



Übung Datenbanksysteme Normalformen

19.1.2004

Final-Klausur

- Termin: Freitag, 13. Februar 2004
- Uhrzeit: ab 16.15 Uhr
- Ort: Hörsaal PH HS 1 (PH 2501)

Funktionale Abhängigkeiten

- Ab jetzt wieder recht theoretisch
- Teil der Normalformtheorie (mehr dazu im nächsten Übungsblatt)
- Formal:
 $\alpha \rightarrow \beta$ α, β Mengen von Spalten
- Anschaulich:
Wenn die Werte von α gleich sind,
dann müssen auch die Werte von β gleich sein
- Wie findet man fkt. Abhängigkeiten:
 - Verständnis der Daten
 - systematisches Suchen
- Wichtig:
- Funktionale Abhängigkeiten sind nicht abhängig von den konkret vorhandenen Tupeln

Beispiel für funktionale Abhängigkeiten

Nach gleichen Werten suchen !

A	B	C	D
A1	B1	C1	D1
A1	B1	C2	D2
A2	B2	C4	D3
A3	B2	C4	D3

Im Beispiel:

$A \rightarrow B$

$BC \rightarrow D$

$BD \rightarrow C$

...

Normalformen

- Normalformen bestimmen die „Qualität“ eines Schemas im Bezug auf
 - Redundanz
 - Update-Anomalien
- Bestimmung auf Grundlage von funktionalen Abhängigkeiten
- Normalformen bauen aufeinander auf
- Wir betrachten 1.-3. NF, BCNF
- Relationen müssen zerlegt werden, um zu höheren Normalformen zu kommen
- Wir verwenden einige Algorithmen, die sicher zum Ergebnis führen, allerdings etwas aufwendig sind
- Geht meist auch „mit Hand“

Definition 1. bis 3. NF

- Definition 1 NF
„Alle Attribute haben atomare Werte“
- Im relationalen Modell immer erfüllt
- Definition 2 NF:
 - 1 NF
 - Nicht-Schlüsselattribute sind **voll funktional abhängig** von allen Kandidatenschlüsseln
- Voll funktional abhängig:
Funktionale Abhängigkeit kann nicht weiter verkleinert werden (durch Herausnahme von Attr.)

Definition 3. Normalform

- R ist in 3NF, wenn
 - R in 2 NF
 - Keine transitiven fkt. Abhängigkeiten in R

Armstrong-Axiome

- Wie berechnet man Zusammenhänge der funktionalen Abh. ?
- U : Menge aller Attribute
- $X, Y, Z \subseteq U$
- 1. Reflexivität:**
 - $Y \subseteq X: X \rightarrow Y$
- 2. Erweiterung:**
 - $X \rightarrow Y: XZ \rightarrow YZ$
- 3. Transitivität:**
 - $X \rightarrow Y, Y \rightarrow Z: X \rightarrow Z$
- Hilfsregeln**
- Abgeleitet von 1. - 3.
- 4. Vereinigung:**
 - $X \rightarrow Y, X \rightarrow Z: X \rightarrow YZ$
- 5. Pseudotransitivität:**
 - $X \rightarrow Y, YW \rightarrow Z: XW \rightarrow Z$
- 6. Zerlegung:**
 - $X \rightarrow Y, Z \subseteq Y: X \rightarrow Z$

Erzeugung der Attributhülle (1)

- Was ist Attributhülle ?
- Input:
 - Menge Attribute X
 - Funktionale Abhängigkeiten F
- Output:
 - Alle von den gegebenen Attributen funktional abhängigen Attribute
- Formal: $X \rightarrow X^+$
- Wozu notwendig ?
 - “komplette” Abhängigkeiten wichtig für Schlüssel, Berechnung höherer NF
 - Einfacher als Anwendung der Armstrong-Axiome

Erzeugung der Attributhülle (2)

- In Schreibweise der Vorlesung
- Kemper/Eickler ist identisch, aber anders formuliert

