Einzelprüfung „Datenbanksysteme / Softwaretechnologie (vertieft)“

Einzelprüfungsnummer 66116 / 2016 / Frühjahr

Thema 1 / Teilaufgabe 1 / Aufgabe 2 (*Polizei*)

Stichwörter: Entity-Relation-Modell, SQL mit Übungsdatenbank, Relationale Algebra, SQL, VIEW, WITH, UNION

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 ]}
```

Additum: Übungsdatenbank

```
CREATE TABLE Fall (
  AkZ VARCHAR (30) PRIMARY KEY,
  Titel VARCHAR (30),
  Beschreibung VARCHAR (50),
  Status VARCHAR (30)
);

CREATE TABLE Dienststelle (
  DSID INTEGER PRIMARY KEY,
  Name VARCHAR (50),
  Strasse VARCHAR (30),
  HausNr VARCHAR (30),
  Stadt VARCHAR (30)
);

CREATE TABLE Polizist (
  PersNr INTEGER Primary KEY,
  DSID INTEGER REFERENCES Dienststelle(DSID),
  Vorname VARCHAR(30),
  Nachname VARCHAR(30),
  Dienstgrad VARCHAR(30),
  Gehalt INT
);

CREATE TABLE Arbeitet_An (
  PersNr INTEGER REFERENCES Polizist(PersNr),
  AkZ VARCHAR(30) REFERENCES Fall(AkZ),
  Von DATE,
  Bis DATE,
  PRIMARY KEY (PersNr, AkZ)
```

```
);

CREATE TABLE Vorgesetzte (
  PersNr INTEGER REFERENCES Polizist(PersNr),
  PersNr_Vorgesetzter INTEGER REFERENCES Polizist(PersNr),
  PRIMARY KEY (PersNr, PersNr_Vorgesetzter)
);

INSERT INTO Dienststelle VALUES
  (10, 'Dienstelle München (Marienplatz)', NULL, NULL, 'München'),
  (11, 'Dienststelle Nürnberg (Mitte)', NULL, NULL, 'Nürnberg'),
  (12, 'Dieststelle Augsburg Ost', NULL, NULL, 'Augsburg');

INSERT INTO Polizist VALUES
  (1, 10, 'Hans', 'Müller', 'Polizeimeister', 40000),
  (2, 11, 'Josef', 'Fischer', 'Polizeihauptmeister', 45000),
  (3, 10, 'Andreas', 'Schmidt', 'Polizeikommisar', 50000),
  (4, 12, 'Stefan', 'Hoffmann', 'Polizeidirektor', 70000),
  (5, 11, 'Sebastian', 'Wagner', 'Polizeioberkommisar', 60000);

INSERT INTO Fall VALUES
  ('VR30932', 'Mord im Fussballstadion', 'Toter BVB-Fan', 'bearbeitet'),
  ('XZ1508', 'Steuerhinterziehung', 'Durchsuchung eines Hauses', 'bearbeitet');

INSERT INTO Arbeitet_An
  (PersNr, AkZ, Von, Bis)
VALUES
  (1, 'VR30932', '2011-02-15', '2011-06-06'),
  (2, 'VR30932', '2011-02-15', '2011-06-06'),
  (2, 'XZ1508', '2012-02-13', '2012-02-14');

INSERT INTO Vorgesetzte
  (PersNr, PersNr_Vorgesetzter)
VALUES
  (1, 3),
  (1, 4),
  (2, 5),
  (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.

Lösungsvorschlag

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

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

Lösungsvorschlag

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

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

Lösungsvorschlag

```
SELECT COUNT(*) AS Anzahl_Polizisten
FROM Polizist p, Dienststelle d
WHERE
  p.DSID = d.DSID AND
  d.Stadt = 'München' AND
  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.

Lösungsvorschlag

```
SELECT DISTINCT d.Name
FROM Dienststelle d, Polizist p, Arbeitet_An a
WHERE
  a.AkZ = 'XZ1508' AND
  p.PersNr = a.PersNr AND
  p.DSID = d.DSID AND
  a.Von <= '2012-02-14' AND
  a.Bis >= '2012-02-14'
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.

Lösungsvorschlag

```
CREATE VIEW ErstrebenswerteDienstgrade AS (
  SELECT DISTINCT p.Dienstgrad
  FROM Polizist p, Dienststelle d
  WHERE
    p.DSID = d.DSID AND
    d.Stadt = 'München'
  GROUP BY Dienstgrad
```

```
  HAVING (AVG(Gehalt) > 2500)
);

SELECT * FROM ErstrebenswerteDienstgrade;


   dienstgrad
-----------------
 Polizeikommisar
 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.

Lösungsvorschlag

Vorarbeiten:

```
  SELECT p.Vorname, p.Nachname
  FROM Polizist p, Arbeitet_An a, Fall f
  WHERE
    p.PersNr = a.PersNr AND
    a.AkZ = f.Akz AND
    f.Titel LIKE '%Fussball%';
```

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

Lösungsansatz 1

```
WITH RECURSIVE Fussball_Vorgesetzte (PersNr, VN, NN, DG, PN_VG, VN_VG, NN_VG,
↪  DG_VG) AS
(
  SELECT
    p1.PersNr,
    p1.Vorname AS VN,
    p1.Nachname AS NN,
    p1.Dienstgrad AS DG,
    p2.PersNr AS PN_VG,
    p2.Vorname AS VN_VG,
    p2.Nachname AS NN_VG,
```

