



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  
11.12.2013

## 8. Aufgabenblatt: SQL & ER-Modell

### Vorbereitung

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

<b>Province</b>	Informationen über Provinzen
<i>Name:</i>	Name der Provinz
<i>Country:</i>	Ländercode
<i>Population:</i>	Einwohnerzahl der Provinz
<i>Area:</i>	Fläche der Provinz
<i>Capital:</i>	Hauptstadt der Provinz

### Aufgaben, die nicht bewertet werden

#### Übung 1

Betrachten Sie die folgenden Tabellendefinitionen und Trigger:

```
CREATE TABLE Z1 (  
  K1 CHAR(2),  
  K2 CHAR(2),  
  PRIMARY KEY (K1) );  
  
CREATE TABLE Z2 (  
  K2 CHAR(2),  
  K1 CHAR(2),  
  PRIMARY KEY (K2) );  
  
INSERT INTO Z1 (K1, K2) VALUES ('1', '2');  
INSERT INTO Z1 (K1, K2) VALUES ('3', '4');  
INSERT INTO Z1 (K1, K2) VALUES ('5', '6');  
INSERT INTO Z1 (K1, K2) VALUES ('7', '8');  
INSERT INTO Z1 (K1, K2) VALUES ('9', '10');  
  
INSERT INTO Z2 (K2, K1) VALUES ('2', '3');  
INSERT INTO Z2 (K2, K1) VALUES ('4', '5');  
INSERT INTO Z2 (K2, K1) VALUES ('6', '7');  
INSERT INTO Z2 (K2, K1) VALUES ('8', '9');  
INSERT INTO Z2 (K2, K1) VALUES ('10', '1');
```

```
CREATE OR REPLACE TRIGGER Z1_delete
AFTER DELETE ON Z1 REFERENCING OLD as oldrow
FOR EACH ROW
BEGIN
    DELETE FROM Z2 WHERE K1 = :oldrow.K1;
END;
```

```
CREATE OR REPLACE TRIGGER Z2_delete
AFTER DELETE ON Z2 REFERENCING OLD as oldrow
FOR EACH ROW
BEGIN
    DELETE FROM Z1 WHERE K2 = :oldrow.K2;
END;
```

---

- Geben Sie den zu erwartenden Inhalt der beiden Tabellen nach Ausführung der Anweisung `DELETE FROM Z1 WHERE K1 = '1'`; an. Wieso lässt sich das Beispiel in Oracle so nicht ausführen?
- Ergänzen Sie die Tabellen um Fremdschlüssel mit referentiellen Aktionen so, dass der Effekt von (a) auch ohne Trigger auftritt. Sie können dies über die Funktionalität des Oracle SQL Developers versuchen oder auch mittels der Anweisung `ALTER TABLE table_name ADD CONSTRAINT xyz ...`.

## Übung 2

Überprüfen Sie die Integritätsbedingung "Alle Länder sind Mitglied der Organisation *World Health Organization*" mittels einer SQL-Anfrage. Wenn die Datenbank die Integritätsbedingung erfüllt, dann soll Ihre Anfrage die Zeichenkette *'Alle Länder sind in der WHO'* liefern. Andernfalls soll die Anzahl der Länder ausgegeben werden, die die Integrität verletzen.

*Hinweis:* Sie können die Default-Tabelle DUAL von Oracle verwenden, die gerade eine Zeile und ein Attribut enthält. Ihre Anfrage kann somit die Struktur `"SELECT <hier der Test> FROM DUAL;"` haben.

## Aufgaben, die bewertet werden (20 Punkte)

### Übung 3 (5 Punkte)

Überprüfen Sie die Integritätsbedingung *"Die Hauptstadt eines Landes ist die Stadt mit den meisten Einwohnern des Landes"* mittels einer SQL-Anfrage. Wenn die Datenbank die Integritätsbedingung erfüllt, dann soll Ihre Anfrage die Zeichenkette *'Die Hauptstädte sind die größten Städte der jeweiligen Länder'* liefern. Andernfalls soll Ihre Anfrage die Fehlernachricht *'Verletzung der Integrität'* liefern.

*Hinweis:* Berücksichtigen Sie nur die Länder, für die die Hauptstädte bekannt sind.