$X_0 := X$

Beginn mit Startattributmeng

$X_{i+1} := X_i \cup$

**$\{A \mid \exists Y \rightarrow Z \in F:$
 $Y \subseteq X_i \wedge A \in Z\}$**

Attribute dazu, die aus den bisherigen
mit fkt. Abhängigkeiten bestimmt werden können

bis $X_{i+1} = X_i =: X^+$

Solange neue Attribute dazukommen (Fixpunkt)

Attributmeng X danach = Ergebnis X^+

Aufgabe 1: Berechnung Attributhülle

- Relation (A,B,C,D,E,G)
 - Funktionale Abhängigkeiten F:
 - $AB \rightarrow C$
 - $D \rightarrow EG$
 - $C \rightarrow A$
 - $BE \rightarrow C$
 - $BC \rightarrow D$
 - $CG \rightarrow BD$
 - $ACD \rightarrow B$
 - $CE \rightarrow AG$
 - Berechnung AttrH (F,BD)
 - $X = \emptyset$
 - $X = BD$ Startregel
 - $X = BDEG$ $D \rightarrow EG$
 - $X = BCDEG$ $BE \rightarrow C$
 - $X = ABCDEG$ $CE \rightarrow AG$
(oder auch
 $C \rightarrow A$)
- Keine Änderungen mehr
Ergebnis ABCDEG
Bei jedem Schritt Test bei allen FD

Aufgabe 1 (2)

- Wir haben gezeigt, daß BD für die gegebenen funkt. Abhängigkeiten ABCDEG bestimmt.
- Damit ist BD ein sog. Superschlüssel.
- Ein Superschlüssel bestimmt alle Attribute der Relation. Er ist aber nicht notwendigerweise minimal.

Kanonische Überdeckung

- Ziel: Vereinfachung der funktionalen Abhängigkeiten
- Formale Def im Kemper/Eickler-Buch
- Eigenschaften:
 - Attributhülle kan. Überdeckung gleich Attributhülle Fd
 - Keine überflüssigen Attribute
 - Linke Seiten einzigartig

Algorithmus für kanon. Überdeckung

1. 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 $\beta \subseteq \text{AttrHülle}(F, \alpha - A)$ gilt. Falls dies der Fall ist, ersetze $\alpha \rightarrow \beta$ durch $(\alpha - A) \rightarrow \beta$.
2. Führe für jede (verbliebene) FD die Rechtsreduktion durch, also:
 - Überprüfe für alle $B \in \beta$, ob
 - $B \in \text{AttrHülle}(F - (\alpha \rightarrow \beta) \cup (\alpha \rightarrow (\beta - B)), \alpha)$ gilt. Falls dies der Fall ist, ist B auf der rechten Seite überflüssig und kann eliminiert werden, d.h. ersetze $\alpha \rightarrow \beta$ durch $\alpha \rightarrow (\beta - B)$.
3. Entferne die FDs der Form $\alpha \rightarrow \emptyset$, die im 2. Schritt möglicherweise entstanden sind.
4. Fasse mittels der Vereinigungsregel FDs der Form $\alpha \rightarrow \beta_1, \dots, \alpha \rightarrow \beta_n$ zusammen, so dass $\alpha \rightarrow (\beta_1 \cup \dots \cup \beta_n)$ verbleibt.

Aufgabe 2a) (1)

- R1 (A,B,C,D,E);
R2 (A,C,F)
- $A \rightarrow B, E$
- $A \rightarrow D$
- $F \rightarrow A$
- $A, C \rightarrow F$
- $B, C \rightarrow E$
- $C \rightarrow A$
- Schlüsselkandidaten bestimmen:
- Entweder per Hand oder per kanonischer Überdeckung
- „Per Hand“
- R2: C, da $C \rightarrow A, AC \rightarrow F$)
- R1: C, da $C \rightarrow A, A \rightarrow BE, A \rightarrow D$

Aufgabe 2a) – kanonische Überdeckung (1)

1. Linksreduktion:

- $A, C \rightarrow F$:
 $F \subseteq \text{AttrHülle}(\text{FD}, C) = (C), (A, C), (A, C, F) ?$
 $\Rightarrow F$ enthalten, damit A eliminieren
 $\Rightarrow C \rightarrow F$ neue fkt. Abh.
- $B, C \rightarrow E$
 $E \subseteq \text{AttrHülle}(\text{FD}, C) = (C), (A, C), (ABCE) ?$
 $\Rightarrow C$ enthalten, damit B eliminieren
 $\Rightarrow C \rightarrow E$ neue fkt. Abh.

