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>Total</td>
<td>50</td>
<td></td>
</tr>
</tbody>
</table>
Problem 1 (10 points)

We wish to horizontally partition two relations $R(A, C)$ and $T(B, C)$ across two sites $S_1$ and $S_2$. The partitions will be based on a parameter $X$ (that we will choose) as follows:

- Site $S_1$ contains fragments $R_1 = \sigma_{A>X} R$ and $T_1 = \sigma_{B>X} T$;
- Site $S_2$ contains fragments $R_2 = \sigma_{A\leq X} R$ and $T_2 = \sigma_{B\leq X} T$.

To decide on a good value of $X$ we will use a simple cost model. Given a value of $X$, if a query instance needs to execute at a single site, its cost is 1 unit; if it executes at 2 sites, its cost is 2 units. The total cost of a workload (set of query instances) is the sum of the costs of the individual queries. For example, if a workload has 3 query instances, one that runs at a single site, and two that need to run at both sites, the workload cost is 5.

For this problem we consider 4 types of queries:

- $QT_1 : \sigma_{A \leq 20}(R)$
- $QT_2 : \sigma_{A > 20}(R)$
- $QT_3 : \sigma_{B \leq 10}(T)$
- $QT_4 : \sigma_{B > 10}(T)$

(a) Consider the following workload $W_1$:

<table>
<thead>
<tr>
<th>query type</th>
<th>number of instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>$QT_1$</td>
<td>75</td>
</tr>
<tr>
<td>$QT_2$</td>
<td>25</td>
</tr>
</tbody>
</table>

If $X = 5$, how many query instances (out of the 100 in the workload) involve both $S_1$ and $S_2$?

Number of query instances: ___________

**Solution:** 75

When $X = 5$, only $QT_1$ queries need to run at both sites.

(b) What is the cost of workload $W_1$ (defined in part (a))? 

Cost: ___________

**Solution:** 175

That is, $25 + 2 * 75$. 

2
(c) What value of $X$ minimizes the cost of workload $W_1$? What is the minimum cost for that value of $X$?

Best $X$ value for $W_1$: ___________

Cost: ___________

**Solution:** $X = 20$, Cost: 100.

Both $QT_1$ and $QT_2$ queries run at one site, in this case.

(d) Next consider the following workload $W_2$:

<table>
<thead>
<tr>
<th>query type</th>
<th>number of instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>$QT_1$</td>
<td>30</td>
</tr>
<tr>
<td>$QT_2$</td>
<td>10</td>
</tr>
<tr>
<td>$QT_3$</td>
<td>45</td>
</tr>
<tr>
<td>$QT_4$</td>
<td>15</td>
</tr>
</tbody>
</table>

What value of $X$ minimizes the cost of workload $W_2$? What is the minimum cost for that value of $X$?

Best $X$ value for $W_2$: ___________

Cost: ___________

**Solution:** $X = 20$, Cost: 115.

Only $QT_4$ queries need to run at both sites.
Problem 2 (10 points)

Consider four transactions:

- \( T_0 : W_0(V) \)
- \( T_1 : W_1(Y)W_1(V) \)
- \( T_2 : R_2(X)R_2(Y)W_2(Z) \)
- \( T_3 : W_3(X) \)

(a) Assume that the four transactions successfully committed under a timestamp ordering (TO) scheduler using timestamps:

\[ ts(T_0) < ts(T_1) < ts(T_2) < ts(T_3) \]

In addition assume that (i) Thomas’ write rule was not applied; and (ii) the scheduler is not necessarily strict (locks can be released early).

