



4. Übungsblatt (Musterlösung)

Einführung in Datenbanksysteme
Datenbanken für die Bioinformatik

Heinz Schweppe, Katharina Hahn

Aufgabe 1 (Funktionale Abhängigkeiten, Axiome von Armstrong)

Punkte

Eine Möglichkeit:

- aus $AB \rightarrow C$ und $C \rightarrow D$ folgt $AB \rightarrow D$ (Transitivität)
- aus $AB \rightarrow D$ folgt $ABE \rightarrow DE$ (Erweiterung)
- aus $A \rightarrow BE$ und $ABE \rightarrow DE$ folgt $A \rightarrow DE$ (Transitivität)
- aus $A \rightarrow DE$ und $DE \rightarrow D$ (Reflexivität) folgt $A \rightarrow D$ (Transitivität)

Weitere Möglichkeit:

- aus $A \rightarrow BE$ folgt aus $A \rightarrow B$ und $A \rightarrow E$ (Dekompositionsregel)
- aus $A \rightarrow B$ und $AB \rightarrow C$ folgt $AA \rightarrow C$ (Pseudotransitivitätsregel)
- aus $AA \rightarrow C$ und $C \rightarrow D$ folgt $AA \rightarrow D$ (Transitivität)
- aus $AA \rightarrow D$ und $AA \rightarrow A$ folgt $A \rightarrow D$ (Reflexivität)

Aufgabe 2 (Relationale Algebra)

Punkte

a)

$$(1) \quad \Pi_{Regie} (Filme \bowtie_{Titel=Titel} Programm)$$

Finde alle Regisseure von Filmen, die aktuell im Programm laufen.

$$(2) \quad \Pi_{Kino} (\sigma_{Titel=Blind\ Side} (Programm)) \bowtie_{Kino=Kino} \Pi_{Kino} (\sigma_{Titel=Date\ Night} (Programm))$$

Gib alle Kinos aus, in denen *The Blind Side* und *Date Night* aktuell im Programm laufen.

$$(3) \quad \Pi_{Kino, Adresse} (Orte \bowtie_{Kino=Titel} (\Pi_{Titel} (\sigma_{Schauspieler=Humphrey\ Bogart} (Filme)) \setminus \Pi_{Titel} (Programm)))$$

Finde alle Kinos mit Name und Adresse, die so heißen, wie ein Film mit *Humphrey Bogart*, der nicht mehr aktuell im Programm läuft (*Casablanca*).

b) Geben Sie folgende Anfragen in relationaler Algebra an:

(1) In welchen Kinos (Name und Adresse) laufen Filme mit *Sandra Bullock*?

$$\Pi_{Kino, Adresse} (Orte \bowtie_{Kino=Kino} (Programm \bowtie_{Titel=Titel} (\sigma_{Schauspieler=Sandra\ Bullock} (Filme))))$$

(2) In welchen Filmen (Titel) spielt *George Clooney* mit oder führt Regie?

$$\Pi_{Titel} ((\sigma_{Schauspieler=George\ Clooney} (Filme)) \cup (\sigma_{Regie=George\ Clooney} (Filme)))$$

(3) Welche Schauspieler haben noch nie mit dem Regisseur *James Cameron* gearbeitet?

$$\Pi_{Schauspieler} (Filme) \setminus \Pi_{Schauspieler} (\sigma_{Regie=James\ Cameron} (Filme))$$

(4) In welchen laufenden Filmen (Titel) spielen **ausschließlich** Schauspieler mit, die schon mit *Steven Spielberg* gearbeitet haben?

$$\Pi_{Titel} (Programm) \setminus \Pi_{Titel} (Filme \bowtie_{Schauspieler=Schauspieler} (\Pi_{Schauspieler} (Filme) \setminus \Pi_{Schauspieler} ((\sigma_{Schauspieler=Steven\ Spielberg} (Filme)) \cup (\sigma_{Regie=Steven\ Spielberg} (Filme)))))$$

Aufgabe 3 (SQL Anfragen)

