CSE 544: Homework 3

Due: Sunday, June 3, 2012, 11:59pm

Name: ______________________________

Question	Points	Score
1	30	
2	50	
3	70	
4	30	
5	10	
6	10	
Total:	200	

You may turn in this homework either electronically (in pdf or word format), or print a copy of the assignment, write your answers in the spaces provided, then turn it in to Paris.

1 Parallel Query Processing

1. (30 points)

Consider the following relations $R(A, B), S(B, C)$ and the following query:

```
select R.A, sum(S.C)
from R, S
where R.B = S.B
group by R.A
```

Assume that the block size is 10^4 bytes, and assume the following statistics:

$$
\begin{aligned}
B(R) &= 10^9 && \text{/* about 10 Terabytes */}\\
B(S) &= 10^9 && \text{/* about 10 Terabytes */}\\
B(R \bowtie S) &= B(R) + B(S) = 2 \cdot 10^9 &&\\
V(A, R) &= 10^5 && R.A \text{ has } 10^5 \text{ distinct values}\\
P &= 10^5 && \text{/* number of servers */}\\
M &= 10^5 && \text{/* about 1GB local memory at each server */}
\end{aligned}
$$

At the beginning of the execution, both relations R and S are distributed evenly among the P servers and are stored on their local disks. In the questions below you will be asked to compute the running time measured in number of disk I/O's: for that, assume the standard implementation of the Map-Reduce operator.

(a) (10 points) The query is evaluated in parallel using two map/reduce steps. We will call this PLAN 1.

Map/reduce Job 1. The *map* phase redistributes both R and S by their B attribute. The *reduce* phase performs a main-memory join of their R and S fragments.

Map/reduce Job 2. The *map* phase redistributes the join $R \bowtie S$ by the A attribute. The *reduce* phase performs a main-memory group-by A, computing the sum of the C values on the local fragment, then writes the result to disk.

Compute the total parallel I/O cost for PLAN 1. You have to turn in an integer number.

(a) ____________

Your answer:

(b) (10 points) Now assume that the query is evaluated as follows (not longer using the map/reduce framework). We call this PLAN 2.

Step 1: Redistribute both R and S by their B attribute. The receiving severs will hold both table fragments in main memory.

Step 2: Perform a main-memory join the local R and S, and keep the result in main memory.

Step 3: Redistribute the join $R \bowtie S$ by the A attribute. The receiving servers keep their fragments in main memory.

Step 4: Each server performs a main-memory group-by A, computing the sum of the C values on the local fragment.

Compute the total parallel I/O cost for PLAN 2. You have to turn in an integer number.

(b) ____________

Your answer:

(c) (10 points) For each of the statement below indicate whether it is true or false:

i. PLAN 1 is always more efficient than PLAN 2 because it uses the map/reduce infrastructure.

i. ____________

True or False ?

ii. If any server fails during the execution of PLAN 1, then we do not need to restart the entire query execution, but instead can re-execute only a fragment of the computation.

ii. ____________

True or False ?

iii. If any server fails during the execution of PLAN 2, then we do not need to restart the entire query execution, but instead can re-execute only a fragment of the computation.

iii. ____________

True or False ?

iv. When R and S are distributed by their B attribute, we should use a different hash function for $R.B$ from $S.B$, to better achieve load balance.

iv. ____________

True or False ?

v. Even if the hash function $h(A)$ used to distribute the join result $R \bowtie S$ (in preparation of the group-by) is perfectly uniform, the distribution may be heavily skewed if the data is skewed.

v. ____________

True or False ?