For each schedule below, answer YES if that schedule is possible and NO otherwise:

1. \( R_2(X)W_3(X)W_1(Y)W_0(V)R_2(Y)W_1(V)W_2(Z) : \) ____________
   **Solution:** YES

2. \( R_2(X)W_3(X)R_2(Y)W_0(V)W_1(Y)W_1(V)W_2(Z) : \) ____________
   **Solution:** NO (\( W_1(Y) \) after \( R_2(Y) \))

3. \( W_1(Y)R_2(X)W_0(V)W_3(X)R_2(Y)W_2(Z)W_1(V) : \) ____________
   **Solution:** YES

4. \( W_0(V)R_2(X)R_2(Y)W_3(X)W_1(Y)W_1(V)W_2(Z) : \) ____________
   **Solution:** NO (\( W_1(Y) \) after \( R_2(Y) \))
(b) Assume that the four transactions successfully committed under a TO scheduler resulting in the following schedule:

\[ W_1(Y)W_3(X)R_2(X)R_2(Y)W_0(V)W_2(Z)W_1(V) \]

In addition assume that (i) Thomas’ write rule was not applied; and (ii) the scheduler is not necessarily strict (locks can be released early).

For each set of timestamps answer YES if the transactions could have committed under these timestamps and NO otherwise:

\[ \text{ts}(T_0) < \text{ts}(T_3) < \text{ts}(T_1) < \text{ts}(T_2): \]

\[ \text{Solution: YES} \]

\[ \text{ts}(T_0) < \text{ts}(T_1) < \text{ts}(T_2) < \text{ts}(T_3): \]

\[ \text{Solution: NO} \]

\[ \text{ts}(T_3) < \text{ts}(T_2) < \text{ts}(T_0) < \text{ts}(T_1): \]

\[ \text{Solution: NO} \]

\[ \text{ts}(T_0) < \text{ts}(T_1) < \text{ts}(T_3) < \text{ts}(T_2): \]

\[ \text{Solution: YES} \]

\[ \text{ts}(T_3) < \text{ts}(T_0) < \text{ts}(T_1) < \text{ts}(T_2): \]

\[ \text{Solution: YES} \]

\[ \text{ts}(T_1) < \text{ts}(T_3) < \text{ts}(T_2) < \text{ts}(T_0): \]

\[ \text{Solution: NO} \]

Based on the schedule, we can infer the following relations:

\[ \text{ts}(T_0) < \text{ts}(T_1) < \text{ts}(T_2) \text{ AND } \text{ts}(T_3) < \text{ts}(T_2) \]
Problem 3 (10 points)

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

- $R_1 = \sigma_{A \leq 50}(R)$
- $R_2 = \sigma_{50 < A \leq 100}(R)$
- $R_3 = \sigma_{A > 100}(R)$
- $S_1 = \pi_{A,C}(S)$
- $S_2 = \pi_{A,D}(S)$

(a) Perform decomposition and localization to transform the following query into an optimized operator tree on fragments. Optimize the query as much as possible.

SELECT $A$, $B$
FROM $R$, $S$
WHERE $R.A = S.A$

Solution

```
U
  └─ R_1
        └─ S_1 (or S_2)

U
  └─ R_2
        └─ S_1 (or S_2)

U
  └─ R_3
        └─ S_1 (or S_2)
```

or

```
U
  └─ R_1
        └─ S_1 (or S_2)

U
  └─ R_2
        └─ S_1 (or S_2)

U
  └─ R_3
        └─ S_1 (or S_2)
```

(b) Say that
- $R_1$ contains 50 tuples and resides at computer $RN_1$,
- $R_2$ contains 100 tuples and resides at computer $RN_2$,
- $R_3$ contains 200 tuples and resides at computer $RN_3$,
- $S_1$ contains 100 tuples and resides at computer $SN_1$,
- $S_2$ contains 100 tuples and resides at computer $SN_1$.

The $A$ attribute is 10 bytes in size; all other attributes are 100 bytes. Thus, for example, each $R$ tuple is 110 bytes in size, and each $S$ tuple is 210 bytes in size. Furthermore, assume that 20% of the tuples in each $R$ fragment join with $S$. For instance, 10 tuples in $R_1$ have $A$ values found in $S$.

