamma$.
  - Seien $R \in \mathrm{Sat}(F)$ eine gültige Relationsausprägung, $\alpha, \beta, \gamma \subseteq X$ und $\alpha \rightarrow \beta \in F$. Außerdem seien $t_1, t_2 \in R$ zwei beliebige Tupel mit $t_1[\alpha \cup \gamma] = t_2[\alpha \cup \gamma]$. Dann gilt auch $t_1[\beta] = t_2[\beta]$ und $t_1[\gamma] = t_2[\gamma]$. Daraus folgt $t_1[\beta \cup \gamma] = \mathrm{t}_2[\beta \cup \gamma]$. Insgesamt: $\alpha \cup \gamma \rightarrow \beta \cup \gamma$ gilt auch in R.

Armstrong-Kalkül – Korrektheit

- Korrektheit der Transitivitätsregel (A_3): Für $\alpha \to \beta, \beta \to \gamma$ gilt auch $\alpha \to \gamma$
  - Seien $R \in \mathtt{Sat}(F)$ eine gültige Relationenausprägung, $\alpha, \beta, \gamma \subseteq X$ und $\alpha \to \beta,\ \beta \to \gamma \in F$. Außerdem seien $t_1, t_2 \in R$ zwei beliebige Tupel mit $t_1[\alpha] = t_2[\alpha]$. Wegen $\alpha \to \beta$, gilt also auch $t_1[\beta] = t_2[\beta]$. Wegen $\beta \to \gamma$, gilt dann auch $t_1[\gamma] = t_2[\gamma]$. Insgesamt: $\alpha \to \gamma$ gilt auch in R.

Armstrong-Kalkül – Erweiterung

Es ist für den Herleitungsprozess komfortabel, weitere Regeln hinzuzunehmen

- Erweiterung der Armstrong-Axiome um drei Regeln:

  Seien $\alpha, \beta, \gamma, \delta \subseteq X$ Attributmengen.

  - Vereinigung (A_4): Gelten $\alpha \rightarrow \beta$ und $\alpha \rightarrow \gamma$, so gilt auch $\alpha \rightarrow \beta\gamma$.
  - Dekomposition (A_5): Falls $\alpha \rightarrow \beta\gamma$ gilt, so gelten auch $\alpha \rightarrow \beta$ und $\alpha \rightarrow \gamma$.
  - Pseudotransitivität (A_6): Falls $\alpha \rightarrow \beta$ und $\beta\gamma \rightarrow \delta$ gilt, so gilt auch $\alpha\gamma \rightarrow \delta$.

Armstrong-Kalkül – Erweiterung

- Ableitung der Vereinigungsregel (A_4) :Gelten $\alpha \to \beta$ und $\alpha \to \gamma$, so gilt auch $\alpha \to \beta\gamma$.
  - Seien $\alpha \to \beta$ und $\alpha \to \gamma \in F$. Über die Grundregeln erhalten wir

$$(A_2) : \text{Da } \alpha \to \beta \text{ gilt } \alpha\gamma \to \beta\gamma$$

$$(A_2) : \text{Da } \alpha \to \gamma \text{ gilt } \alpha \to \alpha\gamma$$

$$(A_3) : \text{Da } \alpha \to \alpha\gamma \text{ und } \alpha\gamma \to \beta\gamma \text{ gilt } \alpha \to \beta\gamma$$

Armstrong-Kalkül – Erweiterung

- Ableitung der Dekompositionsregel ($A_{5)}$):

  Falls $\alpha \rightarrow \beta\gamma$ gilt, so gelten auch $\alpha \rightarrow \beta$ und $\alpha \rightarrow \gamma$.

  – Sei $\alpha \rightarrow \beta\gamma \in F$. Über die Grundregeln erhalten wir

$$(A_1) : \text{Da } \beta\gamma \coloneqq \beta \cup \gamma \text{ gilt } \beta\gamma \rightarrow \beta$$

$$(A_1) : \text{Da } \beta\gamma \coloneqq \beta \cup \gamma \text{ gilt } \beta\gamma \rightarrow \gamma$$

$$(A_3) : \text{Da } \alpha \rightarrow \beta\gamma \text{ und } \beta\gamma \rightarrow \beta \text{ gilt } \alpha \rightarrow \beta$$

$$(A_3) : \text{Da } \alpha \rightarrow \beta\gamma \text{ und } \beta\gamma \rightarrow \gamma \text{ gilt } \alpha \rightarrow \gamma$$

Armstrong-Kalkül – Erweiterung

- Ableitung der Pseudotransitivitätsregel (A_6) :

  Falls $\alpha \rightarrow \beta$ und $\beta\gamma \rightarrow \delta$ gilt, so gilt auch $\alpha\gamma \rightarrow \delta$

  Sei $\alpha \rightarrow \beta$ und $\beta\gamma \rightarrow \delta \in F$. Über die Grundregeln erhalten wir

$$(A_2) : \text{Da } \alpha \rightarrow \beta \text{ gilt } \alpha\gamma \rightarrow \beta\gamma$$

$$(A_3) : \text{Da } \alpha\gamma \rightarrow \beta\gamma \text{ und } \beta\gamma \rightarrow \delta \text{ gilt } \alpha\gamma \rightarrow \delta$$

Armstrong-Kalkül – Vereinigung

$$\text{ProfessorenAdressen(PersNr, Name, Rang, Raum, Ort, Straße, Hausnummer, PLZ, Vorwahl, Bundesland)}$$

- Funktionale Abhängigkeiten F:
  - $\text{Ort, Bundesland} \rightarrow \text{Vorwahl}$
  - $\text{Ort, Bundesland, Straße, Hausnummer} \rightarrow \text{PLZ}$
  - …

- Beispiel: $f = \text{Ort, Bundesland, Straße, Hausnummer} \rightarrow \text{PLZ, Vorwahl}$ ist ebenfalls aus F ableitbar
  - $(A_1) : f_1 = (\text{Ort, Bundesland, Straße, Hausnummer} \rightarrow \text{Ort, Bundesland})$
  - $(A_3) : f_2 = (\text{Ort, Bundesland, Straße, Hausnummer} \rightarrow \text{Vorwahl})$
  - $\boldsymbol{(A_4) : f = f_3 = (\textbf{Ort, Bundesland, Strasse, Hausnummer} \rightarrow \textbf{PLZ, Vorwahl})}$

Attributhülle

- Oft ist man nicht an der gesamten Hülle F^+ interessiert:
  - Welche Attribute sind unter einer gegebenen Menge von FDs F von einer bestimmten Attributmenge α funktional bestimmt?
  - Man nennt α^+ die Attributhülle von α unter F

$$\alpha^+ = \{\, x \mid \text{es gibt } \alpha \to \beta \in F^+ \text{ mit } x \in \beta\}$$

  - Algorithmus zur Bestimmung von $\mathrm{AttrHülle}(F, \alpha)$:
    - $Erg := \alpha$

      $\textbf{while}$ (Änderungen an Erg) $\textbf{do}$

      $\quad\textbf{foreach}$ FD $\beta \to \gamma$ $\textbf{in}$ F $\textbf{do}$

      $\quad\quad\textbf{if}$ $\beta \subseteq Erg$ $\textbf{then}$ $Erg := Erg \cup \gamma;$

      Ausgabe $\alpha^+ = Erg;$

$\kappa \subseteq X$ ist genau dann ein Superschlüssel, falls gilt:
$$\kappa^+ = X$$

Attributhülle – Beispiel

- Attributhülle

$$\alpha^+ = \{\, x \mid \text{es gibt } \alpha \to \beta \in F^+ \text{ mit } x \in \beta\}$$

- Beispiel:

$$F = \{A \to C, B \to A, AB \to C\}$$

  – $\mathrm{AttrHülle}(\mathrm{F}, \{\mathrm{B}\})$:

  $Erg = \{B\}$

  Durchlaufe F:

  $A \to C$: $\{B\}$, $B \to A$: $\{A, B\}$, $AB \to C$: $\{A, B, C\}$

  Durchlaufe F nochmal:

  keine Änderung

  Ausgabe $B^+ = \{A, B, C\}$

Attributhülle

$$\alpha^+ = \{\, x \mid \text{es gibt } \alpha \to \beta \in F^+ \text{ mit } x \in \beta\}$$

- Lemma **L1**: Die Attributhülle besitzt folgende Eigenschaft.

