UNIVERSITÄT DES SAARLANDES
Fachbereich 14– Informatik
Prof. Dr.–Ing. G. Weikum
Dipl.–Inform. A. Kraiß

Datenbanksysteme WS 1998/99

Dritte Teilklausur — Beispiellösung

Aufgabe 1: Ausführung von Semi-Joins

Gegeben sei die verteilte Datenbank einer Fluggesellschaft mit den beiden Relationen

Passagiere(FNr, TicketNr, Name, Wohnort)
Flüge(FNr, Startort, Zielort, Startzeit)

Die unterstrichenen Attribute bilden den Primärschlüssel. Die Relation *Passagiere* sei in Saarbrükken (SB) gespeichert, und die Relation *Flüge* sei in Frankfurt/Main (FFM) gespeichert. Die Attributlängen sind in folgender Tabelle gegeben:

FNr, TicketNr	jeweils 8 Bytes
Name	40 Bytes
Wohnort	24 Bytes
Startort, Zielort	jeweils 8 Bytes
Startzeit	6 Bytes

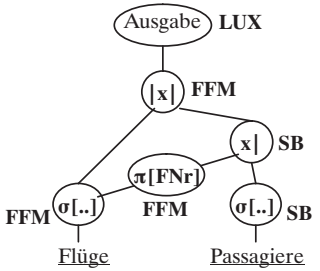
Von einem Rechner in Luxemburg (LUX) aus werde die folgende Anfrage an die verteilte Datenbank gestellt:

(σ[Wohnort='Merzig'](Passagiere)) |x| (σ[Startort='LUX'](Flüge))

Es seien die folgenden Kardinalitäten bekannt:

card(*Passagiere*)	200'000 Tupel
card(*Flüge*)	1'000 Tupel
card(σ[Wohnort='Merzig'](Passagiere))	10'000 Tupel
card(σ[Startort='LUX'](Flüge))	100 Tupel

(a) Berechnen Sie die Kommunikationskosten (in Anzahl zu versendender Bytes) für den folgenden Ausführungsplan:

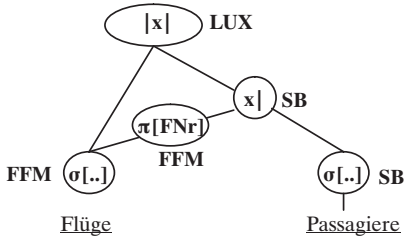


Lösung:
Annahme: Die Passagiere verteilen sich gleichmäßig und unabhängig über die Flüge.
An einem Flug nehmen somit im Mittel 200 Passagiere teil.
5% der Passagiere werden selektiert, 10% der Flüge werden selektiert.

- 100 Flugnummern * 8 Bytes von FFM nach SB: **800 Bytes**
- in SB Semi-Join mit *10'000* Passagier-Tupeln: Passagiere werden um 10% reduziert, da nur 1/10 der Flüge selektiert wurden: Semi-Join liefert *1000 Ergebnistupel* à 80 Bytes.
- Resultat von SB nach FFM: **80'000 Bytes**
- In FFM Join mit selektierten Flügen: 1000 Ergebnistupel à 102 Bytes
- Ergebnistupel mit **102'000 Bytes** werden von FFM nach LUX gesendet.
- Gesamt-Kommunikationskosten: **182'800 Bytes**

(b) Welches ist der bzgl. der Kommunikationskosten optimale Ausführungsplan? Begründen Sie Ihre Antwort.

Lösung:



- 100 Flüge-Tupel * 30 Bytes von FFM nach LUX: **3000 Bytes**
- 100 Flüge-Tupel * 8 Bytes von FFM nach SB: **800 Bytes**
- in SB Semi-Join mit Passagiere, liefert 1000 Ergebnistupel à 80 Bytes: **80'000 Bytes** von SB nach LUX
- Gesamt-Kommunikationskosten: **83'800 Bytes**
- Begründung: die 100 Flüge-Tupel lassen sich nicht reduzieren und müssen einmal über das Netz. Am besten direkt nach LUX. Die Passagiere-Tupel lassen sich um 10% reduzieren, was den Aufwand lohnt, denn 100*8+1000*80 < 10'000*80

Aufgabe 2: Concurrency Control (2+2=4)

Betrachten Sie den folgenden Schedule mit den Operationen von vier Transaktionen T_1, T_2, T_3 und T_4 auf den beiden Zählerwerten (z.B. Kontostände) *x* und *y*.

$$incr_1(x)\ incr_2(y)\ write_3(x)\ read_4(y)\ incr_3(y)\ incr_1(y)$$

Die *incr*-Operation inkrementiert einen Zählerwert um Eins (ohne den Zählerwert zurückzuliefern). Eine *incr*-Operation steht nicht in Konflikt zu einer anderen *incr*-Operation, sie steht jedoch in Konflikt zu einer *read*- und einer *write*-Operation.

