Staatsexamen 66116 / 2018 / Frühjahr / Thema Nr. 1 / Teilaufgabe Nr. 2 / Aufgabe Nr. 4

Aufgabe 4: SQL [Triathlon]

Gegeben sind folgende Relationen aus einer Datenbank zur Verwaltung von Triathlon-Wettbewerben.

```
Athlet : {[ ID, Vorname, Nachname ]}

Ergebnis : {[ Athlet[Athlet], Wettbewerb[Wettbewerb], Schwimmzeit, Radzeit, Laufzeit ]}

Wettbewerb : {[ Name, Jahr ]}
```

```sql
CREATE TABLE Athlet (
  ID INTEGER PRIMARY KEY,
  Vorname VARCHAR(20),
  Nachname VARCHAR(20)
);

CREATE TABLE Wettbewerb (
  Name VARCHAR(40) PRIMARY KEY,
  Jahr INTEGER
);

CREATE TABLE Ergebnis (
  Athlet INTEGER REFERENCES Athlet(ID),
  Wettbewerb VARCHAR(40) REFERENCES Wettbewerb(Name),
  Schwimmzeit INTEGER NOT NULL,
  Radzeit INTEGER,
  Laufzeit INTEGER,
  PRIMARY KEY (Athlet, Wettbewerb)
);

INSERT INTO Athlet VALUES
  (1, 'Boris', 'Stein'),
  (2, 'Trevor', 'Wurtele'),
  (3, 'Reichelt', 'Horst'),
  (12, 'Mitch', 'Kibby');

INSERT INTO Wettbewerb VALUES
  ('Zürichsee', 2018),
  ('Ironman Vichy', 2018),
  ('Challenge Walchsee', 2018),
  ('Triathlon Alpe d’Huez', 2017);

INSERT INTO Ergebnis VALUES
  (1, 'Zürichsee', 14, 10, 11),
  (2, 'Zürichsee', 13, 10, 11),
  (3, 'Zürichsee', 12, 10, 11),
  (12, 'Zürichsee', 11, 10, 11),
  (2, 'Challenge Walchsee', 12, 10, 11),
  (3, 'Challenge Walchsee', 11, 10, 11),
  (12, 'Triathlon Alpe d’Huez', 9, 10, 11);
```

Verwenden Sie im Folgenden nur Standard-SQL und keine produktspezifischen Erweiterungen. Sie dürfen bei Bedarf Views anlegen. Geben Sie einen Datensatz, also eine Entity, nicht mehrfach aus.

(a) Schreiben Sie eine SQL-Anweisung, die die Tabelle „Ergebnis“ anlegt. Gehen Sie davon aus, dass die Tabellen „Athlet“ und „Wettbewerb“ bereits existieren. Verwenden Sie sinnvolle Datentypen.

```sql
CREATE TABLE IF NOT EXISTS Ergebnis (
  Athlet INTEGER REFERENCES Athlet(ID),
  Wettbewerb INTEGER REFERENCES Wettbewerb(Name),
  Schwimmzeit INTEGER NOT NULL,
  Radzeit INTEGER,
  Laufzeit INTEGER,
```

```
  PRIMARY KEY (Athlet, Wettbewerb)
);
```

(b) Schreiben Sie eine SQL-Anweisung, die die Radzeit des Teilnehmers mit der ID 12 beim Wettbewerb „Zürichsee“ um eins erhöht.

```
-- Nur für Test-Zwecke
SELECT * FROM Ergebnis WHERE Athlet = 12 AND Wettbewerb = 'Zürichsee';

UPDATE Ergebnis
SET Radzeit = Radzeit + 1
WHERE Athlet = 12 AND Wettbewerb = 'Zürichsee';

-- Nur für Test-Zwecke
SELECT * FROM Ergebnis WHERE Athlet = 12 AND Wettbewerb = 'Zürichsee';
```

