

Aufgabe 5: SQL

Gegeben seien die folgenden drei Relationen. Diese Relationen erfassen die Mitarbeiterverwaltung eines Unternehmens. Schlüssel sind fett dargestellt und Fremdschlüssel sind kursiv dargestellt. So werden Mitarbeiter, Abteilungen und Unternehmen jeweils durch ihre Nummer identifiziert. AbtNr ist die Nummer der Abteilung, in der ein Mitarbeiter arbeitet. Manager ist die Nummer des Mitarbeiters, der die Abteilung leitet. UntNr ist die Nummer des Unternehmens, dem eine Abteilung zugeordnet ist.

- 1 Mitarbeiter(Nummer, Name, Alter, Gehalt, AbtNr)
- 2 Abteilung(Nummer, Name, Budget, Manager, UntNr)
- 3 Unternehmen(Nummer, Name, Adresse)

- (a) Wie hoch ist das Durchschnittsalter der Abteilung „Personal Care“ im Unternehmen „Test.com“?

Musterlösung:

GROUP BY nicht nötig, AS nicht vergessen.

```
1 SELECT AVG(m.Alter) AS Durchschnittsalter
2 FROM Unternehmen u, Abteilung a, Mitarbeiter m
3 WHERE
4   a.Name = 'Personal Care' AND
5   u.Name = 'Test.com' AND
6   u.Number = a.UntNr AND
7   m.AbtNr = a.Number;
```

- (b) Geben Sie für jedes Unternehmen das Durchschnittsalter der Mitarbeiter an!

Musterlösung:

Statt a.UntNr kann u.Number verwendet werden. a.UntNr nur deshalb, weil man dann eventuell den Join über die Unternehmenstabelle sparen kann.

Alles was ausgegeben werden soll, muss auch in GROUP BY enthalten sein.

```
1 SELECT a.UntNr, u.Name, AVG(m.Alter) as Durchschnittsalter
2 FROM Unternehmen u, Abteilung a, Mitarbeiter m
3 WHERE
4   u.Number = a.UntNr AND
5   m.AbtNr = a.Number
6 GROUP BY a.UntNr, u.Name;
```

- (c) Wie viele Mitarbeiter im Unternehmen „Test.com“ sind älter als ihr Chef? (D.h. sind älter als der Manager der Abteilung, in der sie arbeiten.)

Musterlösung:

```
1 SELECT COUNT(*)
2 FROM Mitarbeiter m, Abteilung a, Unternehmer u
3 WHERE
4   m.AbtNr = a.Number AND
5   a.UntNr = u.Number AND
```

```

6      u.Name = 'Test.com'
7  AND m.Alter > (
8      SELECT ma.Alter
9      FROM Mitarbeiter ma, Abteilung ab
10     WHERE
11         ma.Nummer = ab.Manager AND
12         a.Nummer = ab.Nummer
13 );

```

oder einfacher:

```

1  SELECT COUNT(*)
2  FROM Mitarbeiter m, Abteilung a, Unternehmer u
3  WHERE
4      m.AbtNr = a.Nummer AND
5      a.UntNr = u.Nummer AND
6      u.Name = 'Test.com'
7  AND m.Alter > (
8      SELECT ma.Alter
9      FROM Mitarbeiter ma
10     WHERE ma.Nummer = a.Manager
11 );

```

Alternativ Lösung ohne Unterabfragem, mit Self join:

```

1  SELECT COUNT(*)
2  FROM Mitarbeiter m, Abteilung a, Unternehmer u, Mitarbeiter m2
3  WHERE
4      m.AbtNr = a.Nummer AND
5      a.UntNr = u.Nummer AND
6      u.Name = 'Test.com' AND
7      a.Manager = m2.Nummer AND
8      m.Alter > m2.Alter;

```

- (d) Welche Abteilungen haben ein geringeres Budget als die Summe der Gehälter der Mitarbeiter, die in der Abteilung arbeiten?

Musterlösung:

```

1  SELECT a.Name, a.Nummer
2  FROM Abteilung a
3  WHERE a.Budget < (
4      SELECT SUM(m.Gehalt)
5      FROM Mitarbeiter m
6      WHERE a.Nummer = m.AbtNr
7  );

```

Ohne Unterabfrage

```

1  SELECT a.Name, a.Nummer
2  FROM Abteilung a, Mitarbeiter m
3  WHERE a.Nummer = m.AbtNr
4  GROUP BY a.Nummer, a.Name, a.Budget
5  HAVING a.Budget < SUM(m.Gehalt);

```

- (e) Versetzen Sie den Mitarbeiter „Wagner“ in die Abteilung „Personal Care“!

Musterlösung:

```

1 UPDATE Mitarbeiter m
2 SET AbtNr = (
3     SELECT a.Nummer FROM
4     Abteilung a
5     WHERE a.Name = 'Personal Care'
6 )
7 WHERE m.Name = 'Wagner';

```

- (f) Löschen Sie die Abteilung „Personal Care“ mit allen ihren Mitarbeitern!