Ist der Schedule konflikt-serialisierbar

(a) unter Ausnutzung der *incr*-Semantik ?
Lösung:
Konfliktpaare:
$<incr_1(x), write_3(x)>$, $<incr_2(y), read_4(y)>$, $<read_4(y), incr_3(y)>$, $<read_4(y), incr_1(y)>$
Serialisierungsreihenfolge ist damit T2 –> T4 –> T1 –> T3

(b) ohne Ausnutzung der *incr*-Semantik, d.h. wenn jede *incr*-Operation durch eine *read*- und eine unmittelbar darauffolgende *write*-Operation ersetzt wird ?
Lösung: Resultierender Schedule
$r_1(x)\ \mathbf{w_1(x)}\ r_2(y)\ w_2(y)\ \mathbf{w_3(x)}\ r_4(y)\ r_3(y)\ \mathbf{w_3(y)}\ \mathbf{r_1(y)}\ w_1(y)$
hat u.a. die Konfliktpaare $<w_1(x), >w_3(x)>$ und $<w_3(y), r_1(y)>$.
Damit ergibt sich ein Zyklus T1 <—> T3 und der Schedule ist nicht serialisierbar.

Aufgabe 3: Relationale Entwurfstheorie (4+3+1=8)

Gegeben seien die folgenden Relationen einer Bibliotheks-Datenbank mit den zugehörigen Funktionalabhängigkeiten:

Bücher(ISBN,Autor,Titel,Verlag)	1. ISBN → Autor, Titel, Verlag 2. Autor,Verlag → Titel 3. Autor,Titel,Verlag → ISBN
Exemplare(ISBN,ExplNr,Standort,Titel,Status)	1. ISBN,ExplNr → Standort,Status 2. ISBN → Titel 3. Standort → Status 4. Standort,Titel → ExplNr
Leser(LeserID,Name,PLZ,Ort)	1. LeserId → Name,PLZ,Ort 2. PLZ → Ort
Ausleihen(LeserID,ISBN,ExplNr,Datum)	1. ISBN,Datum → ExplNr 2. LeserId,ISBN → Datum

(a) Welche der Relationen sind in BCNF, welche in 3NF, und welche sind noch nicht einmal in 3NF? Begründen Sie jeweils Ihre Antwort.

Lösung:

- Relation *Bücher:* Schlüsselkandidaten sind *ISBN* und {*Autor,Verlag*}. *Bücher* ist in BCNF, da die linke Seite aller FDs einen Schlüsselkandidaten enthält.
- Relation *Exemplare*: Schlüsselkandidaten sind {*ISBN, ExplNr*} und {*ISBN,Standort*}. *Exemplare* ist wegen 2., 3. und 4. nicht in BCNF und ist wegen 2. und 3. auch nicht in 3NF
- Relation *Leser*: Schlüsselkandidat ist *LeserId*. *Leser* ist wegen 2. nicht in 3NF.
- Relation *Ausleihen*: Schlüsselkandidat ist {LeserId, ISBN}. *Ausleihen* ist wegen 1. nicht in 3NF.

(b) Zerlegen Sie die Relationen so, daß alle Relationen des Resultatschemas in BCNF sind.

Lösung:

Exemplare: entlang 2. in *E1*(ISBN,Titel) und
Ex(ISBN,ExplNr,Standort,Status)
entlang 3. in S(Standort,Status) und *Ex*(ISBN,ExplNr,Standort)

Leser: entlang 2. in *L*(LeserID,Name,PLZ) und *O*(PLZ, Ort)

Ausleihen: entlang 1. in *A1*(ISBN,Datum,ExplNr) und *A2*(ISBN,LeserId,Datum)

(c) Ist Ihr Resultatschema

- abhängigkeitsbewahrend
- verlustfrei bzgl. des ursprünglichen Schemas

Begründen Sie jeweils Ihre Antwort.

Lösung:

Das Resultatschema ist nicht abhängigkeitsbewahrend, da die 4. FD von *Exemplare* verlorengeht. BCNF-Dekompositionen sind jedoch immer verlustfrei.