



Universität Freiburg  
Institut für Informatik  
Prof. Dr. G. Lausen  
Alexander Schätzle  
Martin Przyjaciel-Zablocki

Georges-Köhler Allee, Geb. 51  
D-79110 Freiburg  
lausen@informatik.uni-freiburg.de  
schaetzle@informatik.uni-freiburg.de  
zablocki@informatik.uni-freiburg.de

Übungen zur Vorlesung  
*Datenbanken und Informationssysteme*  
Wintersemester 2013/2014  
19.11.2013

## 5. Aufgabenblatt: SQL - Sichten & referentielle Integrität

### Vorbereitung

Auf der Vorlesungs-Webseite finden Sie das Skript zur Erstellung der Tabelle `Borders`. Laden Sie die Tabelle in Ihre Datenbank (F5 - als Skript ausführen). Das Schema der Tabelle ist wie folgt:

<b>Borders</b>	Grenzen zwischen Ländern
<i>Country1</i>	Ländercode
<i>Country2</i>	Ländercode
<i>Length</i>	Länge der Grenze in Kilometer

### Aufgaben, die nicht bewertet werden

#### Übung 1

Sei gegeben die folgende Tabellendefinition:

```
CREATE TABLE T (  
  A NUMERIC NOT NULL,  
  B NUMERIC NOT NULL );
```

- Geben Sie eine SQL-Anfrage an, die als Antwort gerade alle Zeilen  $z$  von  $T$  liefert, die die folgende Eigenschaft erfüllen: *Jede Zeile  $z'$  in  $T$ , die in  $A$  mit  $z$  übereinstimmt, stimmt auch in  $B$  mit  $z$  überein.*
- Geben Sie eine SQL-Anfrage an, die als Antwort gerade alle Zeilen von  $T$  liefert, die eindeutig über ihren  $A$ -Wert identifiziert werden können.

Berücksichtigen Sie jeweils die Möglichkeit von Duplikaten in  $T$ .

#### Übung 2

Angenommen, ein Benutzer benötigt häufig die Information, welche Stadt in welchem Land liegt. Dabei interessiert er sich aber nicht für Ländercodes sondern für die Namen der Länder. Definieren Sie eine entsprechende Sicht, die diese Informationen zur Verfügung stellt. Würde es sich in diesem Fall anbieten, eine materialisierte Sicht zu erstellen? Begründen Sie Ihre Antwort.

### Übung 3

Betrachten Sie die beiden folgenden Tabellen:

Parent:	PNr	Child:	CNr	Pnr
	1		1	1
	2		2	2
	3		3	3
			4	1
			5	2
			6	3

- Erzeugen Sie die beiden Tabellen mittels `CREATE TABLE` und definieren Sie dabei auch die referentielle Abhängigkeit (ohne Angabe der referentiellen Aktion). Löschen Sie anschliessend mittels `DELETE` die Zeilen 4 und 5 aus der Child Tabelle. Welche Auswirkungen hat das auf die Parent Tabelle? Löschen Sie dann die erste Zeile der Parent Tabelle und beschreiben Sie die Auswirkungen.
- Löschen Sie die Child Tabelle mittels `DROP TABLE` und erstellen Sie sie neu. Verwenden Sie aber dieses Mal die referentielle Aktion `CASCADE` für das Löschen. Führen Sie die Löschoperationen aus Aufgabenteil (a) erneut aus. Wie verändert sich das Verhalten?
- Wie Aufgabenteil (b), verwenden Sie dieses Mal aber die referentielle Aktion `SET NULL` für das Löschen. Wie verändert sich das Verhalten?

### Aufgaben, die bewertet werden (20 Punkte)

#### Übung 4 (2+2 Punkte)

Begründen Sie, warum die folgenden SQL-Anfragen fehlerhaft sind und geben Sie jeweils eine entsprechend korrigierte SQL-Anfrage an.

- Grösste Landesfläche mit zugehörigem Ländercode.  

```
SELECT MAX(area), code
FROM Country ;
```
- Liste der Ländernamen mit Anzahl ihrer Städte.  

```
SELECT Country.name, count(*) AS anzahl
FROM Country JOIN City ON Country.code = City.country
GROUP BY Country.code
ORDER BY anzahl DESC;
```

#### Übung 5 (4 Punkte)

Zeigen Sie, dass die SQL-Anfrage

```
SELECT * FROM R WHERE R.A NOT IN (SELECT A FROM S)
```

immer eine leere Tabelle als Resultat liefert, wenn S mindestens eine Zeile mit Wert `null` für A enthält.

#### Übung 6 (7 Punkte)

Angenommen, Sie betreiben eine Internetseite, auf der Sie Informationen über Landreisen in Europa zur Verfügung stellen. U. a. möchten Sie Ihren Besuchern dabei ermöglichen herauszufinden, wie viele Grenzübergänge mindestens notwendig sind, um von einem Land in Europa zu einem anderen Land in Europa zu gelangen.

Erstellen Sie eine Sicht basierend auf der Tabelle `Borders`, die für alle Paare von europäischen Ländern angibt, wie viele Grenzübergänge mindestens notwendig sind, um vom einen Land zum anderen zu gelangen. Sortieren Sie die Sicht zunächst nach dem Ausgangsland, dann nach Anzahl der nötigen Grenzübergänge und schließlich nach dem Zielland. Zur Vereinfachung soll die Sicht nur diejenigen Paare enthalten, die maximal drei Grenzübergänge erfordern.

*Hinweis:* Die Tabelle `Borders` ist nicht symmetrisch. Erstellen Sie sich zunächst, analog zur Vorlesung, eine symmetrische Sicht der Tabelle und verwenden sie diese anschliessend zum Lösen der Aufgabe.

### Übung 7 (5 Punkte)

Mit Hilfe des folgenden SQL-Skripts soll demonstriert werden, dass das Löschen einer einzigen Zeile unter Umständen alle Datenbankinhalte löschen kann. Fügen Sie hierfür geeignete INSERT Befehle zum Skript hinzu. In beide Tabellen sollen dabei jeweils mindestens 6 Zeilen eingefügt werden. Nehmen Sie keine weiteren Änderungen am Skript vor!

```
CREATE TABLE Z1 (  
K1 CHAR(2),  
K2 CHAR(2),  
PRIMARY KEY (K1) );  
  
CREATE TABLE Z2 (  
K2 CHAR(2),  
K1 CHAR(2),  
PRIMARY KEY (K2) );  
  
-- ADD INSERTS HERE  
  
ALTER TABLE Z1 ADD CONSTRAINT fk1  
FOREIGN KEY (K2) REFERENCES Z2(K2) ON DELETE CASCADE;  
  
ALTER TABLE Z2 ADD CONSTRAINT fk2  
FOREIGN KEY (K1) REFERENCES Z1(K1) ON DELETE CASCADE;  
  
DELETE FROM Z1 WHERE K1 = '1';
```

Abzugeben durch Einwurf in den Briefkasten Raum 01-025 Gebäude 51 bis spätestens 25.11.2013, 12:00 Uhr.