



Übung zur Vorlesung *Grundlagen: Datenbanken* im WS18/19

Moritz Sichert, Lukas Vogel (gdb@in.tum.de)

<https://db.in.tum.de/teaching/ws1819/grundlagen/>

Blatt Nr. 11

Hausaufgabe 1

Gegeben sei die folgende SQL-Anfrage:

```
select distinct a.PersNr, a.Name
from Assistenten a, Studenten s, pruefen p
where s.MatrNr = p.MatrNr
      and a.Boss = p.PersNr
      and s.Name = 'Jonas';
```

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 die relevanten Relationen aus.

Verwenden Sie hierfür die folgenden aus der Vorlesung bekannten Optimierungstechniken:

- Aufbrechen von Selektionen
- Verschieben von Selektionen nach “unten” im Plan
- Zusammenfassen von Selektionen und Kreuzprodukten zu Joins
- Bestimmung der Joinreihenfolge

Hausaufgabe 2

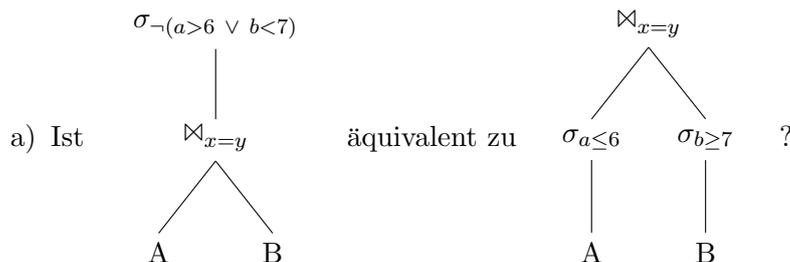
Gegeben seien die Relationen

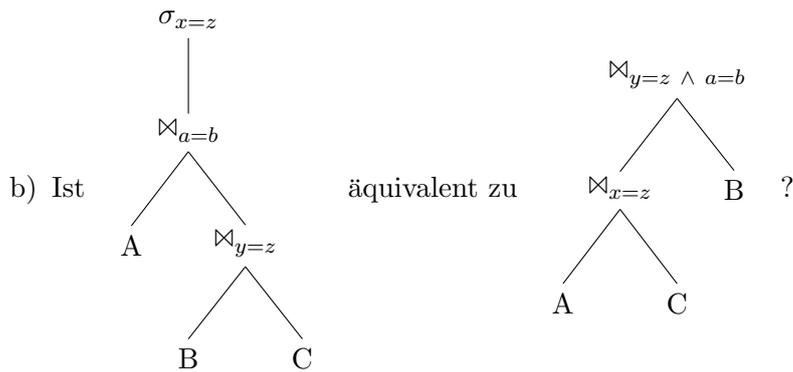
$A : \{[a, x]\}$

$B : \{[b, y]\}$

$C : \{[c, z]\}$

Im Folgenden sehen Sie jeweils zwei Operatorbäume der relationalen Algebra. Sind diese äquivalent zueinander? Beweisen oder widerlegen Sie mithilfe der zwölf äquivalenzerhaltenden Transformationsregeln aus der Vorlesung.





Hausaufgabe 3

Sie möchten folgende SQL-Anfrage auf dem bekannten Unischema auf einer Datenbank ausführen, die abhängige Unteranfragen („dependent subqueries“) nicht auflösen kann:

```
select s.matrnr, s.name from studenten s
where (
  select sum(v.sws) from vorlesungen v, hoeren h
  where v.vorlnr = h.vorlnr and h.matrnr = s.matrnr
) > 10
```

Die Unteranfrage im **where**-Teil ist abhängig von der äußeren Anfrage, da sie das Attribut **s.matrnr** verwendet.

- Warum ist diese Anfrage auf solch einer Datenbank ineffizient? Warum versuchen Datenbanken, abhängige Unteranfragen in unabhängige zu transformieren?
- Dekorrellieren („unnest“) Sie die Anfrage. Erstellen Sie dazu eine neue, äquivalente Anfrage, die keine abhängigen Unteranfragen enthält.

Hausaufgabe 4

Gegeben sind die beiden Relationenausprägungen:

<i>R</i>	
	A
...	0
...	5
...	7
...	8
...	8
...	10
⋮	⋮

<i>S</i>	
B	
5	...
6	...
7	...
8	...
8	...
11	...
⋮	⋮

Werten Sie den Join $R \bowtie_{R.A=S.B} S$ mithilfe des Nested-Loop- sowie des Sort/Merge-Algorithmus aus. Machen Sie deutlich, in welcher Reihenfolge die Tupel der beiden Relationen verglichen werden und kennzeichnen Sie die Tupel, die in die Ergebnismenge übernommen werden. Vervollständigen Sie hierzu die beiden folgenden Tabellen:

		<i>S.B</i>					
		5	6	7	8	8	11
<i>R.A</i>	0	1	2	3			
	5						
	7						
	8						
	8						
	10						

Nested-Loop-Join

		<i>S.B</i>					
		5	6	7	8	8	11
<i>R.A</i>	0	1					
	5	2✓					
	7						
	8						
	8						
	10						

Sort/Merge-Join