2 Datalog

2. (50 points)

   (a) (10 points) Consider the following two datalog programs computing the transitive closure:

   P1:

```
T(x,y) :- R(x,y)
T(x,y) :- T(x,z),R(z,y)
```

   P2:

```
T(x,y) :- R(x,y)
T(x,y) :- T(x,z),T(z,y)
```

   Suppose R is a graph that consists of a single path: $R(a_0, a_1), R(a_1, a_2), \ldots, R(a_{n-1}, a_n)$. Thus, the transitive closure T computed by both programs consists of all $\binom{n}{2}$ ground facts for the form $T(a_i, a_j)$, for $1 \leq i < j \leq n$. Assume that we evaluate both programs using the semi-naive evaluation algorithm.

   i. For a fixed $m = 1, \ldots n-1$, how many times will the fact $T(a_1, a_{m+1})$ be discovered by `P1`?

   i. ____________

   Your answer:

   ii. How many times will the fact $T(a_1, a_{m+1})$ be discovered by `P2`?

   ii. ____________

   Your answer:

   (b) (10 points) Consider the following datalog program that finds the set of nodes accessible from those in a collection `Source`:

   P1:

```
T(x,y) :- R(x,y)
T(x,y) :- T(x,z),R(z,y)
A(x)   :- Source(x)
A(y)   :- Source(x),T(x,y)
```

   A much more efficient way to compute the predicate $A(x)$ is by the following datalog program:

   P1':

```
A(x) :- Source(x)
A(y) :- A(x),R(x,y)
```

   i. Suppose that the graph is a chain $R(a_0, a_1), R(a_1, a_2), \ldots, R(a_{n-1}, a_n)$, and `Source` contains a single node a_i. How many IDB facts does P1 compute? How many IDB facts does P1' compute?

   ii. The transformation from `P1` to `P1'` is called *magic set optimization*. Now consider the following datalog program computing all nodes in the same generation as those in `Source`:

```
P2:
  S(u,v) :- R(x,u),R(x,v)
  S(u,v) :- S(x,y),R(x,u),R(y,v)
  A(x)   :- Source(x)
  A(v)   :- Source(u), S(u,v)
```

Write a different program `P2'` that computes the same answer A as `P2` and is more efficient, in that it computes fewer facts than `P1`. Your solution `P2'` should be a best-effort solution. Think about the case when R is a binary tree, and design your program accordingly. Hint: you may google for magic sets optimizations, but you may find magic sets quite confusing, so it may be easier if you come up with `P2'` yourself.

(c) (10 points) Consider a graph where each node x is either a leaf, or has two outgoing edges $(x, y), (x, z)$. In the former case, we store x in a relation $L(x)$; in the latter case we store the triple (x, y, z) in a relation T. (Thus, x is a key in $T(x, y, z)$.) Consider the following game with two players. Players take turns in moving a pebble on the graph. If the pebble is on a node x, then the player whose turn it is may move it to one of the two children, y or z. A player wins if it is her turn to move and the pebble is on a leaf. Write a datalog program to computes the set starting nodes from which player 1 has a winning strategy. That is, your program should compute a relation $P1(x)$ that returns all nodes x such that, if player 1 starts the game on x (and plays smartly!) then she is guaranteed to win the game.

(d) (10 points) Consider the following startified datalog program that computes the complement of the transitive closure:

```
T(x,y) :- R(x,y)
T(x,y) :- T(x,z),T(z,y)

CT(x,y) :- Node(x),Node(y),not T(x,y)
```

Write a datalog program that computes the complement of the transitive closure assuming inflationary fixpoint semantics. Note: this problem is challenging. You may want to google for the answer, but make sure you understand it.

(e) (10 points) Every stratified datalog program can be expressed in inflationary datalog. This is not obvious at all, but the main step in the proof is the inflationary program that computes the complement of the transitive closure, which you have written for the previous question; so by now you know that stratified datalog can be translated to inflationary datalog. In this problem we are concerned with the other direction. Many years ago, a researcher (or set of researchers) X published a paper where she/he/they proved that the converse also holds: every inflationary datalog program can be rewritten into a stratified datalog program. However, another researcher (or other reserchers) Y found a flaw in the proof, and published another paper that presented a datalog program Z in inflationary datalog, and proved that Z cannot be expressed in stratified datalog. Thus, X was (were) wrong! Stratified datalog is strictly less expressive than inflationary datalog, and the query that separates the two classes is Z. In this question you will determine X, Y, and Z. You need to google a lot, but your answer will be very short: you will write down two sets of names X and Y, and a datalog program Z (you don't have to write the program but only say what it does). X and Y may be unfamiliar names to you, but you have encountered Z many times already.