Aufgabe 2a) – kanonische Überdeckung (2)

- 2. Rechtsreduktion
- $C \rightarrow E$
 $E \in \text{AttrHülle}(\text{FD} \setminus \{C \rightarrow E\} \cup \{C \rightarrow \emptyset\}, C) =$
 $(C), (A,C), (ABCE) ?$
 $\Rightarrow E$ enthalten, damit E eliminieren
 $\Rightarrow C \rightarrow \emptyset$ neue fkt. Abh.
- $C \rightarrow A$
 $A \in \text{AttrHülle}(\text{FD} \setminus \{C \rightarrow A\} \cup \{C \rightarrow \emptyset\}, C)$
 $= (C), (CF), (ACF) ?$
 $\Rightarrow A$ enthalten, damit A eliminieren
 $\Rightarrow C \rightarrow \emptyset$ neue fkt. Abh.
- Andere Eliminationsversuche führen zu keinem Ergebnis

Aufgabe 2a) – kanonische Überdeckung (3)

- 3. Elimination von $X \rightarrow \emptyset$
- $C \rightarrow \emptyset$ entfernen
- 4. Vereinigung gleicher linker Seiten:
- $A \rightarrow BE, A \rightarrow D \Rightarrow$
 $A \rightarrow B, D, E$
- $C \rightarrow F$ (unverändert)
- $F \rightarrow A$ (unverändert)
- Schlüssel wiederum darauf bestimmen

Afg2a) Welche NF haben die Schemata?

- R1(A,B,C,D,E):
 - 1 NF: (klar)
 - 2 NF: ja, da ABDE voll funktional abhängig von C
 - 3 NF: nein, da BDE transitiv abhängig von C
- R2 (A,C,F):
 - 1 NF: (klar)
 - 2 NF: ja, da AF voll funktional abhängig von C
 - 3 NF: nein, da A transitiv abhängig von C (nach kan. Überdeckung) bzw. F transitiv abhängig von C (vor kan. Überdeckung)

Algorithmus zur Zerlegung einer Relation in 3NF

1. Bestimme die kanonische Überdeckung F_c zu F .
2. Für jede funktionale Abhängigkeit $\alpha \rightarrow \beta \in F_c$:
 - Kreiere ein Relationenschema $R_\alpha := \alpha \cup \beta$
 - Ordne R_α die FDs $F_\alpha := \{\alpha' \rightarrow \beta' \in F_c \mid \alpha' \cup \beta' \subseteq R_\alpha\}$ zu.
3. Falls eines der in Schritt 2. erzeugten Schemata einen Kandidatenschlüssel von R bzgl. F_c enthält, sind wir fertig. Sonst wähle einen Kandidatenschlüssel $\kappa \subseteq R$ aus und definiere folgendes Schema:
 - $R_\kappa := \kappa$
 - $F_\kappa := \emptyset$
4. Eliminiere diejenigen Schemata R_α , die in einem anderen Relationenschema $R_{\alpha'}$ enthalten sind, d.h.,
 - $R_\alpha \subseteq R_{\alpha'}$

Aufgabe 2a) (3)

Umformung in 3NF

- Kanon. Überdeckung (schon erledigt)
- Relationen erstellen
- R2: (FA), (CF)
- R1: (ABDE)
- Schlüssel ergänzen – nicht notwendig
- Relationen eliminieren – nicht notwendig
- Wo ist das C von R1 ?
 - kam von $C \rightarrow A$
 - In R2, da $C \rightarrow F, F \rightarrow A$

Aufgabe 2b)

S1 (A,B,C,D)

FD: A,C \rightarrow D A \rightarrow B

- Schlüsselkandidaten: (AC)
- 2 NF: nein, da B nicht voll funktional abhängig von AC (nur von A !)

