



Übungen zur Vorlesung “Datenbanksysteme II”

SS 2014

Benjamin Dietrich (b.dietrich@uni-tuebingen.de)

9. Übungsblatt




Ausgabe: 4. Juni 2014 · Abgabe: 17. Juni 2014

Aufgabe 1: Set Operators

(4 Punkte)

Hash- und sortierungsbasierte Ansätze können nicht nur für Joins, sondern auch für andere Operatoren der relationalen Algebra eingesetzt werden. An den folgenden Beispielen soll die Semantik der beiden Operatoren *relationale Vereinigung* (\cup) und *Schnittmenge* (\cap) erläutert werden:

- Vereinigung (\cup)

 R ₁			 R ₂			 R _{res}		
foo ₁	foo ₂		foo ₁	foo ₂		foo ₁	foo ₂	
2	4	∪	2	4	=	2	4	
6	1		3	5		6	1	
10	5		4	4		10	5	
8	4		4	4		8	4	
			10	5		3	5	
			6	5		4	4	
						6	5	

- Schnittmenge (\cap)

\mathbb{R}_1			\mathbb{R}_2				\mathbb{R}_{res}		
foo ₁	foo ₂		foo ₁	foo ₂			foo ₁	foo ₂	
2	4		2	4			2	4	
6	1		3	5					
10	5		4	4					
8	4		10	5			10	5	
			6	5					

Skizzieren Sie Algorithmen in Pseudocode (wie auf Folie 39, Evaluation of Relational Operators), welche diese beiden Operatoren der relationalen Algebra implementieren. Schreiben Sie dabei jeweils sowohl einen Algorithmus mit einem hash-, sowie einem sortierungsbasierten Vorgehen.

Aufgabe 2: Outer Joins

(4 Punkte)

In der Vorlesung haben wir verschiedene Implementationen von *inner joins* besprochen. Im Folgenden wollen wir diese erweitern, um *left outer joins*¹ auszuwerten:

$S \text{ NATURAL LEFT OUTER JOIN } R$

Eine ineffiziente Strategie deren Verhalten zu implementieren, wäre es, zunächst einen *inner join* zwischen R und S auszuführen und in einem zweiten Schritt die noch fehlenden, mit **NULL**-Werten aufgefüllten Tupel hinzuzufügen. Beschreiben Sie, wie man stattdessen die *block nested loop join* und *index nested loop join* Algorithmen verändern kann, um *left outer joins* **effizient** zu berechnen.

Aufgabe 3: Selectivity and I/O Costs

(8 Punkte)

Es sei $R(a, b, c, d)$ eine Relation mit a als Kandidatenschlüssel. R enthält 1.000.000 Tupel, wobei 10 Tupel auf genau eine Seite passen. Die Werte in R sind unabhängig voneinander und ihre Verteilung ist uniform.

Folgende Informationen seien im Katalog des RDBMS enthalten:

SYSCAT			
Attribut	max Wert	min Wert	Anzahl unterschiedlicher Werte
a	2.000.000	10	1.000.000
b	10.000.000	1	30
c	1.000	-30.000	31.000
d	2.500.000	27	232

1. Berechnen Sie die Selektivitäten für die folgenden Prädikate.

- (a) $a = 23$
- (b) $b > 12 \wedge b > 42$
- (c) $a > 13 \wedge (c > 12 \vee b < 30)$
- (d) $a > 56 \wedge (b = 23 \wedge c = 21)$
- (e) $d = 12 \vee c < 12$

2. Berechnen Sie die I/O Kosten für die Selektionen der Prädikate aus der ersten Teilaufgabe, basierend auf den folgenden Situationen:

- (a) Unsortierte Heap File, kein Index.
- (b) Ein clustered B+ Baum Index, mit Schlüssel (a, b, c) und Höhe 4.
- (c) Hash Index auf dem Attribut a , mit 1, 2 I/Os pro Lookup.

(Hinweis: Ignorieren Sie eventuelle Kosten der Operatoren \cup^{rid} und \cap^{rid})

¹Falls erforderlich, gibt Wikipedia Auskunft über die Semantik von Left Outer Joins: http://en.wikipedia.org/wiki/Outer_join#Left_outer_join

Aufgabe 4: Volcano Iterator Model

(6 Punkte)

In der Vorlesung wurde das *Volcano Iterator Model* vorgestellt. Hierbei handelt es sich um ein Design-Pattern das gut in den Kontext der Auswertung von relationalen Operatoren passt und sogenanntes pipelining ermöglicht.

Jeder Volcano Style Operator verfügt über eine Schnittstelle mit den Methoden:

- `open ()`
- `next ()`
- `close ()`

1. Geben Sie Volcano Style Algorithmen in Pseudocode für die folgenden Operatoren an:

- (a) **UNION ALL**: Vereinigung zweier Relation R und S *ohne* Duplikateliminierung.
- (b) **UNION**: Vereinigung zweier Relationen R und S *mit* Duplikateliminierung.
- (c) **SUM()**-Aggregation: Addition der Tuplel einer Relationen R *ohne* Gruppierung.

2. Klassifizieren Sie die Operatoren der vorhergehenden Aufgabe in *blockierende* und *nicht blockierende* Operatoren.