Friedrich-Schiller-Universität Jena
Datenbanken und Informationssysteme
Prof. Dr. Viktor Leis

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. 12

Hausaufgabe 1

a) Was ist ein Equi-Join?

b) Bei welchen Join-Prädikaten ($<, =, >$) kann man sinnvoll einen Hashjoin einsetzen?

c) Gegeben die Relation Profs = {PersNr, Name} und Raeume = {PersNr, RaumNr}.

1) Skizzieren Sie eine geschickte Möglichkeit, den Equi-Join Profs ⋈ Raeume durchzuführen.

2) In welchem Fall wäre selbst ein Ausdruck wie

$$\text{Profs} \bowtie_{\text{Profs.Persnr} \,<\, \text{Raeume.PersNr}} \text{Raeume}$$

effizient auswertbar?

d) 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 2

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	1	2	3			
	5						
	7						
	8						
	8						
	10						

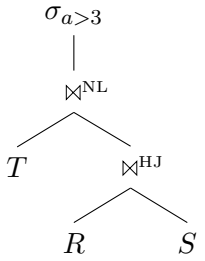
Nested-Loop-Join

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

Sort/Merge-Join

Hausaufgabe 3

Gegeben Sei der folgende physische Anfrageplan:



Hier steht $\bowtie^{\mathrm{NL}}$ für den Nested-Loop-Join und $\bowtie^{\mathrm{HJ}}$ für den Hash-Join. Geben Sie an, wie oft die `next`-Funktion von R aufgerufen wird, wenn ein Datenbanksystem diesen Anfrageplan mit dem Iteratorkonzept ausführt. Gehen Sie von folgenden Kardinalitäten aus: $|R| = 10$, $|S| = 20$ und $|T| = 5$. Gehen Sie davon aus, dass beim Nested-Loop-Join die linke Eingabe zuerst geöffnet wird und dass beim Hash-Join aus der linken Eingabe eine Hashtabelle erzeugt wird.