3 Conjunctive Queries

3. (70 points)

   (a) (10 points) Find a full semi-join reduction for the query below.

$$q(x) : -R(x, y), S(y, z), T(y, u)$$

(b) (60 points)

i. Indicate for each pair of queries q, q' below, whether $q \subseteq q'$. If the answer is yes, provide a proof; if the answer is no, give a database instance I on which $q(I) \not\subseteq q'(I)$.

1.

$$\begin{aligned} q(x) &:- \; R(x,y), R(y,z), R(z,x) \\ q'(x) &:- \; R(x,y), R(y,z), R(z,u), R(u,v), R(v,z) \end{aligned}$$

2.

$$\begin{aligned} q(x,y) &:- \; R(x,u,u), R(u,v,w), R(w,w,y) \\ q'(x,y) &:- \; R(x,u,v), R(v,v,v), R(v,w,y) \end{aligned}$$

3.

$$\begin{aligned} q() &:- \; R(u,u,x,y), R(x,y,v,w), v \neq w \\ q'() &:- \; R(u,u,x,y), x \neq y \end{aligned}$$

4.

$$\begin{aligned} q(x) &:- \; R(x,y), R(y,z), R(z,v) \\ q'(x) &:- \; R(x,y), R(y,z), y \neq z \end{aligned}$$

ii. Consider the two conjunctive queries below, and notice that $q_1 \subset q_2$.

$$q_1(x) = R(x,y), R(y,z)$$
$$q_2(x) = R(x,y)$$

1. Find a conjunctive query $r(x)$ s.t. $q_1 \subset r \subset q_2$

2. **Challenging question** Extend your answer: find an infinite set of queries $r_1(x), r_2(x), \ldots$ such they are inequivalent $r_i \not\equiv r_j$ for $i \neq j$, and for every i, $q_1 \subset r_i \subset q_2$. If you answer this question then you do not need to answer the preceding question.

4 Transactions

4. (30 points)

   (a) (10 points) A *static database* is a database where no insertions or deletions are performed. A *dynamic database* is a database that allows insertions and deletions. We consider a scheduler that has one lock for each record in the database (like SQL Server). Answer the questions below:

      i. In a static database, strict two phase locking guarantees that the schedule is serializable while two phase locking does not.

      i. ____________

      True or false

      ii. Strict two phase locking guarantees that the schedule is recoverable, while two phase locking does not.

      ii. ____________

      True or false

      iii. In a dynamic database, strict two phase locking can prevent phantoms, while two phase locking cannot.

      iii. ____________

      True or false

      iv. Strict two phase locking is more difficult to implement and most database system do not support it.

      iv. ____________

      True or false

      v. Strict two phase locking holds all the locks until the end of a transaction, while two phase locking may release the locks earlier.

      v. ____________

      True or false

vi. In both two phase locking and strict two phase locking all locks must precede all unlocks.

vi. ____________

True or false

vii. In strict two phase locking deadlocks are not possible, while in two phase locking deadlocks are possible.

vii. ____________

True or false

viii. If the database uses shared locks for read operations, then if all transactions are read-only then no deadlocks are possible.

viii. ____________

True or false

ix. SQL Server checks for deadlocks at regular intervals, and if it detects a deadlock then aborts a transaction

ix. ____________

True or false

x. Suppose that the table R has 1000 records of which 50 have $A =' abc'$. Then the transaction below needs to acquire 1000 locks (assuming the database system has locks only for records, i.e. it has not table lock):

```
begin transaction;
select * from R where A='abc';
commit;
```

x. ____________

True or false

(b) (10 points) i. R&G, pp. 598, Exercise 18.4 Consider the execution shown in the following table (same as Fig. 18.7 in the book).

