

Einzelprüfung „Rechnerarchitektur / Datenbanken / Betriebssysteme (vertieft)“

Einzelprüfungsnummer 66113 / 2003 / Frühjahr

Thema 1 / Aufgabe 5

(Mitarbeiterverwaltung)

Stichwörter: SQL, SQL mit Übungsdatenbank

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.

Mitarbeiter(Nummer, Name, Alter, Gehalt, AbtNr)

Abteilung(Nummer, Name, Budget, Manager, UntNr)

Unternehmen(Nummer, Name, Adresse)

```
CREATE TABLE unternehmen (  
    Nummer integer NOT NULL PRIMARY KEY,  
    Name VARCHAR(20) DEFAULT NULL,  
    Adresse VARCHAR(50) DEFAULT NULL  
);  
  
CREATE TABLE abteilung (  
    Nummer integer NOT NULL PRIMARY KEY,  
    Name VARCHAR(20) DEFAULT NULL,  
    Budget float DEFAULT NULL,  
    Manager VARCHAR(20) NOT NULL,  
    UntNr integer DEFAULT NULL REFERENCES unternehmen (Nummer)  
);  
  
CREATE TABLE mitarbeiter (  
    Nummer integer NOT NULL PRIMARY KEY,  
    Name VARCHAR(20) NOT NULL,  
    Alter integer NOT NULL,  
    Gehalt float NOT NULL,  
    AbtNr integer NOT NULL REFERENCES abteilung (Nummer)  
);  
  
INSERT INTO unternehmen (Nummer, Name, Adresse) VALUES  
    (1, 'Test.com', 'Alter Hafen 11'),  
    (2, 'Party.de', 'Technostraße 3'),  
    (3, 'IT.ch', 'Sequelweg 1');  
  
INSERT INTO abteilung (Nummer, Name, Budget, Manager, UntNr) VALUES  
    (1, 'Personal_Care', 20000, 'Huber', 1),  
    (11, 'Tequilla_Mix', 50000, 'Taylor', 2),  
    (21, 'Nerds', 500, 'Gates', 3);  
  
INSERT INTO mitarbeiter (Nummer, Name, Alter, Gehalt, AbtNr) VALUES  
    (1, 'Müller', 30, 30000, 1),  
    (2, 'Huber', 45, 80000, 1),
```

```
(3, 'Habermeier', 62, 40000, 1),
(4, 'Leifsson', 27, 50000, 1),
(5, 'Taylor', 37, 85000, 11),
(6, 'Smith', 61, 34000, 11),
(7, 'Pitt', 36, 40000, 11),
(8, 'Thompson', 54, 52000, 11),
(9, 'Gates', 69, 15000000, 21),
(10, 'Zuckerberg', 36, 10000000, 21),
(11, 'Jobs', 99, 14000000, 21),
(12, 'Nakamoto', 66, 5000000, 21);
```

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

Lösungsvorschlag

GROUP BY nicht nötig, AS nicht vergessen.

```
SELECT AVG(m.Alter) AS Durchschnittsalter
FROM Unternehmen u, Abteilung a, Mitarbeiter m
WHERE
  a.Name = 'Personal Care' AND
  u.Name = 'Test.com' AND
  u.Nummer = a.UntNr AND
  m.AbtNr = a.Nummer;
```

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

Lösungsvorschlag

Statt a.UntNr kann u.Nummer 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.

```
SELECT a.UntNr, u.Name, AVG(m.Alter) as Durchschnittsalter
FROM Unternehmen u, Abteilung a, Mitarbeiter m
WHERE
  u.Nummer = a.UntNr AND
  m.AbtNr = a.Nummer
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.)

Lösungsvorschlag

```
SELECT COUNT(*)
FROM Mitarbeiter m, Abteilung a, Unternehmen u
WHERE
  m.AbtNr = a.Nummer AND
  a.UntNr = u.Nummer AND
  u.Name = 'Test.com'
AND m.Alter > (
  SELECT ma.Alter
  FROM Mitarbeiter ma, Abteilung ab
  WHERE
```

