Pflichtmodul Informationssysteme (SS 2019)

Prof. Dr. Jens Teubner

Leitung der Übungen: Thomas Lindemann, Christoph Stahl

Übungsblatt Nr. 14

Ausgabe: 03.06.2019 Abgabe: —

Dieses Übungsblatt bildet eine Sammlung von Aufgaben, die teilweise Aufgaben in vergangenen Klausuren ähnlich sind. Es dient zur Selbstkontrolle und soll einen Überblick über einiger der über das Semester hinweg behandelten Themen verschaffen. Zudem wurde jede Aufgabe mit Punkten annotiert. Diese geben an, wieviel Zeit für die Bearbeitung der Aufgabe vorgesehen wird. Ein Punkt entspricht einer Minute Bearbeitungszeit. Insgesamt gibt es 60 Punkte zu verteilen, für das Blatt sind somit 60 Minuten vorgesehen.

Der Stoff dieses Übungsblattes umfasst zwar ein groSSen Teil des Stoffes der Veranstaltung, soll aber keineswegs als vollständige Liste an klausurrelevanten Themen verstanden werden.

Aufgabe 1 (ER-Modellierung (*Entity Relation Models*) (3 + 4 + 3 Punkte))

In einem Schulungszentrum werden verschiedene Kurse angeboten, wobei ausgewählte Kurse den Inhalt anderer Kurse voraussetzen. Je Kurs gibt mindestens eine Lehrkraft, die beliebig viele Kurse betreuen kann.

1. Erstellen Sie für diesen Diskursbereich ein ER-Diagramm. Verwenden Sie für die Funktionalität der Beziehungstypen die Min-/Max-Notation. *(3 Punkte)*

2. Überführen Sie das erhaltene ER-Diagram mittels `CREATE TABLE`-Statements in SQL Tabelle und geben Sie die Primär- und Fremdschlüsselbedingungen an. Überlegen Sie sich dazu geeignete Attributnamen. *(4 Punkte)*

3. Die Schulungsleitung entscheidet nun, dass jeder Kurs von genau einer Lehrkraft betreut werden soll und diese auch nur genau einen Kurs betreuen sollen. Beschreiben Sie kurz, wie sich das ER-Modell ändert, und welche Optimierungen an der SQL-Repräsentation vorgenommen werden können. *(3 Punkte)*

Aufgabe 2 (Relationen-Algebra (*Relational Algebra*) (4 + 8 Punkte))

1. Gegeben sei folgende Datenbank.

R			S			T		
A	B	C	A	B	C	A	B	C
1	1	x	1	3	x	1	3	y
1	4	y	1	1	x	4	3	y
5	1	x	4	3	y	1	1	x

Werten Sie folgende Anfragen in relationaler Algebra aus. *(2 + 2 Punkte)*

(a) $R \div \pi_{B \leftarrow A, C}(S)$

(b) $(S - R) \bowtie T$

2. Der folgende Auschnitt eines Datenbankschemas modeliert eine Datenbank an Schachspielenden. Die Relation *Player* modeliert einen Schachspielenden, die Relation *Game* ein Spiel zwischen zwei Schachspielenden. Das Attribut *Remis* gibt an, ob es sich bei dem Spiel um ein Unendschieden handelt, es kann *true* oder *false* sein.

$$\mathsf{sch}(\text{Player}) = (\underline{\text{PID}}, \text{Vorname}, \text{Nachname}, \text{Geburtsdatum})$$
$$\mathsf{sch}(\text{Game}) = (\underline{\text{GID}}, \text{SiegerID}, \text{VerliererID}, \text{Remis})$$

Hierbei ist *SiegerID* und *VerliererID* ein Fremdschlüssel für Player.

Geben Sie die folgenden natürlichsprachlichen Anfragen als Anfragen der relationalen Algebra an. *(2 + 2 + 4 Punkte)*

(a) Welche Schachspielenden haben mindestens einmal gewonnen?. (Ausgabe: *Vorname, Nachname*)

(b) Welche *verschiedenen* Schachspielende wurden am selben Tag geboren? (Ausgabe: PID_1, PID_2)

(c) Welche Schachspielende haben in mindestens zwei Spielen gewonnen? (Ausgabe: *SiegerID*)

Aufgabe 3 (Tupel-Relationen-Kalkül (*Tuple Relational Calculus*) (2 + 3 + 3 Pt))

Übersetzen Sie die folgenden Ausdrücke der relationalen Algebra mit Hilfe der gegebenen Übersetzungsregeln im Anhang des Übungsblattes in äquivalente Ausdrücke des Safe TRC. Dabei soll jeder einzelne Übersetzungsschritt klar erkennbar sein.

