Relational algebra and query execution

CSE 444, fall 2010 — section 8 worksheet

November 18, 2010

1 Relational algebra warm-up

1. Given this database schema:

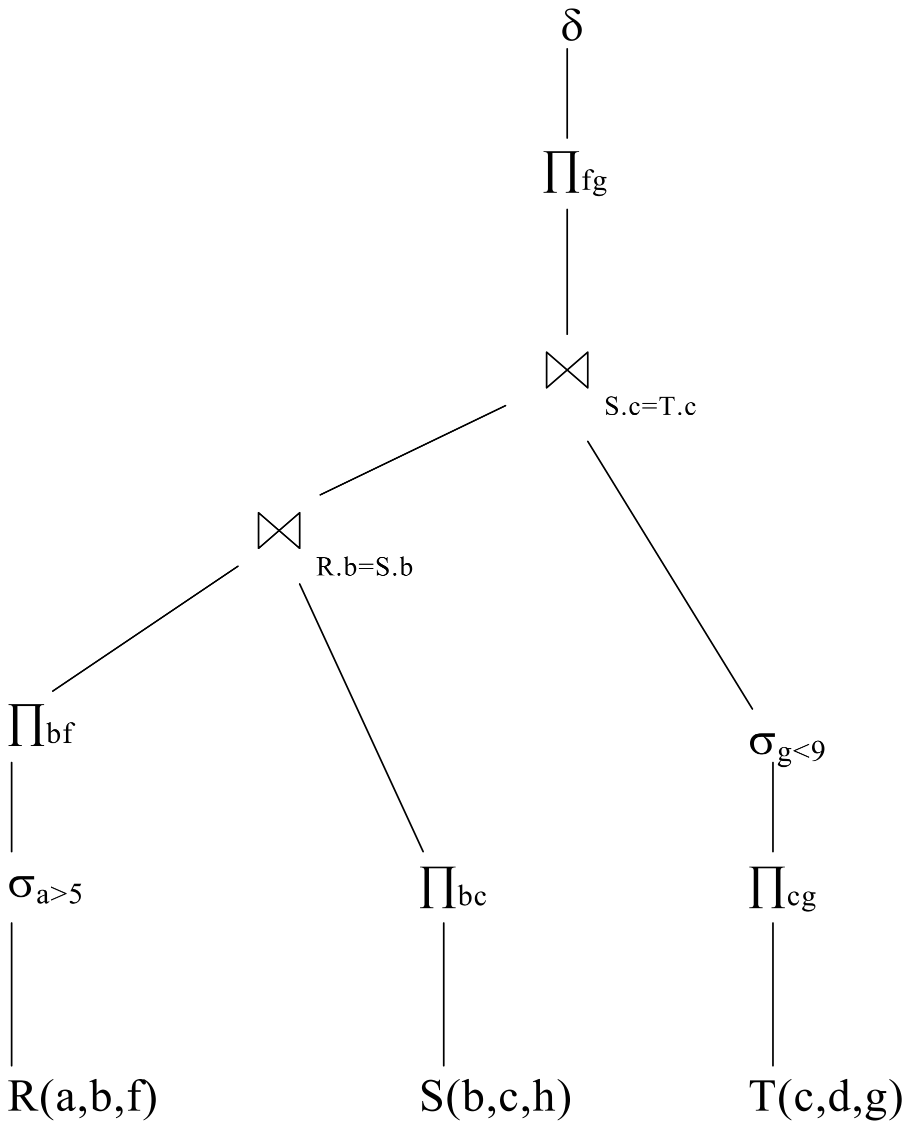
   - Product (pid, name, price)
   - Purchase (pid, cid, store)
   - Customer (cid, name, city)

   draw the logical query plan for each of the following SQL queries.

