

Datenbanken und Informationssysteme (WS 2017/2018)

Jürgen Durner (3740521)
Steven Gilgin (4349366)
Felix Karg (4342014)
Gruppe 2

2. Dezember 2017

Blatt 8

Aufgabe 1

```
CREATE OR REPLACE VIEW SYMBORDERS AS
SELECT C1 AS COUNTRY1, C2 AS COUNTRY2, LENGTH FROM (
SELECT COUNTRY1 AS C1, COUNTRY2 AS C2, LENGTH FROM BORDERS
UNION
SELECT COUNTRY2 AS C1, COUNTRY1 AS C2, LENGTH FROM BORDERS);
```

```
CREATE OR REPLACE VIEW EUROPEAN AS
SELECT COUNTRY FROM ENCOMPASSES WHERE CONTINENT = 'Europe';
```

```
CREATE OR REPLACE VIEW GRENZÜBERGANG AS
SELECT DISTINCT * FROM (
-- all the countries directly next to each other
SELECT B.COUNTRY1, B.COUNTRY2, 1 AS DISTANCE
FROM SYMBORDERS B, EUROPEAN E, EUROPEAN F
WHERE B.COUNTRY1 IN E.COUNTRY AND B.COUNTRY2 IN F.COUNTRY
UNION
-- all the countries with one inbetween, excluding circles
SELECT B.COUNTRY1, C.COUNTRY2, 2 AS DISTANCE
FROM SYMBORDERS B INNER JOIN SYMBORDERS C ON B.COUNTRY2 = C.COUNTRY1,
    EUROPEAN E, EUROPEAN F
WHERE B.COUNTRY1 IN E.COUNTRY AND C.COUNTRY2 IN F.COUNTRY
```

```

        AND B.COUNTRY1 <> C.COUNTRY2
UNION
-- all the countries with two inbetween, excluding circles or partial circles
SELECT B.COUNTRY1, D.COUNTRY2, 3 AS DISTANCE
FROM SYMBORDERS B INNER JOIN
    (SYMBORDERS C INNER JOIN SYMBORDERS D ON C.COUNTRY2 = D.COUNTRY1)
    ON B.COUNTRY2 = C.COUNTRY1, EUROPEAN E, EUROPEAN F
WHERE B.COUNTRY1 IN E.COUNTRY AND D.COUNTRY2 IN F.COUNTRY
    AND B.COUNTRY1 <> C.COUNTRY2 AND B.COUNTRY1 <> D.COUNTRY2
)
WHERE COUNTRY1 <> COUNTRY2
ORDER BY COUNTRY1, DISTANCE, COUNTRY2;

```

Aufgabe 2

a)

Die Tabelle Erreichbar würde nach der ersten Iteration mit Aufruf Erreichbar-Von('0') eine 1 bei Spalte "nach" und eine 1 bei Spalte "Anzahl" ausgeben. Nach der 2. Iteration würde zu den vorigen Werten bei Spalte "nach" noch eine 2 und eine 4 und bei Spalte "Anzahl" jeweils eine 2 dazu kommen. Die 3. Iteration fügt in Spalte "nach" eine 3 und in Spalte "Anzahl" auch eine 3 hinzu. Bei der 4. und letzten Iteration würde Spalte "nach" eine 5 und Spalte "Anzahl" eine 4 hinzugefügt. Da in der Tabelle "Borders" sowohl 3 als auch 1 von "country1" ein Gegenstück bzw. "Grenzland" 4 in "country2" haben, wird Land 3 nie in Tabelle "Borders" berücksichtigt.

b)

```

-- Um einen Pfad aufzubauen reicht uns analog wie bei einer
-- Linked List das vorhergehende Element.
create table Erreichbar(
    nach CHAR(2),
    anzahl NUMBER,
    prev CHAR(2)
);

CREATE OR REPLACE PROCEDURE ErreichbarVon
    (von CHAR)
IS
    alt INTEGER;
    neu INTEGER;
BEGIN

```

```

DELETE FROM Erreichbar;
INSERT INTO Erreichbar
    SELECT sb.country2 AS nach, 1 AS anzahl,
           sb.country1 AS prev
    FROM SymBorders sb WHERE sb.country1 = von;
alt := 0;
SELECT COUNT(*) INTO neu FROM Erreichbar;
WHILE (alt <> neu) LOOP
    alt := neu;
    INSERT INTO Erreichbar
        SELECT DISTINCT sb.country2 AS nach,
                         (e.anzahl + 1) AS anzahl,
                         sb.country1 AS prev
        FROM Erreichbar e, SymBorders sb
        WHERE e.nach = sb.country1 AND sb.country2 <> von
              AND sb.country2 NOT IN (SELECT nach FROM Erreichbar);
    SELECT COUNT(*) INTO neu FROM Erreichbar;
END LOOP;
END;

```

Aufgabe 3

```

CREATE TABLE langlist(
    country CHAR(2),
    percentage NUMBER
);

CREATE OR REPLACE PROCEDURE SpokenLanguage
    (lang CHAR)
IS
BEGIN
    DELETE FROM langlist;
    INSERT INTO langlist
        SELECT country, percentage FROM LANGUAGE WHERE NAME = lang;
END SpokenLanguage;

```

Aufgabe 4

```
CREATE TABLE T (  
    A NUMBER,  
    B NUMBER,  
    CHECK (0 = SELECT COUNT(*) FROM T T1 CROSS JOIN T T2  
           WHERE T1.A = T2.A AND T1.B <> T2.B),  
    CHECK (EXISTS SELECT 1 AS ERR FROM T L WHERE SUM(L.B) > SUM(L.A))  
);
```

```
CREATE ASSERTION AssertT (  
    CHECK (0 = SELECT COUNT(*) FROM T T1 CROSS JOIN T T2  
           WHERE T1.A = T2.A AND T1.B <> T2.B),  
    CHECK (EXISTS SELECT 1 AS ERR FROM T L WHERE SUM(L.B) > SUM(L.A))  
);
```