### Übung 4 (2+3 Punkte)

Man betrachte folgendes (Teil)schema einer Filmdatenbank:

```
Studio(
```

```
    name: string,
    address: string,
    presC#: integer)
```

```
MovieExec(
```

```
    name: string,
    address: string,
    cert#: integer,
    netWorth: integer)
```

Dabei steht `MovieExec` für *Movie Executive*, wobei die *Certificate Number* Schlüssel ist. `presC#` ist Fremdschlüssel und bezieht sich auf den Präsidenten eines Studios. Es soll nun gesichert werden, daß niemand Präsident eines Studios werden kann, dessen `netWorth` nicht mindestens \$ 1.000.000 beträgt.

- Geben Sie eine entsprechende allgemeine Bedingung (Assertion) in SQL an.
- Geben Sie Basistabellenbedingungen mit gleicher Wirkung an. Welche Probleme können bei diesem Ansatz auftreten und wie könnte man sie lösen?

### Übung 5 (1+1+2+2 Punkte)

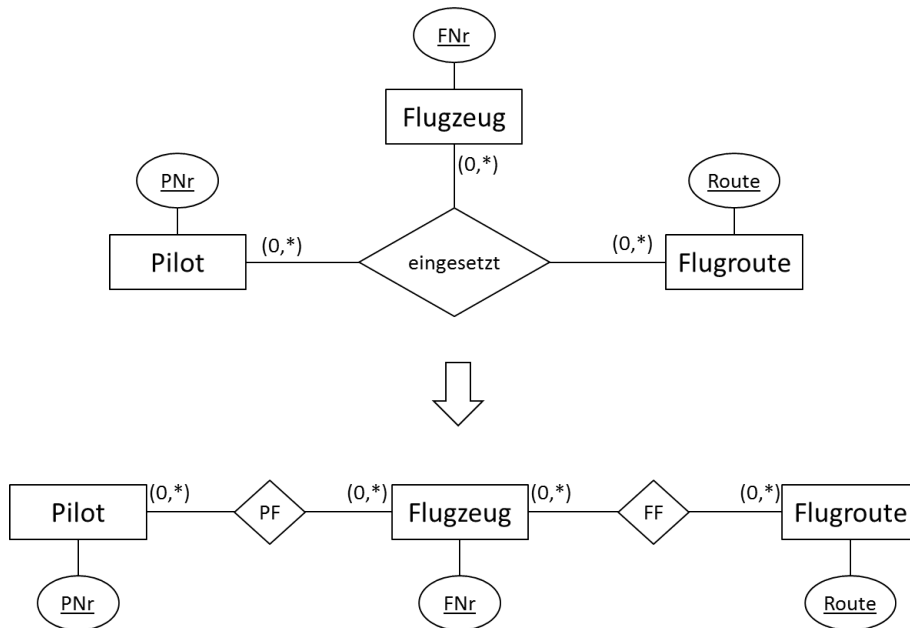
Betrachten Sie die folgende Sicht:

```
CREATE OR REPLACE VIEW myCube AS
SELECT E.Continent, C.Code, P.Name AS PName, Ci.Name AS CiName, COUNT(*) AS Num,
       GROUPING(E.Continent) AS KontTest,
       GROUPING(C.Code) AS LandTest,
       GROUPING(P.Name) AS ProvTest,
       GROUPING(Ci.Name) AS StadtTest
FROM Encompasses E
  LEFT OUTER JOIN Country C ON E.Country = C.Code
  LEFT OUTER JOIN Province P ON C.Code = P.Country
  LEFT OUTER JOIN City Ci ON P.Name = Ci.Province AND P.Country = Ci.Country
GROUP BY CUBE(E.Continent, C.Code, P.Name, Ci.Name)
ORDER BY KontTest, LandTest, ProvTest, StadtTest ;
```

- Welche Aufgabe hat die Funktion GROUPING? Beschreiben Sie den Informationsgehalt der Sicht.
- Geben Sie eine SQL-Anfrage an, die zu jedem Kontinent die Anzahl seiner Länder bestimmt.
- Wie (b), jedoch darf in der Anfrage kein GROUP BY verwendet werden.
- Geben Sie eine SQL-Anfrage an, die die Provinz mit den meisten Städten bestimmt.

### Übung 6 (4 Punkte)

Betrachten Sie folgende Zerlegung eines ternären Beziehungstyps in zwei binäre Beziehungstypen:



Ist die angegebene Zerlegung zulässig? Begründen Sie, wieso die Zerlegung zulässig ist oder zeigen Sie anhand einer Beispiel-Instanz, dass die Zerlegung nicht zulässig ist.

Abzugeben durch Einwurf in den Briefkasten Raum 01-025 Gebäude 51 bis spätestens 16.12.2013, 12:00 Uhr.