

Aufgabe 5 (Check-Up)

Ein Supermarkt speichert seine Bestellungen in nachfolgender Tabelle:

```
{ ARTBEST[ ArtNr; ArtName; ArtArt; Hersteller; HerstTel; Lieferant; LiefTel;  
BestDat; Anzahl; EP ] }
```

Es existieren folgende funktionale Abhängigkeiten:

$$F = \left\{ \begin{array}{l} \{ \text{ArtNr} \} \rightarrow \{ \text{ArtName}, \text{ArtArt}, \text{Hersteller}, \text{HerstTel} \}, \\ \{ \text{Hersteller} \} \rightarrow \{ \text{HerstTel} \}, \\ \{ \text{Lieferant} \} \rightarrow \{ \text{LiefTel} \}, \\ \{ \text{ArtNr}, \text{Lieferant}, \text{BestDat} \} \rightarrow \{ \text{Anzahl} \}, \end{array} \right\}$$

EP „Bestdat“ steht für Bestelldatum, „EP“ für Einkaufspreis

- (a) Erläutern Sie, warum nur Relationen mit einem zusammengesetzten Schlüsselkandidaten die 2. Normalform verletzen können!

Eine Relation ist genau dann in der zweiten Normalform, wenn kein Nichtprimärattribut funktional von einer echten Teilmenge eines Schlüsselkandidaten abhängt.

Anders gesagt: Jedes nicht-primäre Attribut ist jeweils von allen ganzen Schlüsseln abhängig, nicht nur von einem Teil eines Schlüssels.

Bei nicht zusammengesetzten Schlüsselkandidaten, d. h. Schlüsselkandidaten mit nur einem Attribut, können Nichtprimärattribute nur von diesem einen Schlüsselkandidaten abhängen, sonst wäre es ja kein Schlüsselkandidat / Primärschlüssel.

- (b) Finden Sie den einzigen Schlüsselkandidaten von ARTBEST.

Ich wähle ArtNr, Lieferant, BestDat aus, da diese Attribute auf keiner rechten Seite einer FD vorkommen. Außerdem wähle ich EP aus, da EP in keiner FD vorkommt.

$\text{AttrHülle}(F, \{ \text{ArtNr}, \text{Lieferant}, \text{BestDat}, \text{EP} \}) =$
 $\{ \text{ArtNr}, \text{Lieferant}, \text{BestDat}, \text{EP}, \text{ArtName}, \text{ArtArt}, \text{Hersteller},$
 $\text{HerstTel}, \text{LiefTel}, \text{Anzahl} \} = R$

Damit ist gezeigt, dass ArtNr, Lieferant, BestDat, EP ein Superschlüssel ist.

Ich teste mit Hilfe der Attributhülle, ob man den Superschlüssel noch weiter verkleinern kann.

ohne ArtNr

$\text{AttrHülle}(F, \{ \text{Lieferant}, \text{BestDat}, \text{EP} \}) =$
 $\{ \text{Lieferant}, \text{BestDat}, \text{EP}, \text{LiefTel} \} \neq R$

ohne Lieferant

```
AttrHülle(F, { ArtNr, BestDat, EP }) =  
{ ArtNr, BestDat, EP, ArtName, ArtArt, Hersteller, HerstTel } ≠  
R
```

ohne BestDat

```
AttrHülle(F, { ArtNr, Lieferant, EP }) =  
{ ArtNr, Lieferant, EP, ArtName, ArtArt, Hersteller, HerstTel, LiefTel  
} ≠ R
```

ohne EP

```
AttrHülle(F, { ArtNr, Lieferant, BestDat }) =  
{ ArtNr, Lieferant, BestDat, ArtName, ArtArt, Hersteller,  
HerstTel, LiefTel, Anzahl } ≠ R
```

Der Superschlüssel ArtNr, Lieferant, BestDat, EP kann nicht mehr weiter verkleinert werden. Er ist bereits minimal. ArtNr, Lieferant, BestDat, EP ist der einzige Schlüsselkandidat und damit der Primärschlüssel.

