



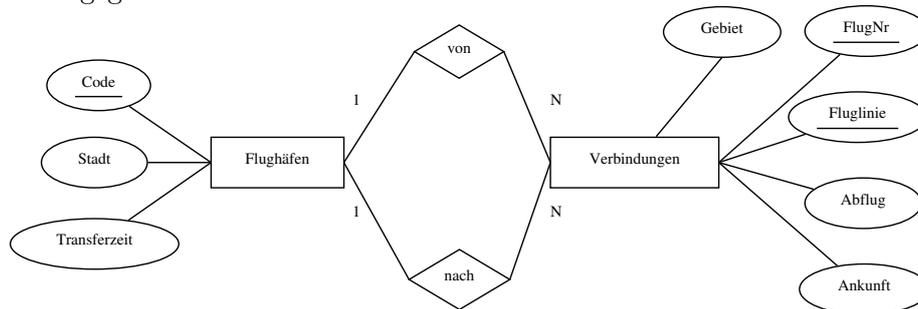
Übung zur Vorlesung *Grundlagen: Datenbanken im WS20/21*

Christoph Anneser, Josef Schmeißer, Moritz Sichert, Lukas Vogel (gdb@in.tum.de)
<https://db.in.tum.de/teaching/ws2021/grundlagen/>

Blatt Nr. 10

Hausaufgabe 1

Betrachten Sie die Anfrage „Finde alle Flüge von New-York nach Sydney mit einmaligem Umsteigen“. Dazu sei das nachfolgende (vereinfachte) ER-Diagramm eines Fluginformationssystems gegeben:



- Geben Sie eine SQL-Query für die oben genannte Anfrage an.
- Führen Sie die kanonische Übersetzung des SQL-Statements in die relationale Algebra durch.
- Schätzen Sie die Relationsgrößen sinnvoll ab (z.B. so wie in den Beispielen der Vorlesung) und transformieren Sie den kanonischen Operatorbaum aus Teilaufgabe b) zur optimalen Form. Wie haben sich die Kosten dabei geändert? (Kosten = Anzahl der Zwischenergebnistupel)

Hausaufgabe 2

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

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

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 3

Sie verwenden ein Datenbanksystem, welches die folgenden Joinalgorithmen unterstützt:

- (Blockwise-)Nested-Loop-Join
- Index-Join
- Sort-Merge-Join
- Hash-Join

Geben Sie für jeden der vier Algorithmen ein Beispiel an, bei dem dieses Datenbanksystem diesen Algorithmus verwenden würde. Geben Sie dazu für jedes Beispiel ein Schema und eine SQL-Anfrage an. Begründen Sie, warum der gewählte Joinalgorithmus den anderen vorgezogen wird.

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						
	5						
	7						
	8						
	8						
	10						

Nested-Loop-Join

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

Sort/Merge-Join

Hausaufgabe 5

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.