```
    p2.Dienstgrad AS DG_VG
  FROM Polizist p1, Fall f, Arbeitet_An a, Vorgesetzte v
  LEFT JOIN Polizist p2 ON v.PersNr_Vorgesetzter = p2.PersNr
  WHERE
    p1.PersNr = a.PersNr AND
    a.AkZ = f.Akz AND
    f.Titel LIKE '%Fussball%' AND
    p1.PersNr = v.PersNr

  UNION ALL

  SELECT
    m.PersNr,
    m.VN AS VN,
    m.NN AS NN,
    m.DG AS DG,
    p.PersNr AS PN_VG,
    p.Vorname AS VN_VG,
    p.Nachname AS NN_VG,
    p.Dienstgrad AS DG_VG
  FROM Fussball_Vorgesetzte m, Vorgesetzte v
  LEFT JOIN Polizist p ON v.PersNr_Vorgesetzter = p.PersNr
  WHERE m.PN_VG = v.PersNr
)

SELECT VN, NN, DG, VN_VG, NN_VG, DG_VG
FROM Fussball_Vorgesetzte
ORDER BY NN DESC;
```

```
  vn   |   nn    |         dg          |   vn_vg   |  nn_vg   |        dg_vg
-------+---------+---------------------+-----------+----------+-------------------
 Hans  | Müller  | Polizeimeister      | Andreas   | Schmidt  | Polizeikommisar
 Hans  | Müller  | Polizeimeister      | Stefan    | Hoffmann | Polizeidirektor
 Josef | Fischer | Polizeihauptmeister | Stefan    | Hoffmann | Polizeidirektor
 Josef | Fischer | Polizeihauptmeister | Sebastian | Wagner   | Polizeioberkommis
(4 rows)
```

Lösungsansatz 2

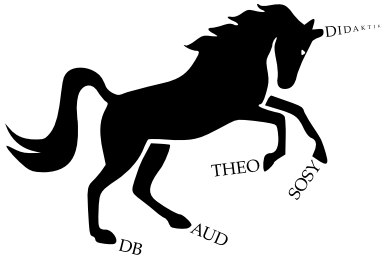
```
CREATE VIEW naechste_Vorgesetzte AS
  SELECT
    p.PersNR,
    p.Vorname,
    p.Nachname,
    p.Dienstgrad,
    v.PersNr_Vorgesetzter AS Vorgesetzter
  FROM Polizist p LEFT JOIN Vorgesetzte v
  ON p.PersNr = v.PersNr;

WITH RECURSIVE Fussball_Vorgesetzte (VN, NN, DG, VN_VG, NN_VG, DG_VG) AS (
  SELECT
```

```
    x.Vorname AS VN,
    x.Nachname AS NN,
    x.Dienstgrad AS DG,
    y.Vorname AS VN_VG,
    y.Nachname AS NN_VG,
    y.Dienstgrad AS DG_VG
  FROM naechste_Vorgesetzte x, Fall f, Arbeitet_An a,
  naechste_Vorgesetzte y
  WHERE
    f.Titel LIKE '%Fussball%' AND
    f.AkZ = a.AkZ AND
    x.PersNr = a.PersNr AND
    x.Vorgesetzter = y.PersNr
  UNION ALL
  SELECT
    a.Vorname AS VN,
    a.Nachname AS NN,
    a.Dienstgrad AS DB,
    Vorname AS VN_VG,
    Nachname AS NN_VG,
    Dienstgrad AS DG_VG
  FROM naechste_Vorgesetzte a INNER JOIN Fussball_Vorgesetzte
  ON a.Vorgesetzter = PersNr
)

SELECT *
FROM Fussball_Vorgesetzte;
```

```
  vn   |   nn    |         dg          |   vn_vg   |  nn_vg   |        dg_vg
-------+---------+---------------------+-----------+----------+--------------------
 Hans  | Müller  | Polizeimeister      | Andreas   | Schmidt  | Polizeikommisar
 Hans  | Müller  | Polizeimeister      | Stefan    | Hoffmann | Polizeidirektor
 Josef | Fischer | Polizeihauptmeister | Stefan    | Hoffmann | Polizeidirektor
 Josef | Fischer | Polizeihauptmeister | Sebastian | Wagner   | Polizeioberkommis
(4 rows)
```



Die Bschlangaul-Sammlung
Hermine Bschlangauland Friends

Eine freie Aufgabensammlung mit Lösungen von Studierenden für Studierende zur Vorbereitung auf die 1. Staatsexamensprüfungen des Lehramts Informatik in Bayern.

Hilf mit! Die Hermine schafft das nicht allein! Das ist ein Community-Projekt! Verbesserungsvorschläge, Fehlerkorrekturen, weitere Lösungen sind herzlich willkommen - egal wie - per Pull-Request oder per E-Mail an hermine.bschlangaul@gmx.net.Der TEX-Quelltext dieses Dokuments kann unter folgender URL aufgerufen werden: `https://github.com/bschlangaul-sammlung/examens-aufgaben/blob/main/Staatsexamen/66116/2016/03/Thema-1/Teilaufgabe-1/Aufgabe-2.tex`