  Für jede Teilmenge $V \subseteq \alpha^+$ gilt auch $\alpha \to V \in F^+$.

- Beweis (Skizze):

  Wir beschränken uns auf den Fall $V = \{a_1, a_2\}$. Da $V \subseteq \alpha^+$ ist, gibt es $\beta_1 = \{a_1, \dots\}$ und $\beta_2 = \{a_2, \dots\}$ mit $\alpha \to \beta_1,\ \alpha \to \beta_2 \in F^+$. Mit der Reflexivitätsregel (A_1) sind auch $\beta_1 \to \{a_1\}, \beta_2 \to \{a_2\} \in F^+$. Mit der Transitivitätsregel (A_3) und

$$\alpha \to \beta_1, \qquad \beta_1 \to \{a_1\}\alpha \to \beta_2, \qquad \beta_2 \to \{a_2\}$$

  sind auch $\alpha \to \{a_1\}, \alpha \to \{a_2\} \in F^+$. Mit der Vereinigungsregel (A_4) erhalten wir

$$\alpha \to \{a_1, a_2\} \in F^+.$$

Armstrong-Kalkül – Vollständigkeit

- Das Armstrong-Kalkül ist vollständig, d.h. jede von funktionalen Abhängigkeiten F implizierte FD f lässt sich mithilfe des Armstrong-Kalküls ableiten. Es gilt also

$$F \vDash f \Rightarrow F \vdash f$$

- Beweis (durch Kontraposition):
  - Wir zeigen die Aussage $\mathrm{F} \nvdash f \Rightarrow F \nvDash f$. D.h. ist eine funktionale Abhängigkeit f nicht ableitbar, so wird sie auch nicht von F impliziert. Sei dazu $f = \alpha \rightarrow \beta$ eine FD mit $F \nvdash f$ und X die Menge aller Attribute aus F und f.
  - Um zu zeigen, dass $F \nvDash f$ gilt, konstruieren wir eine Relation R, in der F gilt, aber f nicht.

  Konstruktion:

  Seien $\alpha^+ = \{a_1, \dots, a_n\}$ und $X \setminus \alpha^+ = \{b_1, \dots, \mathrm{b}_m\}$.

  Konstruiere R wie rechts.

R	$a_1, \dots, a_n$	$b_1, \dots, b_m$
r	$1, \dots, 1$	$1, \dots, 1$
t	$1, \dots, 1$	$0, \dots, 0$

Armstrong-Kalkül – Vollständigkeit

- Zeige für Vollständigkeit: $F \nvdash f \Rightarrow F \nvDash f$
- Hilfslemma 1:

  Es gilt $R \vDash F$, d.h. alle FDs in F werden von R erfüllt.

- Beweis (durch Widerspruch):

  Angenommen es existiert $V \rightarrow W \in F$ mit $R \nvDash V \rightarrow W$. Dann muss für die beiden Tupel $r, t \in R$ gelten, dass $r[V] = t[V]$ und $r[W] \neq t[W]$ ist. Nach Konstruktion von R kann $r[V] = t[V]$ nur gelten, wenn $V \subseteq \{a_1, \ldots, a_n\} = \alpha^+$ ist. Ebenso impliziert $r[W] \neq t[W]$, dass $W \nsubseteq \alpha^+$ ist.

  Nach dem vorigen Lemma **L1** folgt aus $V \subseteq \alpha^+$, dass $\alpha \rightarrow V \in F^+$ ist. Mit der

  Transivititätsregel (A_3) erhalten wir aus $\alpha \rightarrow V,\ V \rightarrow W \in F^+$, dass auch $\alpha \rightarrow W \in F^+$ ist.

  Dies liefert einen Widerspruch zu $W \nsubseteq \alpha^+$.

Armstrong-Kalkül – Vollständigkeit

- Zeige für Vollständigkeit: $F \nvdash f \Rightarrow F \nvDash f$
- Hilfslemma 2:

  Es gilt $R \nvDash f$, d.h. f wird von R nicht erfüllt.

- Beweis:

  Da f nicht aus F ableitbar ist $(F \nvdash f)$ gilt insbesondere $\beta \nsubseteq \alpha^+$. Also existiert ein $b_k \in \beta$ mit $b_k \in X \setminus \alpha^+$. Direkt aus der Konstruktion von R folgt dann, dass $r[\alpha] = t[\alpha]$ ist, aber $r[b_k] = 1 \neq 0 = t[b_k]$.

  Insgesamt erhalten wir, dass $f = \alpha \to \beta$ von R nicht erfüllt wird.

- Es gilt also $R \vDash F$, aber $R \nvDash f$. Darauf folgt $F \nvDash f$.

Kanonische Überdeckung – Motivation

- Um für zwei Mengen F und G zu entscheiden, ob sie äquivalent sind ($\mathrm{Sat}(F) = \mathrm{Sat}(G)$), reicht es $F^+ = G^+$ zu überprüfen.

Warum?
Freiwillige Übung

- Im Allgemeinen ist die Hülle F^+ einer Menge von FDs sehr groß
- Vor allem bei Datenbankmodifikationen:
  - Überprüfen der Konsistenz anhand von F^+ sehr aufwändig (auch viele triviale Abhängigkeiten)
  - minimale Menge von „erzeugenden“ funktionalen Abhängigkeiten wünschenswert

- Statt der Hülle F^+: kanonische Überdeckung

Kanonische Überdeckung (Definition)

- Sei F eine Menge von funktionalen Abhängigkeiten über einer Attributmenge X. Dann heißt F_c eine *kanonische Überdeckung* von F, falls gilt:
  1. $F_c^+ = F^+$, d.h. $\mathrm{Sat}(F_c) = \mathrm{Sat}(F)$ (äquivalent)
  2. In F_c existieren keine FDs $\alpha \to \beta$ bei denen α oder β überflüssige Attribute enthalten. D.h.
     - Für alle $A \in \alpha$: $\left(F_c \setminus (\alpha \to \beta)\right) \cup (\alpha \setminus A \to \beta) \not\equiv F_c$ (nicht äquivalent)
     - Für alle $B \in \beta$: $\left(F_c \setminus (\alpha \to \beta)\right) \cup (\alpha \to \beta \setminus B) \not\equiv F_c$
  3. Jede linke Seite einer FD ist einzigartig in F_c (sonst ersetze durch Vereinigungen)

- Beispiel:

$$F = \{A \to C, B \to A, AB \to C\}$$

Eine kanonische Überdeckung ist $\mathrm{F_c} = \{A \to C, B \to A\}$.
(Algorithmus: nächste Folie)

