

Einzelprüfung „Datenbanksysteme / Softwaretechnologie (vertieft)“

Einzelprüfungsnummer 66116 / 2019 / Frühjahr

## Thema 2 / Teilaufgabe 1 / Aufgabe 6

(Normalisierung)

**Stichwörter:** Normalformen, Schlüsselkandidat, Synthese-Algorithmus

Gegeben sei das Relationsschema  $R(A, B, C, D, E, F)$ , sowie die Menge der zugehörigen funktionalen Abhängigkeiten:

$$FA = \left\{ \begin{array}{l} \{B\} \rightarrow \{F\}, \\ \{C, D\} \rightarrow \{E\}, \\ \{C\} \rightarrow \{A\}, \\ \{C, D\} \rightarrow \{A\}, \\ \{D\} \rightarrow \{F\}, \\ \{D\} \rightarrow \{B\}, \end{array} \right\}$$

- (a) Bestimmen Sie den Schlüsselkandidaten der Relation  $R$  und begründen Sie, warum es keine weiteren Schlüsselkandidaten gibt.

Lösungsvorschlag

Der Schlüsselkandidat ist  $\{C, D\}$ , da  $\{C, D\}$  auf keiner rechten Seiten der funktionalen Abhängigkeiten vorkommt. Außerdem ist  $\{C, D\}$  ein Superschlüssel da gilt:  $\text{AttrHülle}(F, \{C, E\}) = \{A, B, C, D, E, G\} = R$

- (b) Überführen Sie das Relationsschema  $R$  mit Hilfe des Synthesealgorithmus in die dritte Normalform. Führen Sie hierfür jeden der vier Schritte durch und kennzeichnen Sie Stellen, bei denen nichts zu tun ist. Benennen Sie alle Schritte und begründen Sie eventuelle Reduktionen.

Lösungsvorschlag

(i) **Kanonische Überdeckung**

— Die kanonische Überdeckung - also die kleinst mögliche noch äquivalente Menge von funktionalen Abhängigkeiten kann in vier Schritten erreicht werden. \_\_\_\_\_

i. **Linksreduktion**

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

$$FA = \left\{ \right.$$

$$\left. \begin{array}{l} \{ B \} \rightarrow \{ F \}, \\ \{ C, D \} \rightarrow \{ E \}, \\ \{ C \} \rightarrow \{ A \}, \\ \{ C \} \rightarrow \{ A \}, \\ \{ D \} \rightarrow \{ F \}, \\ \{ D \} \rightarrow \{ B \}, \end{array} \right\}$$

**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. —

$$\text{FA} = \left\{ \begin{array}{l} \{ B \} \rightarrow \{ F \}, \\ \{ C, D \} \rightarrow \{ E \}, \\ \{ C \} \rightarrow \{ \emptyset \}, \\ \{ C \} \rightarrow \{ A \}, \\ \{ D \} \rightarrow \{ \emptyset \}, \\ \{ D \} \rightarrow \{ B \}, \end{array} \right\}$$

**iii. Löschen leerer Klauseln**

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

$$\text{FA} = \left\{ \begin{array}{l} \{ B \} \rightarrow \{ F \}, \\ \{ C, D \} \rightarrow \{ E \}, \\ \{ C \} \rightarrow \{ A \}, \\ \{ D \} \rightarrow \{ B \}, \end{array} \right\}$$

**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. —

$\emptyset$  Nichts zu tun

**(ii) Relationsschemata formen**

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

- $R_1(B, F)$
- $R_2(C, D, E)$
- $R_3(C, A)$
- $R_4(D, B)$

**(iii) Schlüssel hinzufügen**

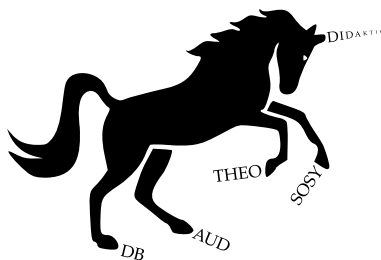
— Falls eines der in Schritt 2. erzeugten Schemata  $R_\alpha$  einen Schlüsselkandidaten von  $\mathcal{R}$  bezüglich  $F_c$  enthält, sind wir fertig, sonst wähle einen Schlüsselkandidaten  $\mathcal{K} \subseteq \mathcal{R}$  aus und definiere folgendes zusätzliche Schema:  $\mathcal{R}_\mathcal{K}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

$\emptyset$  Nichts zu tun

**(iv) Entfernung überflüssiger Teilschemata**

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

$\emptyset$  Nichts zu tun

**Die Bschlangaul-Sammlung**

Hermine Bschlangaul and Friends

Eine freie Aufgabensammlung mit Lösungen von Studierenden für Studierende zur Vorbereitung auf die 1. Staatsexamensprüfungen des Lehramts Informatik in Bayern.



Diese Materialsammlung unterliegt den Bestimmungen der Creative Commons Namensnennung-Nicht kommerziell-Share Alike 4.0 International-Lizenz.

Hilf mit! Die Hermine schafft das nicht allein! Das ist ein Community-Projekt! Verbesserungsvorschläge, Fehlerkorrekturen, weitere Lösungen sind herzlich willkommen - egal wie - per Pull-Request oder per E-Mail an [hermine.bschlangaul@gmx.net](mailto:hermine.bschlangaul@gmx.net). Der  $\text{\LaTeX}$ -Quelltext dieser Aufgabe kann unter folgender URL aufgerufen werden: <https://github.com/bschlangaul-sammlung/examens-aufgaben/blob/main/Staatsexamen/66116/2019/03/Thema-2/Teilaufgabe-1/Aufgabe-6.tex>