```
ma.Nummer = ab.Manager AND
a.Nummer = ab.Nummer
);
```

oder einfacher:

```
SELECT COUNT(*)
FROM Mitarbeiter m, Abteilung a, Unternehmen u
WHERE
  m.AbtNr = a.Nummer AND
  a.UntNr = u.Nummer AND
  u.Name = 'Test.com'
AND m.Alter > (
  SELECT ma.Alter
  FROM Mitarbeiter ma
  WHERE ma.Nummer = a.Manager
);
```

Alternativ Lösung ohne Unterabfragen, mit Self join:

```
SELECT COUNT(*)
FROM Mitarbeiter m, Abteilung a, Unternehmen u, Mitarbeiter m2
WHERE
  m.AbtNr = a.Nummer AND
  a.UntNr = u.Nummer AND
  u.Name = 'Test.com' AND
  a.Manager = m2.Nummer AND
  m.Alter > m2.Alter;
```

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

Lösungsvorschlag

```
SELECT a.Name, a.Nummer
FROM Abteilung a
WHERE a.Budget < (
  SELECT SUM(m.Gehalt)
  FROM Mitarbeiter m
  WHERE a.Nummer = m.AbtNr
);
```

Ohne Unterabfrage

```
SELECT a.Name, a.Nummer
FROM Abteilung a, Mitarbeiter m
WHERE a.Nummer = m.AbtNr
GROUP BY a.Nummer, a.Name, a.Budget
HAVING a.Budget < SUM(m.Gehalt);
```

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

Lösungsvorschlag

```
UPDATE Mitarbeiter m
SET AbtNr = (
  SELECT a.Nummer FROM
  Abteilung a
  WHERE a.Name = 'Personal Care'
)
WHERE m.Name = 'Wagner';
```

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

Lösungsvorschlag

```
DELETE FROM Mitarbeiter
WHERE AbtNr = (
  SELECT a.Nummer
  FROM Abteilung a
  WHERE a.Name = 'Personal Care'
);

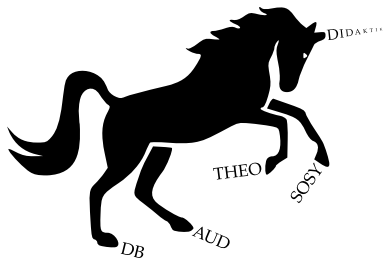
DELETE FROM Abteilung
WHERE Name = 'Personal Care';
```

- (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?

Lösungsvorschlag

```
CREATE VIEW LowBudget AS (
  SELECT Nummer
  FROM Abteilung
  WHERE Nummer NOT IN (
    SELECT a.Nummer
    FROM Abteilung a
    WHERE
      a.Budget < (
        SELECT SUM(Gehalt)
        FROM Mitarbeiter m Abteilung A
        WHERE m.AbtNr = A.Nummer AND
              a.Nummer = A.Nummer
      )
  )
)

UPDATE Mitarbeiter
SET Gehalt = 1.1 * Gehalt
WHERE Nummer IN (
  SELECT Manager
  FROM LowBudget, Abteilung
  WHERE LowBudget.Manager = Abteilung.Nummer
)
```



Die Bschlangaul-Sammlung Hermine Bschlangauland Friends

Eine freie Aufgabensammlung mit Lösungen von Studierenden für Studierende zur Vorbereitung auf die 1. Staatsexamensprüfungen des Lehramts Informatik in Bayern.



Diese Materialsammlung unterliegt den Bestimmungen der Creative Commons Namensnennung-Nicht kommerziell-Share Alike 4.0 International-Lizenz.

Hilf mit! Die Hermine schafft das nicht allein! Das ist ein Community-Projekt! Verbesserungsvorschläge, Fehlerkorrekturen, weitere Lösungen sind herzlich willkommen - egal wie - per Pull-Request oder per E-Mail an hermine.bschlangaul@gmx.net. Der TeX-Quelltext dieses Dokuments kann unter folgender URL aufgerufen werden: <https://github.com/bschlangaul-sammlung/examens-aufgaben/blob/main/Staatsexamen/66113/2003/03/Thema-1/Aufgabe-5.tex>