Kanonische Überdeckung – Algorithmus

- Eingabe: Menge von FDs F, Ausgabe: Kanonische Überdeckung F_c
  1. Setze $F_c = F$
  2. Führe für jede FD $\alpha \rightarrow \beta \in F_c$ eine Linksreduktion durch:
     - Überprüfe für alle $A \in \alpha$, ob A überflüssig ist. D.h. ob gilt: $\beta \subseteq \mathrm{AttrHülle}(F_c, \alpha \setminus A)$

       Falls ja: ersetze $\alpha \rightarrow \beta$ durch $\alpha \setminus A \rightarrow \beta$.
  3. Führe für jede FD $\alpha \rightarrow \beta \in F_c$ eine Rechtsreduktion durch:
     - Überprüfe für alle $B \in \beta$, ob B überflüssig ist. D.h. ob gilt:

       $$B \in \mathrm{AttrHülle}\big((F_c \setminus (\alpha \rightarrow \beta)) \cup (\alpha \rightarrow \beta \setminus B), \alpha\big)$$

       Falls ja: ersetze $\alpha \rightarrow \beta$ durch $\alpha \rightarrow \beta \setminus B$.
  4. Entferne die im 3. Schritt entstandenen FDs der Form $\alpha \rightarrow \emptyset$
  5. Fasse über die Vereinigungsregel FDs der Form $\alpha \rightarrow \beta_1, \ldots, \alpha \rightarrow \beta_n$

     $$\alpha \rightarrow \beta_1 \cup \cdots \cup \beta_n$$

Kanonische Überdeckung – Beispiel

- Beispiel:

$$F = \{A \to C, B \to A, AB \to C\}$$

1. Linksreduktion:
   - $A \to C$: C ist nicht in $(\{A\} \setminus \{A\})^+ = \emptyset^+$. Ok
   - $B \to A$: Ok
   - $AB \to C$: Überprüfe A. Ist $C \in (\{A, B\} \setminus \{A\})^+$? Ja, denn

$$B^+ = \{ \qquad \}$$

   Ersetze $AB \to C$ durch $B \to C$.

   B muss nun in $B \to C$ nicht mehr überprüft werden.

- Zwischenergebnis:

$$F_c = \{A \to C, B \to A, B \to C\}$$

Kanonische Überdeckung – Beispiel

- Beispiel:

$$F = \{A \to C, B \to A, AB \to C\}$$

1. Linksreduktion:
   - $A \to C$: C ist nicht in $(\{A\} \setminus \{A\})^+ = \emptyset^+$. Ok
   - $B \to A$: Ok
   - $AB \to C$: Überprüfe A. Ist $C \in (\{A, B\} \setminus \{A\})^+$? Ja, denn

   $$B^+ = \{B, A, C\}$$

   Ersetze $AB \to C$ durch $B \to C$.

   B muss nun in $B \to C$ nicht mehr überprüft werden.

- Zwischenergebnis:

$$F_c = \{A \to C, B \to A, B \to C\}$$

Kanonische Überdeckung – Beispiel

- Zwischenergebnis:

$$F_c = \{A \to C, B \to A, B \to C\}$$

2. Rechtsreduktion:
   - $A \to C$: Überprüfe C. $\mathrm{AttrHülle}(\{A \to \emptyset, B \to A, B \to C\},\ A)$

     $= \{$

     Also ist C rechts nicht überflüssig.
   - $B \to A$: Analog: A ist rechts nicht überflüssig.
   - $B \to C$: Überprüfe C. $\mathrm{AttrHülle}(\{A \to C, B \to A, B \to \emptyset\},\ B)$

     $= \{$

     Also ist C auf der rechten Seite überflüssig. Ersetze $B \to C$ durch $B \to \emptyset$.

- Neues Zwischenergebnis:

$$F_c = \{A \to C, B \to A, B \to \emptyset\}$$

Kanonische Überdeckung – Beispiel

- Zwischenergebnis:

$$F_c = \{A \to C, B \to A, B \to C\}$$

2. Rechtsreduktion:
   - $A \to C$: Überprüfe C. $\mathrm{AttrHülle}(\{A \to \emptyset, B \to A, B \to C\},\ A)$

     $$= \{A\}$$

     Also ist C rechts nicht überflüssig.
   - $B \to A$: Analog: A ist rechts nicht überflüssig.
   - $B \to C$: Überprüfe C. $\mathrm{AttrHülle}(\{A \to C, B \to A, B \to \emptyset\},\ B)$

     $$= \{$$

     Also ist C auf der rechten Seite überflüssig. Ersetze $B \to C$ durch $B \to \emptyset$.

- Neues Zwischenergebnis:

$$F_c = \{A \to C, B \to A, B \to \emptyset\}$$

Kanonische Überdeckung – Beispiel

- Zwischenergebnis:

$$F_c = \{A \to C, B \to A, B \to C\}$$

2. Rechtsreduktion:

   - $A \to C$: Überprüfe C. $\text{AttrHülle}(\{A \to \emptyset, B \to A, B \to C\},\ A)$

     $$= \{A\}$$

     Also ist C rechts nicht überflüssig.

   - $B \to A$: Analog: A ist rechts nicht überflüssig.

   - $B \to C$: Überprüfe C. $\text{AttrHülle}(\{A \to C, B \to A, B \to \emptyset\},\ B)$

     $$= \{B, A, C\}$$

     Also ist C auf der rechten Seite überflüssig. Ersetze $B \to C$ durch $B \to \emptyset$.

- Neues Zwischenergebnis:

$$F_c = \{A \to C, B \to A, B \to \emptyset\}$$

Kanonische Überdeckung – Beispiel

- Zwischenergebnis:

$$F_c = \{A \to C, B \to A, B \to \emptyset\}$$

3. Entferne $\alpha \to \emptyset$:
   - Entferne die Abhängigkeit $B \to \emptyset$

4. Vereinigen:
   - Hier ist nichts zu tun.

- Ergebnis:

$$F_c = \{A \to C, B \to A\}$$

6. Relationale Entwurfstheorie

Informatik 5 - Informationssysteme und Datenbanken

RWTH AACHEN UNIVERSITY

Zerlegung

- funktionale Abhängigkeiten: mächtiges Werkzeug um Konsistenzbedingungen zu modellieren
- In der Sprache der FDs kann nun ausgedrückt werden, welche Schemata „gut“ und welche „schlecht“ sind (siehe „Normalformen und Normalisierungen“)
- „schlechte“ Schemata können durch Zerlegung/Dekomposition in die Normalformen überführt werden.
- Welche Anomalien können bei einem schlechtem Relationenschema auftreten?

Zerlegung

- nicht zusammenpassende Informationen
- Basis jeder Normalisierung:

  Zerlege das Relationenschema $\mathcal{R}$ in Schemata $\mathcal{R}_1, \ldots, \mathcal{R}_n$

- Kriterien für eine korrekte Zerlegung?

1. Verlustlosigkeit:
   - Die Informationen einer Relationsausprägung R von $\mathcal{R}$ müssen aus den resultierenden Ausprägungen $R_1, \ldots, R_n$ wieder komplett rekonstruiert werden können.
2. Abhängigkeitserhaltung:
   - Die für $\mathcal{R}$ geltenden funktionalen Abhängigkeiten müssen sich auf $\mathcal{R}_1, \ldots, \mathcal{R}_n$ übertragen lassen

(Formalisierung auf den nächsten Folien)

Gültig, Verlustlos

Betrachte die Zerlegung von $\mathcal{R}$ mit FDs F in $\mathcal{R}_1, \mathcal{R}_2$.
Seien X, X_1 und X_2 die entsprechenden Attributmengen.

