



## 5. Übungsblatt

(Testwoche: 18. - 20. Mai 2010)

Einführung in Datenbanksysteme  
Datenbanken für die Bioinformatik

Heinz Schweppe, Katharina Hahn

### Aufgabe 1 (Kardinalitäten von Ergebnisrelationen)

6 Punkte

Gegeben sind die Relationen  $A$ ,  $B$  und  $C$  mit den Attributen  $a_1, a_2, \dots$  bzw.  $b_1, b_2, \dots$  bzw.  $c_1, c_2, \dots$  und den Kardinalitäten  $|A|$ ,  $|B|$  und  $|C|$ .

a) Zwischen welchen Schranken liegt die Kardinalität der folgenden Ergebnismengen. Geben Sie möglichst scharfe Schranken an.

- (1) `SELECT * FROM A JOIN B ON (a1 = b1) JOIN C ON (b2 = c3)`
- (2) `SELECT a1, a2 FROM A UNION SELECT b2, b1 FROM B`
- (3) `SELECT a1 FROM A WHERE a1 NOT IN (SELECT b2 FROM B)`
- (4) `SELECT * FROM A, B, C`

b) Nehmen Sie an, dass  $a_i$  Schlüssel von  $A$ ,  $b_j$  ein Fremdschlüssel auf  $A$  und  $c_k$  Schlüssel von  $C$  ist. Zwischen welchen Schranken liegt die Kardinalität der folgenden Mengen. Geben Sie möglichst scharfe Schranken an.

- (1) `SELECT * FROM A JOIN B ON (a_i = b_1)`
- (2) `SELECT * FROM A JOIN B ON (a_i = b_j) JOIN C ON (b_2 = c_k)`

### Aufgabe 2 (Advanced SQL)

14 Punkte

Gegeben ist das Kinodatenbankschema des letzten Übungsblatts.

```
KinoDB (
  Orte(Kino, Adresse, Telefon),
  Filme(Titel, Regie, Schauspieler),
  Programm(Kino, Titel, Zeit)
)
```

Formulieren Sie folgende Anfragen in SQL.

- a) Welche Filme (Titel) laufen nicht mehr aktuell im Kino?
- b) Wieviele Filme laufen aktuell, bei denen *Sofia Coppola* Regie geführt hat?
- c) Welche Filme (Titel) laufen nur zu einer Uhrzeit.
- d) In wie vielen Kinos läuft *A Single Man* nach 18h?
- e) In welchen laufenden Filmen spielen **ausschließlich** Schauspieler mit, die schon mal mit *Steven Spielberg* gearbeitet haben?

### Aufgabe 3 (Advanced SQL, Mondial Datenbank)

15 Punkte

Formulieren Sie folgende Anfragen in SQL. Einige der Anfragen lassen sich unter zu Hilfenahme der Mengenquantoren ALL, ANY und EXISTS formulieren.

Geben Sie jeweils **gültige** SQL-Statements an.

- a) Welche Länder liegen nicht an Meeren, wohl aber alle ihre Nachbarländer?
- b) Welche Länder (Name) mit mehr als 10 Millionen Einwohnern haben eine höhere Bevölkerungsdichte (Einwohner/Fläche) als alle ihre Nachbarländer?
- c) Geben Sie alle Flüsse an, die in ein Meer münden. Das Meer soll dabei nur an eines der Landesteile grenzen, durch die der Fluss fließt. Das Meer kann aber an andere Landesteile grenzen, durch die der Fluss nicht fließt.
- d) Welche Länder, außer dem bevölkerungsreichsten Land haben mehr als 100 Millionen Einwohner? Lassen Sie Ländername, Bevölkerungszahl und Gesamtfläche ausgeben.
- e) Welche Länder haben eine mindestens fünfmal größere Gesamtfläche als jeder der deutschen Nachbarstaaten. Lassen Sie Ländername und Gesamtfläche ausgeben.

#### **Aufgabe 4 (SQL Gruppenprädikate)**

**4 Punkte**

Kann man – wenn ja, wie; wenn nein, warum nicht – das *HAVING*-Prädikat für Gruppen durch Zeilenprädikate ausdrücken?

#### **Aufgabe 5 (SQL DATE und TIMESTAMP)**

**12 Punkte**

Machen Sie sich mit den Datum- und Zeitstempelformaten von Postgres vertraut, z.B. unter <http://developer.postgresql.org/pgdocs/postgres/functions-datetime.html> für die Version 9. Betrachten Sie nun folgende Tabelle:

```
CREATE TABLE users (  
    username VARCHAR(15) PRIMARY KEY,  
    firstname VARCHAR(15),  
    lastname VARCHAR(19),  
    birthday DATE,  
    registered TIMESTAMP  
);
```

Beantworten Sie folgende Anfragen in SQL. Geben Sie jeweils **gültige** SQL Statements an.

- a) Geben Sie den ältesten Benutzer mit Namen, Vornamen und Geburtsdatum aus. Falls es mehrere gibt, sollen alle sortiert nach Namen ausgegeben werden.
- b) Geben Sie alle Benutzer, die älter als 18 Jahre sind, unter Verwendung  
(1) von *CURRENT\_DATE*,  
(2) der *AGE*-Funktion aus.
- c) Geben Sie alle Benutzer aus, die beigetreten (Attribut *registered*) sind, bevor andere Benutzer geboren wurden.
- d) Geben Sie alle Benutzer aus, die nach 16h beigetreten sind.
- e) Geben Sie die Paare von Benutzern aus, die am gleichen Tag Geburtstag haben (beachten Sie, dass diese nicht im gleichen Jahr geboren sein müssen). Geben Sie jeweils die Nachnamen, Vornamen und Geburtsdaten der Benutzer aus. Achten Sie darauf, dass ein solches Paar von Benutzern (*u1*, *u2*) nicht zusätzlich als (*u2*, *u1*) ausgegeben wird.