- (c) Erläutern Sie, inwiefern obiges Schema die 3. Normalform verletzt! Zeigen Sie anhand obiger Relation ARTBEST zwei mögliche Anomalien auf, die bei fehlender Normalisierung auftreten können.

In der dritten Normalform darf kein Nichtschlüsselattribut von einem Schlüsselkandidaten transitiv abhängen. In der Relation ARTBEST hängt HerstTel funktional von Hersteller und Hersteller hängt wiederum funktional von dem Primärschlüssel / Schlüsselkandidaten ArtNr, Lieferant, BestDat, EP ab.

$\text{ArtNr, Lieferant, BestDat, EP} \rightarrow \text{Hersteller} \rightarrow \text{HerstTel}$

Update-Anomalie

Es kann zur Update-Anomalie kommen. Ändert sich zum Beispiel die Telefonnummer eines Herstellers, so müssen in allen Datensätzen die Telefonnummer geändert werden.

Delete-Anomalie

Wird die Datenbank aufgeräumt, d. h. alte Bestellungen gelöscht, so verschwindet auch die Hersteller-Telefonnummer von manchen Herstellern.

- (d) Überführen Sie das obige Relationenschema schrittweise in die 3. Nor-

malform! Erläutern Sie die dazu durchzuführenden Schritte jeweils kurz!

(i) **Kanonische Überdeckung**

i. **Linksreduktion**

— Führe für jede funktionale Abhängigkeit $\alpha \rightarrow \beta \in F$ die Linksreduktion durch, überprüfe also für alle $A \in \alpha$, ob A überflüssig ist, d. h. ob $\beta \subseteq \text{AttrHülle}(F, \alpha - A)$. —

Die einzige FD mit einer Determinante bestehend aus mehreren Attributen, ist $\{ \text{ArtNr}, \text{Lieferant}, \text{BestDat} \} \rightarrow \{ \text{Anzahl} \}$

- ohne ArtNr

$\text{Anzahl} \notin \text{AttrHülle}(F, \{ \text{Lieferant}, \text{BestDat} \}) = \{ \text{Lieferant}, \text{BestDat}, \text{LiefTel} \}$

- ohne Lieferant

$\text{Anzahl} \notin \text{AttrHülle}(F, \{ \text{ArtNr}, \text{BestDat} \}) = \{ \text{ArtNr}, \text{BestDat}, \text{ArtName}, \text{ArtArt}, \text{Hersteller}, \text{HerstTel} \}$

- ohne BestDat

$\text{Anzahl} \notin \text{AttrHülle}(F, \{ \text{ArtNr}, \text{Lieferant} \}) = \{ \text{ArtNr}, \text{Lieferant}, \text{ArtName}, \text{ArtArt}, \text{Hersteller}, \text{HerstTel}, \text{LiefTel} \}$

Die linke Seiten der FDs können nicht reduziert werden.

ii. **Rechtsreduktion**

— Führe für jede (verbliebene) funktionale Abhängigkeit $\alpha \rightarrow \beta$ die Rechtsreduktion durch, überprüfe also für alle $B \in \beta$, ob $B \in \text{AttrHülle}(F - (\alpha \rightarrow \beta) \cup (\alpha \rightarrow (\beta - B)), \alpha)$ gilt. In diesem Fall ist B auf der rechten Seite überflüssig und kann eliminiert werden, d. h. $\alpha \rightarrow \beta$ wird durch $\alpha \rightarrow (\beta - B)$ ersetzt. —

Das einzige Attribut, dass auf der rechten Seite der FDs doppelt vorkommt ist **HerstTel**