- *Gültigkeit*:

  Die Zerlegung von $\mathcal{R}$ heißt *gültig*, falls gilt

$$X = X_1 \cup X_2.$$

- *Verlustlosigkeit*:

  Sei $R \in \mathrm{Sat}(F)$ eine beliebige gültige Ausprägung von $\mathcal{R}$. Definiere

$$R_1 \coloneqq \Pi_{X_1}(R), \qquad R_2 \coloneqq \Pi_{X_2}(R).$$

  Die Zerlegung von $\mathcal{R}$ in $\mathcal{R}_1, \mathcal{R}_2$ heißt *verlustlos*, falls für alle $R \in \mathrm{Sat}(F)$ gilt

$$R = R_1 \bowtie R_2$$

Verlustlos – Beispiel

- Verlust von Informationen:

  Relationsschema $\text{Biertrinker}(\text{Kneipe}, \text{Gast}, \text{Bier})$ mit FD $\{\text{Kneipe}, \text{Gast} \rightarrow \text{Bier}\}$

Biertrinker		
Kneipe	Gast	Bier
Domkeller	Kemper	Pils
Domkeller	Eickler	Hefeweizen
Die Kiste	Eickler	Pils

  (Kneipe, Gast und Bier werden eindeutig durch ihren Namen bestimmt)

  – Welches Getränk ein Gast trinkt hängt von der besuchten Kneipe ab

  – Zerlegung in $\text{Besucht}(\text{Kneipe}, \text{Gast})$, $\text{Trinkt}(\text{Gast}, \text{Bier})$

Besucht	
Kneipe	Gast
Domkeller	Kemper
Domkeller	Eickler
Die Kiste	Eickler

Trinkt	
Gast	Bier
Kemper	Pils
Eickler	Hefeweizen
Eickler	Pils

Nicht verlustlose Zerlegung – ein Beispiel

Besucht	
Kneipe	**Gast**
Domkeller	Kemper
Domkeller	Eickler
Die Kiste	Eickler

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

$\bowtie$

Besucht $\bowtie$ Trinkt		
Kneipe	**Gast**	**Bier**
Domkeller	Kemper	Pils
Domkeller	*Eickler*	*Hefeweizen*
Domkeller	Eickler	Pils
Die Kiste	*Eickler*	*Hefeweizen*
Die Kiste	Eickler	Pils

$\text{Biertrinker} \neq \text{Besucht} \bowtie \text{Trinkt}$

Ein Mehr an Tupeln bedeutet einen Verlust an Informationen!

Kriterien für Verlustlosigkeit

- Finde Bedingungen unter denen eine verlustlose Zerlegung garantiert ist

  Seien $\mathcal{R}, \mathcal{R}_1, \mathcal{R}_2$ Relationenschemata mit den Attributmengen X, X_1 und X_2 wie vorher definiert.

- Hinreichend:

  Die Zerlegung von $\mathcal{R}$ ist verlustlos, falls gilt

$$(X_1 \cap X_2) \rightarrow X_1 \in F^+ \quad \text{oder} \quad (X_1 \cap X_2) \rightarrow X_2 \in F^+$$

- In anderen Worten:

  Die Zerlegung von $\mathcal{R}$ ist verlustlos, falls gilt

$$X_1 \subseteq (X_1 \cap X_2)^+ \quad \text{oder} \quad X_2 \subseteq (X_1 \cap X_2)^+$$

- Intutition: Joinattribut bestimmt eines der beiden Teilschemata

- Im Biertrinker-Beispiel: einzige nicht-triviale Abhängigkeit war

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

Verlustlos – Beispiel

- Verlustlose Zerlegung:

Relationsschema $\mathrm{Eltern}(\ \mathrm{Vater}, \mathrm{Mutter}, \mathrm{Kind}\ )$ mit FDs $\{\ \mathrm{Kind} \rightarrow \mathrm{Vater},\ \ \mathrm{Kind} \rightarrow \mathrm{Mutter}\ \}$

(biol.) Eltern		
Vater	Mutter	Kind
Johann	Martha	Else
Johann	Maria	Theo
Heinz	Martha	Cleo

(Personen sind durch ihren Vornamen eindeutig bestimmt)

Verlustlose Zerlegung

$\mathrm{Väter}(\ \mathrm{Vater}, \mathrm{Kind}\ )$

$\mathrm{Mütter}(\ \mathrm{Mutter}, \mathrm{Kind}\ )$

Väter	
Vater	Kind
Johann	Else
Johann	Theo
Heinz	Cleo

Mütter	
Mutter	Kind
Martha	Else
Maria	Theo
Martha	Cleo

Verlustlos – Beispiel

Väter	
Vater	Kind
Johann	Else
Johann	Theo
Heinz	Cleo

Mütter	
Mutter	Kind
Martha	Else
Maria	Theo
Martha	Cleo

$\bowtie$

Väter $\bowtie$ Mütter		
Vater	Mutter	Kind
Johann	Martha	Else
Johann	Maria	Theo
Heinz	Martha	Cleo

$\text{Kind} \rightarrow \text{Vater},$
$\text{Kind} \rightarrow \text{Mutter}$

Abhängigkeitserhaltung

- Gegeben Zerlegung von $\mathcal{R}$ mit FDs F in die Schemata $\mathcal{R}_1, \dots, \mathcal{R}_n$ mit entsprechenden Attributmengen $X, X_1, \dots, X_n$

  Die Überprüfung der Konsistenzbedinungen bei Einfügen, Löschen und Updaten auf

  $$\mathcal{R}_1 \bowtie \cdots \bowtie \mathcal{R}_n$$

  benötigt Joins, falls funktionale Abhängigkeiten sich nach der Zerlegung auf Attribute verschiedenen Relationenschemata beziehen (Überprüfung bei jeder Transaktion). Ineffzient!
- Wunsch: alle FDs, die für das Schema $\mathcal{R}$ gelten, sollen *lokal* (ohne Joins) auf den einzelnen Schemata $\mathcal{R}_1, \dots, \mathcal{R}_n$ überprüfbar sein.
- Formal*: Abhängigkeitserhaltend*:

  Sei $F_{\mathcal{R}_i}$ die Menge der FDs $\alpha \to \beta \in F^+$ mit $\alpha, \beta \subseteq X_i$, (also deren Attribute aus X_i sind) .

  Die Zerlegung von $\mathcal{R}$ in $\mathcal{R}_1, \dots, \mathcal{R}_n$ heißt *abhängigkeitserhaltend*, falls $F^+ = \left(F_{\mathcal{R}_1} \cup \cdots \cup F_{\mathcal{R}_n}\right)^+$

  Diese Eigenschaft wird auch *Hüllentreue* der Zerlegung genannt.