   (a)
   ```
   SELECT DISTINCT x.store
   FROM Purchase x, Customer y
   WHERE x.cid = y.cid
         and y.city = 'Seattle'
   ```

(b)
```
SELECT z.city, sum(x.price)
FROM Product x, Purchase y, Customer z
WHERE x.pid = y.pid and y.cid = z.cid
     and y.store = 'Wal-Mart'
GROUP BY z.city
HAVING count(*) > 100
```

2. Write a SQL query that is equivalent to the logical plan below:



3. Consider two tables $R(A, B, C)$ and $S(D, E, F)$, and the logical query plan P:

$$P = \sigma_{A>9}\left(\gamma_{A,\mathrm{sum}(F)}\left(R \bowtie_{C=D} S\right)\right)$$

Indicate which of the following three query plans are equivalent to P.

$$P_1 = \gamma_{A,\mathrm{sum}(F)}\left((\sigma_{A>9}R) \bowtie_{C=D} \left(\gamma_{D,\mathrm{sum}(F)}S\right)\right)$$
$$P_2 = \gamma_{A,\mathrm{sum}(F)}\left((\sigma_{A>9}R) \bowtie_{C=D} \left(\gamma_{D,E,\mathrm{sum}(F)}S\right)\right)$$
$$P_3 = \gamma_{A,\mathrm{sum}(F)}\left((\sigma_{A>9}R) \bowtie_{C=D} \left(\gamma_{D,E,\mathrm{sum}(F)}\left((\sigma_{A>9}R) \bowtie_{C=D} (S)\right)\right)\right)$$

Hint: Draw out each plan in tree form before solving.

2 Spring 2009 final, problem 3 (parts a-b)

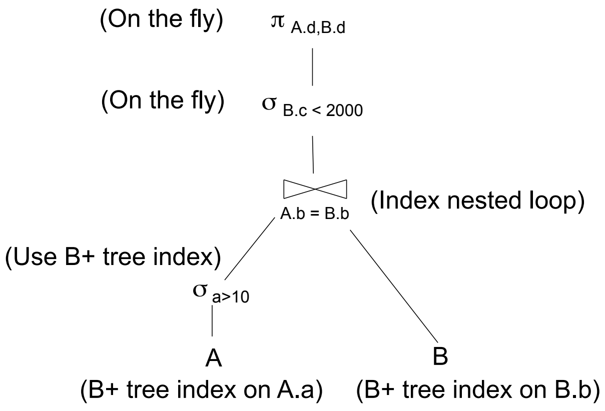
Consider four tables $R(a, b, c)$, $S(d, e, f)$, $T(g, h)$, $U(i, j, k)$.

(a) Consider the following SQL query:

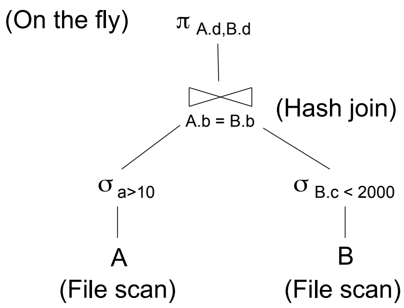
```
SELECT   R.b, avg(U.k) as avg
FROM     R, S, T, U
WHERE    R.a = S.d
         AND S.e = T.g
         AND T.h = U.i
         AND U.j = 5
         AND (R.c + S.f) < 10
GROUP BY R.b
```

Draw a *logical* plan for the query. You may choose any plan as long as it is correct (i.e. no need to worry about efficiency).

(b) Consider the following two physical query plans. Give **two** reasons why plan B may be faster than plan A. **Explain** each reason.



