

Aufgabe 2

Gehen Sie dabei von dem dazugehörigen relationalen Schema aus:

Polizist : {[PersNr, DSID, Vorname, Nachname, Dienstgrad, Gehalt]}

Dienststelle : {[DSID, Name, Strasse, HausNr, Stadt]}

Fall : {[AkZ, Titel, Beschreibung, Status]}

Arbeitet_An : {[PersNr, AkZ, Von, Bis]}

Vorgesetzte : {[PersNr, PersNr, Vorgesetzter]}

```
1 CREATE TABLE Fall (
2     AkZ VARCHAR (30) PRIMARY KEY,
3     Titel VARCHAR (30),
4     Beschreibung VARCHAR (50),
5     Status VARCHAR (30)
6 );
7
8 CREATE TABLE Dienststelle (
9     DSID INTEGER PRIMARY KEY,
10    Name VARCHAR (50),
11    Strasse VARCHAR (30),
12    HausNr VARCHAR (30),
13    Stadt VARCHAR (30)
14 );
15
16 CREATE TABLE Polizist (
17     PersNr INTEGER Primary KEY,
18     DSID INTEGER REFERENCES Dienststelle(DSID),
19     Vorname VARCHAR(30),
20     Nachname VARCHAR(30),
21     Dienstgrad VARCHAR(30),
22     Gehalt INT
23 );
24
25 CREATE TABLE Arbeitet_An (
26     PersNr INTEGER REFERENCES Polizist(PersNr),
27     AkZ VARCHAR(30) REFERENCES Fall(AkZ),
28     Von DATE,
29     Bis DATE,
30     PRIMARY KEY (PersNr, AkZ)
31 );
32
33 CREATE TABLE Vorgesetzte (
34     PersNr INTEGER REFERENCES Polizist(PersNr),
35     PersNr_Vorgesetzter INTEGER REFERENCES Polizist(PersNr),
36     PRIMARY KEY (PersNr, PersNr_Vorgesetzter)
37 );
38
39 INSERT INTO Dienststelle VALUES
40 (10, 'Dienststelle München (Marienplatz)', NULL, NULL, 'München'),
41 (11, 'Dienststelle Nürnberg (Mitte)', NULL, NULL, 'Nürnberg'),
42 (12, 'Dienststelle Augsburg Ost', NULL, NULL, 'Augsburg');
43
44 INSERT INTO Polizist VALUES
45 (1, 10, 'Hans', 'Müller', 'Polizeimeister', 40000),
46 (2, 11, 'Josef', 'Fischer', 'Polizeihauptmeister', 45000),
47 (3, 10, 'Andreas', 'Schmidt', 'Polizeikommisar', 50000),
```

```

48 (4, 12, 'Stefan', 'Hoffmann', 'Polizeidirektor', 70000),
49 (5, 11, 'Sebastian', 'Wagner', 'Polizeioberkommissar', 60000);
50
51 INSERT INTO Fall VALUES
52 ('VR30932', 'Mord im Fussballstadion', 'Toter BVB-Fan', 'bearbeitet'),
53 ('XZ1508', 'Steuerhinterziehung', 'Durchsuchung eines Hauses', 'bearbeitet');
54
55 INSERT INTO Arbeitet_An
56 (PersNr, AkZ, Von, Bis)
57 VALUES
58 (1, 'VR30932', '2011-02-15', '2011-06-06'),
59 (2, 'VR30932', '2011-02-15', '2011-06-06'),
60 (2, 'XZ1508', '2012-02-13', '2012-02-14');
61
62 INSERT INTO Vorgesetzte
63 (PersNr, PersNr_Vorgesetzter)
64 VALUES
65 (1, 3),
66 (1, 4),
67 (2, 5),
68 (2, 4);

```

Gegeben sei folgendes ER-Modell, welches Polizisten, deren Dienststelle und Fälle, an denen sie arbeiten, speichert:

- (a) Formulieren Sie eine Anfrage in relationaler Algebra, welche den *Vornamen* und *Nachnamen* von Polizisten zurückgibt, deren Dienstgrad „*Polizeikommissar*“ ist und die mehr als 1500 Euro verdienen.

$$\pi_{\text{Vorname,Nachname}}(\sigma_{\text{Dienstgrad}='Polizeikommissar' \wedge \text{Gehalt} > 1500}(\text{Polizist}))$$