(c) Schreiben Sie eine SQL-Anweisung, die die Namen aller Wettbewerbe des Jahres 2018 ausgibt, absteigend sortiert nach Name.

```
SELECT Name
FROM Wettbewerb
WHERE Jahr = 2018
ORDER BY Name DESC;
```

(d) Schreiben Sie eine SQL-Anweisung, die die Namen aller Wettbewerbe ausgibt, in der die durchschnittliche Schwimmzeit größer als 10 ist.

```
SELECT Wettbewerb, AVG(Schwimmzeit)
FROM Ergebnis
GROUP BY Wettbewerb
HAVING AVG(Schwimmzeit) > 10;
```

(e) Schreiben Sie eine SQL-Anweisung, die die IDs aller Athleten ausgibt, die im Jahr 2017 an keinem Wettbewerb teilgenommen haben.

```
(SELECT DISTINCT Athlet FROM Ergebnis)
EXCEPT
(SELECT DISTINCT Athlet FROM Ergebnis e, Wettbewerb w
WHERE e.Wettbewerb = w.name AND w.Jahr = 2017);
```

(f) Schreiben Sie eine SQL-Anweisung, die die Nachnamen aller Athleten ausgibt, die mindestens 10 Wettbewerbe gewonnen haben, das heißt im jeweiligen Wettbewerb die kürzeste Gesamtzeit erreicht haben. Die Gesamtzeit ist die Summe aus Schwimmzeit, Radzeit und Laufzeit. Falls zwei Athleten in einem Wettbewerb die gleiche Gesamtzeit erreichen, sind beide Sieger.

vermutlich falsch

```
CREATE VIEW Gesamtzeiten AS
SELECT e.Athlet AS NameAthlet, e.Radzeit + e.Schwimmzeit + e.Laufzeit
AS Gesamtzeit, w.NameWettbewerb
FROM Ergebnis e, Wettbewerb w
WHERE e.Wettbewerb = w.Name
CREATE VIEW Sieger AS
SELECT g1.NameAthlet
FROM Gesamtzeiten g1, Gesamtzeiten g2
GROUP BY g1.NameWettbewerb
HAVING g1.Gesamtzeit < g2.Gesamtzeit
SELECT NameAthlet
FROM Sieger
GROUP BY NameAthlet
HAVING count(*) > 10;
```

(g) Schreiben Sie eine SQL-Anweisung, die die Top-Ten der Athleten mit der schnellsten Schwimmzeit des Wettbewerbs „Paris“ ausgibt. Ausgegeben werden sollen die Platzierung (1 bis 10) und der Nachname des Athleten, aufsteigend sortiert nach Platzierung. Gehen Sie davon aus, dass keine zwei Athleten die gleiche Schwimmzeit haben und verwenden Sie keine produktspezifischen Anweisungen wie beispielsweise rownum, top oder limit.

```sql
CREATE VIEW AthletenParis AS
SELECT a.Nachname, e.Schwimmzeit
FROM Athlet a, Ergebnis e INNER JOIN Wettbewerb W ON e.Wettbewerb = w.Name
WHERE w.Name = "Paris" AND a.ID = e.Athlet
SELECT a.Nachname COUNT(*) + 1 AS Platzierung
FROM AthletenParis a, AthletenParis b
WHERE a.Schwimmzeit < b.Schwimmzeit
GROUP BY a.Nachname
HAVING Platzierung <= 10;
```

(h) Schreiben Sie einen Trigger, der beim Einfügen neuer Tupel in die Tabelle „Ergebnis“ die Schwimmzeit auf den Wert 0 setzt, falls diese negativ ist.

```sql
CREATE TRIGGER update_Ergebnis AFTER UPDATE ON Ergebnis AS
IF(UPDATE Schwimmzeit AND Schwimmzeit < 0) BEGIN UPDATE Ergebnis
SET Schwimmzeit = 0
WHERE (Athlet, Wettbewerb) IN (SELECT DISTINCT (Athlet, Wettbewerb) FROM inserted)
END;
```

Github: `Staatsexamen/66116/2018/09/Thema-1/Teilaufgabe-2/Aufgabe-4.tex`