

KFK GC/DM Grid Computing - Datenbanksysteme	schriftliche Einzelprüfung	06.02.2006	1
--	-------------------------------	------------	---

Aufgabe 1 [Relationale Abfragen: 25 Punkte]

Gegeben ist folgendes Relationenschema:

ort (ortsname, einwohnerzahl, flaeche)

person (sozialversicherungsnr, vorname, zuname, geburtsdatum)

bereist (ortsname, sozialversicherungsnr, datum, bezahlt)

bereist.ortsnameort, bereist.sozialversicherungsnrperson

bereist.bezahlt IN {'privat', 'firma', 'sponsor'}

Formulieren Sie die folgende Abfrage in Relationenalgebra (ohne Erweiterungen für Aggregatfunktionen zu verwenden):

- (8 Punkte) Ermitteln Sie Sozialversicherungsnummer und Zuname aller Personen, die zumindest 3 mal an den gleichen Ort gereist sind.

Formulieren Sie die folgenden Abfragen in SQL99:

- (5 Punkte) Ermitteln Sie die Einwohnerzahlen aller Orte, die schon auf private Kosten und auf Firmenkosten bereist wurden, niemals aber auf Kosten eines Sponsors.
- (7 Punkte) Ermitteln Sie die Sozialversicherungsnummern der Personenpaare (d. h. jeweils 2 Personen), die den selben Vornamen aber verschiedene Zunamen haben, wobei zumindest ein Ort existiert, den beide Personen bereist haben.
- (5 Punkte) Ermitteln Sie die Vornamen und Zunamen aller Personen, für die die Gesamtfläche aller von der jeweiligen Person bereisten Orte kleiner ist als 100000.

Aufgabe 2 [Datalog: 15 Punkte]

Folgendes Relationenschema ist gegeben:

ort(name, flaeche),

verbindung(ort1, ort2),

verbindung.ort1ort,

verbindung.ort2ort.

Die Relation speichert ein Straßennetz.

Finden Sie eine Datalog-Abfrage, die für die zwei Orte "A-Dorf" und "B-Dorf" ermittelt, ob es eine Verbindung zwischen diesen beiden Orten gibt.

Aufgabe 3 [Query Abarbeitung: 15 Punkte]

Welche Algorithmen zur Durchführung von Joins kennen Sie? Beschreiben Sie die verschiedenen Algorithmen kurz.

Aufgabe 4 [Query Optimierung: 20 Punkte]

Gegeben ist die folgende relationale Algebra Abfrage auf dem Relationenschema R1(A, D), R2(B, D), R3(C, E):

$\sigma_{(A>0 \wedge B<5)}(R1 * (R2_{(D<E)}R3))$

- (2 Punkte)

Stellen Sie den Ausdruck grafisch dar.

- (8 Punkte)

Führen Sie eine heuristische Optimierung mit Hilfe der in der Vorlesung angegebenen Äquivalenzumformungsregeln durch, mit dem Ziel den Abarbeitungsaufwand zu minimieren.

Stellen Sie den umgeformten Ausdruck grafisch dar und geben Sie die Reihenfolge der verwendeten Äquivalenzumformungsregeln an.

Begründen Sie Ihre Vorgangsweise.

- (10 Punkte)

Nehmen Sie an, dass die Relation R1 1000, die Relation R2 5000 und die Relation R3 100 Datensätze enthält, wobei die Blockgröße für alle Relationen 10 ist.

Für den Join wird das Nested-Loop Verfahren verwendet (Memorygröße 1 Block) und die Selektivität der Selektionen ist jeweils 1/10 (Annahme der Unabhängigkeit).

Geben Sie die Kosten (Anzahl von Blockzugriffen) für

- den ursprünglichen Ausdruck und
- den umgeformten Ausdruck an.

Nehmen Sie an, dass die Abarbeitung der Ausdrücke Pipelining nützt.

Aufgabe 5 [XML: 15 Punkte]

Gegeben ist die folgende DTD:

```
<!DOCTYPE db [  
  <!ELEMENT db (Fahrzeug*)>  
  <!ELEMENT Fahrzeug (Typ, Seriennummer, Vorbesitzer*)>  
  <!ELEMENT Vorbesitzer (Vorname, Zuname, Adresse)>  
  <!ELEMENT Typ( #PCDATA )>  
  <!ELEMENT Seriennummer( #PCDATA )>  
  <!ELEMENT Vorname( #PCDATA )>  
  <!ELEMENT Zuname( #PCDATA )>  
  <!ELEMENT Adresse( #PCDATA )>  
>
```

a. (5 Punkte)

Geben Sie ein gültiges XML Dokument an, das auf dieser DTD beruht und in dem alle Elemente der DTD vorkommen.

b. (5 Punkte) Schreiben Sie eine XPath Abfrage, die alle Fahrzeug-Elemente liefert, die weniger als 2 Vorbesitzer anführen.

c. (5 Punkte)

Schreiben Sie eine XQuery Abfrage, die Seriennummer und Typ aller Fahrzeuge sortiert nach Seriennummer liefert, die einen Vorbesitzer namens 'Fred Feuerstein' haben.

Aufgabe 6 [Serialisierbarkeit: 10 Punkte]

a. (5 Punkte) Wann ist ein Schedule konflikt-serialisierbar? Geben Sie jeweils ein konkretes Beispiel für einen konflikt-serialisierbaren und einen nicht konflikt-serialisierbaren Schedule an.

b. (5 Punkte) Wann ist ein Schedule view-serialisierbar? Geben Sie ein konkretes Beispiel für einen view-serialisierbaren Schedule an, der nicht konflikt-serialisierbar ist.