Determine the execution plan for the query of part (a) that minimizes the amount of data transmitted. Ignore the steps that assemble the final result at a single node (the cost of those steps is the same for all plans). Write your plan in the table below, indicating the number of bytes transmitted by each step. For each step, use the notation $expression @ node \rightarrow node$. For example, “$\sigma_{D>20}(S) @ SN_2 \rightarrow RN_2, RN_3$” means that node $SN_2$ sends its $S$ tuples with $D > 20$ both to nodes $RN_2$ and $RN_3$. If a step computes part of the final answer, you can omit the “$\rightarrow node$” part (and it will send zero bytes). You may not need all the steps provided in the table.

**Solution**

<table>
<thead>
<tr>
<th>step</th>
<th>transmitted bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\pi_A(R_1) @ RN_1 \rightarrow SN_1$</td>
<td>500</td>
</tr>
<tr>
<td>$\pi_A(R_1)$ semijoin $S_1 @ SN_1 \rightarrow RN_1$</td>
<td>100</td>
</tr>
<tr>
<td>$\pi_A(S_1) @ SN_1 \rightarrow RN_2, RN_3$</td>
<td>2000</td>
</tr>
</tbody>
</table>

A **better solution** (2 bonus points)
\begin{center}
\begin{tabular}{|c|c|}
\hline
\textbf{step} & \textbf{transmitted bytes} \\
\hline
\(\sigma_{A \leq 50} \pi_A S_1 @ S N_1 \rightarrow R N_1\) & [100, 400] \\
\hline
\(\sigma_{50 < A \leq 100} \pi_A S_1 @ S N_1 \rightarrow R N_2\) & [200, 500] \\
\hline
\(\sigma_{A > 100} \pi_A S_1 @ S N_1 \rightarrow R N_3\) & [400, 700] \\
\hline
& \text{total cost of the above three steps} = 1000 \\
\hline
\end{tabular}
\end{center}

(c) Now let us change some of the fragment sizes:
\begin{itemize}
\item \(R_1\) contains 10 tuples and resides at computer \(RN_1\),
\item \(R_2\) contains 10 tuples and resides at computer \(RN_2\),
\item \(R_3\) contains 10 tuples and resides at computer \(RN_3\),
\item \(S_1\) contains 100 tuples and resides at computer \(SN_1\),
\item \(S_2\) contains 100 tuples and resides at computer \(SN_1\).
\end{itemize}
The \(A\) attribute is still 10 bytes in size; all other attributes are still 100 bytes. Again assume that 20\% of the tuples in each \(R\) fragment join with \(S\).

Determine the execution plan for the query of part (a) that minimizes the amount of data transmitted, given the new fragment sizes. Use same assumptions and notation as in part (b). You may not need all the steps provided in the table.

**Solution**
\begin{center}
\begin{tabular}{|c|c|}
\hline
\textbf{step} & \textbf{transmitted bytes} \\
\hline
\(\pi_A(R_1) @ R N_1 \rightarrow S N_1\) & 100 \\
\hline
\(\pi_A(R_2) @ R N_2 \rightarrow S N_1\) & 100 \\
\hline
\(\pi_A(R_3) @ R N_3 \rightarrow S N_1\) & 100 \\
\hline
\(\pi_A(R_1)\) semijoin \(S_1 @ S N_1 \rightarrow R N_1\) & 20 \\
\hline
\(\pi_A(R_2)\) semijoin \(S_1 @ S N_1 \rightarrow R N_2\) & 20 \\
\hline
\(\pi_A(R_3)\) semijoin \(S_1 @ S N_1 \rightarrow R N_3\) & 20 \\
\hline
\end{tabular}
\end{center}
Problem 4 (10 points)

Consider the linear 2-phase commit protocol described on Slide 40 of CS347 Notes 06. In this protocol, each node does its work and enters a “pre-commit” state (state W), and then asks its immediate neighbor to do its work. The last node in the chain commits the transactions and then sends a commit message along the chain, in reverse.

