This exam is open book and notes. You can use a calculator and your laptop to access course notes and videos (but not to communicate with other people). You have 70 minutes to complete the exam.

Print your name:______________________________________________________________

The Honor Code is an undertaking of the students, individually and collectively:

1. that they will not give or receive aid in examinations; that they will not give or receive unpermitted aid in class work, in the preparation of reports, or in any other work that is to be used by the instructor as the basis of grading;

2. that they will do their share and take an active part in seeing to it that others as well as themselves uphold the spirit and letter of the Honor Code.

The faculty on its part manifests its confidence in the honor of its students by refraining from proctoring examinations and from taking unusual and unreasonable precautions to prevent the forms of dishonesty mentioned above. The faculty will also avoid, as far as practicable, academic procedures that create temptations to violate the Honor Code.

While the faculty alone has the right and obligation to set academic requirements, the students and faculty will work together to establish optimal conditions for honorable academic work.

I acknowledge and accept the Honor Code.

Signed:______________________________________________________________

<table>
<thead>
<tr>
<th>Problem</th>
<th>Points</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>50</strong></td>
<td></td>
</tr>
</tbody>
</table>
Problem 1 (10 points)

We are given two tables $R(A, B, C)$ and $S(A, D, E)$. Assume that $A$ is the primary key of both $R$ and $S$. The two tables are fragmented as follows:

- $R1 = \sigma_{(A < 40) \land (B > 30)} (R)$
- $R2 = \sigma_{(40 \leq A \leq 100) \land (B > 30)} (R)$
- $R3 = \sigma_{(A > 100) \land (B > 30)} (R)$
- $R4 = \sigma_{B \leq 30} (R)$

and

- $S1 = \pi_{A,D} (S)$
- $S2 = \pi_{A,E} (S)$

Perform decomposition and localization to transform the following queries into an optimized operator tree on fragments. Optimize each query as much as possible:

(a) SELECT A, B, C
    FROM R
    WHERE B > 70 AND A \leq 100

    ANSWER:

(b) SELECT A, B, C
    FROM R, S
    WHERE R.A = S.A AND B \leq 30

    ANSWER:

(c) SELECT A, D
    FROM R, S
    WHERE R.A = S.A AND A < 40

    ANSWER:
Problem 2 (10 points)

Consider a distributed database with two nodes N1 and N2. Let $R(AB)$ with 10K tuples reside at N1 and let $S(AC)$ with 50K tuples reside at N2. Each tuple of relation $R$ has a distinct value of $A$; relation $S$ has 5K distinct values of $A$. Let the attribute $A$ be 4 bytes wide, the attribute $B$ be 200 bytes wide, and the attribute $C$ be 150 bytes wide. The estimate for $R \bowtie S$ is 1K tuples, the estimate for $S \bowtie R$ is 10K tuples, and the estimate for $R \bowtie S$ is 10K tuples.

Suppose the nodes are connected over the Internet and communication is expensive, say 1 unit per 1K bytes. Computation is much cheaper so we assume computation cost is zero. Also assume that there is no cost in distributing the query plan to the nodes.

The answers to the questions below are query plans for $R \bowtie S$. The plans are described using a table, where each row represents one step. For example, the table below is a one step plan where node N1 sends $T_1 = R$ to node N2, and node N2 computes the join $T_1 \bowtie S$. In this case, the result is produced at N2.

<table>
<thead>
<tr>
<th>from node</th>
<th>to node</th>
<th>what is sent</th>
<th>number of tuples sent</th>
<th>cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>N1</td>
<td>N2</td>
<td>$T_1 = R$</td>
<td>10K</td>
<td>2040</td>
</tr>
</tbody>
</table>

Total cost for this plan: 2040

(a) Find the optimal plan for computing $R \bowtie S$, where the final result is produced at node N1. Describe this optimal plan in the table below. (Note: not all rows shown might be necessary.)

<table>
<thead>
<tr>
<th>from node</th>
<th>to node</th>
<th>what is sent</th>
<th>number of tuples sent</th>
<th>cost</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$T_1 =$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$T_2 =$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$T_3 =$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$T_4 =$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$T_5 =$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Total cost for this plan: _____________________
(b) Find the optimal plan for computing $R \bowtie S$, where the final result is now produced at node N2. Describe this optimal plan in the table below. (Note: not all rows shown might be necessary.)

<table>
<thead>
<tr>
<th>from node</th>
<th>to node</th>
<th>what is sent</th>
<th>number of tuples sent</th>
<th>cost</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$T_1 =$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$T_2 =$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$T_3 =$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$T_4 =$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$T_5 =$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Total cost for this plan: ______________________
Problem 3 (10 points)

A distributed database runs across two nodes and you have been asked to allocate fragments to the nodes. (Fragments are not replicated for this problem.) You are given an access matrix that specifies which transaction classes access which fragments.