Musterlösung:

```

1 DELETE FROM Mitarbeiter
2 WHERE AbtNr = (
3     SELECT a.Nummer
4     FROM Abteilung a
5     WHERE a.Name = 'Personal Care'
6 );
7
8 DELETE FROM Abteilung
9 WHERE Name = 'Personal Care';

```

- (g) Geben Sie den Managern aller Abteilungen, die ihr Budget nicht überziehen, eine 10 Prozent Gehaltserhöhung. (Das Budget ist überzogen, wenn die Gehälter der Mitarbeiter höher sind als das Budget der Abteilung.) Zusatzfrage: Was passiert mit Mitarbeitern, die Manager von mehreren Abteilungen sind?

Musterlösung:

```

1 CREATE VIEW LowBudget AS (
2     SELECT Nummer
3     FROM Abteilung
4     WHERE Nummer NOT IN (
5         SELECT a.Nummer
6         FROM Abteilung a
7         WHERE
8             a.Budget < (
9                 SELECT SUM(Gehalt)
10                FROM Mitarbeiter m Abteilung A
11                WHERE m.AbtNr = A.Nummer AND
12                a.Nummer = A.Nummer
13            )
14     )
15 )
16
17 UPDATE Mitarbeiter
18 SET Gehalt = 1.1 * Gehalt
19 WHERE Nummer IN (
20     SELECT Manager
21     FROM LowBudget, Abteilung
22     WHERE LowBudget.Manager = Abteilung.Nummer
23 )

```

```

1 TABLE `abteilung` (
2   `Nummer` int(11) NOT NULL,
3   `Name` varchar(20) DEFAULT NULL,
4   `Budget` float DEFAULT NULL,
5   `Manager` varchar(20) NOT NULL,

```

```

6      `UntNr` int(11) DEFAULT NULL
7  ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
8
9  INSERT INTO `abteilung` (`Nummer`, `Name`, `Budget`, `Manager`, `UntNr`) VALUES
10 (1, 'Personal_Care', 20000, 'Huber', 1),
11 (11, 'Tequilla_Mix', 50000, 'Taylor', 2),
12 (21, 'Nerds', 500, 'Gates', 3);
13
14 CREATE TABLE `mitarbeiter` (
15   `Nummer` int(11) NOT NULL,
16   `Name` varchar(20) NOT NULL,
17   `Alter` int(11) NOT NULL,
18   `Gehalt` float NOT NULL,
19   `AbtNr` int(11) NOT NULL
20 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
21
22 INSERT INTO `mitarbeiter` (`Nummer`, `Name`, `Alter`, `Gehalt`, `AbtNr`) VALUES
23 (1, 'Müller', 30, 30000, 1),
24 (2, 'Huber', 45, 80000, 1),
25 (3, 'Habermeier', 62, 40000, 1),
26 (4, 'Leifsson', 27, 50000, 1),
27 (5, 'Taylor', 37, 85000, 11),
28 (6, 'Smith', 61, 34000, 11),
29 (7, 'Pitt', 36, 40000, 11),
30 (8, 'Thompson', 54, 52000, 11),
31 (9, 'Gates', 69, 15000000, 21),
32 (10, 'Zuckerberg', 36, 10000000, 21),
33 (11, 'Jobs', 99, 14000000, 21),
34 (12, 'Nakamoto', 66, 5000000, 21);
35
36 CREATE TABLE `unternehmen` (
37   `Nummer` int(11) NOT NULL,
38   `Name` varchar(20) DEFAULT NULL,
39   `Adresse` varchar(50) DEFAULT NULL
40 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
41
42 INSERT INTO `unternehmen` (`Nummer`, `Name`, `Adresse`) VALUES
43 (1, 'Test.com', 'Alter Hafen 11'),
44 (2, 'Party.de', 'Technosträße 3'),
45 (3, 'IT.ch', 'Sequelweg 1');
46
47 ALTER TABLE `abteilung`
48   ADD PRIMARY KEY (`Nummer`),
49   ADD KEY `UntNr` (`UntNr`);
50
51 ALTER TABLE `mitarbeiter`
52   ADD PRIMARY KEY (`Nummer`),
53   ADD KEY `AbtNr` (`AbtNr`);
54
55 ALTER TABLE `unternehmen`
56   ADD PRIMARY KEY (`Nummer`);
57
58 ALTER TABLE `abteilung`
59   ADD CONSTRAINT `abteilung_ibfk_1` FOREIGN KEY (`UntNr`) REFERENCES
60     ↳ `unternehmen` (`Nummer`);
61
62 ALTER TABLE `mitarbeiter`
63   ADD CONSTRAINT `mitarbeiter_ibfk_1` FOREIGN KEY (`AbtNr`) REFERENCES
64     ↳ `abteilung` (`Nummer`);
65 COMMIT;

```