

## 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', 'Polizeikommissar', 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 VALUES
56   (1, 'VR30932', '2012-02-15', '2012-04-12'),
57   (2, 'XZ1508', '2012-02-13', '2012-02-15');
58
59 INSERT INTO Vorgesetzte VALUES
60   (1,3),
61   (1,4),
62   (2,5),
63   (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.

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

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

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

- (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';

```

- (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;

```

- (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 );
```

- (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 meins (PersNr, VN, NN, DG, VN_VG, NN_VG, DG_VG) AS (  
2     SELECT p1.PersNr, p1.Vorname AS VN, p1.Nachname AS VN, p1.Dienstgrad  
3     ↪ AS DG_VG  
4     FROM Polizist p1, Fall f, Arbeitet_An a, Polizist p2, Vorgesetzte v  
5     WHERE  
6         p1.PersNr = a.PersNr AND  
7         a.AkZ = f.Akz AND  
8         f.Titel LIKE '%Fussball%' AND  
9         v.PersNr_Vorgesetzter = p2.PersNr  
10    UNION ALL  
11    SELECT m.PersNr, m.Vorname, m.Nachname, p.Vorname AS VN,  
12    p.Nachname AS VN, p.Dienstgrad AS DG_VG  
13    FROM meins m, Polizist p, Vorgesetzte v  
14    WHERE m.PersNr = v.PersNr AND p.PersNr_Vorgesetzter = p.PersNr  
15 )  
16 SELECT VN, NN, NN, DG, VN_VG, NN_VG, DG_VG  
17 FROM meins  
18 ORDER BY NN DESC;
```