For example, consider this access matrix for transactions classes 1, 2 and 3 and fragments A, B, C and D:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

The matrix indicates, for instance, that transactions of class 1 access fragments A and B, while transactions of class 3 access fragment A only.

Your task is to assign each fragment to one of the available nodes. In doing so, there are two critical factors for your application:

- **Number of Global Transaction Classes:** A transaction class is global if it must access fragments that are at more than one node. For instance, given the matrix above, if you assign fragments A and B to node 1 and fragments C and D to node 2, class 2 is global (and classes 1 and 3 are not). Your database system does not provide a global commit protocol, so for each global transaction class, you will have to hand code a commit protocol, which is a pain. Thus, your goal is to minimize the number of global transactions classes.

- **Load Imbalance:** If all you want to do is minimize the number of global classes, then you can place all fragments on a single node, which may lead to poor performance. Thus you also want to consider the load at each node: Let us assume that the number of transactions running of each class is the same, and that the load at a node is the number of fragment accesses at the node. In our example, the load at node 1 (fragments A, B) is 4 (the number of 1's in the A, B rows), and the load at node 2 (fragments 3, 4) is 2. The load imbalance is the difference between the maximum load at a node and the minimum node. Thus, in our example, the load imbalance is $4 - 2 = 2$. 
(a) Consider the following access matrix for a 2-node system:

\[
\begin{array}{ccccc}
& 1 & 2 & 3 & 4 & 5 \\
A & 1 & 1 & 0 & 0 & 1 \\
B & 0 & 1 & 0 & 1 & 1 \\
C & 1 & 0 & 1 & 0 & 0 \\
D & 0 & 0 & 1 & 1 & 0 \\
\end{array}
\]

Give a fragment allocation that yields zero load imbalance. For this allocation, what is the number of global transaction classes?

Place at node 1 fragments: ______________________

Place at node 2 fragments: ______________________

Number of global transaction classes: ______________________

(b) Consider all possible fragment allocations for the same access matrix. Which of these fragment allocations minimizes the number of global transaction classes? Also show the number of global classes and the load imbalance in this case.

Place at node 1 fragments: ______________________

Place at node 2 fragments: ______________________

Number of global transaction classes: ______________________

Load imbalance: ______________________

The next step, which you do not have to do here, would be to define a metric that combines both goals (e.g., the sum of the number of global transaction classes and the load imbalance) and then write an optimization program that finds the best fragment allocation. But thankfully you do not have to do all that for this exam!!
Problem 4 (10 points)

In this problem we consider a distributed two-phase commit protocol. This protocol is analogous to the centralized 2PC one described in class, except that there is no coordinator. All sites play both the role of a coordinator and of a participant.

The state transition diagram for this protocol is shown below, except that some information has been omitted. All sites use this same diagram. Your job is to fill in the missing information. The notation is as used in class.

Initially, one of the sites gets a “go” message, and sends out to all participants both an “exec” message (telling them to do their part of the transaction) and an “ok” message (telling them that this initial site has done its work and is prepared to commit). Note that the initial “go” message, and the “exec” messages include the list of all participants, so all sites will know all the participants in the protocol.

When a site times out waiting for another site to respond (“t” message), it sends out a “ping” message, as shown in the diagram. (The message goes to the node(s) that have failed to respond.) One of your jobs is to figure out how to respond to the ping messages.

Complete the state transition diagram below that shows the behavior of any node running the distributed 2PC. Note that some transitions are missing both the triggering message (above the line) and the response (below the line). Other transitions are just missing the response.

You do not need to add any other transitions (arrows) or any other message types to make the protocol work.

One final note: Do not send unnecessary messages. For example, in the timeout transitions we only send out ping messages to sites that have not responded, not to everybody (i.e., we do not use “ping *”).
Problem 5 (10 points)

Consider the following schedule on transactions $T_1$, $T_2$ and $T_3$ that runs on Sites A and B. (The actions on the left execute at Site A, while the actions on the right run on Site B.)

(a) Show the precedence graph for this schedule:

(b) Is this schedule serializable?

Answer (Yes/No):

(c) Can this schedule be produced by a two phase locking system? (Remember, a schedule can be produced by 2PL if one can add lock and unlock actions and obtain a legal schedule with well-formed and two-phase locking transactions.)

Answer (Yes/No):
(d) Can this schedule be produced by a timestamp ordering algorithm, assuming the timestamp order is $ts(T_1) < ts(T_2) < ts(T_3)$?

Answer (Yes/No):_____________

(e) Can this schedule be produced by a timestamp ordering algorithm, assuming the timestamp order is $ts(T_3) < ts(T_2) < ts(T_1)$?

Answer (Yes/No):_____________

(f) Is there an ordering of the timestamps $ts(T_1), ts(T_2), ts(T_3)$ so that this schedule can be produced by a strict timestamp ordering mechanism? [Hint: Consider specifically the strict TO algorithm presented in the class notes. Could this algorithm yield the schedule on the previous page?]

Answer (Yes/No):_____________

If Yes, give ordering:_____________