CSE 444 Final

December 12, 2005

8:30 - 10:20am

Name: _____

Total number of points: 100

Total time: 1h 50'

1. [**20 points**] Consider the following relational schema:

```
Person(name, age, city)
Trusts(name1, name2)
```

The Trusts relation tells us who trusts whom. This relation is not necessarily symmetric: i.e., if 'John' trusts 'Mary', then it is not necessarily the case that 'Mary' trusts 'John'. This relation is not necessarily transitive: i.e., if 'John' trusts 'Mary' and if 'Mary' trusts 'Fred', it is not necessarily the case that John' trusts 'Fred'.

Consider the following types of people:

- A "loner" is a person who trusts no one but himself.
- A "loyal" is a person who trusts only those who trust him.
- A "ruler" is a person who trusts only those who trust only him.

Indicate for each of the queries below whether it computes the loners, or the loyals, or the rulers, or none of the above:

(a)
```
select distinct x.name1
from Trusts x
where x.name1 not in
               (select y.name1
                from Trusts y
                where y.name2 in
                            (select z.name1
                             from Trusts z
                             where z.name2 != y.name1))
```

This query computes:

(b)
```
select distinct x.name1
from Trusts x
where x.name2 not in
                (select y.name2
                 from Trusts y
                 where x.name1=y.name1
                       and y.name2 != x.name1)
```

This query computes:

(c)
```
select distinct x.name1
from Trusts x
where x.name1 in
                (select y.name1
                 from Trusts y
                 where y.name2 not in
                             (select z.name1
                              from Trusts z
                              where z.name2 = y.name1))
```

This query computes:

(d)
```
select distinct x.name1
from Trusts x
where x.name1 not in
               (select y.name1
                from Trusts y
                where y.name2 not in
                             (select z.name1
                              from Trusts z
                              where z.name2 = y.name1))
```

This query computes:

2. **[30 points]** Consider an XML document with the following DTD:

```
<!ELEMENT root     (category*)>
<!ELEMENT category (name, prod*)>
<!ELEMENT prod     (name, price, store*)>
<!ELEMENT store    (name, city)>
```

All elements that are not indicated above have content `#PCDATA`.

(a) **[5 points]** Write an XPath expression that returns all products under the category with name "toy", whose price is < 200 and that are sold at some store in "New York City".

(b) **[10 points]** Write an XQuery query that computes for each city the total number of product categories that are sold at stores in that city.

(c) [**10 points**] Write an XQuery that restructures the data into the following DTD:

```
<!ELEMENT output (store*)>
<!ELEMENT store (name, city, product*)>
<!ELEMENT product (name, price, category)>
```

All elements that are not indicated above have content `#PCDATA`. Each store in the input XML document is uniquely identified by its name, and should occur exactly once in the output document.

(d) [**5 points**] Consider the three XQuery expressions below. They refer to the DTD of the input XML data (with top element `root`), defined at the beginning of this question.

```
Q1:
for $x in document()/root/category/prod[price/text()>5]
                                        [store/city/text()=''NYC'']
return $x/name

Q2:
for $c in document()/root/category,
    $x in $c/prod[price/text() > 5][store/city/text()=''NYC'']
return $x/name

Q3:
for $x in document()/root/category/prod,
    $c in $x/store/city
where $x/price/text() > 5 and $c/text()=''NYC''
return $x/name
```

Indicate which queries are equivalent and which are different. You have to fill in the three entries below with $=$ or $\neq$:

	Q1	Q2	Q3
Q1	$=$		
Q2	■	$=$	
Q3	■	■	$=$

3. **[20 points]** Transactions.

   (a) **[5 points]** The recovery manager uses undo logging. In each of the logs below there is an `<START CKPT(...)>` entry, but the matching `<END CKPT>` is missing. That is, the checkpoint ended, but the corresponding log entry was somehow omitted. Please indicate where the `<END CKPT>` entry should go. In the case where there may be several places, indicate the earliest possible place where the entry `<END CKPT>` can be. In addition, please indicate how far back the recovery manager has to read in the log, if the crash occurs after the last entry in the log. Indicate your answer with arrows in the logs below.

<START S>
<S,A,60>
<START CKPT(S)>
<COMMIT S>
<START T>
<T,A,10>
<START U>
<U,B,20>
<T,C,30>
<START V>
<U,D,40>
<V,F,70>
<COMMIT U>
<T,E,50>
<COMMIT T>
<V,B,80>
<COMMIT V>

<START S>
<S,A,60>
<COMMIT S>
<START T>
<T,A,10>
<START CKPT(T)>
<START U>
<U,B,20>
<T,C,30>
<START V>
<U,D,40>
<V,F,70>
<COMMIT U>
<T,E,50>
<COMMIT T>
<V,B,80>
<COMMIT V>

(b) [**5 points**] Repeat the same question above, assuming that the recovery manager is uses redo logging.