Complete the state transition diagram below that shows the behavior of any node running the linear 2PC. All nodes use this same logic.

The three transitions out of the I state have already been filled in. For the failure transitions (dotted lines), the triggering message is given, you only need to fill in the response (under the line). Note that there are two failure transitions out of state W (one for a timeout, and one for receipt of an exec message).

In this diagram, we use “N” to represent the next node down the chain, and “P” the previous node. Thus, “msg P” sends message “msg” to the previous node. If the first node in the chain tries to send “msg P”, then no message is sent.

For simplicity, we can assume that nodes can indefinitely remember whether they committed or aborted a transaction. Thus, there is no need to transition to a finish state where the outcome can be forgotten.

You do not need to add any other transitions (arrows) or any other message types to make the protocol work. Look for the horizontal lines in the diagram that are missing the triggering message or output message: these are the items you need to fill in.

Solution
Problem 5 (10 points)

The fragments of relation \( R \) are located at three sites, \( A, B, \) and \( C \), and the relation is not partitioned on attribute \( K \). We wish to repartition \( R \) using range partitioning on the integer attribute \( K \) and store the resulting fragments at two new sites \( D \) and \( E \). A central coordinator is in charge of computing the partition vector, trying to ensure that \( D \) and \( E \) have the same amount of data. The following statistics (histogram, minimum, maximum) are stored locally at \( A, B, \) and \( C \), respectively:

### Site Statistics

<table>
<thead>
<tr>
<th>Site</th>
<th>Histogram</th>
<th>( min(K) )</th>
<th>( max(K) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( R_A )</td>
<td><img src="image" alt="Histogram" /></td>
<td>3</td>
<td>16</td>
</tr>
<tr>
<td>( R_B )</td>
<td><img src="image" alt="Histogram" /></td>
<td>9</td>
<td>21</td>
</tr>
<tr>
<td>( R_C )</td>
<td><img src="image" alt="Histogram" /></td>
<td>7</td>
<td>26</td>
</tr>
</tbody>
</table>

(a) First, let us assume that the sites only report to the coordinator the minimum and maximum \( K \)-values, as well as the total number of tuples at the site. Write down the rules for repartitioning \( R \) (the select statements corresponding to the new fragments), based on a threshold \( K \)-value computed by the coordinator, following the strategy presented in class. (Note: Since we are dealing with integers, the expected partition sizes may not be identical. Just try to make them as close as possible.)

**Partition \( R_D \):**

**Solution:** \( \sigma_{K \leq 14}(R) \)

**Partition \( R_E \):**

**Solution:** \( \sigma_{K > 14}(R) \)

Site \( R_A \) hosts 14 tuples. If we assume a uniform distribution across the integer values of \( K \) in the range \([3, 16]\), we can infer a distribution of 1 tuple per value. Note that the number of integer values in the range \([Z, Y]\) is \( Y - (Z - 1) \), e.g., \( 16 - (3 - 1) = 14 \) values in \([3, 16]\). Following the same principle, we can infer a distribution of 2 tuples per value in sites \( R_B \) and \( R_C \). Hence if we “split” at \( X \) the number of tuples going to one site is:
\[ L(X) = (X - (3 - 1)) + 2 \times (X - (9 - 1)) + 2 \times (X - (7 - 1)) \]

we want \( L(X) \) to be as close to \( \frac{80}{2} = 40 \) as possible; 80 is the total number of tuples across all sites. In fact, if we solve the equation, we get an integer \( X = 14 \).

(b) Second, let us assume that the sites report their full histograms shown above to the coordinator. What partitioning would the coordinator generate in this case?

Partition \( R_D \): _____________  
Solution: \( \sigma_{K \leq 12}(R) \)

Partition \( R_E \): _____________  
Solution: \( \sigma_{K > 12}(R) \)

Looking at the histograms, if we split at \( X = 12 \), there are exactly 40 tuples going on each side.