=> Damit auch nicht 3 NF

- Händische Zerlegung:
(ACD), (AB)

Zur Kontrolle Kanonische Überdeckung:

- LR:
 - AC \rightarrow D keine Änderung
- RR:
 - AC \rightarrow D keine Änderung
 - A \rightarrow B keine Änderung
- Elimination und Zusammenfassung keine Änderung
=> Lösung ist OK

Aufgabe 2c)

- S2 (A,B,C)
- $A, B \rightarrow C$ $C \rightarrow A$
- Schlüsselkandidaten
 1. AB (klar)
 2. BC (da, $C \rightarrow A \Rightarrow BC \rightarrow AB \Rightarrow BC \rightarrow A$)
- 2 NF
 - 1. SK: ja, da C voll fkt. abh. von AB
 - 2. SK: nein, da A nicht voll fkt. abh. von BC (nur C)
- 3 NF
 - 1. SK: ja, da keine transitive Abh
 - Zum 2.SK kommen wir noch in der BCNF !

Aufgabe 3

- Auftrag (ProduktNr, ProduktName, KundenNr, KundenName, Datum, Stückpreis, Anzahl, Nettopreis, Mehrwertsteuersatz, Bruttopreis)

Letztes Übungsblatt:

- FD:
 - {Stückpreis, Anzahl} → {Nettopreis}
 - {Nettopreis, Mehrwertsteuersatz} → {Bruttopreis}
 - {KundenNr} → {KundenName}
 - {KundenName} → {KundenNr}
 - {ProduktName} → {ProduktNr}
 - {ProduktNr} → {ProduktName, Stückpreis, Mehrwertsteuersatz}
 - {ProduktNr, KundenNr, Datum} → {Anzahl}
- SK:
 1. {ProduktNr, KundenNr, Datum}
 2. {ProduktName, KundenName, Datum}

Aufgabe 3:

- Normalform der Relation
- 2 NF: nein, da bei 1. SK z.B. Kundenname nicht voll funktional abhängig (nur von KundenNR)
- Weitere NF: nein
- Kanonische Überdeckung:
=> Erhält alle fkt. Abhängigkeiten

Synthesealgorithmus

1. Zerlegung nach kan. Überdeckung:
 - Produkt: (ProduktNr, ProduktName, Stückpreis, Mehrwertsteuersatz)
{ProduktName} → {ProduktNr} enthalten (oder bei 4. eliminieren)
 - Kunden: (KundenNr, KundenName)
{KundenName} → {KundenNr} enthalten (oder bei 4. eliminieren)
 - OrderItem: (ProduktNr, KundenNr, Datum, Anzahl)
 - Nettopreis: (Stückpreis, Anzahl, Nettopreis)
 - Bruttopreis: (Nettopreis, Mehrwertsteuersatz, Bruttopreis)
2. OrderItem enthält kompletten Schlüssel
=> keine weitere Tabelle nötig
3. Tabellen disjunkt
=> keine Elimination von Tabellen
- Vergabe der Schlüssel in den Tabellen ähnlich am Anfang

Definition BCNF

- R in BCNF, wenn
 - jeder Determinator SK ist
- Determinator:
Attributmengende A_1 - A_n zu Attribut A_x ,
so dass
 A_x voll funktional abhängig von $A_1..A_n$

Aufgabe 2 - BCNF

a)

- R1 (ABDE),
 - Determinator B,D,E: A ist Schlüssel
- R2: (CF), (FA)
 - Determinator F: C ist Schlüssel
 - Determinator A: F ist Schlüssel

b) (ACD), (AB)

- Determinator von D: AC ist Schlüssel von ACD
- Determinator von B: A ist Schlüssel von AB

=> ACD und AB sind in BCNF

c) (ABC), $C \rightarrow A$

- Determinator von A ist C: nicht Schlüsselkandidat

=> Nicht in BCNF

Erstellung von BCNF

- Solange Determinator d in R vorhanden, der nicht Schlüsselkandidat ist
- Erstelle neue Relation (d , Attribute def. von d)
- Entferne (Attribute def. von d) aus R

- Aufgabe 2c)
- (ABC), $\text{Det}(A) = C$
- (CA), (BC) in BCNF (und damit auch 3NF)
- Fkt. Abh. $AB \rightarrow C$ nicht mehr direkt enthalten !
=> Problem bei BCNF !