

Aufgabe 4

Gegeben sind folgende Relationen:

Mitarbeiter (MitarbeiterID, Vorname, Nachname, Adresse, Gehalt, Vorgesetzter [Mitarbeiter] NOT NULL, AbteilungsID[Abteilung])

Abteilung (AbteilungsID, Bezeichnung UNIQUE NOT NULL)

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

```
1 CREATE TABLE Abteilung(  
2     AbteilungsID INTEGER PRIMARY KEY,  
3     Bezeichnung VARCHAR(30) UNIQUE NOT NULL  
4 );  
5  
6 CREATE TABLE Mitarbeiter(  
7     MitarbeiterID INTEGER PRIMARY KEY,  
8     Vorname VARCHAR(30),  
9     Nachname VARCHAR(30),  
10    Adresse VARCHAR(60),  
11    Gehalt DECIMAL(7, 2),  
12    Vorgesetzter INTEGER NOT NULL REFERENCES Mitarbeiter(MitarbeiterID),  
13    AbteilungsID INTEGER REFERENCES Abteilung(AbteilungsID)  
14 );  
15  
16 INSERT INTO Abteilung VALUES  
17     (1, 'Buchhaltung'),  
18     (42, 'Vertrieb');  
19  
20 INSERT INTO Mitarbeiter VALUES  
21     (1, 'Karl', 'Landsbach', 'Sigmaringstraße 4, 87153 Farnbach', 2467.23, 1, 42),  
22     (2, 'Lisa', 'Grätzner', 'Scheidplatz 6, 18434 Tullach', 5382.2, 1, 42),  
23     (3, 'Sarah', 'Riedel', 'Am Angera 3, 79527 Töll', 7382.2, 1, 42),  
24     (4, 'Franz', 'Rudolf', 'Strewitzstraße 4, 45507 Strewith', 2382.2, 1, 42),  
25     (5, 'Sergej', 'Puschkin', 'Radolf 4, 12507 Radstadt', 1382.2, 1, 1);
```

- (a) Schreiben Sie eine SQL-Anweisung, die die Tabelle Mitarbeiter anlegt. Gehen Sie davon aus, dass die Tabelle Abteilung bereits existiert.

Siehe oben

- (b) Schreiben Sie eine SQL-Anweisung, die Vor- und Nachnamen der Mitarbeiter der Abteilung mit der Bezeichnung Vertrieb ausgibt, absteigend sortiert nach MitarbeiterID.

```
1 SELECT m.Vorname, m.Nachname  
2 FROM Mitarbeiter m, Abteilung a  
3 WHERE  
4     a.AbteilungsID = m.AbteilungsID AND  
5     a.Bezeichnung = 'Vertrieb'  
6 ORDER BY m.MitarbeiterID DESC;
```

- (c) Schreiben Sie eine SQL-Anweisung, die Vor- und Nachnamen sowie das Gehalt von Mitarbeitern ausgibt, die mehr verdienen als ihr Chef. Sortieren Sie die Ausgabe absteigend nach dem Gehalt.

```

1  SELECT m.Vorname, m.Nachname, m.Gehalt
2  FROM Mitarbeiter m, Mitarbeiter n
3  WHERE
4      m.Vorgesetzter = n.MitarbeiterID AND
5      m.Gehalt > n.Gehalt
6  ORDER BY m.Gehalt DESC;

```

- (d) Schreiben Sie eine SQL-Anweisung, die das Gehalt von allen Mitarbeitern aus der Abteilung mit der ID 42 um 10% erhöht.

```

1  UPDATE Mitarbeiter
2  SET Gehalt = Gehalt * 1.1
3  WHERE AbteilungsID = 42;
4  SELECT * FROM Mitarbeiter;

```

- (e) Schreiben Sie eine SQL-Anweisung, welche den Vornamen, die Nachnamen und das Gehalt der sieben bestbezahlten Mitarbeiter aus der Buchhaltung ausgibt. Standardkonforme Sprachkonstrukte, die eine Beschränkung der Ausgabe bewirken, sind erlaubt.

```

1  SELECT m.Vorname, m.Nachname, m.Gehalt
2  FROM Mitarbeiter m, Mitarbeiter n, Abteilung a
3  WHERE
4      m.Gehalt <= n.Gehalt AND
5      a.AbteilungsID = m.AbteilungsID AND
6      a.AbteilungsID = n.AbteilungsID AND
7      a.Bezeichnung = 'Buchhaltung'
8  GROUP BY m.Vorname, m.Nachname, m.Gehalt
9  HAVING COUNT(*) <= 7
10 ORDER BY m.Gehalt DESC;

```

- (f) Schreiben Sie eine SQL-Anweisung, die für jede Abteilung die Mitarbeiter ermittelt, die am wenigsten verdienen. Dabei sollen Vorname, Nachname und die Abteilungsbezeichnung der Mitarbeiter ausgegeben werden.

```

1  SELECT m.Vorname, m.Nachname, m.Gehalt, a.Bezeichnung
2  FROM Mitarbeiter m, Mitarbeiter n, Abteilung a
3  WHERE
4      m.Gehalt >= n.Gehalt AND
5      m.AbteilungsID = n.AbteilungsID AND
6      m.AbteilungsID = a.AbteilungsID
7  GROUP BY m.Vorname, m.Nachname, m.Gehalt, m.AbteilungsID,
8      ↪ a.Bezeichnung
9  HAVING COUNT(*) <= 1
10 ORDER BY m.Gehalt DESC;

```