

Staatsexamen 66116 / 2019 / Frühjahr / Thema Nr. 1 / Teilaufgabe Nr. 2 / Aufgabe Nr. 3

## Aufgabe 3 [Game of Thrones]

Formulieren Sie folgende Anfragen in SQL gegen die angegebene Datenbank aus einer imaginären Serie.

Figur : {[ Id, Name, Schwertkunst, Lebendig, Titel ]}

gehört\_zu : {[ Id, Familie, FK (Id) references Figur(Id), FK (Familie) references Familie(Id) ]}

Familie : {[ Id, Name, Reichtum, Anführer ]}

Drache : {[ Name, Lebendig ]}

besitzt : {[ Id, Name, FK (Id) references Figur(Id), FK (Name) references Drache(Name) ]}

Festung : {[ Name, Ort, Ruine ]}

besetzt : {[ Familie, Festung, FK (Familie) references Familie(Id), FK (Festung) references Festung(Name) ]}

lebt : {[ Id, Name, FK (Id) references Figur(Id), FK (Name) references Festung(Name) ]}

```
1 CREATE TABLE Figur (  
2     Id integer PRIMARY KEY,  
3     Name VARCHAR(20),  
4     Schwertkunst integer,  
5     Lebendig boolean,  
6     Titel VARCHAR(50)  
7 );  
8  
9 CREATE TABLE Familie (  
10    Id integer PRIMARY KEY,  
11    Name VARCHAR(20),  
12    Reichtum numeric(11,2),  
13    Anführer VARCHAR(20)  
14 );  
15  
16 CREATE TABLE gehört_zu (  
17     Id integer REFERENCES Figur(id),  
18     Familie integer REFERENCES Familie(id)  
19 );  
20  
21 CREATE TABLE Drache (  
22     Name VARCHAR(20) PRIMARY KEY,  
23     Lebendig boolean  
24 );  
25  
26 CREATE TABLE besitzt (  
27     Id integer REFERENCES Figur(id),  
28     Drache integer REFERENCES Drache(Name),  
29     Familie integer REFERENCES Familie(Id),  
30     Festung integer REFERENCES Festung(Name)
```

```

27     Id integer REFERENCES Figur(Id),
28     Name VARCHAR(20) REFERENCES Drache(Name)
29 );
30
31 CREATE TABLE Festung (
32     Name VARCHAR(20) PRIMARY KEY,
33     Ort VARCHAR(30),
34     Ruine boolean
35 );
36
37 CREATE TABLE besetzt (
38     Familie integer REFERENCES Familie(Id),
39     Festung VARCHAR(20) REFERENCES Festung(Name)
40 );
41
42 CREATE TABLE lebt (
43     Id integer REFERENCES Figur(Id),
44     Name VARCHAR(20) REFERENCES Festung(Name)
45 );
46
47 INSERT INTO Figur VALUES
48     (1, 'Eddard Stark', 5, FALSE, 'Lord von Winterfell'),
49     (2, 'Rodd Stark', 4, FALSE, 'Lord von Winterfell'),
50     (3, 'Tywin Lennister', 5, FALSE, 'Lord von Casterlystein'),
51     (4, 'Cersei Lennister', 2, TRUE, 'Lady von Casterlystein'),
52     (5, 'Brandon Stark', 0, TRUE, 'König der Andalen'),
53     (6, 'Jon Schnee', 5, TRUE, 'König-jenseits-der-Mauer');
54
55 INSERT INTO Familie VALUES
56     (1, 'Haus Stark', 76873.12, 'Eddard Stark'),
57     (2, 'Haus Lennister', 82345.43, 'Tywin Lennister');
58
59 INSERT INTO gehört_zu VALUES
60     (1, 1),
61     (2, 1),
62     (3, 2),
63     (4, 2),
64     (5, 1),
65     (6, 1);
66
67 INSERT INTO Festung VALUES
68     ('Roter Bergfried', 'Westeros', FALSE),
69     ('Casterlystein', 'Westeros', FALSE),
70     ('Winterfell', 'Westeros', FALSE);
71
72 INSERT INTO besetzt VALUES
73     (1, 'Winterfell'),
74     (2, 'Roter Bergfried'),
75     (2, 'Casterlystein');
76
77 INSERT INTO lebt VALUES
78     (1, 'Winterfell'),
79     (2, 'Winterfell'),
80     (3, 'Casterlystein'),
81     (4, 'Roter Bergfried'),
82     (5, 'Winterfell'),
83     (6, 'Winterfell');

```

(a) Geben Sie für alle Figuren an, wie oft alle vorhandenen Titel vorkommen.

```

1 SELECT
2   Name,
3   Titel,
4   (SELECT COUNT(*) FROM Figur g WHERE g.Titel = f.Titel) AS
    ↳ Anzahl_Titel
5 FROM Figur f;