LSN	Action	pLSN	uLSN
00	update: T1 writes P2		
10	update: T1 writes P1		
20	update: T2 writes P5		
30	update: T3 writes P3		
40	T3 commit		
50	update: T2 writes P5		
60	update: T2 writes P3		
70	T2 abort		

Expand the figure to show `prevLSN` and `undonextLSN` values. Then show the actions taken to rollback transaction T2. Finally, show the log after T2 is rolled back, including all `prevLSN` and `undonextLSN` values in the log records.

ii. R&G, Exercise 18.5 Consider the execution showin in the table below (same as Figure 18.8 in the book):

LSN	Action	pLSN	uLSN
00	begin_checkpoint		
10	end_checkpoing		
20	update: T1 writes P1		
30	update: T2 writes P2		
40	update: T3 writes P3		
50	T2: commit		
60	update: T3 writes P2		
70	T2: end		
80	update: T1 writes P5		
90	T3: abort		
CRASH, RESTART			

In addition, the system crashes during recovery after writing two log records to stabel storage and again after writing another two log records.

1. What is the value of the LSN stored in the master log record?
2. Whas is done during Analysis?
3. What is done during Redo?
4. What is done during Undo?
5. Show the log when recovery is complete, including all non-null prevLSN and undonextLSN values in the log records.

(c) (10 points) For each of the statements below indicate whether it is true of false about the ARIES recovery manager:

i. During the normal operation of a database (i.e. not during recovery) no update records in the log are undone.

i. ______________

true or false ?

ii. During the normal operation of a database (i.e. not during recovery) no update records in the log are redone.

ii. ______________

true or false ?

iii. During the normal operation of a database (i.e. not during recovery) no CLR records are ever written to the log.

iii. ______________

true or false ?

iv. During recovery from a crash no update record is processed both during the redo and during the undo phase. (In other words, an update record is either redone or undone, but not both.)

iv. ______________

true or false ?

v. During recovery from a crash the CLR records in the log are neither undone nor redone. (In other words the CLR records server a different purpose, not to to be redone or undone during recovery.)

v. ______________

true or false ?

vi. Suppose that two update log entries were written in the log on behalf of a transaction, then the system crashed while the transaction was active. Even if the system crashes repeatedly during recovery, there will never be more than two CLR records written on behalf of this transaction.

vi. ______________

true or false ?

vii. All log entries preceeding the last `begin_checkpoint` can be safely deleted from the log in order to reclaim their disc space.

vii. ______________

true or false ?

5 Provenance

5. (10 points)

   (a) (10 points) Consider a relational database schema $R(A), S(B, C)$. All queries mentioned below are assumed to be monotone queries.

      i. One of the answers a of a relational query Q has the provenance polynomial $x_1y_1^2 + 2x_2y_1y_2$, where x_1, x_2 are annotations of two tuples in R and y_1, y_2 are annotations of two tuples in S.

         1. Assume that the input relations are sets. If we evaluate the query under bag semantics, how many copies of a will be in the query's answer ?
         2. Assume that the input relations are bags, where each tuple occurs exactly twice. If we evaluate Q under bag semantics, how many copies of a are there in the query's answer ?
         3. What is the smallest number of tuples that need to be removed from the database instance in order to remove a from the answer to Q ?
         4. Suppose that the tuple annotated with y_2 was incorrect (it contains some incorrect value). Is the answer a correct, or did it become incorrect ?

      ii. Consider the following instance:

R

A	
a_1	x_1
a_2	x_2

S

A	B	
a_1	b_1	y_1
a_1	b_2	y_2
a_2	b_2	y_3

For each polynomial below, write a Boolean conjunctive query having that provenance polynomial. Your queries should not have constants or the inequality predicates $\neq, <, \leq$.

$$P_1 = x_1y_1 + x_1y_2 + x_2y_3$$
$$P_2 = x_1y_1^2 + x_1y_2^2 + x_1y_2y_3 + x_2y_2y_3 + x_2y_3^2$$
$$P_3 = x_1^2y_1^2 + x_1^2y_2^2 + 2x_1x_2y_2y_3 + x_2^2y_3^2$$

      iii. Now we consider arbitrary instances $R(A)$, $S(B, C)$, and we assume that the tuples in R are annotated with variables $x_1, x_2, \ldots$ and the tuples in S are annotated with $y_1, y_2, \ldots$ In each case below give an example of a Boolean conjunctive query whose provenance polynomial P has the property stated below. Your conjunctive query may use the predicate $\neq$.