$\text{HerstTel} \in \text{AttrHülle}(F - \{ \text{ArtNr} \} \rightarrow \{ \text{HerstTel} \}, \{ \text{ArtNr} \})$
 $=$
 $\{ \text{ArtNr}, \text{ArtName}, \text{ArtArt}, \text{Hersteller}, \text{HerstTel} \}$

iii. **Löschen leerer Klauseln**

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

Nichts zu tun

iv. **Vereinigung**

— Fasse mittels der Vereinigungsregel funktionale Abhängigkeiten der Form $\alpha \rightarrow \beta_1, \dots, \alpha \rightarrow \beta_n$, so dass $\alpha \rightarrow \beta_1 \cup \dots \cup \beta_n$ verbleibt. —

Nichts zu tun

Kanonische Überdeckung:

$F_c = \{$
 $\{ \text{ArtNr} \} \rightarrow \{ \text{ArtName}, \text{ArtArt}, \text{Hersteller} \},$
 $\{ \text{Hersteller} \} \rightarrow \{ \text{HerstTel} \},$
 $\{ \text{Lieferant} \} \rightarrow \{ \text{LiefTel} \},$
 $\{ \text{ArtNr}, \text{Lieferant}, \text{BestDat} \} \rightarrow \{ \text{Anzahl} \},$
 $\}$

(ii) **Neues Relationenschema**

— Erzeuge für jede funktionale Abhängigkeit $\alpha \rightarrow \beta \in F_c$ ein Relationenschema $\mathcal{R}_\alpha := \alpha \cup \beta$. —

R1(ArtNr, ArtName, ArtArt, Hersteller)

R2(Hersteller, HerstTel)

R3(Lieferant, LiefTel)

R4(ArtNr, Lieferant, BestDat, Anzahl)

(iii) **Hinzufügen einer Relation**

— Falls eines der in Schritt 2. erzeugten Schemata \mathcal{R}_α einen Schlüsselkandidaten von \mathcal{R} bezüglich F_c enthält, sind wir fertig, sonst wähle einen Schlüsselkandidaten $\mathcal{K} \subseteq \mathcal{R}$ aus und definiere folgendes zusätzliche Schema: $\mathcal{R}_\mathcal{K} := \mathcal{K}$ und $\mathcal{F}_\mathcal{K} := \emptyset$ —

Es muss noch eine Relation hinzugefügt werden, nämlich kommt das Attribut EP bisher in keiner Relation vor.

R1(ArtNr, ArtName, ArtArt, Hersteller)

R2(Hersteller, HerstTel)

R3(Lieferant, LiefTel)

R4(ArtNr, Lieferant, BestDat, Anzahl)

R5(ArtNr, Lieferant, BestDat, EP)

(iv) **Entfernung überflüssiger Teilschemata**

— Eliminiere diejenigen Schemata \mathcal{R}_α , die in einem anderen Relationenschema $\mathcal{R}_{\alpha'}$ enthalten sind, d. h. $\mathcal{R}_\alpha \subseteq \mathcal{R}_{\alpha'}$. —

Nicht zu tun.

Ergebnis:

R1(ArtNr, ArtName, ArtArt, Hersteller)

R2(Hersteller, HerstTel)

R3(Lieferant, LiefTel)

R4(ArtNr, Lieferant, BestDat, Anzahl)

R5(ArtNr, Lieferant, BestDat, EP)

- (e) Erläutern Sie, inwiefern sich eine vollständige Normalisierung nachteilig auf die Geschwindigkeit der Anfragebearbeitung auswirken kann und wie darauf reagiert werden kann!

Eine vollständige Normalisierung hat den Effekt, dass die Daten auf mehr Relation bzw. Tabellen aufgeteilt werden. In der Regel geht damit einher, dass bei Abfragen mehr Joins durchgeführt werden müssen, was in der Regel mit mehr Speicherbedarf und Rechenzeit der Anfragen einhergeht.

Man könnte auf eine Normalisierung verzichten, oder nur teilweise normalisieren und somit zwischen Performance und Redundanz abwägen.