



## 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. 13

#### 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 sei die Anfrage:

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

Desweiteren 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.

### **Hausaufgabe 3**

Wofür stehen die vier Buchstaben ACID? Erklären Sie für jeden der vier Konzepte, warum es für eine Datenbank wichtig ist. Geben Sie dazu jeweils ein Beispiel an, was passieren könnte, wenn dieses Konzept nicht gelten würde.

### **Hausaufgabe 4**

Formulieren Sie die folgende Anfrage auf dem bekannten Unischema in SQL: Ermitteln Sie für jede Vorlesung, wie viele Studenten diese vorgezogen haben. Ein Student hat eine Vorlesung vorgezogen, wenn er in einem früheren Semester ist als der „Modus“ der Semester der Hörer dieser Vorlesung. Der Modus ist definiert als der Wert, der am häufigsten vorkommt – für diese Anfrage also das Semester, in dem die meisten Hörer dieser Vorlesung sind. Falls es mehrere Semester dieser Art gibt, soll nur das niedrigste zählen.

Beachten Sie, dass auch Vorlesungen ohne Hörer, sowie Vorlesungen deren Hörer alle im gleichen Semester sind, ausgegeben werden sollen.

Geben Sie für jede Vorlesung die Vorlesungsnummer, den Titel und die Anzahl der „Vorzieher“ aus.