



## Übung zur Vorlesung *Grundlagen: Datenbanken* im WS16/17

Harald Lang, Linnea Passing (gdb@in.tum.de)

<http://www-db.in.tum.de/teaching/ws1617/grundlagen/>

### Blatt Nr. 13

#### 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

Für einen Join-Baum  $T$  sei folgende Kostenfunktion gegeben

$$C_{out}(T) = \begin{cases} 0 & \text{falls } T \text{ eine Basisrelation } R_i \text{ ist} \\ |T| + C_{out}(T_1) + C_{out}(T_2) & \text{falls } T = T_1 \bowtie T_2 \end{cases}$$

Die Kardinalität sei dabei

$$|T| = \begin{cases} |R_i| & \text{falls } T \text{ eine Basisrelation } R_i \text{ ist} \\ (\prod_{R_i \in T_1, R_j \in T_2} f_{i,j}) |T_1| |T_2| & \text{falls } T = T_1 \bowtie T_2 \end{cases}$$

Sei  $p_{i,j}$  das Join Prädikat zwischen  $R_i$  und  $R_j$ , dann sei

$$f_{i,j} = \frac{|R_i \bowtie_{p_{i,j}} R_j|}{|R_i \times R_j|}$$

und die Kardinalität eines Join-Resultats ist  $|R_i \bowtie_{p_{i,j}} R_j| = f_{i,j} |R_i| |R_j|$ .

Gegeben sei eine Anfrage über die Relationen  $R_1, R_2, R_3$  und  $R_4$  mit  $|R_1| = 10, |R_2| = 20, |R_3| = 20, |R_4| = 10$ . Die Selektivitäten der Joins seien  $f_{1,2} = 0.01, f_{2,3} = 0.5, f_{3,4} = 0.01$ , alle nicht gegebenen Selektivitäten sind offensichtlich 1 (Warum?). Berechnen Sie den optimalen (niedrigste Kosten) Join-Tree. Als Vereinfachung reicht es, wenn Sie nur Joins mit Prädikat und keine Kreuzprodukte betrachten.

### Hausaufgabe 3

Gegeben sind die beiden Relationenausprägungen:

$R$	
	A
...	0
...	5
...	7
...	8
...	8
...	10
⋮	⋮

$S$	
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:

		$S.B$					
		5	6	7	8	8	11
$R.A$	0						
	5						
	7						
	8						
	8						
	10						

		$S.B$					
		5	6	7	8	8	11
$R.A$	0						
	5	2✓					
	7						
	8						
	8						
	10						

Nested-Loop-Join

Sort/Merge-Join

### Hausaufgabe 4

- Was ist ein Equi-Join?
- Bei welchen Join-Prädikaten ( $<$ ,  $=$ ,  $>$ ) kann man sinnvoll einen Hashjoin einsetzen?
- Gegeben die Relation  $Prof s = \{\underline{PersNr}, Name\}$  und  $Raeume = \{\underline{PersNr}, RaumNr\}$ .
  - Skizzieren Sie eine geschickte Möglichkeit, den Equi-Join  $Prof s \bowtie_{Prof s.Persnr = Raeume.PersNr} Raeume$  durchzuführen.
  - In welchem Fall wäre selbst ein Ausdruck wie

$$Prof s \bowtie_{Prof s.Persnr < Raeume.PersNr} Raeume$$

effizient auswertbar?

- Der Student Maier hat einen Algorithmus gefunden, der den Ausdruck  $A \times B$  in einer Laufzeit von  $O(|A|)$  materialisiert. Was sagen Sie Herrn Maier?