

Staatsexamen 66116 / 2020 / Frühjahr

Thema 1 / Teilaufgabe 2 / Aufgabe 7

(Zehnkampf)

Stichwörter: SQL, SQL mit Übungsdatenbank, Top-N-Query

Gegeben sei die Relation *Zehnkampf*, welche die Ergebnisse eines Zehnkampfwettkampfes verwaltet. Eine beispielhafte Ausprägung ist in nachfolgender Tabelle gegeben.

Hinweise: Jeder Athlet kann in jeder Disziplin maximal ein Ergebnis erzielen. Außerdem können Sie davon ausgehen, dass jeder Name eindeutig ist.

Name	Disziplin	Leistung	Einheit	Punkte
John	100m	10.21	Sekunden	845
Peter	Hochsprung	213	Zentimeter	812
Peter	100m	10.10	Sekunden	920
Hans	100m	10.21	Sekunden	845
Hans	400m	44.12	Sekunden	910
...

```

1 CREATE TABLE Zehnkampf (
2   Name VARCHAR(30),
3   Disziplin VARCHAR(30),
4   Leistung FLOAT,
5   Einheit VARCHAR(30),
6   Punkte INTEGER,
7   PRIMARY KEY(Name, Disziplin, Leistung)
8 );
9
10 INSERT INTO Zehnkampf VALUES
11 ('John', '100m', 10.21, 'Sekunden', 845),
12 ('Peter', 'Hochsprung', 213, 'Zentimeter', 812),
13 ('Peter', '100m', 10.10, 'Sekunden', 920),
14 ('Hans', '100m', 10.21, 'Sekunden', 845),
15 ('Hans', '400m', 44.12, 'Sekunden', 910);

```

- (a) Bestimmen Sie alle funktionale Abhängigkeiten, die **sinnvollerweise** in der Relation *Zehnkampf* gelten.

$$FA = \left\{ \begin{array}{l} \{ Disziplin \} \rightarrow \{ Einheit \}, \\ \{ Disziplin, Leistung \} \rightarrow \{ Punkte \}, \\ \{ Name, Disziplin \} \rightarrow \{ Leistung \}, \end{array} \right\}$$

- (b) Normalisieren Sie die Relation *Zehnkampf* unter Beachtung der von Ihnen identifizierten funktionalen Abhängigkeiten. Unterstreichen Sie alle Schlüssel des resultierenden Schemas.

```

R1 : {[ Disziplin, Einheit ]}
R2 : {[ Disziplin, Leistung, Punkte ]}
R3 : {[ Name, Disziplin, Leistung ]}

```

- (c) Bestimmen Sie in SQL den Athleten (oder bei Punktgleichheit, die Athleten), der in der Summe am meisten Punkte in allen Disziplinen erzielt hat. Benutzen Sie dazu die noch nicht normalisierte Ausgangsrelation *Zehnkampf*.

```

1 CREATE VIEW GesamtPunkte AS
2   SELECT Name, SUM(Punkte) As Punkte
3   FROM Zehnkampf
4   GROUP BY Name;
5

```

```
6 SELECT g1.Name, g1.Punkte, COUNT(*) AS Rang
7 FROM GesamtPunkte g1, GesamtPunkte g2
8 WHERE g1.Punkte <= g2.Punkte
9 GROUP BY g1.Name, g1.Punkte
10 HAVING COUNT(*) = 1;
```

```
1  name | punkte | rang
2  -----+-----
3  Hans |   1755 |    1
4  (1 row)
```