Plan A



Plan B

3 Summer 2009 final, problem 2

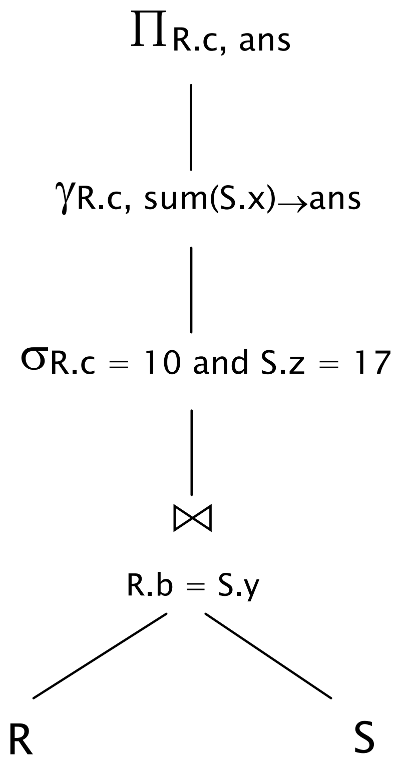
This problem and the next concern two relations $R(a, b, c)$ and $S(x, y, z)$ that have the following characteristics:

$B(R) = 600$	$B(S) = 800$
$T(R) = 3000$	$T(S) = 4000$
$V(R, a) = 300$	$V(S, x) = 100$
$V(R, b) = 100$	$V(S, y) = 400$
$V(R, c) = 50$	$V(S, z) = 40$

We also have $M = 1000$ (number of memory blocks).

Relation R has a clustered index on attribute a and an unclustered index on attribute c. Relation S has a clustered index on attribute x and an unclustered index on attribute z. All indices are B+ trees.

Now consider the following logical query plan for a query involving these two relations.



$\Pi_{\text{R.c, ans}}$
$\gamma_{\text{R.c, sum(S.x)}\to\text{ans}}$
$\sigma_{\text{R.c = 10 and S.z = 17}}$
$\bowtie$
R.b = S.y
R
S

(Answer the questions about this query on the following page.)

(a) Write a SQL query that is equivalent to the logical query plan on the previous page.

(b) Change or rearrange the original logical query plan to produce one that is equivalent (has the same final results), but which is estimated to be significantly faster, if possible. Recall that logical query optimization does not consider the final physical operators used to execute the query, but only things at the logical level, such as the sizes of relations and estimated sizes of intermediate results.

You should include a brief but specific explanation of how much you expect your changes to improve the speed of the query and why.

Draw your new query plan and write your explanation below. *If* you can clearly and easily show the changes on the original diagram, you may do so, otherwise draw a new diagram here.

4 Summer 2009 final, problem 3

This problem uses the same two relations and statistics as the previous one, but for a different operation not related to the previous query.

As before, the relations are $R(a, b, c)$ and $S(x, y, z)$. Here are the statistics again:

$B(R) = 600$	$B(S) = 800$
$T(R) = 3000$	$T(S) = 4000$
$V(R, a) = 300$	$V(S, x) = 100$
$V(R, b) = 100$	$V(S, y) = 400$
$V(R, c) = 50$	$V(S, z) = 40$

$M = 1000$ (number of memory blocks)

Also as before, relation R has a clustered index on attribute a and an unclustered index on attribute c. Relation S has a clustered index on attribute x and an unclustered index on attribute z. All indices are B+ trees.

Your job for this problem is to specify and justify a good physical plan for performing the join

$$R \bowtie_{a=z} S \qquad\qquad \text{(i.e., join } R \text{ and } S \text{ using the condition } R.a = S.z\text{)}$$

Your answer should specify the physical join operator used (hash, nested loop, sort-merge, or other) and the access methods used to read relations R and S (sequential scan, index, etc.). Be sure to give essential details: i.e., if you use a hash join, which relations(s) are included in hash tables; if you use nested loops, which relation(s) are accessed in the inner and outer loops, etc.

Give the estimated cost of your solution in terms of number of disk I/O operations needed (you should ignore CPU time and the cost to read any index blocks).

You should give a brief justification why this is the best (cheapest) way to implement the join. You do not need to exhaustively analyze all the other possible join implementations and access methods, but you should give a brief discussion of why your solution is the preferred one compared to the other possibilities.

(Write your answer on the next page. You may remove this page for reference if you wish.)

Give your solution and explanation here.