

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]}

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.Name = 'München' AND
6   p.Nachname = 'Schmidt';
```

- (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 >= '14.02.2012' AND
8   a.Bis <= '14.02.2012' AND
9 ORDER BY d.Name ASC;
```

- (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 d.Dienstgrad
3     FROM Polizist p, Dienststelle d
4     WHERE
5         p.DSID = d.DSID AND
6         d.Dienststelle = 'München';
7     GROUP BY Dienstgrad
8     HAVING (AVG(Gehalt) > 2500);
9 )

```

- (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.

```

1 WITH RECURSIVE meus (PersNr, VN, NN, DG, VN_VG, NN_VG, DG_VG)
2     SELECT PersNr, Vorname AS VN, Nachname AS NN, Dienstgrad AS DG_VG
3     FROM Polizist p1, Fall f, Arbeitet_An a, Polizist p2
4     WHERE
5         p1.PersNr = a.PersNr AND
6         a.AkZ = f.Akz AND
7         f.Titel CONTAINS "Fussball" AND
8         p1.Vorgesetzter = p2.PersNr
9     UNION ALL
10    SELECT m.PersNr, m.Vorname, m.Nachname, p.Vorname AS VN,
11           p.Nachname AS NN, p.Dienstgrad AS DG_VG
12    FROM meus m, Polizist p, Vorgesetzte v
13    WHERE m.PersNr = v.PersNr AND p.PersNr_Vorgesetzter = p.PersNr;
14
15 SELECT VN, NN, NN, DG, VN_VG, NN_VG, DG_VG
16 FROM meus
17 ORDER BY NN DESC;

```

```

1 CREATE TABLE Dienststelle(
2     DSID INT PRIMARY KEY,
3     Name VARCHAR (30),
4     Strasse VARCHAR (30),
5     HausNr VARCHAR (30),
6     Stadt VARCHAR (30));
7
8 INSERT INTO Dienststelle VALUES
9 (10, 'Stadt Mitte', NULL, NULL, 'Muenchen'),
10 (11, 'Stadt Mitte', NULL, NULL, 'Nuernberg'),
11 (12, 'Stadt Mitte', NULL, NULL, 'Augsburg');
12
13 CREATE TABLE Polizist(
14     PersNr INT Primary KEY,
15     DSID INT REFERENCES Dienststelle(DSID),
16     Vorname VARCHAR (30),
17     Nachname VARCHAR (30),
18     Dienstgrad VARCHAR (30),
19     Gehalt INT);

```

```

20
21 INSERT INTO Polizist
22 VALUES
23 (1, 10, 'Hans', 'Müller', 'Polizeimeister', 40000),
24 (2, 11, 'Josef', 'Fischer', 'Polizeihauptmeister', 45000),
25 (3, 10, 'Andreas', 'Schmidt', 'Polizeikommissar', 50000),
26 (4, 12, 'Stefan', 'Hoffmann', 'Polizeidirektor', 70000),
27 (5, 11, 'Sebastian', 'Wagner', 'Polizeioberkommissar', 60000);
28
29 CREATE TABLE Fall(
30 AkZ VARCHAR (30) PRIMARY KEY,
31 Titel VARCHAR (30),
32 Beschreibung VARCHAR (50),
33 Status VARCHAR (30));
34
35 INSERT INTO Fall
36 VALUES
37 ('VR30932', 'Mord im Fussballstadion', 'Toter BVB-Fan', 'bearbeitet');
38
39 CREATE TABLE Arbeitet_An(
40 PersNr INT References Polizist(PersNr),
41 AkZ VARCHAR (30) References Fall(AkZ),
42 Von DATE,
43 Bis DATE,
44 PRIMARY KEY (PersNr, AkZ));
45
46 INSERT INTO Arbeitet_An VALUES
47 (1, 'VR30932', '2012-02-15', '2012-04-12'),
48 (2, 'VR30932', '2012-02-15', '2012-04-12');
49
50 CREATE TABLE Vorgesetzte(
51 PersNr INT References Polizist(PersNr),
52 PersNr_Vorgesetzter INT References Polizist(PersNr),
53 PRIMARY KEY (PersNr, PersNr_Vorgesetzter));
54
55 INSERT INTO Vorgesetzte VALUES
56 (1,3),
57 (1,4),
58 (2,5),
59 (2,4);

```