<START S>
<S,A,60>
<START CKPT(S)>
<COMMIT S>
<START T>
<T,A,10>
<START U>
<U,B,20>
<T,C,30>
<START V>
<U,D,40>
<V,F,70>
<COMMIT U>
<T,E,50>
<COMMIT T>
<V,B,80>
<COMMIT V>

<START S>
<S,A,60>
<COMMIT S>
<START T>
<T,A,10>
<START CKPT(T)>
<START U>
<U,B,20>
<T,C,30>
<START V>
<U,D,40>
<V,F,70>
<COMMIT U>
<T,E,50>
<COMMIT T>
<V,B,80>
<COMMIT V>

(c) **[10 points]** Consider a concurrency control manager by timestamps. Below are several sequences of events, including *start* events, where st_i means that transaction T_i starts. These sequences represent real time, and the timestamp-based scheduler will allocate timestamps to transactions in the order of their starts. In each case below tell what happens with the last *write* request. You have to choose between one of the following four possible answers: (1) the request is accepted, (2) is ignored, (3) the transaction is delayed, (4) the transaction is rolled back.

i.

$$st_1; st_2; r_1(A), r_2(B); w_2(A); w_1(B)$$

The system will perform the following action for $w_1(B)$:

ii.

$$st_1; r_1(A); st_2; w_2(B); r_2(A); w_1(B)$$

The system will perform the following action for $w_1(B)$:

iii.

$$st_1; st_2; st_3; r_1(A); r_2(B); w_1(C); r_3(B); w_2(B); w_3(A)$$

The system will perform the following action for $w_3(A)$:

iv.

$$st_1; st_3; st_2; r_1(A); r_2(B); r_3(B); w_2(A); w_2(B); w_3(A)$$

The system will perform the following action for $w_3(A)$:

4. **[30 points]** Query Execution and Optimization.

   (a) **[8 points]** Consider two tables $R(A, B)$ and $S(C, D)$ with the following statistics:

$$\begin{aligned} B(R) &= 5 \\ T(R) &= 200 \\ V(R, A) &= 10 \\ B(S) &= 100 \\ T(S) &= 400 \\ V(S, C) &= 50 \\ M &= 1000 \end{aligned}$$

   There is a clustered index on $S.C$ and an unclustered index on $R.A$. Consider the logical plan:

$$P = \sigma_{A=77}(R) \bowtie_{B=C} S$$

   There are two logical operators, $S = \sigma_{A=77}$ and $J = \bowtie_{B=C}$, and for each we consider two physical operators:

$$\begin{aligned} s1 &= \text{one pass table scan} \\ s2 &= \text{index-based selection} \\ j1 &= \text{main memory hash join} \\ j2 &= \text{index-based join} \end{aligned}$$

   Both $s1$ and $s2$ are pipelined, i.e. the result of the select operator is not materialized. For each of the resulting four physical plans compute its cost in terms number of disc I/0s, expressed as a function of the statistics above. Your answer should consists of four expressions, e.g. $\texttt{COST}(s1j1) = B(R)B(S)/M + V(R, A)$ (not the real answer).

i. $\texttt{COST}(s1j1) =$

ii. $\texttt{COST}(s2j1) =$

iii. $\texttt{COST}(s1j2) =$

iv. $\texttt{COST}(s2j2) =$

(b) **[2 points]** Indicate the cheapest plan of the four, together with its cost expressed as a number. You will get credit for this point only if you compute correctly all four expressions above.

(c) **[10 points]** Consider two tables $R(A, B, C)$ and $S(D, E, F)$, and the query plan P below. Indicate which of the query plans P_1, P_2, P_3, P_4 are equivalent to P. The symbol $\rtimes$ in P_4 represents the right outer join, i.e. $R \rtimes S = S \ltimes R$.

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

$$P_1 = \gamma_{A,sum(F)}(\sigma_{A>9}(R) \bowtie_{C=D} \gamma_{D,sum(F)}S)$$

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

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

$$P_4 = \gamma_{A,sum(F)}(\sigma_{A>9}(R) \bowtie_{C=D} \gamma_{D,E,sum(F)}(\sigma_{A>9}(R) \rtimes_{C=D} S))$$

(d) [**10 points**] Consider the following query, where $\bowtie$ denotes the natural join:

$$R(A,B) \bowtie S(B,C) \bowtie T(C,D) \bowtie U(D,E)$$

We consider only left linear plans below, and do not distinguish between join order, i.e. $R(A,B) \bowtie S(B,C)$ is the same as $S(B,C) \bowtie R(A,B)$.

i. How many different left linear plans exist for this query ?

ii. Show two different left linear plans without cartesian products.

iii. How many different plans without cartesian product exist for this query ?