



Übung zur Vorlesung *Einführung in die Informatik 2 für Ingenieure (MSE)*

Alexander van Renen (renen@in.tum.de)
<http://db.in.tum.de/teaching/ss20/ei2/>

Blatt Nr. 12

Dieses Blatt wird am Montag, den 20. Juli 2020 besprochen.

Aufgabe 1: SQL als DML

Gegeben sei ein erweitertes Universitätsschema mit den folgenden zusätzlichen Relationen *StudentenGF* und *ProfessorenF*:

StudentenGF : {[MatrNr : integer, Name : varchar(20), Semester : integer,
Geschlecht : char, FakName : varchar(20)]}
ProfessorenF : {[PersNr : integer, Name : varchar(20), Rang : char(2),
Raum : integer, FakName : varchar(20)]}

Die erweiterten Tabellen sind bereits auf der Webschnittstelle unter

<http://hyper-db.com/interface.html>

angelegt.

- Ermitteln Sie den Männeranteil an den verschiedenen Fakultäten in SQL! Beachten Sie dabei, dass es auch Fakultäten ohne Männer geben kann.
- Ermitteln Sie in SQL die Studenten, die alle Vorlesungen ihrer Fakultät hören. Geben Sie zwei Lösungen an, höchstens eine davon darf auf Abzählen basieren.

Aufgabe 2: Outer Join

In der Vorlesung haben wir den **left outer join** kennen gelernt:

```
select *  
from Studenten s left outer join  
hoeren h on s.matrNr = h.matrNr;
```

Ist es möglich eine semantisch äquivalente Anfrage zu formulieren ohne einen **left outer join** zu benutzen (selbverständlich ist auch der **right** und **full outer join** verboten).

Aufgabe 3: Fleißige Studenten

Formulieren Sie eine SQL-Anfrage, um die Studenten zu ermitteln, die mehr SWS belegt haben als der Durchschnitt. Berücksichtigen Sie dabei auch Totalverweigerer, die gar keine Vorlesungen hören.

Aufgabe 4: Sichten und Rekursion

Für diese Aufgabe fügen wir eine Spalte (Attribut/Column) zu der Professoren Relation hinzu: doktorElternteilPersNr. Diese gibt die Personennummer der Professorin an bei der eine Professorin ihren Dokortitel erhalten hat. Um eine endlose Kette von Professorinnen zu verhindern erlauben wir in dieser Spalte null Werte.

Erstellen Sie eine Sicht mit folgendem Schema: [persNr, name, generationen]. Die Generation einer Professorin ist durch die Anzahl von bekannten Doktoreltern definiert. Hat eine Professorin zum Beispiel einen null Eintrag in doktorElternteilPersNr, dann ist ihre Generation 0. Sind dagegen 2 Doktoreltern bekannt ist die Generation auch 2.

Das Webinterface beinhaltet diese zusätzliche Spalte nicht, falls Sie dennoch ihre Anfragen testen möchten, können Sie dies in einer lokalen Datenbank tun. Der Folgende Code kann in das die Schemadefinition eingearbeitet werden um die Tabelle mit der zusätzlichen Spalte zu erstellen.

```
CREATE TABLE Professoren
(PersNr          INTEGER PRIMARY KEY,
Name            VARCHAR(30) NOT NULL,
Rang            CHAR(2) CHECK (Rang in ('C2', 'C3', 'C4')),
Raum           INTEGER UNIQUE,
DoktorElternteil INTEGER,
FOREIGN KEY     (DoktorElternteil) REFERENCES Professoren);

INSERT INTO Professoren VALUES
(2125, 'Sokrates', 'C4', 226, null);
INSERT INTO Professoren VALUES
(2134, 'Augustinus', 'C3', 309, 2125);
INSERT INTO Professoren VALUES(2137, 'Kant', 'C4', 007, 2134);
INSERT INTO Professoren VALUES(2126, 'Russel', 'C4', 232, 2137);
INSERT INTO Professoren VALUES(2133, 'Popper', 'C3', 052, 2137);
INSERT INTO Professoren VALUES
(2127, 'Kopernikus', 'C3', 310, null);
INSERT INTO Professoren VALUES(2136, 'Curie', 'C4', 036, 2127);
```