

Staatsexamen 66116 / 2020 / Frühjahr

Thema 1 / Teilaufgabe 2 / Aufgabe 8

(Universitätsschema)

Stichwörter: SQL, SQL mit Übungsdatenbank

Gegeben sei das Universitätsschema. Formulieren Sie folgende Anfragen in SQL-92:

```
1 CREATE TABLE Studierende (
2     MatrNr INTEGER PRIMARY KEY,
3     Name VARCHAR(15),
4     Semester INTEGER
5 );
6
7 CREATE TABLE Professoren (
8     PersNr INTEGER PRIMARY KEY,
9     Name VARCHAR(30),
10    Rang VARCHAR(30),
11    Raum INTEGER
12 );
13
14 CREATE TABLE Assistenten (
15     PersNr INTEGER PRIMARY KEY,
16     Name VARCHAR(20),
17     Fachgebiet VARCHAR(30),
18     Boss INTEGER,
19     FOREIGN KEY (Boss) REFERENCES Professoren(PersNr)
20 );
21
22 CREATE TABLE Vorlesungen (
23     VorlNr INTEGER PRIMARY KEY,
24     Titel VARCHAR(30),
25     SWS INTEGER,
26     gelesenVon INTEGER,
27     FOREIGN KEY (gelesenVon) REFERENCES Professoren(PersNr)
28 );
29
30 CREATE TABLE hören (
31     MatrNr INTEGER,
32     VorlNr INTEGER,
33     PRIMARY KEY (MatrNr, VorlNr),
34     FOREIGN KEY (MatrNr) REFERENCES Studierende(MatNr),
35     FOREIGN KEY (VorlNr) REFERENCES Vorlesungen(VorlNr)
36 );
37
38 CREATE TABLE prüfen (
39     MatrNr INTEGER,
40     VorlNr INTEGER,
41     PersNr INTEGER,
42     Note INTEGER,
43     PRIMARY KEY (MatrNr, VorlNr, PersNr),
44     FOREIGN KEY (MatrNr) REFERENCES Studierende(MatNr),
45     FOREIGN KEY (VorlNr) REFERENCES Vorlesungen(VorlNr),
46     FOREIGN KEY (PersNr) REFERENCES Professoren(PersNr)
47 );
48
49 CREATE TABLE voraussetzen (
50     Vorgänger INTEGER,
51     Nachfolger INTEGER,
52     PRIMARY KEY (Vorgänger, Nachfolger),
53     FOREIGN KEY (Vorgänger) REFERENCES Vorlesungen(VorlNr),
54     FOREIGN KEY (Nachfolger) REFERENCES Vorlesungen(VorlNr)
55 );
56
57 INSERT INTO Studierende
58 (MatrNr, Name, Semester)
59 VALUES
60 (24002, 'Xenokrates', 18),
61 (25403, 'Jonas', 12),
```

```

62  (26120, 'Fichte', 10),
63  (26830, 'Aristoxenos', 8),
64  (27550, 'Schopenhauer', 6),
65  (28106, 'Carnap', 3),
66  (29120, 'Theophrastos', 2),
67  (29555, 'Feuerbach', 2);
68
69  INSERT INTO Professoren
70  (PersNr, Name, Rang, Raum)
71  VALUES
72  (2125, 'Sokrates', 'C4', 226),
73  (2126, 'Russel', 'C4', 226),
74  (2127, 'Kopernikus', 'C3', 226),
75  (2133, 'Popper', 'C3', 226),
76  (2134, 'Augustinus', 'C3', 226),
77  (2136, 'Curie', 'C4', 226),
78  (2137, 'Kant', 'C4', 226);
79
80  INSERT INTO Assistenten
81  (PersNr, Name, Fachgebiet, Boss)
82  VALUES
83  (3002, 'Platon', 'Ideenlehre', 2125),
84  (3003, 'Aristoteles', 'Syllogistik', 2125),
85  (3004, 'Wittgenstein', 'Sprachtheorie', 2126),
86  (3005, 'Rhetikus', 'Planetenbewegung', 2127),
87  (3006, 'Newton', 'Kaplarsche Gesetze', 2127),
88  (3007, 'Spinosä', 'Gott und Natur', 2134);
89
90  INSERT INTO Vorlesungen
91  (VorlNr, Titel, SWS, gelesenVon)
92  VALUES
93  (4052, 'Logik', 4, 2125),
94  (4630, 'Die 3 Kritiken', 4, 2137),
95  (5001, 'Grundzüge', 4, 2137),
96  (5022, 'Glaube und Wissen', 2, 2134),
97  (5041, 'Ethik', 4, 2125),
98  (5043, 'Erkenntnistheorie', 3, 2126),
99  (5049, 'Mäeutik', 2, 2125),
100  (5052, 'Wissenschaftstheorie', 3, 2126),
101  (5216, 'Bioethik', 2, 2126),
102  (5259, 'Der Wiener Kreis', 2, 2133);
103
104  INSERT INTO hören
105  (MatrNr, VorlNr)
106  VALUES
107  (25403, 5022),
108  (26120, 5001),
109  (27550, 4052),
110  (27550, 5001),
111  (28106, 5041),
112  (28106, 5052),
113  (28106, 5216),
114  (28106, 5259),
115  (29120, 5001),
116  (29120, 5041),
117  (29120, 5049),
118  (29555, 5001),
119  (29555, 5022),
120  (28106, 5001),
121  (28106, 5022);
122
123  INSERT INTO prüfen
124  (MatrNr, VorlNr, PersNr, Note)
125  VALUES
126  (28106, 5001, 2126, 1),
127  (25403, 5041, 2125, 2),
128  (27550, 4630, 2137, 2),
129  (25403, 4630, 2137, 5);
130
131  INSERT INTO voraussetzen
132  (Vorgänger, Nachfolger)