Abhängigkeitserhaltung – Beispiel

- Beispiel: Verlustlose aber nicht abhängigkeitserhaltende Zerlegung

  Betrachte das Relationsschema

$$\text{PLZVerzeichnis}(\ \text{BLand}, \text{Ort}, \text{Straße}, \text{PLZ}\ )$$

  - Annahme:
    - Ortsnamen sind innerhalb eines Bundeslandes eindeutig
    - PLZ’n ändern sich nicht innerhalb einer Straße
  - Funktionale Abhängigkeiten:
    - $\text{PLZ} \rightarrow \text{Ort}, \text{BLand}$
    - $\text{BLand}, \text{Ort}, \text{Straße} \rightarrow \text{PLZ}$
  - Betrachte die Zerlegung in

$$\text{Straßen}(\ \text{PLZ}, \text{Straße}\ )$$

$$\text{Orte}(\ \text{PLZ}, \text{Ort}, \text{BLand}\ )$$

Abhängigkeitserhaltung – Beispiel

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

Straßen	
PLZ	Straße
60313	Goethestraße
60437	Galgenstraße
15234	Goethestraße

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

Abhängigkeitserhaltung – Beispiel

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

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

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

Siehe funktionale Abhängigkeit: BLand, Ort, Straße→PLZ

Ungültiger Eintrag kann nur durch $\mathrm{PLZVerzeichnis} = \mathrm{Straßen} \bowtie \mathrm{Orte}$ überprüft werden

Abhängigkeitserhaltung – Beispiel

- Zerlegung von $\mathrm{PLZVerzeichnis}(\ \mathrm{BLand}, \mathrm{Ort}, \mathrm{Straße}, \mathrm{PLZ}\ )$ in

$$\mathrm{Straßen}(\ \mathrm{PLZ}, \mathrm{Straße}\ ), \qquad \mathrm{Orte}(\ \mathrm{PLZ}, \mathrm{Ort}, \mathrm{BLand}\ )$$

- Also:
  - Die Zerlegung ist verlustlos, da $\{\ \mathrm{PLZ}, \mathrm{Straße}\ \} \cap \{\ \mathrm{Ort}. \mathrm{Straße}, \mathrm{PLZ}\ \} = \{\ \mathrm{PLZ}\ \}$ ist und die funktionale Abhängigkeit

    $$\mathrm{PLZ} \rightarrow \mathrm{PLZ}, \mathrm{Ort}, \mathrm{BLand}$$

    gilt.
  - Die Zerlegung ist nicht abhängigkeitserhaltend, da die funktionale Abhängigkeit

    $$\mathrm{Ort}, \mathrm{BLand}, \mathrm{Straße} \rightarrow \mathrm{PLZ}$$

    keiner der neuen Relationen $\mathrm{Straßen}$ oder Orte zugeordnet werden kann.

6. Relationale Entwurfstheorie

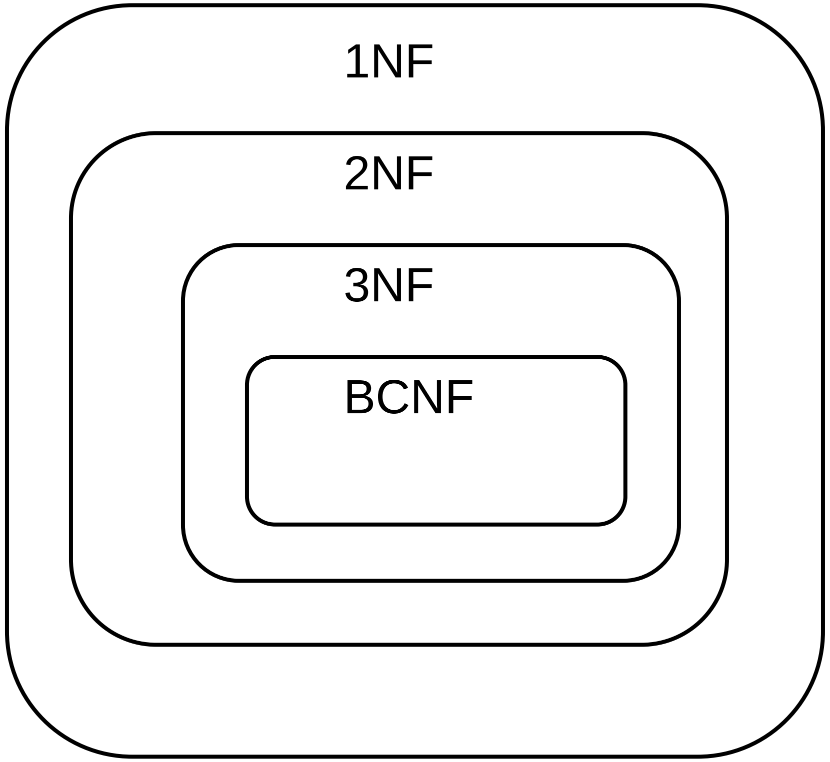
1. Funktionale Abhängigkeiten
2. Armstrong-Kalkül
3. Zerlegung von Relationen
4. **Normalformen und Normalisierungen**

Normalformen und Normalisierungen

- Vermeidung von Redundanzen und damit zusammenhängender Anomalien

- Normalformen unterscheiden sich in der Strenge der Anforderungen an die funktionalen Abhängigkeiten

- Normalisierungsalgorithmen berechnen zu einem Schema eine <u>verlustlose</u> und, *wenn möglich*, auch <u>abhängigkeitserhaltende</u> Zerlegung



Erste Normalform

- Alle Attribute haben atomare Wertebereiche (zB String, Integer, …)
  - zusammengesetzte, mengenwertige oder relationenwertige Attribute sind nicht erlaubt

- Beispiel: nicht in 1NF

Eltern-1		
Vater	Mutter	Kind
Johann	Martha	$\{\text{Else}, \text{Lucia}\}$
Johann	Maria	$\{\text{Theo}, \text{Jose}\}$
Heinz	Martha	$\{\text{Cleo}\}$

in 1NF

Eltern-2		
Vater	Mutter	Kind
Johann	Martha	Else
Johann	Martha	Lucia
Johann	Maria	Theo
Johann	Maria	Jose
Heinz	Martha	Cleo

- Im Folgenden gehen wir stets von Relationen in 1NF aus

NF²-Modelle:
non-first-normal-form
Modelle

Zweite Normalform

- Betrachte wieder das Beispiel

ProfVorl						
PersNr	Name	Rang	Raum	VorlNr	Titel	SWS
2125	Sokrates	W3	226	5041	Ethik	4
2125	Sokrates	W3	226	5049	Mäeutik	2
2125	Sokrates	W3	226	4052	Logik	4
…	…	…	…	…	…	…

- Anschaulich ist die 2NF verletzt, wenn eine Relation Informationen aus mehreren Konzepten enthält (hier: $\mathrm{ProfVorl}$ entspricht $\mathrm{Professor} \bowtie \mathrm{Vorlesung}$)
- Attribute dürfen nicht von Teilmengen von Schlüsselkandidaten abhängen
- FDs: $\mathrm{PersNr}, \mathrm{VorlNr} \rightarrow \mathrm{PersNr}, \dots, \mathrm{SWS}$

  $\mathrm{PersNr} \rightarrow \mathrm{Name}, \mathrm{Rang}, \mathrm{Raum}$

  $\mathrm{VorlNr} \rightarrow \mathrm{Titel}, \mathrm{SWS}$

Zweite Normalform (Definition)

- *Nichtschlüssel-Attribut (NSA)*:

  Seien $\kappa_1, \dots, \kappa_m$ die Schlüsselkandidaten einer gegebenen Relation $\mathcal{R}$ und X die zugehörige Attributmenge. Dann heißt $X \setminus (\kappa_1 \cup \cdots \cup \kappa_m)$ die Menge aller *Nichtschlüssel-Attribute*.

- *Zweite Normalform*:

  Ein Relationenschema $\mathcal{R}$ mit FDs F ist in *zweiter Normalform*, falls für jedes Nichtschlüssel-Attribut A und jeden Schlüsselkandidaten κ_i gilt:

  - A ist voll funktional abhängig von κ_i, d.h. für kein Attribut $B \in \kappa_i$ gilt $\kappa_i \setminus \{B\} \rightarrow A \in F^+$

- Beispiel:
  - Einziger Schlüsselkandidat von ProfVorl ist $\kappa_1 = \{\mathrm{PersNr}, \mathrm{VorlNr}\}$. Es gilt aber $\mathrm{PersNr} \rightarrow \mathrm{Name} \in F^+$. Also ist Name nicht voll funktional abhängig von κ_1.

