

Staatsexamen 66116 / 2016 / Frühjahr

Thema 2 / Teilaufgabe 1 / Aufgabe 2

(Schulverwaltung)

Stichwörter: SQL mit Übungsdatenbank, SQL, CREATE TABLE, CONSTRAINT, ALTER TABLE, GROUP BY, HAVING

Gegeben sei der folgende Ausschnitt aus dem Schema einer Schulverwaltung:

```
Person : {[
  ID : INTEGER,
  Name : VARCHAR(255),
  Wohnort : VARCHAR(255),
  Typ : CHAR(1)
]}
```

```
Unterricht : {[
  Klassenbezeichnung : VARCHAR(20),
  Schuljahr : INTEGER,
  Lehrer : INTEGER,
  Fach : VARCHAR(100)
]}
```

```
Klasse : {[
  Klassenbezeichnung : VARCHAR(20),
  Schuljahr : INTEGER,
  Klassenlehrer : INTEGER
]}
```

```
Klassenverband : {[
  Schüler : INTEGER,
  Klassenbezeichnung : VARCHAR(20),
  Schuljahr : INTEGER
]}
```

```
1 CREATE TABLE Person (
2   ID INTEGER PRIMARY KEY,
3   Name VARCHAR(255),
4   Wohnort VARCHAR(255),
5   Typ CHAR(1) CHECK(Typ in ('S', 'L'))
6 );
7
8 CREATE TABLE Klasse (
9   Klassenbezeichnung VARCHAR(20),
10  Schuljahr INTEGER,
11  Klassenlehrer INTEGER REFERENCES Person(ID),
12  PRIMARY KEY (Klassenbezeichnung, Schuljahr)
13 );
14
```

```

15 CREATE TABLE Klassenverband (
16     Schüler INTEGER REFERENCES Person(ID),
17     Klassenbezeichnung VARCHAR(20),
18     Schuljahr INTEGER,
19     PRIMARY KEY (Schüler, Schuljahr)
20 );
21
22 CREATE TABLE Unterricht (
23     Klassenbezeichnung VARCHAR(20),
24     Schuljahr INTEGER,
25     Lehrer INTEGER REFERENCES Person(ID),
26     Fach VARCHAR(100),
27     CONSTRAINT Unterricht_PK
28     PRIMARY KEY (Klassenbezeichnung, Schuljahr, Lehrer, Fach)
29 );
30
31 INSERT INTO Person VALUES
32 (1, 'Lehrer Ludwig', 'München', 'L'),
33 (2, 'Schüler Max', 'München', 'S'),
34 (3, 'Schülerin Maria', 'München', 'S'),
35 (4, 'Schülerin Eva', 'Starnberg', 'S'),
36 (5, 'Lehrerin Walter', 'München', 'L'),
37 (6, 'Schüler Karl', 'München', 'S');
38
39 INSERT INTO Klasse VALUES
40 ('1a', 2015, 1),
41 ('1a', 2014, 1),
42 ('1b', 2015, 5);
43
44 INSERT INTO Klassenverband VALUES
45 (2, '1a', 2015),
46 (3, '1a', 2015),
47 (4, '1b', 2015),
48 (6, '1a', 2015),
49 (6, '1a', 2014);

```

Hierbei enthält die Tabelle *Person* Informationen über Lehrer (Typ 'L') und Schüler (Typ 'S'); andere Werte für Typ sind nicht zulässig. *Klasse* beschreibt die Klassen, die in jedem Schuljahr gebildet wurden, zusammen mit ihrem Klassenlehrer. In *Unterricht* wird abgelegt, welcher Lehrer welches Fach in welcher Klasse unterrichtet; es ist möglich, dass derselbe Lehrer mehr als ein Fach in einer Klasse unterrichtet. *Klassenverband* beschreibt die Zuordnung der Schüler zu den Klassen.

- (a) Schreiben Sie eine SQL-Anweisung, die die Tabelle *Unterricht* mit allen ihren Constraints (einschließlich Fremdschlüsselconstraints) anlegt.

```

1 CREATE TABLE IF NOT EXISTS Person(
2     ID INTEGER PRIMARY KEY,
3     Name VARCHAR(255),
4     Wohnort VARCHAR(255),
5     Typ CHAR(1) CHECK(Typ in ('S', 'L'))
6 );
7

```

```

8 CREATE TABLE IF NOT EXISTS Klasse(
9     Klassenbezeichnung VARCHAR(20),
10    Schuljahr INTEGER,
11    Klassenlehrer INTEGER REFERENCES Person(ID),
12    PRIMARY KEY (Klassenbezeichnung, Schuljahr)
13 );
14
15 CREATE TABLE IF NOT EXISTS Unterricht (
16    Klassenbezeichnung VARCHAR(20) REFERENCES Klasse(Klassenbezeichnung),
17    Schuljahr INTEGER REFERENCES Klasse(Schuljahr),
18    Lehrer INTEGER REFERENCES Person(ID),
19    Fach VARCHAR(100),
20    CONSTRAINT Unterricht_PK
21    PRIMARY KEY (Klassenbezeichnung, Schuljahr, Lehrer, Fach)
22 );

```

- (b) Definieren Sie ein geeignetes Constraint, das sicherstellt, dass nur zulässige Werte im Attribut Typ der (bereits angelegten) Tabelle *Person* eingefügt werden können.