- (b) Formulieren Sie eine Anfrage in relationaler Algebra, welche die *Titel* der *Fälle* ausgibt, die von *Polizisten* mit dem *Nachnamen* „*Mayer*“ bearbeitet wurden.

$$\pi_{\text{Titel}}(\sigma_{\text{Nachname}='Mayer'}(\text{Polizist}) \bowtie_{\text{PersNr}} \text{Arbeitet_An} \bowtie_{\text{AkZ}} \text{Fall})$$

- (c) Formulieren Sie eine SQL-Anfrage, welche die Anzahl der Polizisten ausgibt, die in der Stadt „*München*“ arbeiten und mit Nachnamen „*Schmidt*“ heißen.

```

1 SELECT COUNT(*) AS Anzahl_Polizisten
2 FROM Polizist p, Dienststelle d
3 WHERE
4   p.DSID = d.DSID AND
5   d.Stadt = 'München' AND
6   p.Nachname = 'Schmidt';

```

anzahl_polizisten
1

(1 row)

- (d) Formulieren Sie eine SQL-Anfrage, welche die *Namen* der *Dienststellen* ausgibt, die am 14.02.2012 an dem Fall mit dem XZ1508 beteiligt waren. Ordnen Sie die Ergebnismenge alphabetisch (aufsteigend) und achten Sie darauf, dass keine Duplikate enthalten sind.

```

1 SELECT DISTINCT d.Name
2 FROM Dienststelle d, Polizist p, Arbeitet_An a
3 WHERE
4     a.AkZ = 'XZ1508' AND
5     p.PersNr = a.PersNr AND
6     p.DSID = d.DSID AND
7     a.Von <= '2012-02-14' AND
8     a.Bis >= '2012-02-14'
9 ORDER BY d.Name ASC;

```

```

          name
-----
Dienststelle Nürnberg (Mitte)
(1 row)

```

- (e) Definieren Sie die View „*Erstrebenswerte Dienstgrade*“, welche Dienstgrade enthalten soll, die in *München* mit durchschnittlich mehr als 2500 Euro besoldet werden.

```

1 CREATE VIEW ErstrebenswerteDienstgrade AS (
2     SELECT DISTINCT p.Dienstgrad
3     FROM Polizist p, Dienststelle d
4     WHERE
5         p.DSID = d.DSID AND
6         d.Stadt = 'München'
7     GROUP BY Dienstgrad
8     HAVING (AVG(Gehalt) > 2500)
9 );
10
11 SELECT * FROM ErstrebenswerteDienstgrade;

```

```

          dienstgrad
-----
Polizeikommissar
Polizeimeister
(2 rows)

```

- (f) Formulieren Sie eine SQL-Anfrage, welche *Vorname*, *Nachname* und *Dienstgrad* von *Polizisten* mit *Vorname*, *Nachname* und *Dienstgrad* ihrer *Vorgesetzten* als ein Ergebnis-Tupel ausgibt (siehe Beispiel-Tabelle). Dabei sind nur *Polizisten* zu selektieren, die an Fällen gearbeitet haben, deren Titel den Ausdruck „Fussball“ beinhalten. An *Vorgesetzte* sind keine Bedingungen gebunden. Achten Sie darauf, dass Sie nicht nur direkte Vorgesetzte, sondern alle Vorgesetzte innerhalb der Vorgesetzten-Hierarchie betrachten. Ordnen Sie ihre Ergebnismenge alphabetisch (absteigend) nach Nachnamen des Polizisten.

Hinweis: Sie dürfen Views verwenden, um Teilergebnisse auszudrücken.

