



Universität Freiburg
Institut für Informatik
Prof. Dr. Georg Lausen
Io Taxisdou

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

Übungen zur Vorlesung
Datenbanken und Informationssysteme
Wintersemester 2017/2018

Ausgabe: 28.11.2017
Abgabe: 04.12.2017, 12:00 Uhr

8. Aufgabenblatt: SQL - Prozeduren, Funktionen & Integrität

Wichtiger Hinweis: Kommentieren Sie bei komplexeren SQL-Anfragen Ihren Lösungsansatz!
Das vereinfacht den Tutoren die Korrektur und ermöglicht auch die Vergabe von Teilpunkten bei Fehlern.

Hinweis zu PL/SQL Prozeduren und Funktionen:

Ein übersichtliches Tutorial zu PL/SQL finden Sie z.B. unter <http://www.datenbank-plsql.de/>.
Eine ausführliche Dokumentation bietet die *Oracle Database PL/SQL Language Reference*.

In ILIAS finden Sie das Skript zur Erstellung der Tabelle **Borders** mit folgendem Schema. Laden Sie die Tabelle in Ihre Datenbank (F5 - als Skript ausführen):

Borders	Borders between countries
<i>Country1</i>	Country code
<i>Country2</i>	Country code
<i>Length</i>	Length of border in kilometer

Übung 1 (5 Punkte)

Angenommen Sie betreiben eine Internetseite, auf der Sie den Besuchern Informationen über Landreisen in Europa zur Verfügung stellen. Sie möchten Ihren Besuchern dabei u.a. 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 **Grenzuebergang** 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. Zwischenschritte dürfen auch Länder miteinbeziehen, die nicht europäisch sind. Zur Vereinfachung soll die Sicht nur diejenigen Paare enthalten, die maximal **drei** Grenzübergänge erfordern.

Sortieren Sie die Sicht zunächst nach dem Ausgangsland, dann nach Anzahl der nötigen Grenzübergänge und schließlich nach dem Zielland.

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

Übung 2 (3+2=5 Punkte)

In der Vorlesung wurde gezeigt, wie mit einer Funktion zu einem gegebenen Land alle erreichbaren Länder berechnet werden können (Kapitel 4, Seite 56). Die direkte Rückgabe einer Tabelle aus einer Funktion wird von Oracle jedoch nicht unterstützt. Betrachten Sie daher die folgende Prozedur **ErreichbarVon**, die das gleiche Ergebnis berechnet.

Hinweis: Verwenden Sie Ihre Sicht **SymBorders** von Übung 1.

```
1  -- first create the result table
2  CREATE TABLE Erreichbar(
3      nach CHAR(2),
4      anzahl NUMBER
5  );
6
7  -- then create the procedure
8  CREATE OR REPLACE
9  PROCEDURE ErreichbarVon(von CHAR) AS
10     alt INTEGER; neu INTEGER;
11  BEGIN
12     DELETE FROM Erreichbar;
13     INSERT INTO Erreichbar
14         SELECT sb.country2 AS nach, 1 AS anzahl
15         FROM SymBorders sb WHERE sb.country1 = von;
16     alt := 0;
17     SELECT COUNT(*) INTO neu FROM Erreichbar;
18     WHILE (alt <> neu) LOOP
19         alt := neu;
20         INSERT INTO Erreichbar
21             SELECT DISTINCT sb.country2 AS nach, (e.anzahl + 1) AS anzahl
22             FROM Erreichbar e, SymBorders sb
23             WHERE e.nach = sb.country1 AND sb.country2 <> von
24                 AND sb.country2 NOT IN (SELECT nach FROM Erreichbar);
25         SELECT COUNT(*) INTO neu FROM Erreichbar;
26     END LOOP;
27 END;
```

Die Prozedur wird mittels des Kommandos `EXEC ErreichbarVon('...')` ausgeführt, nachdem Sie kompiliert wurde. Das Ergebnis kann anschließend der Tabelle **Erreichbar** entnommen werden, die vor dem Erstellen der Prozedur angelegt werden muss.

a) Angenommen der Inhalt der Tabelle **Borders** sei wie folgt:

	<i>country1</i>	<i>country2</i>
	0	1
	1	2
<i>Borders :</i>	2	3
	3	4
	1	4
	4	5

Protokollieren Sie die beim Aufruf von **ErreichbarVon('0')** zu beobachtenden Änderungen der Tabelle **Erreichbar** nach jeder Iteration. Begründen Sie dies jeweils.

b) Angenommen Sie wollen zu jedem erreichbaren Land auch den konkreten Weg in Form der Folge der Grenzübergänge ermitteln. Wie können Sie dies durch Änderung der Prozedur **ErreichbarVon** erreichen? Sie können die üblichen built-in Funktionen für Zeichenketten verwenden und die Tabelle **Erreichbar** um eine geeignete Spalte erweitern.

Übung 3 (4 Punkte)

Erstellen Sie eine Prozedur **SpokenLanguage(lang CHAR)**, die beim Aufruf von `EXEC SpokenLanguage('...')` eine Tabelle mit allen Ländern liefert, in denen die Sprache (**lang**) gesprochen wird. Die Tabelle soll den Ländernamen sowie den prozentualen Anteil der Sprache enthalten. Erstellen Sie zunächst eine geeignete Tabelle, die das Ergebnis der Prozedur aufnehmen soll.

Übung 4 (3+3=6 Punkte)

Betrachten Sie die folgende Tabellendefinition:

```
CREATE TABLE T (  
  A NUMBER,  
  B NUMBER );
```

Und die folgenden beiden Integritätsbedingungen:

- (1) *Zwei Tupel einer Instanz von T, die den gleichen A-Wert haben, haben auch den gleichen B-Wert.*
- (2) *Die Summe aller Werte von A ist mindestens so groß wie die Summe aller Werte von B.*

- a) Definieren diese Bedingungen jeweils als Tabellenbedingungen.
- b) Definieren diese Bedingungen jeweils als Assertions.

Hinweis: Oracle unterstützt keine Subanfragen in Tabellenbedingungen und auch keine Assertions, wie sie im SQL-92 Standard definiert werden. Sie können Ihre Lösungen daher leider nicht in Oracle testen.