Punkte

a) In welchen Kinos (Name und Adresse) läuft *The Hurt Locker*?

```
SELECT o.Kino, o.Adresse
FROM Orte o, Programm p
```

```
WHERE o.Kino = p.Kino
      AND p.Titel = 'The Hurt Locker'
```

- b) Gib die Liste aller Schauspieler und Regisseure des Films *The Blind Side* aus!

```
SELECT DISTINCT f.schauspieler, f.regie
FROM Film f
WHERE f.titel = 'The Blind Side'
```

--oder

```
SELECT DISTINCT f.schauspieler
FROM Film f
WHERE f.titel = 'The Blind Side'
UNION
SELECT DISTINCT f.regie
FROM Film f
WHERE f.titel = 'The Blind Side'
```

- c) In welchen Kinos (Name und Adresse) laufen Filme mit *Colin Firth*?

```
SELECT o.Kino, o.Adresse
FROM Orte o, Programm p, Film f
WHERE o.Kino = p.Kino
      AND p.Titel = f.Titel
      AND f.Schauspieler = 'Colin Firth'
```

- d) Welcher Regisseur hat mal mit Romy Schneider und James Dean zusammengearbeitet? Beachten Sie, dass die Zusammenarbeit mit den beiden Schauspielern in unterschiedlichen Filmen stattgefunden haben kann.

```
SELECT f1.Regie
FROM Filme f1, Filme f2
WHERE f1.Regie = f2.Regie
      AND f1.Schauspieler = 'Romy Schneider'
      AND f2.Schauspieler = 'James Dean'
```

- e) In welchen Filmen spielt *George Clooney* mit oder führt Regie?

```
SELECT f.Titel
FROM Filme f
WHERE f.Schauspieler = 'George Clooney'
      OR f.Regie = 'George Clooney'
```

- f) Gib die Liste der Filme aus, die um 20h laufen, zusammen mit dem Kino, in dem sie laufen!

```
SELECT p.Titel, p.Kino
FROM Programm p
WHERE p.Zeit = '20h'
-- da kein Typ für die Attribute angegeben ist, wird hier
exemplarisch ein Stringvergleich durchgeführt
```

Aufgabe 4 (SQL-Anfragen, MONDIAL Datenbank)**Punkte**

- a) Finden Sie alle Landesteile (Provinzen), durch die der *Amazonas* fließt. Geben Sie sowohl den Landesteil und das jeweiligen Nachbarland an.

```
SELECT DISTINCT p.name AS Provinzname, c2.name AS Nachbarland
FROM province p, geo_river gr, country c1, country c2, borders
b
WHERE p.name = gr.province
      AND gr.river = 'Amazonas'
      AND gr.country = c1.code
      AND ((c1.code = b.country1 AND c2.code = b.country2)
           OR (c1.code = b.country2 AND c2.code = b.country1))
ORDER BY p.name;
```

- 48 Tupel

- b) Finden Sie alle Länder (Name), die direkt an *Angola* grenzen.

```
SELECT l2.name
FROM country l1, country l2, borders b
WHERE l1.name='Angola'
      AND((b.country1=l1.code
           AND b.country2=l2.code)
          OR( b.country2=l1.code
              AND b.country1=l2.code))
```

- 4 Tupel

- c) Geben Sie eine Liste aus, in der alle Länder und ihre Nachbarn jeweils namentlich auftauchen. (Ihre Ausgaberektion hat also zwei Spalten, z.B. Land1Name und Land2Name. Das Tupel (l1, l2) kann dabei auch nochmals als (l2, l1) auftauchen).

```
SELECT c1.name, c2.name
FROM borders b, country c1, country c2
WHERE b.country1 = c1.code
      AND b.country2 = c2.code
```

- 320 Tupel

- d) Finden Sie die Namen aller Berge, die in mehr als einem Land liegen. Geben Sie den Namen des Berges und die Namen der Länder an, aufsteigend sortiert nach Bergname.

```
SELECT m.name, c.name
FROM geo_mountain gm1, geo_mountain gm2, country c, mountain m
WHERE gm2.country=c.code
      AND gm2.country != gm1.country
      AND gm2.mountain = m.name
      AND gm1.mountain = m.name
ORDER BY m.name;
```

- 66 Tupel