



Übung zur Vorlesung *Grundlagen: Datenbanken* im WS14/15

Harald Lang (harald.lang@in.tum.de)

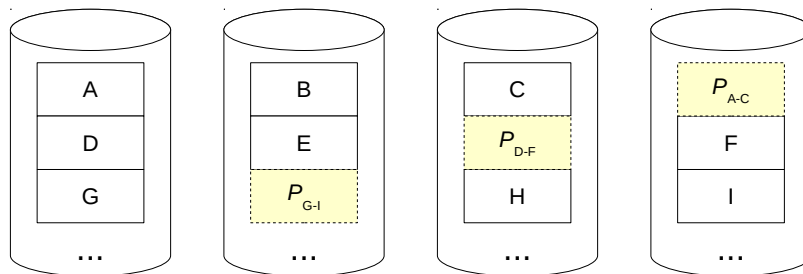
<http://www-db.in.tum.de/teaching/ws1415/grundlagen/>

Blatt Nr. 11

Dieses Blatt wird in KW 52 (Mo-Di) sowie in KW 2 (Mi-Fr) in den Übungen besprochen.

Hausaufgabe 1

Die folgende Abbildung zeigt einen Festplattenverbund bestehend aus vier Laufwerken, auf welchen die Datenblöcke A bis I gespeichert sind. Die Blöcke P_i enthalten Paritätsinformationen.



- Um welches RAID-Level handelt es sich?
- Wieviele Festplatten können ausfallen, ohne dass mit Datenverlust zu rechnen ist? Geben Sie eine allgemeine Lösung für einen Verbund bestehend aus n Festplatten an.
- Kann die Ausfallsicherheit erhöht werden? Begründung?
- Welchen weiteren Vorteil bietet das gezeigte RAID-System neben der Ausfallsicherheit?
- Nach einem Festplattendefekt, enthalten die Datenblöcke die folgenden Binärdaten. Rekonstruieren Sie die Datenblöcke der $Disk_2$ mithilfe der XOR-Verknüpfung.

$Disk_0$	$Disk_1$	$Disk_2$	$Disk_3$
A = 1111	B = 1001	C = - - - -	$P_{A-C} = 1110$
D = 0101	E = 1100	$P_{D-F} = - - - -$	F = 1100
G = 0011	$P_{G-I} = 1110$	H = - - - -	I = 0011

- 5
- 1, unabhängig von n .
- Ja, z.B. mit einem RAID-6 (Ausfall zweier Platten kann kompensiert werden) oder RAID-15 (das RAID-5 wird zusätzlich nochmal gespiegelt).
- Höherer Datendurchsatz.
- Die Rekonstruktion der Datenblöcke unterscheidet sich rechnerisch nicht von der Berechnung der Parität.

$Disk_0$	$Disk_1$	$Disk_2$	$Disk_3$
A = 1111	B = 1001	C = 1000	P_{A-C} = 1110
D = 0101	E = 1100	P_{D-F} = 0101	F = 1100
G = 0011	P_{G-I} = 1110	H = 1110	I = 0011

Hausaufgabe 2

- (a) Was ist ein Equi-Join?
- (b) Was spricht gegen die Verwendung des Hashjoin-Verfahrens bei einem Join, der etwa ein “<”-Zeichen im Prädikat enthält?
- (c) Gegeben die Relation $Prof s = \{ \underline{PersNr}, Name \}$ und $Raeume = \{ \underline{PersNr}, RaumNr \}$.
- 1) Skizzieren Sie eine geschickte Möglichkeit, den Equi-Join $Prof s \bowtie Raeume$ durchzuführen.
 - 2) Wieso ist dies hier „angenehm“ durchführbar?
 - 3) In welchem Fall wäre selbst ein Ausdruck wie $Prof s \bowtie_{Prof s.Persnr < Raeume.PersNr} Raeume$ angenehm durchführbar?
- (d) Der Student Maier hat einen Algorithmus gefunden, der den Ausdruck $A \times B$ in einer Laufzeit von $O(|A|)$ materialisiert. Was sagen Sie Herrn Maier?
- (a) Ein Equi-Join hat eine Äquivalenz als Joinbedingung, etwa die Gleichheit zweier Attribute.
- (b) Ein Hash Join bietet sich nur für Equi-Joins an, da lediglich ein Join-Partner mit gleichem Attributwert effizient auffindbar ist. Das Finden eines Partners, dessen Attributwert beispielsweise kleiner sein soll kann mittels Hashing i.A. nicht effizient bearbeitet werden.
- (c) 1) Offenbar ist das Joinattribut gerade der Primärschlüssel, womit von der Existenz eines Indexes ausgegangen werden kann. Somit bietet sich ein Index-basierter Join an, etwa dadurch, dass die eine Relation Element für Element abgearbeitet wird, während Joinpartner aus der anderen Relation mittels des Indexes gefunden werden.
- 2) Angenehm weil wahrscheinlich Index vorhanden und Equi-Join.
 - 3) Falls der Index sortiert ist, dies wäre etwa bei einem B-Baum der Fall. Dadurch liegen Joinpartner zumindest nacheinander im Index, anders als bei einer Implementierung des Indexes mittels Hash.
- (d) Dies ist mit Sicherheit nicht der Fall, da ein Algorithmus keine bessere Komplexitätsklasse haben kann als sein Ergebnis wächst. Mit anderen Worten, $A \times B$ hat eine Ergebnisgröße von $|A| * |B|$ und dieses Ergebnis kann sicher nicht schneller als in $O(|A| * |B|)$ materialisiert werden.

Hausaufgabe 3

Gegeben sind die beiden Relationenausprägungen:

<i>R</i>	
	A
...	0
...	5
...	7
...	8
...	8
...	10
⋮	⋮

<i>S</i>	
B	
5	...
6	...
7	...
8	...
8	...
11	...
⋮	⋮

Werten Sie den Join $R \bowtie_{R.A=S.B} S$ mithilfe des Nested-Loop- sowie des Sort/Merge-Algorithmus aus. Machen Sie deutlich, in welcher Reihenfolge die Tupel der beiden Relationen verglichen werden und kennzeichnen Sie die Tupel, die in die Ergebnismenge übernommen werden. Vervollständigen Sie hierzu die beiden folgenden Tabellen:

		<i>S.B</i>					
		5	6	7	8	8	11
<i>R.A</i>	0	1	2	3			
	5						
	7						
	8						
	8						
	10						

Nested-Loop-Join

		<i>S.B</i>					
		5	6	7	8	8	11
<i>R.A</i>	0	1					
	5	2✓					
	7						
	8						
	8						
	10						

Sort/Merge-Join

		<i>S.B</i>					
		5	6	7	8	8	11
<i>R.A</i>	0	1	2	3	4	5	6
	5	7✓	8	9	10	11	12
	7	13	14	15✓	16	17	18
	8	19	20	21	22✓	23✓	24
	8	25	26	27	28✓	29✓	30
	10	31	32	33	34	35	36

Nested-Loop-Join

		<i>S.B</i>					
		5	6	7	8	8	11
<i>R.A</i>	0	1					
	5	2✓	3				
	7		4	5✓			
	8			6	7✓	10✓	
	8				8✓	11✓	
	10				9	12	13

Sort/Merge-Join

Ausführliche Lösung:

http://www-db.in.tum.de/teaching/ws1415/grundlagen/Loesung11_sort_merge_join.pdf