

2. SQL und relationale Algebra

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
]}
```

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.

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

```
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 Unterricht (
16     Klassenbezeichnung VARCHAR(20) REFERENCES
17     ↳ Klasse(Klassenbezeichnung),
18     Schuljahr INTEGER REFERENCES Klasse(Schuljahr),
19     Lehrer INTEGER REFERENCES Person(ID),
20     Fach VARCHAR(100),
21     CONSTRAINT Unterricht_PK
22     PRIMARY KEY (Klassenbezeichnung, Schuljahr, Lehrer, Fach)
23 );

```

- (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     s.ID = v.Schueler AND
5     v.Klassenbezeichnung = k.Klassenbezeichnung AND
6     v.Schuler = s.ID
7 GROUP BY k.Klassenlehrer, v.Schueler
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.

Ich habe Schuljahr zum joinen vergessen.

```

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.

$$\begin{aligned}
 &\pi_{\text{Schüler}} (\\
 &\quad \sigma_{\text{Name}='Ludwig Lehrer'}(\text{Person}) \\
 &\quad \bowtie_{\text{Person.ID}=\text{Unterricht.Lehrer}} \\
 &\quad \text{Unterricht} \\
 &\quad \bowtie_{\text{Unterricht.Klassenbezeichnung}=\text{Klassenverband.Klassenbezeichnung}} \\
 &\quad \text{Klassenverband} \\
 &)
 \end{aligned}$$

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

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

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