



## Übung zur Vorlesung *Einsatz und Realisierung von Datenbanken im SoSe25*

Alice Rey, Maximilian Reif, Tobias Goetz (i3erdb@in.tum.de)

<http://db.in.tum.de/teaching/ss25/impldb/>

### Blatt Nr. 12

**Hinweise:** Dies ist das letzte Übungsblatt. Bitte nutzen Sie die Gelegenheit, offene Fragen oder Unklarheiten zu den Übungsblättern in das letzte Tutorium mitzubringen.

### Hausaufgabe 1

Berechnen Sie das Ergebnis des Joins zwischen den Relationen in der folgenden Abbildung:

R		S		T	
A	B	B	C	A	C
a <sub>0</sub>	b <sub>0</sub>	b <sub>0</sub>	c <sub>0</sub>	a <sub>0</sub>	c <sub>0</sub>
a <sub>0</sub>	b <sub>1</sub>	b <sub>0</sub>	c <sub>1</sub>	a <sub>0</sub>	c <sub>1</sub>
a <sub>0</sub>	b <sub>2</sub>	b <sub>0</sub>	c <sub>2</sub>	a <sub>0</sub>	c <sub>2</sub>
a <sub>0</sub>	b <sub>3</sub>	b <sub>0</sub>	c <sub>3</sub>	a <sub>0</sub>	c <sub>3</sub>
a <sub>1</sub>	b <sub>0</sub>	b <sub>1</sub>	c <sub>0</sub>	a <sub>1</sub>	c <sub>0</sub>
a <sub>2</sub>	b <sub>0</sub>	b <sub>2</sub>	c <sub>0</sub>	a <sub>2</sub>	c <sub>0</sub>
a <sub>3</sub>	b <sub>0</sub>	b <sub>3</sub>	c <sub>0</sub>	a <sub>3</sub>	c <sub>0</sub>

- Berechnen Sie das Ergebnis des Joins mithilfe von zwei binären Joins (z.B.  $(R \bowtie S) \bowtie T$ ). Wie viele Tupel enthält das Zwischenergebnis? Wie viele Tupel enthält das Endergebnis?
- Was ist der Vorteil von Worst-Case Optimal Joins? Erklären Sie das Prinzip des Leapfrog Triejoin Algorithmus.
- Bauen Sie aus den Relationen R, S und T Tries.
- Berechnen Sie erneut das Ergebnis des Joins, dieses Mal mithilfe des Leapfrog Triejoin Algorithmus.

### Hausaufgabe 2

Gegeben seien zwei dünnbesetzte Matrizen A und B.

$$A = \begin{pmatrix} 5 & 0 & 0 & 0 \\ 0 & 8 & 8 & 8 \\ 0 & 0 & 3 & 0 \\ 0 & 6 & 0 & 0 \end{pmatrix}$$

$$B = \begin{pmatrix} 1 & 2 & 0 & 0 \\ 0 & 3 & 0 & 4 \\ 0 & 0 & 5 & 6 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Da dünnbestzte Matrizen nur sehr wenige Positionen haben mit Nicht-Null-Einträgen, können diese sehr effizient gespeichert werden. Statt die gesamte Matrix zu speichern, wird für jeden Nicht-Null-Eintrag die Spalte, die Zeile, sowie deren Wert gespeichert:

Matrix A		
row	col	value
0	0	5
1	1	8
2	2	3
3	1	6

Matrix B		
row	col	value
0	0	1
0	1	2
1	1	3
1	4	4
2	2	5
2	3	6

Formulieren Sie eine Anfrage in SQL, die das Produkt der Matrizen A und B ( $A \times B$ ) berechnet.

```

with sparse_matrix_A (row, col, value) as (
  SELECT *
  FROM (VALUES (0,0,5), (1,1,8), (2,2,3), (3,1,6))
),
sparse_matrix_B (row, col, value) as (
  SELECT *
  FROM (VALUES (0,0,1), (0,1,2), (1,1,3), (1,4,4), (2,2,5), (2,3,6))
)

```

### Hausaufgabe 3

Beantworten sie die folgenden Fragen in SQL. Das Schema ist aus dem TPC-H Benchmark. Schreiben sie je zwei Anfragen - eine mit und eine ohne Window-Funktionen.

1. Bestimmen Sie den Rang der Bestellungen aus der orders Relation nach dem Gesamtpreis (totalprice). Der höchste Gesamtpreis soll den Rang 1 erhalten.
2. Berechnen sie für jede Bestellung die laufende Summe über den Gesamtpreis (totalprice) im jeweiligen Jahr (extract(year from o\_orderdate)).