


Beispiel der Nested Loop Join Implementierung

Beispiel: Natural Join der Relationen L und R

L	
A	B
a_1	b_1
a_2	b_2
a_3	b_3
a_4	b_4



R	
B	C
b_1	c_1
b_2	c_2
b_3	c_3
b_4	c_4
b_5	c_5
b_6	c_6

Algorithmus:

for each $l \in L$

for each $r \in R$

if $l.B = r.B$ then

res := res \cup ($l \times r$)

In der Praxis ist der Hauptspeicher begrenzt und Teile der Relationen müssen nacheinander in den Speicher geladen werden:

- In diesem Beispiel passen 2 Tupel in einen Block (Seite)
- Im Hauptspeicher stehen $m = 2$ Blöcke für die Join-Berechnung zur Verfügung
- Das Ergebnis wird direkt ausgegeben (Display, Netzwerk, etc.)
- Es werden $k = 1$ Seiten für die innere Schleife und $m - k = 2 - 1 = 1$ Seiten für die äußere Schleife im Hauptspeicher reserviert
- Für alle Portionen zu $m - k$ Seiten der äußeren Relation L, muss die komplette innere Relation R zu k Portionen durchlaufen werden

In diesem Beispiel wird die *Simple Nested Loop Join* Implementierung mit folgenden Optimierungsvarianten verbessert:

- Pro Durchlauf kann das Einlesen von k Seiten gespart werden wenn die innere Relation im Zick-Zack-Verfahren (*Rocking*) durchlaufen wird.
- *Nestes Blocks*: Es werden nicht nur einzelne Tupel, sondern alle Tupel eines Blocks verarbeitet (Sätze werden immer in ganzen Blöcken gelesen).

Schritt 1:

Hauptspeicher:

Block 1	a_1	b_1	
	a_2	b_2	
Block 2	b_1	c_1	
	b_2	c_2	

Ergebnis:

A	B	C
a_1	b_1	c_1
a_2	b_2	c_2

Schritt 2:

Hauptspeicher:

Block 1	a_1	b_1	
	a_2	b_2	
Block 2	b_3	c_3	
	b_4	c_4	

Ergebnis:

A	B	C
a_1	b_1	c_1
a_2	b_2	c_2

Schritt 3:

Hauptspeicher:

Block 1	a_1	b_1
	a_2	b_2
Block 2	b_5	c_5
	b_6	c_6

Ergebnis:

A	B	C
a_1	b_1	c_1
a_2	b_2	c_2

Schritt 4:

Hauptspeicher (Block 2 muss in diesem Schritt nicht erneut gelesen werden, aufgrund des *Zick-Zack-Verfahrens!*):

Block 1	a_3	b_3
	a_4	b_4
Block 2	b_5	c_5
	b_6	c_6

Ergebnis:

A	B	C
a_1	b_1	c_1
a_2	b_2	c_2

Schritt 5:

Hauptspeicher:

Block 1	a_3	b_3
	a_4	b_4
Block 2	b_3	c_3
	b_4	c_4

Ergebnis:

A	B	C
a_1	b_1	c_1
a_2	b_2	c_2
a_3	b_3	c_3
a_4	b_4	c_4

Schritt 6:

Hauptspeicher:

Block 1	a_3	b_3
	a_4	b_4
Block 2	b_1	c_1
	b_2	c_2

Ergebnis:

A	B	C
a_1	b_1	c_1
a_2	b_2	c_2
a_3	b_3	c_3
a_4	b_4	c_4