         1. The polynomial P factorizes as $P = P_1 \cdot P_2$ where P_1 is a linear polynomial in $x_1, x_2, \ldots$ and P_2 is a linear polynomial in $y_1, y_2, \ldots$
         2. All monomials in P have the form $x_iy_jy_k$ where $j \neq k$.
         3. All monomials in P have the form $x_ix_jy_k$ where $i \neq j$.

6 Differential Privacy

6. (10 points)

   (a) (10 points) Consider a table of patients from a hospital, which contains $n = 10^6$ patient records. (This is a realistic number for the major hospitals in Seattle: the reason it is so high is that the database stores historical data, i.e. each visit of each patient.) Three different statisticians ask the following queries:

   - Statistician 1 wants to compute the number n_1 of patients whose diagnostic is *flue*. Assume that the true answer to this query is: $n_1 = 100,000$.
   - Statistician 2 wants to compute the number n_2 of patients whose diagnostic is *appendicitis*. Assume that the true answer to this query is: $n_2 = 1,000$.
   - Statistician 3 wants to compute the number n_3 of patients whose diagnostic is *the Meckel syndrome*. Assume that the true answer to this query is: $n_3 = 10$.

   To answer these queries, the system uses a differentially private algorithm that adds a random noise to the answer. The noise is computed using the Laplace distribution with parameter b (see the paper) chosen such as to guarantee that each answer is ε-differentially private (see the definition in the paper). We consider three cases: $\varepsilon = 0.1$, $\varepsilon = 0.01$, $\varepsilon = 0.001$. For each of the three levels of privacy answer the following questions:

   i. What is a patient's privacy guarantee? In other words, for a given patient, what is the upper bound on the probability ratio between the answer of the algorithm on the database with, or without that patient's record? We assume that each patient has a single record in the database. (You need to simply apply the definition of differential privacy, and use a calculator.)

   ii. Compute the parameter b for the Laplacian distribution by the differentially private algorithm. (You need apply directly the formula.)

   iii. When the differentially private algorithm answers each query, the perturbed answer $\tilde{n}_i$ that it computes happens to be exactly one standard deviation higher than the mean (i.e. n_i). Compute the answer $\tilde{n}_i = n_i + \sigma$ returned by the algorithm, for each of the three statisticians $i = 1, 2, 3$. (Hint: you need to find out the standard deviation of a Laplacian distribution.)

   iv. What is the probability that the algorithm returns an answer that is at least one standard deviation larger than the mean, $\tilde{n}_i \geq n_i + \sigma$? This answer will be the same for all three statisticians. (Hint: don't use Tchebishev's inequality; instead compute a simple integral which gives you the exact answer.)

   v. Suppose that each statistician wants to know the answer to his/her query within three standard deviations. What is the probability that the differentially private algorithm will return an answer with the desired precision? Note: here "standard deviation" refers to the binomial random variable $B(n, p)$ that characterizes the property inquired, and not to the Laplacian noise. For example, consider the first statistician, who wants to count the number of patients with "flue". This number is a binomial distribution $B(n, p)$, where $n = 1M$ and $p = 0.1$, because each patient in our population has a 1/10 chance of having "flue". The statistician wants to compute the value of $n_1 = 100,00$ within 3 standard deviations of the binomial random variable.

You need to answer each question above for each of the three values of ε. If you don't like numerical computations on a calculator, you may choose to turn only the closed formula (in ε, n, n_i) that represents the answer.

ε	End user privacy	b	$\tilde{n}_i$			$P(\tilde{n}_i \geq n + \sigma)$	Prob. that $\tilde{n}_i$ is within 3 std.dev
			$\tilde{n}_1$	$\tilde{n}_2$	$\tilde{n}_3$		
0.1							
0.01							
0.001							
Generic ε							