Ich habe REFERENCES bei Unterricht Schuljahr vergessen, die referenzierten Tabellen Person und Klasse wurden in der Musterlösung auch nicht angelegt.

```

1 ALTER TABLE Person
2     ADD CONSTRAINT TypLS
3     CHECK(Typ IN ('S', 'L'));

```

- (c) Schreiben Sie eine SQL-Anweisung, die die Bezeichnung der Klassen bestimmt, die im Schuljahr 2015 die meisten Schüler haben.

Falsch: ORDER BY Anzahl;. DESC vergessen.

```

1 SELECT k.Klassenbezeichnung, COUNT(*) AS Anzahl
2 FROM Klasse k, Klassenverband v
3 WHERE
4     k.Schuljahr = 2015 AND
5     k.Klassenbezeichnung = v.Klassenbezeichnung
6 GROUP BY k.Klassenbezeichnung
7 ORDER BY COUNT(*) DESC;

```

- (d) Schreiben Sie eine SQL-Anweisung, die die Namen aller Lehrer bestimmt, die nur Schüler aus ihrem Wohnort unterrichtet haben.

```

1 SELECT DISTINCT l.Name
2 FROM Person l
3 WHERE NOT EXISTS(
4     SELECT DISTINCT *
5     FROM Unterricht u, Klassenverband v, Person s
6     WHERE
7         u.Lehrer = l.ID AND
8         u.Klassenbezeichnung = v.Klassenbezeichnung AND

```

```

9      v.Schüler = s.ID AND
10     l.Wohnort != s.Wohnort
11 );

```

- (e) Schreiben Sie eine SQL-Anweisung, die die Namen aller Schüler bestimmt, die immer den gleichen Klassenlehrer hatten.

```

1  SELECT s.Name
2  FROM Person s, Klasse k, Klassenverband v
3  WHERE
4      v.Klassenbezeichnung = k.Klassenbezeichnung AND
5      k.Schuljahr = v.Schuljahr AND
6      v.Schüler = s.ID
7  GROUP BY s.ID, s.Name
8  HAVING COUNT(*) = 1

```

- (f) Schreiben Sie eine SQL-Anweisung, die alle Paare von Schülern bestimmt, die mindestens einmal in der gleichen Klasse waren. Es genügt dabei, wenn Sie die ID der Schüler bestimmen.

```

1  SELECT DISTINCT s1.ID, s2.ID
2  FROM Klassenverband s1, Klassenverband s2
3  WHERE
4      s1.Schuljahr = s2.Schuljahr AND
5      s1.Schueler <> s2.Schueler AND
6      s1.Klassenbezeichnung = s2.Klassenbezeichnung;

```

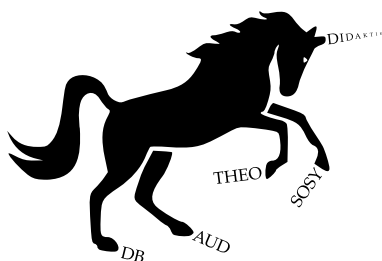
- (g) Formulieren Sie eine Anfrage in der relationalen Algebra, die die ID aller Schüler bestimmt, die mindestens einmal von „Ludwig Lehrer“ unterrichtet wurden.

$$\pi_{\text{Schüler}} \left(\begin{array}{c} \sigma_{\text{Name}='Ludwig Lehrer'}(\text{Person}) \\ \bowtie_{\text{Person.ID}=\text{Unterricht.Lehrer}} \\ \text{Unterricht} \\ \bowtie_{\text{Unterricht.Klassenbezeichnung}=\text{Klassenverband.Klassenbezeichnung}} \\ \text{Klassenverband} \end{array} \right)$$

- (h) Formulieren Sie eine Anfrage in der relationalen Algebra, die Namen und ID der Schüler bestimmt, die von allen Lehrern unterrichtet wurden.

$$\pi_{\text{Name, ID}} \left(\left(\pi_{\text{Lehrer, Schueler}} (\text{Unterricht} \bowtie \text{Klassenverband}) \div \pi_{\text{ID}} (\sigma_{\text{Typ}='L'} (\text{Person})) \right) \bowtie \text{Person} \right)$$

Beachten Sie bei der Formulierung der SQL-Anfragen, dass die Ergebnisrelationen keine Duplikate enthalten dürfen. Sie dürfen geeignete Views definieren.



Die Bschlangaul-Sammlung

Hermine Bschlangaul and Friends

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



Diese Materialsammlung unterliegt den Bestimmungen der Creative Commons Namensnennung-Nicht kommerziell-Share Alike 4.0 International-Lizenz.

Hilf mit! Die Hermine schafft das nicht alleine! 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 \LaTeX -Quelltext dieses Dokuments kann unter folgender URL aufgerufen werden: <https://github.com/hbschlang/lehramt-informatik/blob/main/Staatsexamen/66116/2016/09/Thema-2/Teilaufgabe-1/Aufgabe-2.tex>