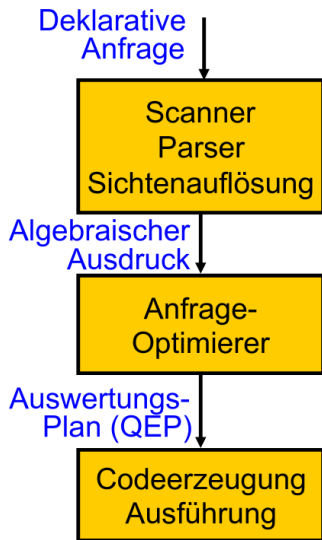


Datenbanken: Tutorium 11

Marvin Jahn

15.01.2020

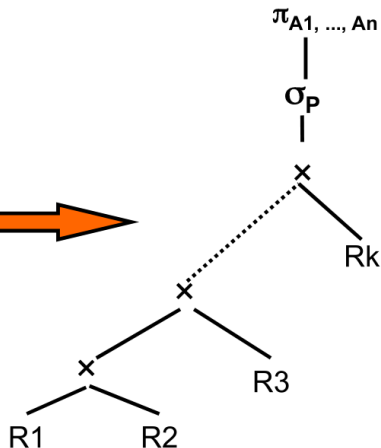
Ablauf der Anfragebearbeitung



Kanonische Übersetzung

Kanonische Übersetzung: Standardübersetzung von SQL in relationale Algebra

select A1, ..., An
from R1, ..., Rk
where P



Kanonische Übersetzung

- `select a, sum(d) as s from ... group by a, b, c`
 $\rightarrow \pi_{a,s}(\Gamma_{a,b,c;s:sum(d)}(C))$, wobei C die kanonische Übersetzung des inneren Teils ist

Kanonische Übersetzung

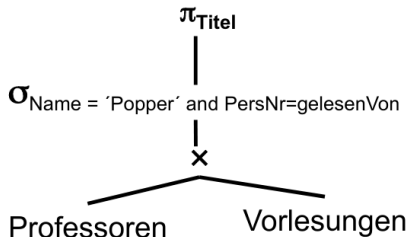
- `select a, sum(d) as s from ... group by a, b, c`
 $\rightarrow \pi_{a,s}(\Gamma_{a,b,c;s:sum(d)}(C))$, wobei C die kanonische Übersetzung des inneren Teils ist
- `select ... having p`
 $\rightarrow \sigma_p(C)$, wobei C die kanonische Übersetzung inklusive *group by* ist

Kanonische Übersetzung

- `select a, sum(d) as s from ... group by a, b, c`
→ $\pi_{a,s}(\Gamma_{a,b,c;s:sum(d)}(C))$, wobei C die kanonische Übersetzung des inneren Teils ist
- `select ... having p`
→ $\sigma_p(C)$, wobei C die kanonische Übersetzung inklusive *group by* ist
- `select ... order by a, b, c`
→ $sort_{a,b,c}(C)$, wobei C die kanonische Übersetzung inklusive *having* ist

Kanonische Übersetzung: Beispiel

select Titel
from Professoren, Vorlesungen
where Name = 'Popper' **and**
PersNr = gelesenVon



$\pi_{\text{Titel}} (\sigma_{\text{Name} = \text{'Popper'} \text{ and } \text{PersNr} = \text{gelesenVon}} (\text{Professoren} \times \text{Vorlesungen}))$

Implementierung der *join* Operation

Verschiedene Algorithmen zur Berechnung der *join* Operation

$R \bowtie_{R.A=S.B} S$:

Implementierung der *join* Operation

Verschiedene Algorithmen zur Berechnung der *join* Operation

$R \bowtie_{R.A=S.B} S$:

(Blockwise-) Nested-Loop-Join (“brute force”)

for each $r \in R$

for each $s \in S$

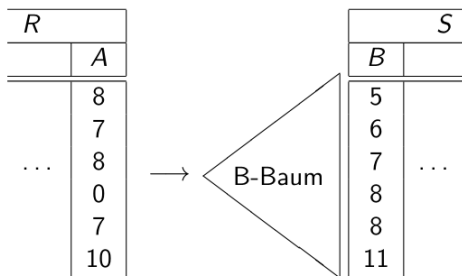
if $r.A = s.B$ **then**

$\text{res} := \text{res} \cup (r \times s)$

Nachteil: i.A. ineffizient

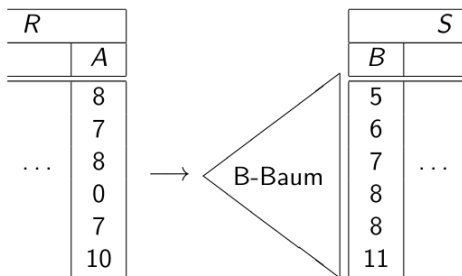
Implementierung der *join* Operation

Index-Join



Implementierung der *join* Operation

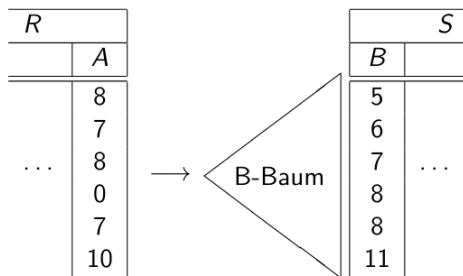
Index-Join



- Nehme Element aus linker Eingabe.