Zweite Normalform – Algorithmus (Skizze)

- Ein Relationenschema $\mathcal{R}$ kann wie folgt in zweite Normalform überführt werden
  a) Fasse alle Nichtschlüsselattribute, die nur von einem Teilschlüssel abhängen, mit diesem Teilschlüssel als Primärschlüssel in einer eigenen Relation zusammen. Alle Attribute, die von demselben Teilschlüssel abhängen, müssen in derselben Relation zusammengefasst werden
  b) Entferne die ausgelagerten Nichtschlüsselattribute aus der Ursprungsrelation und fasse die übriggebliebenen Attribute zu einer neuen Relation zusammen

<u>PersNr</u>	Name	Rang	Raum	<u>VorlNr</u>	Titel	SWS

Zweite Normalform – Algorithmus (Skizze)

- Ein Relationenschema $\mathcal{R}$ kann wie folgt in zweite Normalform überführt werden
  a) Fasse alle Nichtschlüsselattribute, die nur von einem Teilschlüssel abhängen, mit diesem Teilschlüssel als Primärschlüssel in einer eigenen Relation zusammen. Alle Attribute, die von demselben Teilschlüssel abhängen, müssen in derselben Relation zusammengefasst werden

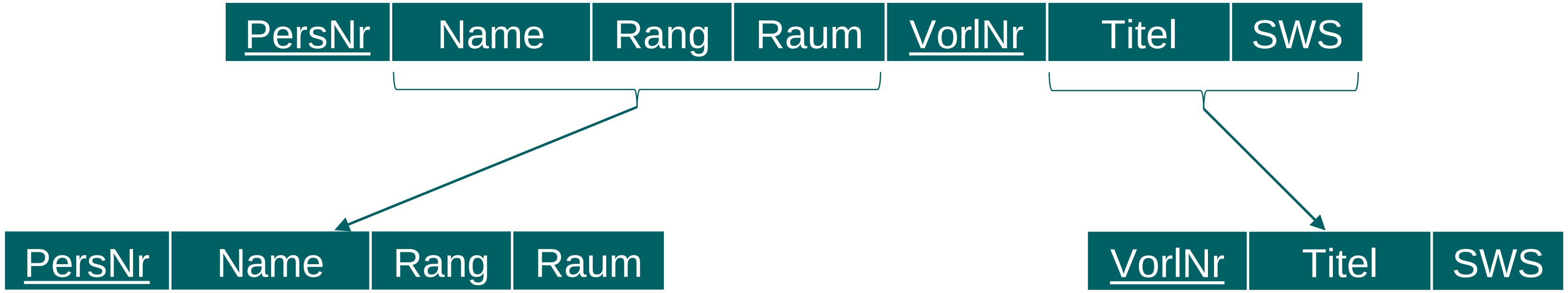
  b) Entferne die ausgelagerten Nichtschlüsselattribute aus der Ursprungsrelation und fasse die übriggebliebenen Attribute zu einer neuen Relation zusammen

<u>PersNr</u>	Name	Rang	Raum	<u>VorlNr</u>	Titel	SWS

<u>PersNr</u>	Name	Rang	Raum

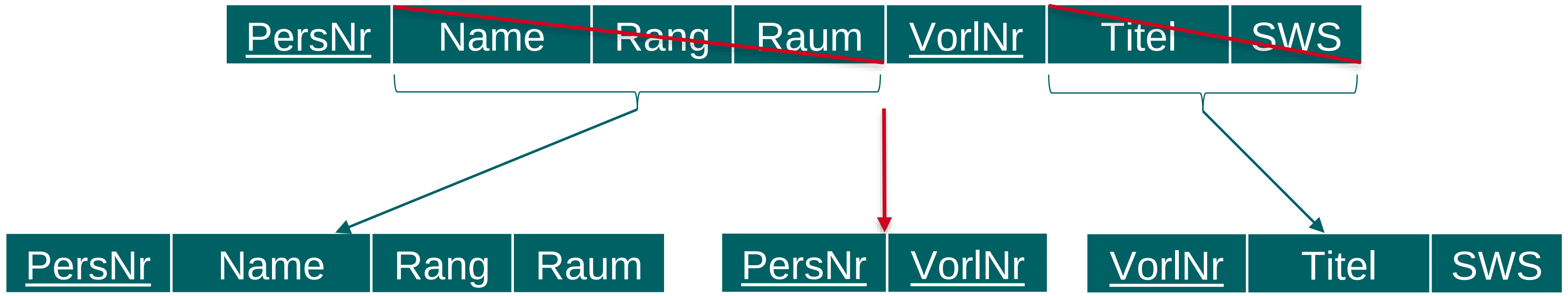
Zweite Normalform – Algorithmus (Skizze)

- Ein Relationenschema $\mathcal{R}$ kann wie folgt in zweite Normalform überführt werden
  a) Fasse alle Nichtschlüsselattribute, die nur von einem Teilschlüssel abhängen, mit diesem Teilschlüssel als Primärschlüssel in einer eigenen Relation zusammen. Alle Attribute, die von demselben Teilschlüssel abhängen, müssen in derselben Relation zusammengefasst werden
  b) Entferne die ausgelagerten Nichtschlüsselattribute aus der Ursprungsrelation und fasse die übriggebliebenen Attribute zu einer neuen Relation zusammen



Zweite Normalform – Algorithmus (Skizze)

- Ein Relationenschema $\mathcal{R}$ kann wie folgt in zweite Normalform überführt werden
  a) Fasse alle Nichtschlüsselattribute, die nur von einem Teilschlüssel abhängen, mit diesem Teilschlüssel als Primärschlüssel in einer eigenen Relation zusammen. Alle Attribute, die von demselben Teilschlüssel abhängen, müssen in derselben Relation zusammengefasst werden
  b) Entferne die ausgelagerten Nichtschlüsselattribute aus der Ursprungsrelation und fasse die übriggebliebenen Attribute zu einer neuen Relation zusammen



Zweite Normalform – Beispiel

PersNr	~~Name~~	~~Rang~~	~~Raum~~	VorlNr	~~Titel~~	~~SWS~~

PersNr	Name	Rang	Raum

PersNr	VorlNr

VorlNr	Titel	SWS

PersNr	Name	Rang	Raum	VorlNr	Titel	SWS
2125	Sokrates	W3	226	5041	Ethik	4
2125	Sokrates	W3	226	5049	Mäeutik	2
2125	Sokrates	W3	226	4052	Logik	4
…	…	…	…	…	…	…

PersNr	Name	Rang	Raum
2125	Sokrates	W3	226
…	…	…	…

PersNr	VorlNr
2125	5041
2125	5049
2125	4052
…	…

VorlNr	Titel	SWS
5041	Ethik	4
5049	Mäeutik	2
4052	Logik	4
…	…	…

Dritte Normalform – Motivation

- Beispiel: Erweiterung des Universitäts-Beispiels um Adressen

$$\mathrm{ProfessorenAdressen}(\ \mathrm{PersNr}, \mathrm{Name}, \mathrm{Rang}, \mathrm{Raum}, \mathrm{Ort}, \mathrm{Straße}, \mathrm{Hausnr}, \mathrm{PLZ}, \mathrm{Vorwahl}, \mathrm{BLand}\ )$$

ProfessorenAdressen						
PersNr	Name	…	Ort	…	PLZ	…
2125	Sokrates	…	Aachen	…	52052	…
2132	Popper	…	Aachen	…	52052	…
…	…	…	…	…	…	…

