



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
29.11.2013

7. Aufgabenblatt: SQL - Integrität, OLAP & Zugriffskontrolle

Vorbereitung

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

<code>is_Member</code>	Mitgliedschaften der Länder
<i>Organization:</i>	Name der Organisation
<i>Country:</i>	Ländercode
<i>Type:</i>	Status der Mitgliedschaft

Aufgaben, die nicht bewertet werden

Übung 1

Definieren Sie die folgenden Tabellen in Oracle:

```
CREATE TABLE T1 (  
    k1 NUMERIC NOT NULL ,  
    PRIMARY KEY (k1));
```

```
CREATE TABLE T2 (  
    k2 NUMERIC NOT NULL ,  
    k1 NUMERIC ,  
    PRIMARY KEY (k2);
```

```
CREATE TABLE T3 (  
    k3 NUMERIC NOT NULL ,  
    k1 NUMERIC ,  
    PRIMARY KEY (k3);
```

```
CREATE TABLE T4 (  
    k4 NUMERIC NOT NULL ,  
    k2 NUMERIC ,  
    k3 NUMERIC ,  
    PRIMARY KEY (k4);
```

- a) Simulieren Sie die Effekte der folgenden referentiellen Aktionen mittels Trigger:
- FOREIGN KEY(k1) REFERENCES T1(k1) ON DELETE CASCADE -- in Tabelle T2
 - FOREIGN KEY(k1) REFERENCES T1(k1) ON DELETE CASCADE -- in Tabelle T3
 - FOREIGN KEY(k2) REFERENCES T2(k2) ON DELETE CASCADE,
FOREIGN KEY(k3) REFERENCES T3(k3) ON DELETE SET NULL -- in Tabelle T4
- b) Wie unter (a), jedoch gelte nun bzgl. Tabelle T4:
- FOREIGN KEY(k2) REFERENCES T2(k2) ON DELETE CASCADE,
FOREIGN KEY(k3) REFERENCES T3(k3) ON DELETE RESTRICT -- in Tabelle T4

Übung 2

Betrachten Sie die beiden Tabellen $R(A, B)$ und $S(B, C)$. Definieren Sie zwei Trigger, sodass das Einfügen eines Tupels in R eine nicht terminierende Sequenz von Trigger-Aufrufen zur Folge hat.

Übung 3

Angenommen eine Datenbank habe vier Benutzer (A, B, C, D). Weiterhin sei der Benutzer A Erzeuger der Tabelle T . Betrachten Sie folgende Zugriffszuweisungen:

```
-- Benutzer A
GRANT Select, Insert, Update, Delete ON T TO B,C WITH GRANT OPTION ;
-- Benutzer B
GRANT Select ON T TO C WITH GRANT OPTION ;
GRANT Insert ON T TO C ;
-- Benutzer C
GRANT Select, Insert, Update ON T TO D ;
```

Geben Sie anhand der folgenden Tabelle an, welcher Benutzer welche Rechte an der Tabelle T besitzt:

Benutzer	Select	Insert	Update	Delete
A				
B				
C				
D				

(√: Ja, ×: Nein)

Anschließend führe Benutzer A die folgenden Anweisungen aus. Geben Sie den Zustand der Rechte-Tabelle nach jeder REVOKE Anweisung an. Begründen Sie dabei, ob und welche Änderungen sich ergeben haben.

```
-- Benutzer A
REVOKE Select, Insert ON T FROM C CASCADE ;

REVOKE Delete ON T FROM B RESTRICT ;

REVOKE GRAND OPTION FOR Select ON T FROM B CASCADE ;

REVOKE Update ON T FROM C RESTRICT ;
```

Aufgaben, die bewertet werden (20 Punkte)

Übung 4 (1+1+1+2+2 Punkte)

Betrachten Sie folgende Tebellendefinitionen:

```
CREATE TABLE Abteilung(  
  AbNr          INTEGER PRIMARY KEY,  
  Bezeichnung   VARCHAR2(40),  
  Budget       NUMBER,  
  AnzAng       INTEGER  
);
```

```
CREATE TABLE Angestellte(  
  AngNr        INTEGER PRIMARY KEY,  
  Name         VARCHAR2(40),  
  Gehalt       NUMBER,  
  AbNr         INTEGER  
);
```

Definieren Sie Integritätsbedingungen für folgende Anforderungen. Verwenden Sie hierfür Spaltenbedingungen, Tabellenbedingungen oder Assertions:

- Ein Angestellter hat mindestens ein Gehalt von 1000 Euro.
- Das Budget einer Abteilung muss mindestens so groß sein wie die Summe aller Gehälter ihrer Angestellten.
- Ein Angestellter kann nur einer existierenden Abteilung zugeteilt werden.
- Das größte Gehalt in einer Abteilung darf maximal doppelt so groß sein wie das durchschnittliche Gehalt der Abteilung.
- Das größte Gehalt in einer Abteilung darf maximal doppelt so groß sein wie das durchschnittliche Gehalt aller Angestellten.

Übung 5 (3+4 Punkte)

Betrachten Sie erneut die Tebellendefinitionen aus Aufgabe 4. Definieren Sie Trigger für folgende Anforderungen:

- Das Attribut `AnzAng` enthält immer die Anzahl aller Angestellten der Abteilung, d.h. es müssen entsprechende Änderungen in der Tabelle `Angestellte` übertragen werden.
- Die Gehaltssumme aller Angestellten einer Abteilung darf 50% des Budgets der Abteilung nicht überschreiten. Überprüfen Sie Änderungen am Budget einer Abteilung sowie Änderungen und Einfügungen in der Tabelle `Angestellte` und weisen Sie unzulässige ab.

Übung 6 (1+3+2 Punkte)

- Die Tabelle `CITY` der Mondial-Datenbank enthält die Attribute `LATITUDE` und `LONGITUDE` jeweils vom Typ `NUMBER`. Definieren Sie eine Sicht `CITY_LAT_LON`, die eine zu `CITY` bis auf den Typ der Spalten `LATITUDE` und `LONGITUDE` identische Tabelle definiert; `LATITUDE` und `LONGITUDE` sollen in der Sicht vom Typ `NUMBER(3)` sein, d.h. Nachkommastellen werden abgeschnitten.
- Erläutern Sie das Ergebnis der folgenden SQL-Anweisung.
Geben Sie eine äquivalente Anfrage unter Verwendung von `GROUP BY` ohne `ROLLUP` an.

```
SELECT ci.country, lat, lon, SUM(population), COUNT(*)  
FROM   City_Lat_Lon ci, is_member  
WHERE  ci.country = is_member.country AND is_member.organization = 'OPEC'  
GROUP BY ROLLUP(ci.country, lat, lon)  
ORDER BY ci.country, lat, lon ;
```

- Erläutern Sie das Ergebnis der folgenden SQL-Anweisung.
Eine Anfrage mit `CUBE` kann in einer äquivalenten Anfrage unter Verwendung von `GROUP BY` ohne `CUBE` analog zu (b) berechnet werden. Geben Sie an, aus wievielen mit `UNION ALL` verbundenen Teilanfragen die Anfrage dann bestehen würde.

```
SELECT ci.country, lat, lon, SUM(population), COUNT(*)  
FROM   City_Lat_Lon ci, is_member  
WHERE  ci.country = is_member.country AND is_member.organization = 'OPEC'  
GROUP BY CUBE(ci.country, lat, lon)  
ORDER BY ci.country, lat, lon ;
```

Abzugeben durch Einwurf in den Briefkasten Raum 01-025 Gebäude 51 bis spätestens 09.12.2013, 12:00 Uhr.