Implementierung der *join* Operation

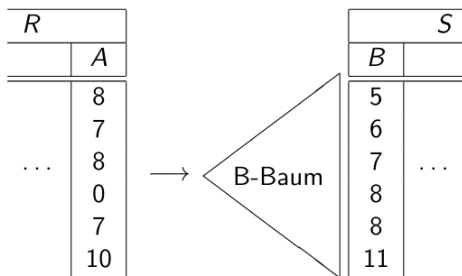
Index-Join



- Nehme Element aus linker Eingabe.
- Schlage Joinattribut im Index nach.

Implementierung der *join* Operation

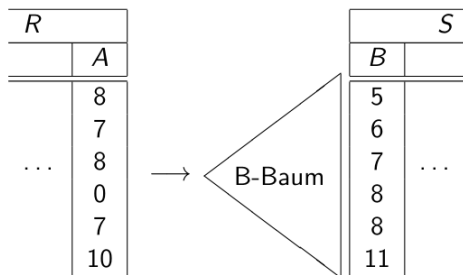
Index-Join



- Nehme Element aus linker Eingabe.
- Schlage Joinattribut im Index nach.
- Bilde join, falls Index ein (weiteres) Tupel zu diesem Wert liefert, ansonsten bewege linke Eingabe vor.

Implementierung der *join* Operation

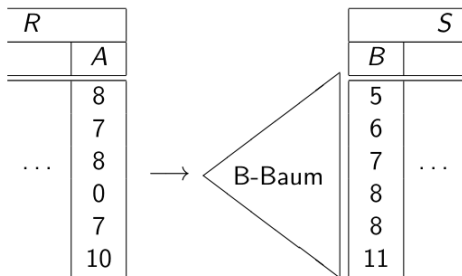
Index-Join



- Nehme Element aus linker Eingabe.
- Schlage Joinattribut im Index nach.
- Bilde join, falls Index ein (weiteres) Tupel zu diesem Wert liefert, ansonsten bewege linke Eingabe vor.
- Iteriere.

Implementierung der *join* Operation

Index-Join

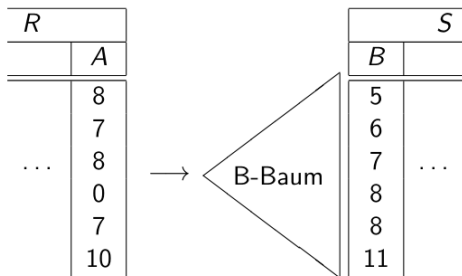


Nachteile:

- auf Zwischenergebnissen existieren keine Indexstrukturen

Implementierung der *join* Operation

Index-Join

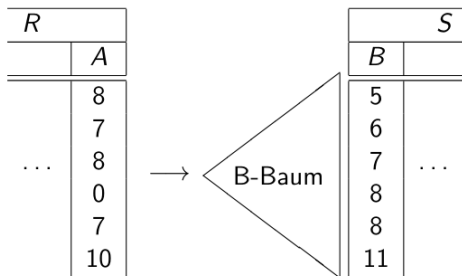


Nachteile:

- auf Zwischenergebnissen existieren keine Indexstrukturen
- temporäres Anlegen i.A. zu aufwendig

Implementierung der *join* Operation

Index-Join



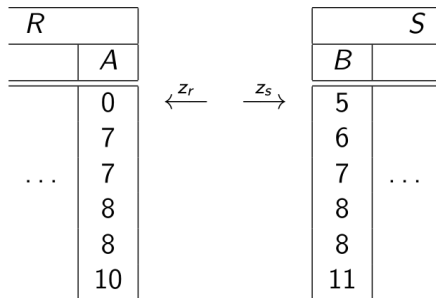
Nachteile:

- auf Zwischenergebnissen existieren keine Indexstrukturen
- temporäres Anlegen i.A. zu aufwendig
- Nachschlagen im Index i.A. zu aufwendig

Implementierung der *join* Operation

Sort-Merge-Join

Voraussetzung: R und S sind sortiert.



Implementierung der *join* Operation

Hash-Join:

- R und S werden mittels der gleichen Hashfunktion h , angewendet auf $R.A$ und $S.B$ auf dieselben Hash-Buckets abgebildet.

Implementierung der *join* Operation

Hash-Join:

- R und S werden mittels der gleichen Hashfunktion h , angewendet auf $R.A$ und $S.B$ auf dieselben Hash-Buckets abgebildet.
- Zu verbindende Tupel befinden sich dann im selben Bucket.

Implementierung der *join* Operation

Hash-Join:

- R und S werden mittels der gleichen Hashfunktion h , angewendet auf $R.A$ und $S.B$ auf dieselben Hash-Buckets abgebildet.
- Zu verbindende Tupel befinden sich dann im selben Bucket.
- Ist i.A. am Schnellsten; Merge-Join ist schneller, falls die Relationen schon vorsortiert sind.