- Diese Relation ist in zweiter Normalform
- Die Information, dass die PLZ 52052 zu Aachen gehört ist hier dennoch redundant
- Ursache: funktionale Abhängigkeit $\mathrm{PLZ} \rightarrow \mathrm{Ort}$ zwischen NSAs

Dritte Normalform (Definition)

- *Dritte Normalform*:

  Ein Relationenschema $\mathcal{R}$ mit FDs F ist in *dritter Normalform*, falls für jedes Nichtschlüssel-Attribut B und jede nicht-triviale Abhängigkeit $\alpha \rightarrow B \in \mathrm{F}^+$ gilt:

  - α ist ein Superschlüssel (d.h. nicht notwendig minimal)

- Die zweite Normalform lässt auch Abhängigkeiten zwischen NSAs zu
  - In dritter Normalform dürfen NSAs im Endeffekt nur von Schlüsselkandidaten abhängen

- Beispiel:

$$\mathrm{ProfessorenAdressen}(\,\mathrm{PersNr}, \mathrm{Name}, \mathrm{Rang}, \mathrm{Raum}, \mathrm{Ort}, \mathrm{Straße}, \mathrm{Hausnr}, \mathrm{PLZ}, \mathrm{Vorwahl}, \mathrm{BLand}\,)$$

  - Sei X die Attributmenge von $\mathrm{ProfessorenAdressen}$. Es gelten die FDs:

$$\mathrm{PersNr} \rightarrow X, \qquad \mathrm{Raum} \rightarrow \mathrm{PersNr}, \qquad \mathrm{Raum} \rightarrow X,$$

$$\mathrm{Ort}, \mathrm{BLand} \rightarrow \mathrm{Vorwahl}, \qquad \mathrm{Ort}, \mathrm{BLand}, \mathrm{Straße}, \mathrm{Hausnr} \rightarrow \mathrm{PLZ}, \qquad \ldots$$

nicht erlaubt!

Dritte Normalform – Synthesealgorithmus

- Synthesealgorithmus:

  Eingabe: Relationenschema $\mathcal{R}$ mit FDs F und Attributmenge X

  Ausgabe: verlustlose, abhängigkeitserhaltende 3NF-Zerlegung $\mathcal{R}_1, \dots, \mathcal{R}_n$

1. Bestimme die kanonische Überdeckung F_c zu F

   Zur Wiederholung:

   a) Linksreduktion der FDs
   b) Rechtsreduktion der FDs
   c) Entfernung der FDs der Form $\alpha \rightarrow \emptyset$
   d) Zusammenfassung von FDs mit gleichen linken Seiten

Dritte Normalform – Synthesealgorithmus

- Synthesealgorithmus:

  Eingabe: Relationenschema $\mathcal{R}$ mit FDs F und Attributmenge X

  Ausgabe: verlustlose, abhängigkeitserhaltende 3NF-Zerlegung $\mathcal{R}_1, \dots, \mathcal{R}_n$

1. Bestimme die kanonische Überdeckung F_c zu F
2. Für jede funktionale Abhängigkeit $\alpha \to \beta \in F_c$:
   - Erstelle Relationenschema $\mathcal{R}_\alpha$ mit Attributmenge $\alpha \cup \beta$
   - Ordne $\mathcal{R}_\alpha$ die FDs $F_\alpha = \{ \alpha' \to \beta' \in F_c \mid \alpha', \beta' \subseteq \alpha \cup \beta \}$ zu
3. Falls ein in Schritt 2 erzeugtes Schema $\mathcal{R}_\alpha$ einen Schlüsselkandidaten von $\mathcal{R}$ enthält:

   fertig mit Schritt 3

   sonst: wähle einen Schlüsselkandidaten κ von $\mathcal{R}$ aus und erstelle das zusätzliche Schema $\mathcal{R}_\kappa$ mit funktionaler Abhängigkeit $F_\kappa = \{\kappa \to \kappa\}$
4. Eliminiere jedes Schema $\mathcal{R}_\alpha$ dessen Attributmenge in der eines größeren Schemas $\mathcal{R}_{\alpha'}$ enthalten ist

Dritte Normalform – Synthesealgorithmus (Beispiel)

- Beispiel:

  $$\text{ProfessorenAdressen}(\ \underline{\text{PersNr}}, \text{Name}, \text{Rang}, \text{Raum}, \text{Ort}, \text{Straße}, \text{Hausnr}, \text{PLZ}, \text{Vorwahl}, \text{BLand}\ )$$

- Funktionale Abhängigkeiten:
  - $\text{PersNr} \rightarrow \text{PersNr}, \text{Name}, \text{Rang}, \text{Raum}, \text{Ort}, \text{Straße}, \text{Hausnr}, \text{PLZ}, \text{Vorwahl}, \text{Bland}$
  - $\text{PersNr} \rightarrow \text{Raum}$
  - $\text{Raum} \rightarrow \text{PersNr}$
  - $\text{Ort}, \text{Straße}, \text{Hausnr}, \text{BLand} \rightarrow \text{PLZ}$
  - $\text{Ort}, \text{BLand} \rightarrow \text{Vorwahl}$
  - $\text{PLZ} \rightarrow \text{Ort}, \text{BLand}$

Dritte Normalform – Synthesealgorithmus (Beispiel)

- Beispiel:

$$\text{ProfessorenAdressen}(\,\underline{\text{PersNr}}, \text{Name}, \text{Rang}, \text{Raum}, \text{Ort}, \text{Straße}, \text{Hausnr}, \text{PLZ}, \text{Vorwahl}, \text{BLand}\,)$$

1. die kanonische Überdeckung F_c enthält die FDs

$f_1 = \text{PersNr} \rightarrow \text{Raum}, \text{Name}, \text{Rang}, \text{Ort}, \text{Straße}, \text{Hausnr}, \text{Bland}$

$f_2 = \text{Raum} \rightarrow \text{PersNr},$ $\quad f_3 = \text{Ort}, \text{Straße}, \text{Hausnr}, \text{BLand} \rightarrow \text{PLZ}$

$f_4 = \text{Ort}, \text{BLand} \rightarrow \text{Vorwahl},$ $\quad f_5 = \text{PLZ} \rightarrow \text{Ort}, \text{Bland}$

2. erzeugte Relationenschemata:

aus f_1: $\text{Professor}(\,\text{PersNr}, \text{Name}, \text{Rang}, \text{Raum}, \text{Ort}, \text{Straße}, \text{Hausnr}, \text{BLand})$ mit $F_1 = \{f_1, f_2\}$

aus f_2: $\mathcal{R}_2(\,\text{Raum}, \text{PersNr}\,)$ mit $F_2 = \{f_2\}$

aus f_3: $\text{PLZverzeichnis}(\,\text{Ort}, \text{Straße}, \text{Hausnr}, \text{BLand}, \text{PLZ}\,)$ mit $F_3 = \{f_3, f_5\}$

aus f_4: $\text{Vorwahlverzeichnis}(\,\text{Ort}, \text{BLand}, \text{Vorwahl}\,)$ mit $F_4 = \{f_4\}$

aus f_5: $\mathcal{R}_5(\,\text{PLZ}, \text{Ort}, \text{BLand}\,)$ mit $F_5 = \{f_5\}$

Dritte Normalform – Synthesealgorithmus (Beispiel)

- Beispiel:

$$\mathrm{ProfessorenAdressen}(\,\underline{\mathrm{PersNr}}, \mathrm{Name}, \mathrm{Rang}, \mathrm{Raum}, \mathrm{Ort}, \mathrm{Straße}, \mathrm{Hausnr}, \mathrm{PLZ}, \mathrm{Vorwahl}, \mathrm{BLand}\,)$$

