

Einzelprüfung „Datenbanksysteme / Softwaretechnologie (vertieft)“

Einzelprüfungsnummer 66116 / 2016 / Herbst

## Thema 1 / Teilaufgabe 1 / Aufgabe 4

(Personalverwaltung)

**Stichwörter:** SQL, SQL mit Übungsdatenbank, VIEW, WITH, DELETE

Gegeben sind folgende Relationen aus einer Personalverwaltung:

Mitarbeiter : {[ MitarbeiterID, Vorname, Nachname, Vorgesetzter[Mitarbeiter], AbteilungsID[Abteilung], Telefonnummer, Gehalt ]}

Abteilung : {[ AbteilungsID, Bezeichnung ]}

### Additum: Übungsdatenbank

```
CREATE TABLE Abteilung(
  AbteilungsID INTEGER PRIMARY KEY,
  Bezeichnung VARCHAR(30)
);

CREATE TABLE Mitarbeiter(
  MitarbeiterID INTEGER PRIMARY KEY,
  Vorname VARCHAR(30),
  Nachname VARCHAR(30),
  Vorgesetzter INTEGER REFERENCES Mitarbeiter(MitarbeiterID),
  AbteilungsID INTEGER REFERENCES Abteilung(AbteilungsID),
  Telefonnummer VARCHAR(50),
  Gehalt DOUBLE PRECISION
);

INSERT INTO Abteilung VALUES
(1, 'Buchhaltung'),
(2, 'Vertrieb'),
(42, 'Managment'),
(4, 'Qualitätskontrolle'),
(5, 'Produktion');

INSERT INTO Mitarbeiter
(MitarbeiterID, Vorname, Nachname, Vorgesetzter, AbteilungsID, Telefonnummer, Gehalt)
VALUES
(1, 'Hans', 'Meier', 11, 4, '023/13432', 2335),
(2, 'Fred', 'Wolitz', 11, 2, '0233/413432', 1233),
(11, 'Lea', 'Müller', NULL, 42, '0343/3452', 5875),
(3, 'Till', 'Fuchs', 2, 1, '023/13344', 2345),
(4, 'Fred', 'Hase', 11, 4, '04/453432', 1334),
(12, 'Gerd', 'Navratil', NULL, 42, '0345/552', 7154),
(6, 'Jürgen', 'Schmidt', 12, 5, '097/dfg854', 654);
```

- (a) Schreiben Sie eine SQL-Anfrage, die Vor- und Nachnamen der Mitarbeiter aller Abteilungen mit der Bezeichnung „Buchhaltung“ ausgibt, absteigend sortiert nach Mitarbeiter-ID.

Lösungsvorschlag

```

SELECT Vorname, Nachname
FROM Mitarbeiter m, Abteilung a
WHERE
    m.AnteilungsID = a.AnteilungsID AND
    a.Bezeichnung = 'Buchhaltung'
ORDER BY m.MitarbeiterID DESC;

```

```

vorname | nachname
-----+-----
Till    | Fuchs
Hans    | Meier
(2 rows)

```

- (b) Schreiben Sie eine SQL-Anfrage, die die Nachnamen aller Mitarbeiter mit dem Nachnamen ihres jeweiligen direkten Vorgesetzten ausgibt. Mitarbeiter ohne Vorgesetzten sollen in der Ausgabe ebenfalls enthalten sein. In diesem Fall soll der Nachname des Vorgesetzten NULL sein.

Lösungsvorschlag

```

SELECT m.Nachname AS Mitarbeiter, v.Nachname AS Vorgesetzter
FROM Mitarbeiter m LEFT OUTER JOIN Mitarbeiter v
ON m.Vorgesetzter = v.MitarbeiterID;

```

```

mitarbeiter | vorgesetzter
-----+-----
Meier       | Müller
Wolitz      | Müller
Müller      |
Fuchs       | Wolitz
Hase        | Müller
Navratil    |
Schmidt     | Navratil
(7 rows)

```

- (c) Schreiben Sie eine SQL-Anfrage, die die 10 Abteilungen ausgibt, deren Mitarbeiter das höchste Durchschnittsgehalt haben. Ausgegeben werden sollen der Rang (1 = höchstes Durchschnittsgehalt bis 10 = niedrigstes Durchschnittsgehalt), die Bezeichnung sowie das Durchschnittsgehalt der Abteilung. Gehen Sie davon dass es keine zwei Abteilungen mit gleichem Durchschnittsgehalt gibt. Sie können der Übersichtlichkeit halber Views oder With-Anweisungen verwenden. Verwenden Sie jedoch keine datenbankspezifischen Erweiterungen wie limit oder rownum.

