CS245 Database System Principles - Winter 2001

- PLEASE write your serial number on the top of your first page. If you have not received your serial number by e-mail, send a message to **orkut@stanford.edu**.
- This assignment is due in class on Tuesday, Feb 20th.
- State all assumptions.
- Subscribe to **cs245-win01** to receive clarifications and changes.
- Email questions to **cs245ta-win01@lists.stanford.edu**.

Assignment 5

Problem 1 (10 points)

Write the following joins as expressions involving selection, projection, and product.

a) $R(a, b, c) \bowtie S(e, a, c) \bowtie T(b)$

b) $R(a,b,c,d) \underset{R.b=e \wedge a<e}{\bowtie} S(b,c,e)$

Problem 2 (15 points)

Using the following "movie" relations:

```
Movie(title, year, length, studioName)
MovieStar(name, address, gender, birthdate)
StarsIn(title, year, starName)
Studio(name, address)
```

Turn the following queries into expressions using the algebraic operators:

a)
```
(SELECT name FROM MovieStar)
      UNION ALL
(SELECT starName FROM StarsIn);
```

b)
```
SELECT startName, SUM(length)
FROM Movie NATURAL JOIN StarsIn
GROUP BY starName
HAVING COUNT(*) >= 3;
```

Problem 3 (10 points)

Starting with an expression $\Pi_{b+f \to x, b+d \to y}$ (R(a,b,c,d) $\bowtie$ S(e,f,b,d))
push the projection down as far as it can go.

Problem 4 (25 points)

Below are the vital statistics for four relations, *R*, *S*, *U* and *X*.

R(a,b)	S(b,c)	U(c,d)	X(d,e)
T(R) = 1500	T(S) = 3000	T(U) = 4500	T(X) = 6000
V(R,a) = 300	V(S,b) = 750	V(U,c) = 750	V(X,d) = 600
V(R,b)= 900	V(S,c) = 1500	V(U,d) = 750	V(X,e) = 1500

Estimate the sizes of relations (number of tuples) that are the results of the following expressions. Explain how you derived each answer and what assumptions you made.

a) $R \bowtie S \bowtie U$
b) $\sigma_{b=10}(S)$
c) $\sigma_{d=20}(U) \bowtie X$
d) $R \times U \times S$
e) $\sigma_{e=30}(X)$
f) $\sigma_{c=40 \wedge d=50}(U)$
g) $\sigma_{c=40 \wedge d>50}(U)$

Problem 5 (40 points)

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 indices:
- a non-clustered non-unique B+-tree index for attribute *B*
- a clustered non-unique B+-tree index for attribute *C*

Furthermore, assume that these two indices are kept in memory (i.e. you do not need to read them from disk). 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:
- 1000 tuples of R are stored per block on disk.
- T(R) = 1200,000 (number of tuples of *R*)

- 100 tuples of S are stored per block on disk.
- T(S) = 360,000 (number of tuples of *S*)
- V(S,B)= 90,000 (image size of attribute *B* in *S*)
- V(S,C)= 90 (image size 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:

```
For every block B of R, retrieved using the clustered index on A for R
    For every tuple r of B
         Use the index on B for S to retrieve all
             of the tuples s of S such that s.B=r.B
             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:

```
For every block B of R, retrieved using the clustered index on A for R
    For every tuple r of B
         Use the index on C for S to retrieve all
             of the tuples s of S such that s.C=r.C
             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
```

You should analyze each of these plans carefully in terms of their behavior regarding accesses to disk. Your analysis should consider the behavior both in terms of the number of I/Os, as well as the total access time. Explain which of the plans is therefore better under what circumstances. Be sure to include in your analysis what accesses to disk are sequential accesses and which ones are random accesses.

- Now consider what happens if instead V(S,C) = 1000. Will one plan perform better than the other?

- Instead of using the V-function to do your estimate, assume that the number of tuples returned is proportional to the size of the domain of an attribute. Let DOM(S,B) = 200,000, and DOM(S,C) = 200. Re-calculate your estimates.