1. $\sigma_{A=5}(R)$ für $\mathsf{sch}(R) = (A, B, C)$

2. $\pi_{A,B}(R \cup S)$ für $\mathsf{sch}(R) = (A, B, C)$ und $\mathsf{sch}(S) = (A, B, C)$

3. $R \bowtie S$ für $\mathsf{sch}(R) = (A, B, C)$ und $\mathsf{sch}(S) = (A, D, E)$

Aufgabe 4 (SQL (*Structured Query Language*) (4 + 4 + 7 Punkte))

1. Im Folgenden werden Ihnen drei Momentaufnahmen von Datenbanken gegeben. Geben Sie an, wie Sie mit möglichst wenigen und möglichst kurzen SQL-Statements diese in einaner überführen können. Hierbei stellt eine Tabelle mit hellgrauem Kopf eine View dar. *(3 + 1 Punkte)*

Von:

R		
A	*B*	*C*
1	1	x
1	4	y
5	1	x

S		
D	*E*	*F*
1	3	x
1	1	x
4	3	y

T		
G	*H*	*I*
1	3	y
4	3	y
1	1	x

Über:

R		
A	*B*	*C*
1	1	x
1	4	y
5	1	x

S		
D	*E*	*F*
1	3	x
1	1	x
4	3	y

T		
G	*H*	*I*
1	3	y
4	3	y
1	1	x

View						
A	*B*	*C*	*D*	*E*	*G*	*H*
1	1	x	1	1	1	1
1	1	x	1	3	1	1
1	4	y	4	3	1	3
1	4	y	4	3	4	3
5	1	x	1	1	1	1
5	1	x	1	3	1	1

Nach:

R		
A	*B*	*C*
1	1	x
1	4	y

S		
D	*E*	*F*
1	3	x
4	3	y

T		
G	*H*	*I*
1	3	y
1	1	x

View						
A	*B*	*C*	*D*	*E*	*G*	*H*
1	1	x	1	3	1	1
1	4	y	4	3	1	3

Von → Über:

Über → Nach:

2. Betrachten Sie den Semijoin Operator $\ltimes$. Gehen Sie davon aus, dass R und S die folgenden Schemata haben.

   - $\mathsf{sch}(R) = \{A, B, C\}$
   - $\mathsf{sch}(S) = \{A, D, E\}$

   (a) Geben Sie $R \ltimes S$ nur mit Basisoperatoren der relationalen Algebra an. *(2 Punkte)*

   (b) Geben Sie ein zu $R \ltimes S$ äquivalentes SQL-Statement an. *(2 Punkte)*
   Hinweis: Es gibt kein Keyword `SEMI JOIN` in SQL.

3. Der folgende Auschnitt eines Datenbankschemas modeliert eine Datenbank an Schachspielenden. Die Relation *Player* modeliert einen Schachspielenden, die Relation *Game* ein Spiel zwischen zwei Schachspielenden. Das Attribut *Remis* gibt an, ob es sich bei dem Spiel um ein Unendschieden handelt, es kann *true* oder *false* sein.

$$\mathsf{sch}(\text{Player}) = (\underline{\text{PID}}, \text{Vorname}, \text{Nachname}, \text{Geburtsdatum})$$
$$\mathsf{sch}(\text{Game}) = (\underline{\text{GID}}, \text{SiegerID}, \text{VerliererID}, \text{Remis})$$

Geben Sie die folgenden natürlichsprachlichen Anfragen als SQL-Anfragen an.

(a) Wieviel Spiele hat *Garri Kasparow* gewonnen? (Ausgabe: *Anzahl*) *(2 Punkte)*

(b) Welche Schachspielende haben noch nie gewonnen? (Ausgabe: *Vorname, Nachname*) *(2 Punkte)*

(c) Wer ist der jüngste Spielende? (Ausgabe: *Vorname, Nachname*) *(3 Punkte)*
Hinweis: Verwenden Sie hier nicht `LIMIT`.