Lösungsvorschlag

```

CREATE VIEW Durchschnittsgehälter AS
SELECT Abteilung.AnteilungsID, Bezeichnung,
    AVG (Gehalt) AS Durchschnittsgehalt
FROM Mitarbeiter, Abteilung

```

```

WHERE Mitarbeiter.AnteilungsID = Abteilung.AnteilungsID
GROUP BY Abteilung.AnteilungsID, Bezeichnung;

SELECT a.Bezeichnung, a.Durchschnittsgehalt, COUNT (*) AS Rang
FROM Durchschnittsgehalter a, Durchschnittsgehalter b
WHERE a.Durchschnittsgehalt <= b.Durchschnittsgehalt
GROUP BY a.AnteilungsID, a.Bezeichnung, a.Durchschnittsgehalt
HAVING COUNT(*) <= 10
ORDER BY Rang ASC;

```

bezeichnung	durchschnittsgehalt	rang
Managment	6514.5	1
Buchhaltung	2340	2
Vertrieb	1283.5	3
Produktion	654	4

(4 rows)

- (d) Schreiben Sie eine SQL-Anfrage, die das Gehalt aller Mitarbeiter aus der Abteilung mit der AbteilungsID 42 um 5% erhöht.

Lösungsvorschlag

```

vorname | nachname | gehalt
-----+-----+-----
Lea      | Müller   | 5875
Gerd     | Navratil | 7154
(2 rows)

SELECT Vorname, Nachname, Gehalt
FROM MITARBEITER
WHERE AbteilungsId = 42
ORDER BY Gehalt;

UPDATE Mitarbeiter
SET Gehalt = 1.05 * Gehalt
WHERE AbteilungsID = 42;

SELECT Vorname, Nachname, Gehalt
FROM MITARBEITER
WHERE AbteilungsId = 42
ORDER BY Gehalt;

vorname | nachname |      gehalt
-----+-----+-----
Lea      | Müller   | 6168.75
Gerd     | Navratil | 7511.700000000001
(2 rows)

```

- (e) Alle Abteilungen mit Bezeichnung „Qualitätskontrolle“ sollen zusammen mit den Datensätzen ihrer Mitarbeiter gelöscht werden. ON DELETE CASCADE ist für keine der Tabellen

gesetzt. Schreiben Sie die zum Löschen notwendigen SQL-Anfragen.

vorname	nachname
Hans	Meier
Fred	Wolitz
Lea	Müller
Till	Fuchs
Fred	Hase
Gerd	Navratil
Jürgen	Schmidt

(7 rows)

abteilungsid	bezeichnung
1	Buchhaltung
2	Vertrieb
42	Managment
4	Qualitätskontrolle
5	Produktion

(5 rows)

```
SELECT Vorname, Nachname FROM Mitarbeiter;
SELECT * FROM Abteilung;

DELETE FROM Mitarbeiter
WHERE AbteilungsID IN (
  SELECT a.AbteilungsID
  FROM Abteilung a
  WHERE a.Bezeichnung = 'Qualitätskontrolle'
);

DELETE FROM Abteilung
WHERE Bezeichnung = 'Qualitätskontrolle';

SELECT Vorname, Nachname FROM Mitarbeiter;
SELECT * FROM Abteilung;
```

vorname	nachname
Fred	Wolitz
Lea	Müller
Till	Fuchs
Gerd	Navratil
Jürgen	Schmidt

(5 rows)

abteilungsid	bezeichnung
--------------	-------------

-----+-----	
1	Buchhaltung
2	Vertrieb
42	Managment
5	Produktion
(4 rows)	

- (f) Alle Mitarbeiter sollen mit SQL-Anfragen nach den Telefonnummern anderer Mitarbeiter suchen können. Sie dürfen jedoch das Gehalt der Mitarbeiter nicht sehen können. Erläutern Sie in zwei bis drei Sätzen eine Möglichkeit, wie dies in einem Datenbanksystem realisiert werden kann, ohne die gegebenen Relationen, die Tabellen als abgelegt sind, zu verändern. Sie brauchen hierzu keinen SQL-Code schreiben.

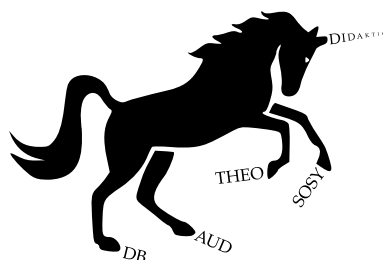
Lösungsvorschlag

Wir könnten eine VIEW erstellen, die zwar Namen und ID der anderen Mitarbeiter, sowie ihre Telefonnummern enthält (evtl. auch Abteilungsbezeichnung und ID), aber eben nicht das Gehalt: Mitarbeiter arbeiten auf eingeschränkter Sicht.

Alternativ mit GRANT:

explizit mit SELECT die Spalten auswählen, die man lesen können soll (auf nicht angegebene Spalten ist kein Zugriff möglich)

```
GRANT SELECT (Vorname, Nachname, Telefonnummer)
ON Mitarbeiter TO postgres;
```



## Die Bschlangaul-Sammlung

Hermine Bschlangaul and 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](mailto:hermine.bschlangaul@gmx.net). Der TeX-Quelltext dieser Aufgabe kann unter folgender URL aufgerufen werden: <https://github.com/bschlangaul-sammlung/examens-aufgaben-tex/blob/main/Examen/66116/2016/09/Thema-1/Teilaufgabe-1/Aufgabe-4.tex>