

# Datenbanken II

## Übungsblatt 7 – WiSe 2019/20

20. Gegeben seien folgende Relationen:

- Relation  $R$ :  $n_R = 15.000$  Tupel, gespeichert auf  $b_R = 2.500$  Blöcken.
- Relation  $S$ :  $n_S = 1.000$  Tupel, gespeichert auf  $b_S = 30$  Blöcken.

Es steht ein Puffer der Größe  $M = 8$  Blöcke zur Verfügung. Es wird angenommen, dass es nur wenige Duplikate in den Join-Attributen gibt.

- a) [*Block-Nested-Loop-Join*]  
Berechnen Sie die Blockzugriffe für einen Block-Nested-Loop-Join  $S \bowtie R$ . Es wird der Zick-Zack-Modus verwendet:  $k = 5$  Block für  $R$ .
- b) [*Block-Nested-Loop-Join*]  
Berechnen Sie die Blockzugriffe für einen Block-Nested-Loop-Join  $R \bowtie S$ . Es wird der Zick-Zack-Modus verwendet:  $k = 5$  Block für  $S$ .
- c) [*Block-Nested-Loop-Join*]  
Berechnen Sie die Blockzugriffe für einen Block-Nested-Loop-Join  $R \bowtie S$ . Es wird der Zick-Zack-Modus verwendet:  $k = 1$  Block für  $S$ .
- d) [*Indexed-Nested-Loop-Join*]  
Berechnen Sie die Blockzugriffe für einen Indexed-Nested-Loop-Join  $S \bowtie R$ . *Annahme*: Auf Relation  $R$  existiert ein  $B^+$ -Baum-Index mit  $m = 2^5 = 32$ .

21. [*Merge-Join*]

Gegeben seien folgende Relationen:

- Relation  $R[A, B]$ :  $n_R = 15.000$  Tupel, gespeichert auf  $b_R = 2.500$  Blöcken.
- Relation  $S[B, C]$ :  $n_S = 1.000$  Tupel, gespeichert auf  $b_S = 30$  Blöcken.

Es steht ein Puffer der Größe  $M = 8$  Blöcke zur Verfügung. Es wird angenommen, dass es nur wenige Duplikate in den Join-Attributen gibt. Berechnen Sie die Blockzugriffe für einen Merge-Join  $R \bowtie S$

- a.) **mit** Clustered  $B^+$ -Baum-Index auf den Join-Attributen.
- b.) **ohne** Clustered  $B^+$ -Baum-Index auf den Join-Attributen.

22. [*Hash-Join*]

Gegeben sind die Relationen und der Puffer aus Aufgabe 21. Berechnen Sie die Blockzugriffe für einen Hash-Join  $R \bowtie S$ . Geben Sie außerdem an, welche Relation als Probe- bzw. Build-Input verwendet wird.

23. [Hash-Join]

Der Join  $R \bowtie S$ ,  $R(A)$ ,  $S(A)$ , soll als Hash-Join ausgeführt werden, wobei  $R = \{1, 6, 11, 18, 25, 31, 28\}$ ,  $S = \{1, 3, 6, 7, 9, 11, 23, 25, 27, 30, 8, 19, 17\}$ . Es stehen  $M = 3$  Blöcke im Puffer zur Verfügung, jeder Block fasst 1 Tupel.

**Anmerkung:** Im Allgemeinen (d.h. es werden mehrere Tupel pro Block gespeichert) kann eine Relation in maximal  $M - 1$  Partitionen zerlegt werden und nur dann gelten die Kosten von  $3 \cdot (b_R + b_S)$ . Da in dieser Aufgabe aber lediglich 1 Tupel pro Block gespeichert wird, können die Kosten von  $3 \cdot (b_R + b_S)$  auch mit einem kleineren Puffer garantiert werden.

- a) Welche Relation wird als Build-Input verwendet?
- b) Verwenden Sie zum Erstellen der Partitionen eine Hashfunktion der Form  $h(x) = x \bmod c$  und bestimmen Sie einen geeigneten Wert für  $c$ . Geben Sie die **Partitionen des Build-Inputs** an.
- c) Führen Sie den Join aus. Für den Hauptspeicher-Hash-Index wird der 5-stellige Binärwert als Hashfunktion und erweiterbares Hashing verwendet. Nehmen Sie an, dass das Verzeichnis nicht im Puffer gespeichert werden muss und leere Buckets keinen Platz benötigen. Visualisieren Sie die **Hash-Indizes** und geben Sie die **Partitionen des Probe-Inputs** sowie das **Join-Ergebnis** an.