Vorarbeiten:

```

1 SELECT p.Vorname, p.Nachname
2 FROM Polizist p, Arbeitet_An a, Fall f
3 WHERE
4     p.PersNr = a.PersNr AND

```

```

5      a.AkZ = f.Akz AND
6      f.Titel LIKE '%Fussball%';

```

```

vorname | nachname
-----+-----
Hans     | Müller
Josef    | Fischer
(2 rows)

```

Lösungsansatz 1

```

1  WITH RECURSIVE Fussball_Vorgesetzte (PersNr, VN, NN, DG, PN_VG, VN_VG,
2  ↪  NN_VG, DG_VG) AS
3  (
4      SELECT
5          p1.PersNr,
6          p1.Vorname AS VN,
7          p1.Nachname AS NN,
8          p1.Dienstgrad AS DG,
9          p2.PersNr AS PN_VG,
10         p2.Vorname AS VN_VG,
11         p2.Nachname AS NN_VG,
12         p2.Dienstgrad AS DG_VG
13     FROM Polizist p1, Fall f, Arbeitet_An a, Vorgesetzte v
14     LEFT JOIN Polizist p2 ON v.PersNr_Vorgesetzter = p2.PersNr
15     WHERE
16         p1.PersNr = a.PersNr AND
17         a.AkZ = f.AkZ AND
18         f.Titel LIKE '%Fussball%' AND
19         p1.PersNr = v.PersNr
20
21     UNION ALL
22
23     SELECT
24         m.PersNr,
25         m.VN AS VN,
26         m.NN AS NN,
27         m.DG AS DG,
28         p.PersNr AS PN_VG,
29         p.Vorname AS VN_VG,
30         p.Nachname AS NN_VG,
31         p.Dienstgrad AS DG_VG
32     FROM Fussball_Vorgesetzte m, Vorgesetzte v
33     LEFT JOIN Polizist p ON v.PersNr_Vorgesetzter = p.PersNr
34     WHERE m.PN_VG = v.PersNr
35 )
36
37 SELECT VN, NN, DG, VN_VG, NN_VG, DG_VG
38 FROM Fussball_Vorgesetzte
39 ORDER BY NN DESC;

```

```

      vn | nn | dg | vn_vg | nn_vg | dg_vg
-----+-----+-----+-----+-----+-----
+-----+
Hans | Müller | Polizeimeister | Andreas | Schmidt | Polizeikommissar
Hans | Müller | Polizeimeister | Stefan | Hoffmann | Polizeidirektor

```

Josef	Fischer	Polizeihauptmeister	Stefan	Hoffmann	Polizeidirektor
Josef	Fischer	Polizeihauptmeister	Sebastian	Wagner	Polizeioberkommissar

(4 rows)

Lösungsansatz 2

```

1 CREATE VIEW naechste_Vorgesetzte AS
2 SELECT
3     p.PersNR,
4     p.Vorname,
5     p.Nachname,
6     p.Dienstgrad,
7     v.PersNr_Vorgesetzter AS Vorgesetzter
8 FROM Polizist p LEFT JOIN Vorgesetzte v
9 ON p.PersNr = v.PersNr;
10
11 WITH RECURSIVE Fussball_Vorgesetzte (VN, NN, DG, VN_VG, NN_VG, DG_VG)
12 ↪ AS (
13     SELECT
14         x.Vorname AS VN,
15         x.Nachname AS NN,
16         x.Dienstgrad AS DG,
17         y.Vorname AS VN_VG,
18         y.Nachname AS NN_VG,
19         y.Dienstgrad AS DG_VG
20 FROM naechste_Vorgesetzte x, Fall f, Arbeitet_An a,
21 naechste_Vorgesetzte y
22 WHERE
23     f.Titel LIKE '%Fussball%' AND
24     f.AkZ = a.AkZ AND
25     x.PersNr = a.PersNr AND
26     x.Vorgesetzter = y.PersNr
27 UNION ALL
28     SELECT
29         a.Vorname AS VN,
30         a.Nachname AS NN,
31         a.Dienstgrad AS DG,
32         Vorname AS VN_VG,
33         Nachname AS NN_VG,
34         Dienstgrad AS DG_VG
35 FROM naechste_Vorgesetzte a INNER JOIN Fussball_Vorgesetzte
36 ON a.Vorgesetzter = PersNr
37 )
38 SELECT *
39 FROM Fussball_Vorgesetzte;

```

vn	nn	dg	vn_vg	nn_vg	dg_vg
Hans	Müller	Polizeimeister	Andreas	Schmidt	Polizeikommissar
Hans	Müller	Polizeimeister	Stefan	Hoffmann	Polizeidirektor
Josef	Fischer	Polizeihauptmeister	Stefan	Hoffmann	Polizeidirektor
Josef	Fischer	Polizeihauptmeister	Sebastian	Wagner	Polizeioberkommissar

(4 rows)