Aufgabe 5 (Schemanormalisierung (*Normalization*) (2 + 6 Punkte))

Gegeben seien das Relationenschema

$$\mathsf{sch}(R) = DBINFO$$

sowie die Menge

$$\mathcal{F} = \{DB \to INFO, \quad I \to NFO, \quad DI \to B, \quad F \to O, \quad N \to D\}$$

von funktionalen Abhängigkeiten.

1. Zeigen Sie, dass das gegebene Schema von R mit den gegebenen funktionalen Abhängigkeiten $\mathcal{F}$ **die Boyce-Codd Normalform verletzt**. *(2 Punkte)*

2. **Zerlegen Sie R mit Hilfe des BCNF-Zerlegungsalgorithmus aus der Vorlesung, sodass alle resultierenden Relationen in BCNF vorliegen.** *(6 Punkte)*

   Geben Sie dabei jeweils an, entlang welcher funktionalen Abhängigkeit Sie das Schema zerlegen. Machen Sie nach Ablauf des Algorithmus die Schemata aller erzeugten Tabellen, die zu dem zerlegten Relationenschema gehören und alle zugehörigen funktionalen Abhängigkeiten kenntlich.

(mehr Platz für Ihre Lösung auf der nächsten Seite)

Aufgabe 6 (Transaktionskontrolle (*Transactions*) (2 + 2 + 1 Punkte))

Gegeben sei die Menge $\mathbf{T} = \{T_1, T_2, T_3\}$ von Transaktionen.

Betrachtet den zu **T** passenden Schedule

$$S = \langle w_1(z), w_1(x), r_2(y), r_2(x), r_3(y), r_3(z), w_2(y), w_3(z), r_1(z) \rangle.$$

1. Stellen Sie die vollständige **Konfliktrelation** $\prec_S$ auf. *(2 Punkte)*

2. Zeichnen Sie den **Konflikgraphen** $G(S)$. *(2 Punkte)*

3. Ist der Schedule konfliktserialisierbar? Geben Sie wenn möglich eine **konfliktäquivalente serielle** Ausführungsreihenfolge der Transaktionen an. *(1 Punkt)*

Anhang (*Appendix*)

In Vorlesung und Übung wurde diskutiert, wie ein beliebiger Ausdruck der relationalen Algebra in einen äquivalenten Ausdruck des Safe TRC übersetzt werden kann:

$$\mathbb{TRC}\Big(\mathit{Exp}\Big) := \Big\{v \mid \mathbb{T}(v, \mathit{Exp})\Big\} \ .$$

$\mathbb{T}(v, \mathit{Exp})$ erzeugt dabei eine Formel, indem ein beliebiger Algebrausdruck *Exp* entlang seiner Struktur gemäSS der folgenden Regeln zerlegt wird:

$$\begin{aligned}
\mathbb{T}(v, R) &:= v \in R \qquad \text{(wobei } R \text{ eine Basisrelation ist)} \\
\mathbb{T}(v, \sigma_p(\mathit{Exp})) &:= \mathbb{T}(v, \mathit{Exp}) \wedge p(v) \\
\mathbb{T}(v, \pi_{\langle A_1,\ldots,A_k\rangle}(\mathit{Exp})) &:= \exists u : \mathbb{T}(u, \mathit{Exp}) \wedge v \leftarrow \langle u.A_1, \ldots, u.A_k\rangle \\
\mathbb{T}(v, \mathit{Exp}_1 \times \mathit{Exp}_2) &:= \exists u : \mathbb{T}(u, \mathit{Exp}_1) \wedge \exists w : \mathbb{T}(w, \mathit{Exp}_2) \wedge v \leftarrow \langle u, w\rangle \\
\mathbb{T}(v, \mathit{Exp}_1 \cup \mathit{Exp}_2) &:= \mathbb{T}(v, \mathit{Exp}_1) \vee \mathbb{T}(v, \mathit{Exp}_2) \\
\mathbb{T}(v, \mathit{Exp}_1 - \mathit{Exp}_2) &:= \mathbb{T}(v, \mathit{Exp}_1) \wedge \neg\mathbb{T}(v, \mathit{Exp}_2)
\end{aligned}$$