```

	name	titel	anzahl_titel
1	ddard Stark	Lord von Winterfell	2
2	odd Stark	Lord von Winterfell	2
3	ywin Lennister	Lord von Casterlystein	1
4	ersei Lennister	Lady von Casterlystein	1
5	randon Stark	König der Andalen	1
6	on Schnee	König-jenseits-der-Mauer	1
7	rows)		

- (b) Welche Figuren (Name ist gesucht) kommen aus „Kings Landing“?

```

1 SELECT
2   f.Name
3 FROM
4   Figur f,
5   Festung b,
6   lebt l
7 WHERE
8   b.Name = l.Name AND
9   f.Id = l.Id AND
10  b.Ort = 'Königsmund';

```

- (c) Geben Sie für jede Familie (Name) die Anzahl der zugehörigen Charaktere und Festungen an.

```

1 SELECT
2   f.Name,
3   (SELECT COUNT(*) FROM Figur fi, gehört_zu ge WHERE fi.id = ge.id AND
    ↳ ge.Familie = f.id) AS Anzahl_Charaktere,
4   (SELECT COUNT(*) FROM Festung fe, besetzt be WHERE fe.Name =
    ↳ be.Festung AND be.Familie = f.id) AS Anzahl_Festungen
5 FROM
6   Familie f;

```

- (d) Gesucht sind die besten fünf Schwertkämpfer aus Festungen aus dem Ort „Westeros“. Es soll der Name, die Schwertkunst und die Platzierung ausgegeben werden. Die Ausgabe soll nach der Platzierung sortiert erfolgen.

```

1 -- Problem: Es gibt 3 mal 3. Platz und nicht 3 mal 1. Platz
2 SELECT f1.Name, f1.Schwertkunst, COUNT(*) FROM Figur f1, Figur f2
3 WHERE f1.Schwertkunst <= f2.Schwertkunst
4 GROUP BY f1.Name, f1.Schwertkunst
5 ORDER BY COUNT(*)
6 LIMIT 5;

```

- (e) Schreiben Sie eine Anfrage, die alle Figuren löscht, die tot sind. Das Attribut *Lebendig* kann dabei die Optionen „ja“ und „nein“ annehmen.

```

1  -- Nur zu Testzwecken auflisten:
2  SELECT * FROM Figur;
3  SELECT * FROM gehört_zu;
4
5  -- PostgreSQL unterstützt kein DELETE JOIN
6  -- DELETE f, g, b, l FROM Figur AS f
7  -- JOIN gehört_zu AS g ON f.id = g.id
8  -- JOIN besitzt AS b ON f.id = b.id
9  -- JOIN lebt AS l ON f.id = l.id
10 -- WHERE f.Lebendig = FALSE;
11
12 DELETE FROM gehört_zu WHERE id IN (SELECT id FROM Figur WHERE Lebendig
   ↳ = FALSE);
13 DELETE FROM besitzt WHERE id IN (SELECT id FROM Figur WHERE Lebendig =
   ↳ FALSE);
14 DELETE FROM lebt WHERE id IN (SELECT id FROM Figur WHERE Lebendig =
   ↳ FALSE);
15 DELETE FROM Figur WHERE Lebendig = FALSE;
16
17 -- Nur zu Testzwecken auflisten:
18 SELECT * FROM Figur;
19 SELECT * FROM gehört_zu;

```

- (f) Löschen Sie die Spalten „Lebendig“ aus der Datenbank.

```

1  ALTER TABLE Figur DROP COLUMN Lebendig;
2  ALTER TABLE Drache DROP COLUMN Lebendig;

```

- (g) Erstellen Sie eine weitere Tabelle mit dem Namen *Waffen*, welche genau diese auflistet. Eine Waffe ist genau einer Figur zugeordnet, hat einen eindeutigen Namen und eine Stärke zwischen 0 und 5. Wählen Sie sinnvolle Typen für die Attribute.

```

1  CREATE TABLE Waffen (
2      Name VARCHAR(20) PRIMARY KEY,
3      Figur integer NOT NULL REFERENCES Figur(Id),
4      Stärke integer NOT NULL CHECK(Stärke BETWEEN 0 AND 5)
5  );
6
7  -- Sollte funktionieren:
8  INSERT INTO Waffen VALUES
9      ('Axt', 1, 5);
10
11 -- Sollte Fehler ausgeben:
12 -- INSERT INTO Waffen VALUES
13 -- ('Schleuder', 1, 6);

```

Github: [Staatsexamen/66116/2019/09/Thema-1/Teilaufgabe-2/Aufgabe-3.tex](https://github.com/Staatsexamen/66116/2019/09/Thema-1/Teilaufgabe-2/Aufgabe-3.tex)