Einzelprüfung „Datenbanksysteme / Softwaretechnologie (vertieft)“

Einzelprüfungsnummer 66116 / 2015 / Herbst

Thema 1 / Teilaufgabe 1 / Aufgabe 3

(*Vater und Mutter*)

Stichwörter: SQL, SQL mit Übungsdatenbank

Gegeben seien folgende Relationen:

`Mensch : {[ ID, MutterID, VaterID ]}`

`Mann : {[ ID ]}`

`Frau : {[ ID ]}`

Additum: Übungsdatenbank

```sql
CREATE TABLE Mensch (
  ID INTEGER PRIMARY KEY,
  MutterID INTEGER,
  VaterID INTEGER
);

CREATE TABLE Mann (
  ID INTEGER PRIMARY KEY
);

CREATE TABLE Frau (
  ID INTEGER PRIMARY KEY
);

INSERT INTO Mensch VALUES
  (1, 42, 41),
  (2, 42, 41),
  (3, 42, 41),
  (4, 42, 41),
  (42, NULL, 1),
  (41, NULL, NULL);

INSERT INTO Mann VALUES
  (1),
  (3),
  (41);

INSERT INTO Frau VALUES
  (2),
  (4),
  (42);

CREATE VIEW VaterKind AS
SELECT Mensch.VaterID, Mensch.ID as KindID
FROM Mensch
```

```
WHERE
  Mensch.VaterID IS NOT NULL;
```

Das zugehörige ER-Modell für dieses relationale Datenbankschema sieht folgendermaßen aus:

Bearbeiten Sie folgende Teilaufgaben:

(a) Finden Sie die Töchter der Frau mit ID 42.

Lösungsvorschlag

```
SELECT Mensch.ID
FROM Mensch, Frau
WHERE
  Mensch.MutterID = Frau.id AND
  Frau.ID = 42;
```

(b) Gibt es Männer, die ihre eigenen Großväter sind? Formulieren Sie eine geeignete SQL-Anfrage.

Lösungsvorschlag

```
SELECT Mensch.ID
FROM Mann, Mensch
WHERE
  Mensch.ID = Mann.id AND (
    Mensch.VaterID IN (SELECT v.ID FROM Mensch v WHERE v.VaterID = Mensch.ID)
    OR
    Mensch.MutterID IN (SELECT v.ID FROM Mensch v WHERE v.VaterID = Mensch.ID)
  );
```

(c) Definieren Sie eine View VaterKind (VaterID; KindID), die allen Vätern (VaterID) ihre Kinder (KinderID) zuordnet. Diese View darf keine NULL-Werte enthalten.

Lösungsvorschlag

```
-- Wir erzeuge bereits beim Erstellen der Datenbank diese View, damit
-- sie für spätere Aufgaben zur Verfügung steht.
DROP VIEW IF EXISTS VaterKind;
CREATE VIEW VaterKind AS
SELECT Mensch.VaterID, Mensch.ID as KindID
FROM Mensch
WHERE
  Mensch.VaterID IS NOT NULL;
SELECT * FROM VaterKind;
```

(d) Verwenden Sie die View aus c), um alle Väter zurückzugeben, absteigend geordnet nach der Anzahl ihrer Kinder.

Lösungsvorschlag

```
SELECT VaterID, COUNT(VaterID) as Anzahl
FROM VaterKind
GROUP BY VaterID
```

```
ORDER BY Anzahl DESC;
```

(e) Hugo möchte mit folgender Anfrage auf Basis der View aus c) alle kinderlosen Männer erhalten:

```
SELECT VaterID
FROM VaterKind
GROUP BY VaterID
HAVING COUNT(KindID) = 0
```

(i) Was ist das Ergebnis von Hugos Anfrage und warum?

Lösungsvorschlag

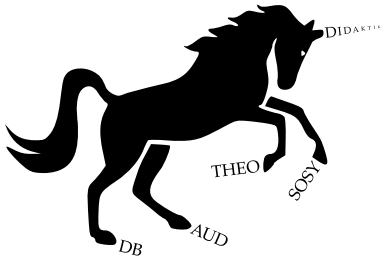
Die Anfrage liefert kein Ergebnis. Da die View laut Angabe keine Null-Werte enthalten darf, sind in der View nur Männer verzeichnet, die wirklich Väter sind.

(ii) Formulieren Sie eine Anfrage, die tatsächlich alle kinderlosen Männer zurückliefert.

Lösungsvorschlag

```
SELECT * FROM Mann
EXCEPT
SELECT VaterID
FROM VaterKind
GROUP BY VaterID;
```

Hinweis: Denken Sie daran, dass SQL auch Mengenoperationen kennt.



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.

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/66116/2015/09/Thema-1/Teilaufgabe-1/Aufgabe-3.tex`