

CS411 Database Systems  
*Spring 2007*

*HW #4*

Due: 1:55pm CST, 4/26/07

Note: Print your name and NetID in the upper right corner of every page of your submission. Handin your homework to Trisha Benson in 4322 SC. In case Trisha is not in office, slide your homework under the door.

To grade homeworks faster, the homework is partitioned into three parts. Please, submit each part separately. For each part, make sure to write down your name and NetID. This homework is partitioned into 3 parts as follows:

Part 1: Problem 1 - Problem 2

Part 2: Problem 3 - Problem 4

Part 3: Problem 5 - Problem 6

There is a bonus credit of 2% if your homework is formatted correctly: Each part must be separable so that it can be independently graded. If you submit a paper copy then each part should i) be separately stapled and ii) include your netid. For I2Cs students: If you submit an electronic copy (or fax) then each part should start on a new page and include your netid at the top of each page.

Handwritten submissions will be graded but they will take longer to grade. For clarity, machine formatted text is preferable: Expect to lose points if your handwritten answer is unclear or misread by the grader.

## Problem 1 Query Execution Suggested reading: Chapter 15

Suppose you have 2 relations,  $R(A, B, C)$  and  $S(B, C, D, E)$ . You have a clustered unique (no duplicate keys) B+-tree index on attribute A for relation R. Assume this index is kept entirely in memory (i.e., you do not need to read it from disk).

For relation S, you have two indexes: (i) a non-clustered non-unique B+-tree index for attribute B, and (ii) a clustered non-unique B+-tree index for attribute C. Assume that these two indexes are also kept in memory. Also, assume that all of the tuples of S that agree on attribute C are stored in sequentially adjacent blocks on disk (that is, if more than one block is needed to store all of the tuples with some value of C, then these blocks will be sequentially located on the disk).

Other relevant data:

- 2000 tuples of R are stored per block on disk.
- $T(R) = 2,000,000$  (number of tuples of R)
- 200 tuples of S are stored per block on disk.
- $T(S) = 200,000$  (number of tuples of S)
- $V(B,S)=40,000$  (number of distinct values of attribute B in S)
- $V(C,S)=800$  (number of distinct values of attribute C in S)

You want to execute the following query:

```
SELECT *  
FROM R, S  
WHERE (R.B=S.B) AND (R.C=S.C)
```

We present you with two query plans:

Plan 1:

- 1: For every block BL of R, retrieved using the clustered index on A for R
- 2: For every tuple r of BL
- 3: Use the index on B for S to retrieve all of the tuples s of S such that  $s.B=r.B$
- 4: For each of these tuples s, if  $s.C=r.C$ , output r.A, r.B, r.C, s.B, s.C, s.D, s.E

Plan 2:

- 1: For every block BL of R, retrieved using the clustered index on A for R
- 2: For every tuple r of BL
- 3: Use the index on C for S to retrieve all of the tuples s of S such that  $s.C=r.C$
- 4: For each of these tuples s, if  $s.B=r.B$ , output r.A, r.B, r.C, s.B, s.C, s.D, s.E

(a) Analyze the above two plans carefully in terms of their behavior regarding accesses to disk, and explain which of the plans is better. Be sure to include in your analysis which accesses to disk are sequential accesses and which ones are random accesses.

(b) Suggest an alternative plan that is better than the above two in terms of disk access? You do not need to submit detail computations for this question. Briefly explain your assumptions and your alternative plan and argue why your plan is better.

