

## Übung zur Vorlesung *Datenbanksysteme I* im WS 19/20

Gabriel Haas (gabriel.haas@uni-jena.de)  
<https://dbis1.github.io/courses/ws19/db1/>

### Blatt Nr. 11

Tool zum Üben von SQL-Anfragen: <https://hyper-db.com/interface.html>.

### Hausaufgabe 1

Gegeben sei die folgende SQL-Anfrage:

```
select p.Name, v.Titel, vi.anzahlHoerer
from Vorlesungen v,
( select vi.VorlNr, count(*) as anzahlHoerer
  from hoeren h join Vorlesungen vi on h.VorlNr = vi.VorlNr
  where vi.sws > 2
  group by vi.VorlNr
  having count(*) > 1) vi,
  Professoren p
where v.VorlNr = vi.VorlNr and p.PersNr = v.gelesenVon
```

Geben Sie die kanonische Übersetzung dieser Anfrage in die relationale Algebra an. Verwenden Sie zur Darstellung des relationalen Algebraausdrucks die Baumdarstellung.

Optimieren Sie Ihren relationalen Algebraausdruck logisch. Gehen Sie dabei von **realistischen** Kardinalitäten (für eine sehr große Universität) für die relevanten Relationen aus. Verwenden Sie hierfür die aus der Vorlesung bekannten Optimierungstechniken.

### Hausaufgabe 2

Gegeben sei ein Array von 1.000.000.000 8-Byte-Integer-Werten und ein Programm, das alle Werte aufsummiert.

Das Programm wird auf einem System mit 16 GB Hauptspeicher und einer herkömmlichen Magnetfestplatte (Größe 1 TB), auf der alle Werte sequentiell gespeichert sind, ausgeführt. Ein Random Access auf die Festplatte dauert 10 ms, beim sequentiellen Lesen hat sie einen Durchsatz von 160 MB/s. Das Summieren zweier Werte im Hauptspeicher dauert 1 ns.

(1 MB =  $10^6$  B und 1 TB =  $10^{12}$  B)

- Gehen Sie davon aus, dass alle Werte bereits im Hauptspeicher liegen. Wie lange läuft das Programm?
- Nun liegen alle Werte ausschließlich auf der Festplatte. Wie lange läuft das Programm jetzt?
- Auf der Festplatte liegt jetzt zusätzlich nach jedem 100.000. Wert die Summe der 100.000 davorliegenden Werte. Wie lange läuft das Programm, wenn es nur diese Summen aufsummiert?

### Hausaufgabe 3

Gegeben sei die folgende Relation **Zehnkampf** mit Athletennamen und den von ihnen erreichten Punkten im Zehnkampf:

Name	Punkte
Eaton	8869
Suarez	8523
Behrenbruch	8126
Hardee	8671
...	...

- Ermitteln Sie die Goldmedaillengewinner in SQL. (Eine Goldmedaille bekommen alle Athleten, für die gilt: es gibt niemand besseren (also mit mehr Punkten).)
- Ermitteln Sie die Silbermedaillengewinner in SQL. (Eine Silbermedaille bekommen alle, für die gilt: es gibt genau eine/n bessere/n.)

HINWEIS: Beachten Sie, dass die Relation **Zehnkampf** in der oben genannten Webschnittstelle nicht existiert. Verwenden Sie die folgende Syntax um temporäre Relationen zu erzeugen:

```
with zehnkampf(name,punkte) as (  
  values  
    ('Eaton', 8869),  
    ('Suarez', 8523),  
    ('Behrenbruch', 8126),  
    ('Hardee', 8671),  
    ('Sebrle', 8869)  
)  
select * from zehnkampf order by punkte desc
```

### Hausaufgabe 4

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

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

Die erweiterten Tabellen sind auch in der Webschnittstelle angelegt.

- Ermitteln Sie den Männeranteil an den verschiedenen Fakultäten in SQL!
- Ermitteln Sie in SQL die Studenten, die alle Vorlesungen ihrer Fakultät hören. Geben Sie zwei Lösungen an, höchstens eine davon darf auf Abzählen basieren.