```

VALUES

```

133 (5001, 5041),
134 (5001, 5043),
135 (5001, 5049),
136 (5041, 5216),
137 (5043, 5052),
138 (5041, 5052),
139 (5052, 5259);
140

```

- (a) Welche Vorlesungen liest der Boss des Assistenten *Platon* (nur Vorlesungsnummer und Titel ausgeben)?

```

1 SELECT v.VorlNr, v.Titel
2 FROM Vorlesungen v, Assistenten a
3 WHERE a.Boss = v.gelesenVon AND a.Name = 'Platon';

1  vorlnr |  titel
2  -----+-----
3      4052 | Logik
4      5041 | Ethik
5      5049 | Mäeutik
6 (3 rows)

```

- (b) Welche Studierende haben sich schon in mindestens einer direkten Voraussetzung von *Wissenschaftstheorie* prüfen lassen?

Wissenschaftstheorie (5052) → Erkenntnistheorie (5043) Ethik (5041) → Jonas (25403)

```

1 SELECT s.Name
2 FROM Vorlesungen l, voraussetzen a, prüfen p, Studierende s
3 WHERE
4   l.Titel = 'Wissenschaftstheorie' AND
5   l.VorlNr = a.Nachfolger AND
6   a.Vorgänger = p.VorlNr AND
7   p.MatrNr = s.MatrNr;

1  name
2  -----
3  Jonas
4 (1 row)

```

- (c) Wie viele Studierende hören *Ethik*?

```

1 SELECT COUNT(*)
2 FROM Vorlesungen v, hören h
3 WHERE
4   v.Titel = 'Ethik' AND
5   v.VorlNr = h.VorlNr;

1  count
2  -----
3      2
4 (1 row)

```

- (d) Welche Studierende sind im gleichen Semester? — Geben Sie Paare von Studierenden aus.

Achten Sie darauf, dass ein/e Studierende/r mit sich selbst kein Paar bildet. — Achten Sie auch darauf, dass kein Paar doppelt ausgegeben wird: wenn das Paar *StudentA*, *StudentB* im Ergebnis enthalten ist, soll nicht auch noch das Paar *StudentB*, *StudentA* ausgegeben werden.

```

1 SELECT s1.Name, s2.Name
2 FROM Studierende s1, Studierende s2
3 WHERE
4   s1.Semester = s2.Semester AND
5   s1.MatrNr < s2.MatrNr;

1  name |  name
2  -----+-----

```

```

3 Theophrastos | Feuerbach
4 (1 row)

```

- (e) In welchen Fächern ist die Durchschnittsnote schlechter als 2? Geben Sie die Vorlesungsnummer und den Titel aus.

```

1 SELECT v.VorlNr, v.Titel
2 FROM prüfen p, Vorlesungen v
3 WHERE p.VorlNr = v.VorlNr
4 GROUP BY v.VorlNr, v.Titel
5 HAVING AVG(p.Note) > 2;

```

```

1 vorlnr |      titel
2 -----+-----
3      4630 | Die 3 Kritiken
4 (1 row)

```

- (f) Finden Sie alle Paare von Studierenden (*MatrNr* duplikatfrei ausgeben), die mindestens zwei Vorlesungen gemeinsam hören.

```

1 SELECT h1.MatrNr, h2.MatrNr
2 FROM hören h1, hören h2
3 WHERE
4     h1.VorlNr = h2.VorlNr AND
5     h1.MatrNr < h2.MatrNr
6 GROUP BY h1.MatrNr, h2.MatrNr
7 HAVING COUNT(*) > 1;

```

```

1 matrnr | matrnr
2 -----+-----
3      28106 | 29120
4      28106 | 29555
5 (2 rows)

```