



Übung zur Vorlesung *Grundlagen: Datenbanken* im WS17/18

Harald Lang, Linnea Passing (gdb@in.tum.de)
<http://www-db.in.tum.de/teaching/ws1718/grundlagen/>

Blatt Nr. 12

Hausaufgabe 1

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 2

Gegeben sind die beiden Relationenausprägungen:

R		S	
	A	B	
...	0	5	...
...	5	6	...
...	7	7	...
...	8	8	...
...	8	8	...
...	10	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	1	2	3			
	5						
	7						
	8						
	8						
	10						

Nested-Loop-Join

		S.B					
		5	6	7	8	8	11
R.A	0	1					
	5	2✓					
	7						
	8						
	8						
	10						

Sort/Merge-Join

Hausaufgabe 3

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

$$Profs \bowtie_{Profs.PersNr < Raeume.PersNr} Raeume$$

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?

Hausaufgabe 4

Gegeben sei die Anfrage:

```
select *
  from R, S, T
 where R.A = S.A and S.B = T.B and T.C = R.A
```

Des Weiteren soll gelten:

- S.A und T.C seien Fremdschlüssel auf R
- S.B sei Fremdschlüssel auf T
- R.A, T.B seien Primärschlüssel von R respektive T
- Ihre Query-Engine “kann” nur nested loops-Join
- Kardinalitäten: $|R|=100$, $|S|=1000$, $|T|=10$
- Es gibt keine Indexe

Bestimmen Sie den günstigsten QEP (query evaluation plan) auch als Baum mit Kosten-/Kardinalitäts-Abschätzungen. Verwenden Sie den in der Vorlesung gezeigten kostenbasierten DP (dynamisches Programmieren)-Optimierer.