**Problem 2** Query Execution Cont Suggested reading: Chapter 15

Consider using simple sort-merge join to join two relations  $\mathbf{R}(\mathbf{x},\mathbf{y})$  and  $\mathbf{S}(\mathbf{x},\mathbf{z})$  on their common attribute  $\mathbf{x}$ . Relation  $\mathbf{R}$  contains 10000 tuples and has 10 tuples per block. Relation  $\mathbf{S}$  contains 4500 tuple and has 5 tuples per block. We have 201 memory buffers available. Attribute  $\mathbf{x}$  takes 6 values (1, 2, 3, 4, 5, 6) in both relation R and S. The distribution of tuples taking different  $\mathbf{x}$  values is given below.

	x=1	x=2	x=3	x=4	x=5	x=6
Number of tuples in R	3000	2000	1000	600	400	3000
Number of tuples in S	1500	1500	100	100	300	1000

Please briefly describe how we can use simple sort-merge join to join these two relations, and calculate the overall cost in terms of disk I/Os. (Attention: these two relations are not sorted on any attributes)

**Problem 3** Algebraic Laws Suggested reading: Chapter 16.2

Exercise 16.2.2 part b) c) d) (Page 809)

**Problem 4** Dynamic Programming Suggested reading: Chapter 16.6

Compute the optimal plan for  $R \bowtie S \bowtie T \bowtie U$  using the dynamic programming algorithm, assuming the following:  $R(a,b,c)$ ,  $S(b,c,d)$ ,  $T(b,d,e)$ ,  $U(e,f)$

$$T(R) = 1000, T(S) = 500, T(T) = 250, T(U) = 125$$

$$V(R,c) = V(S,c) = 10, V(S,b) = V(R,b) = V(T,b) = 5$$

$$V(S,d) = V(T,d) = 20, V(T,e) = V(U,e) = 10$$

The cost of a join is estimated to be the cost of the subplans plus the size of the intermediate results (the same as the cost model we have in Lecture).

(a) Please draw the table for dynamic programming, to show how you compute the optimal plan for all possible join orders allowing all trees. (You only need to show columns with two or more joins)

(b) Please draw the table for dynamic programming, to show how you compute the optimal plan for all possible join orders allowing left-deep trees only. (You only need to show columns with two or more joins)

### **Problem 5** Cost Estimation Suggested reading: Chapter 16.4

Consider two relations  $R(a,b,c)$  and  $S(c,d,e)$  with the following statistics:

$B(R)=100$ ;  $T(R) = 1000$ ;  $V(R,a) = 20$ ;  $V(R,b) = 10$ ;  $V(R,c)=5$

$B(S)= 500$ ;  $T(S)= 1500$ ;  $V(S,c)=30$ ;  $V(S,d)=10$ ;  $V(S,e)=2$

- (a) Estimate the number of tuples in  $\sigma_{a=10}(R)$
- (b) Estimate the number of tuples in  $\sigma_{c=10}(R)$
- (c) Estimate the number of tuples in  $\sigma_{(b=10)\wedge(c=20)}(R)$
- (d) Estimate the number of tuples in  $\sigma_{b<25}(R)$
- (e) Estimate the number of tuples in  $\sigma_{(b=25)\vee(c>5)}(R)$
- (g) Estimate the number of tuples in  $R \bowtie S$
- (h) Estimate the number of tuples in  $R \bowtie_{R.c=S.c \text{ AND } R.b=S.d} S$

### **Problem 6** Failure Recovery Suggested reading: Chapter 17

Consider the following transaction log from the start of the run of a database system that is using undo/redo logging with (Nonquiescent) Checkpointing for crash recovery.

- 1) <Start T1>
- 2) <T1,X, 20,10>
- 3) <T1,Y, 40,0>
- 4) <Start T2>
- 5) <T1,X, 50,20>
- 6) <T2,Z,30,20>
- 7) <Commit T1>
- 8) <Start T3>
- 9) <T3,U,60,30>
- 10) <T2,V,50,25>
- 11) <start ckpt(T2, T3)>
- 12) <T2,Z, 45, 30>
- 13) <commit T2>
- 14) <Start T4>
- 15) <T4,W, 80, 10>
- 16) <commit T3>
- 17) <END CKPT>
- 18) <T4,W,100,80>
- 19) <commit T4>

What is the value for the data items X, Y, Z, U, V and W on disk after recovery:

- (a) if the system crashes just before line 13 is written to disk?
- (b) if the system crashes just before line 14 is written to disk?
- (c) if the system crashes just before line 19 is written to disk?