2. erzeugte Relationenschemata:

aus f_1: $\mathrm{Professor}(\,\mathrm{PersNr}, \mathrm{Name}, \mathrm{Rang}, \mathrm{Raum}, \mathrm{Ort}, \mathrm{Straße}, \mathrm{Hausnr}, \mathrm{BLand})$ mit $F_1 = \{f_1, f_2\}$

aus f_2: $\mathcal{R}_2(\,\mathrm{Raum}, \mathrm{PersNr}\,)$ mit $F_2 = \{f_2\}$

aus f_3: $\mathrm{PLZverzeichnis}(\mathrm{Ort}, \mathrm{Straße}, \mathrm{Hausnr}, \mathrm{BLand}, \mathrm{PLZ})$ mit $F_3 = \{f_3, f_5\}$

aus f_4: $\mathrm{Vorwahlverzeichnis}(\,\mathrm{Ort}, \mathrm{BLand}, \mathrm{Vorwahl}\,)$ mit $F_4 = \{f_4\}$

aus f_5: $\mathcal{R}_5(\,\mathrm{PLZ}, \mathrm{Ort}, \mathrm{BLand}\,)$ mit $F_5 = \{f_5\}$

3. PersNr ist ein Schlüsselkandidat des gesamten Schemas $\mathrm{ProfessorenAdressen}$

   $\Rightarrow$ Es muss keine neue Relation erzeugt werden

4. $\mathcal{R}_2$ und $\mathcal{R}_5$ werden entfernt, da sie in $\mathrm{Professor}$ bzw. $\mathrm{PLZverzeichnis}$ enthalten sind

Boyce-Codd Normalform

- Ziel: Jede Information wird <u>genau einmal</u> gespeichert
- *BCNF*:

  Ein Relationsschema $\mathcal{R}$ mit FDs F ist genau dann in *Boyce-Codd Normalform*, wenn für alle nicht-trivialen FDs $\alpha \rightarrow \beta \in F^+$ gilt:

  - α ist ein Superschlüssel von $\mathcal{R}$

- Beispiel: In dritter Normalform aber nicht in BCNF

$$\text{Städte}(\ \text{Ort}, \text{BLand}, \text{MinisterpräsidentIn}, \text{EW}\ )$$

$$f_3 \qquad f_2 \qquad f_1$$

  – zwei Schlüsselkandidaten

$$\kappa_1 = \{\ \text{Ort}, \text{BLand}\ \}, \qquad \kappa_2 = \{\ \text{Ort}, \text{MinisterpräsidentIn}\}$$

  – ist in 3NF: einziges NSA ist EW. EW kommt nur in $\text{f}_1 = \kappa_1 \rightarrow \text{EW}$ vor

  – nicht in BCNF: $f_3 = \text{Ministerpräsidentin} \rightarrow \text{BLand}$, die linke Seite ist kein Schlüssel

BCNF – Dekompositionsalgorithmus

- Achtung: Die resultierende Zerlegung ist i.A. nicht abhängigkeitserhaltend!

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

  Solange es noch ein Schema $\mathcal{R}_i \in Z$ gibt, das nicht in BCNF ist:

  - finde nicht-triviale, in R_i geltende FD $\alpha \to \beta$ mit
    - $\alpha \cap \beta = \emptyset$ und $\alpha \nrightarrow X_i$ (X_i ist Attributmenge von $\mathcal{R}_i$)
  - Zerlege $\mathcal{R}_i$ in $\mathcal{R}_{i1} = (X_{i1}, F_{i1})$, $\mathcal{R}_{i2} = (X_{i2}, F_{i2})$ *entlang* der FD $\alpha \to \beta$

    mit Attributmengen:

    $X_{i1} = \alpha \cup \beta$ und $X_{i2} = X_i \setminus \beta$

    und FDs:

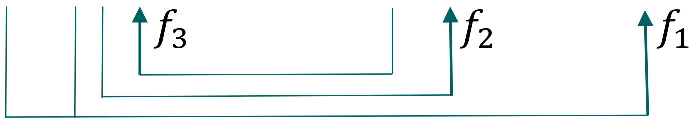
    $F_{i1} = \Pi_{Xi1}(F_i^+)$ und $F_{i2} = \Pi_{Xi2}(F_i^+)$
  - Entferne $\mathcal{R}_i$ aus Z und füge $\mathcal{R}_{i1}, \mathcal{R}_{i2}$ hinzu.

Definition:

$$\Pi_X(F) \coloneqq \{ \alpha \to \beta \in F \mid (\alpha \cup \beta) \subseteq X \}$$

Dekompositionsalgorithmus Beispiel 1

- $\text{Städte}(\ \text{Ort}, \text{BLand}, \text{MinisterpräsidentIn}, \text{EW}\ )$



- Zerlegung *entlang* von $f_3 = \text{MinisterpräsidentIn} \rightarrow \text{Bland}$
  - $\text{Städte1}(\ \text{Ort}, \text{MinisterpräsidentIn}, \text{EW}\ )$ mit FDs:
    - $f_1' = \text{MinisterpräsidentIn}, \text{Ort} \rightarrow \text{EW}$

      wegen f_1' in F^+ **(wichtig!)**
  - $\text{Regierungen}(\ \text{BLand}, \text{MinisterpräsidentIn}\ )$ mit FDs:
    - $f_2 = \text{BLand} \rightarrow \text{MinisterpräsidentIn}$
    - $f_3 = \text{Ministerpräsidentin} \rightarrow \text{BLand}$

in diesem Beispiel ist die Zerlegung abhängigkeitserhaltend, da sich f_1 rekonstruieren lässt.

- **Bitte beachten:** BLand wurde durch MinisterpräsidentIn in f_1' ersetzt!
- Die Relationen Städte1 und Regierungen sind nun in BNCF.

Dekompositionsalgorithmus Beispiel 2

- Beispiel für nicht abhängigkeitserhaltende Zerlegung

$$\mathrm{PLZVerzeichnis}(\ \mathrm{Straße}, \mathrm{Ort}, \mathrm{BLand}, \mathrm{PLZ}\ )$$

  mit funktionalen Abhängigkeiten

$$f_1 = \mathrm{Straße}, \mathrm{Ort}, \mathrm{BLand} \rightarrow \mathrm{PLZ} f_2 = \mathrm{PLZ} \rightarrow \mathrm{Ort}, \mathrm{Bland}$$

  Die Zerlegung von $\mathrm{PLZVerzeichnis}$ entlang von f_2 ergibt

  - $\mathrm{Straßen}(\ \mathrm{Straße}, \mathrm{PLZ}\ )$ mit FDs: $\emptyset$
  - $\mathrm{Orte}(\ \mathrm{Ort}, \mathrm{BLand}, \mathrm{PLZ}\ )$ mit FDs: $f_2 = \mathrm{PLZ} \rightarrow \mathrm{Ort}, \mathrm{Bland}$

  Wie schon diskutiert, geht in diesem Fall die Abhängigkeit f_1 verloren

- Wäre eine Zerlegung nicht abhängigkeitserhaltend, gibt man sich normalerweise mit der dritten Normalform zufrieden

Zusammenfassung

- Ziel der relationalen Entwurfstheorie
  – Was ist ein redundanzfreies relationales Datenbank-Schema?
- Zerlegung von Relationenschemata
  - Verlustlos
  - Abhängigkeitserhaltend
- Normalformen
  – Algorithmen
    - Synthesealgorithmus (3NF)
    - Dekompositionsalgorithmus (BCNF)
  – Abhängigkeitserhaltung ist nur bis zur dritten Normalform garantiert