This exam is open book and notes. In addition to paper versions, you may use a calculator and your laptop. On your laptop, you may only access personal notes and official course material (notes and videos). You may not access other material or use your laptop to communicate with other people. If you find a question to be ambiguous, make and state a reasonable assumption and proceed with answering the question. 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>6</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>60</td>
<td></td>
</tr>
</tbody>
</table>
Problem 1 (10 points)

Consider a database schema that represents store sales. Each sale transaction is represented by a tuple in the following relation:

\[ S(C, D, P, Q) \]

where \( C \) is the customer id, \( D \) is the date, \( P \) is the product id, and \( Q \) is the quantity. For each customer, we have a relation

\[ R(C, N, X) \]

where \( C \) is again the customer id, \( N \) is the name and \( X \) is the sex. There are other tables in the system but we are not concerned with them here. Some additional information:

- \( C \) is the primary key for the customer table \( R \);
- \((C, D)\) is the primary key for the sale table \( S \);
- Attribute \( X \) can only be “m” (male) or “f” (female).

The following predicates appear in the majority of queries:

- \( Q \leq 10, \ Q \leq 5 \)
- \( X = m, \ X = f \)

(a) Write down the fragments that result from performing primary horizontal fragmentation on the \( S, R \) tables, using the predicates given above.

FRAGMENTS FOR \( S \): \( S_1 = \sigma_{Q \leq 5} S \); \( S_2 = \sigma_{Q > 5 \land Q \leq 10} S \); \( S_3 = \sigma_{Q > 10} S \)

FRAGMENTS FOR \( R \): \( R_1 = \sigma_{X = m} R \); \( R_2 = \sigma_{X = f} R \)

(b) One of the two relations can be further fragmented using derived horizontal fragmentation based on the fragmentation of the other relation. Write down the resulting fragments.

RELATION THAT IS FURTHER FRAGMENTED: \( S \)

ITS FRAGMENTS: \( S_{ij} = S_i \rhd R_j \) for \( i = 1, 2, 3 \) and \( j = 1, 2 \)
Problem 2 (10 points)

A database system uses timestamp-ordering concurrency control. The following table shows the contents of the timestamp cache:

<table>
<thead>
<tr>
<th>Object</th>
<th>mrts</th>
<th>mwts</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>5</td>
<td>18</td>
</tr>
<tr>
<td>Y</td>
<td>30</td>
<td>20</td>
</tr>
<tr>
<td>Z</td>
<td>40</td>
<td>22</td>
</tr>
</tbody>
</table>

The following events occur at this point, in this order:

1. Transaction $T_1$ with timestamp 22 tries to read $Y$.
2. Transaction $T_2$ with timestamp 13 tries to read $X$.
3. Transaction $T_3$ with timestamp 30 tries to read $Z$.
4. Transaction $T_4$ with timestamp 40 tries to write $X$.
5. Transaction $T_5$ with timestamp 37 tries to read $X$.

Note that these events are not independent. For example, the actions of $T_1$ may change the timestamp cache, and therefore impact what $T_2$ sees. Among $T_1$ through $T_5$, assume that the transactions whose requests get denied are aborted and the transactions whose requests succeed commit immediately (before the next request is seen). Also, assume that the Thomas write rule is not in use.

(a) Write below “YES” if the transaction is allowed or “NO” if it is not allowed to perform the requested operations:

- $T_1$: YES
- $T_2$: NO
- $T_3$: YES
- $T_4$: YES
- $T_5$: NO
(b) Now assume that before \(T_1\) through \(T_5\) make their requests the timestamp cache is purged with \(min = 35\). Show below the contents of the timestamp cache after this purging. (If an object is not in the cache, leave its line blank.)

There are two acceptable solutions.

(i) If you assume that the cache purge either removes the entire record for an object or leaves it unaltered, the solution is:

<table>
<thead>
<tr>
<th>Object</th>
<th>mrts</th>
<th>mwts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z</td>
<td>40</td>
<td>22</td>
</tr>
</tbody>
</table>

(ii) Alternatively, if you assume that the cache purge can alter each individual max timestamp cell, the solution is:

<table>
<thead>
<tr>
<th>Object</th>
<th>mrts</th>
<th>mwts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z</td>
<td>40</td>
<td></td>
</tr>
</tbody>
</table>

(c) Given that the cache has been purged (\(min = 35\)), indicate below if the transactions are allowed (write “YES”) or not allowed (write “NO”) to perform the requested operations:

For your convenience, we repeat the transactions here. This list is identical with that on the previous page.

1. Transaction \(T_1\) with timestamp 22 tries to read \(Y\).
2. Transaction \(T_2\) with timestamp 13 tries to read \(X\).
3. Transaction \(T_3\) with timestamp 30 tries to read \(Z\).
4. Transaction \(T_4\) with timestamp 40 tries to write \(X\).
5. Transaction \(T_5\) with timestamp 37 tries to read \(X\).

- \(T_1\): NO
- \(T_2\): NO
- \(T_3\): (i) YES or (ii) NO
- \(T_4\): YES
- \(T_5\): NO
Problem 3 (10 points)

A group of four nodes is trying to terminate a transaction $T$ after a coordinator failure. Assume that the network is reliable and without partitions and that nodes are fail-stop. All other nodes, including the original coordinator are not responding. Assume that recovering nodes must wait for an operational node (or for all nodes to recover).

(a) Assume that the basic three phase commit protocol is in use. The table below shows various scenarios, one per line. For each scenario, the state of the four nodes is given, using the terminology used in class. For each scenario, indicate if

- The nodes should try to commit (write “COMMIT” in the space provided);
- The nodes should try abort (write “ABORT”) the transaction;
- The nodes must block (write “BLOCK”) until it can contact an operational node (or until all nodes recover).
- The indicated scenario can never arise (write “IMPOSSIBLE”).

<table>
<thead>
<tr>
<th>Site 1</th>
<th>Site 2</th>
<th>Site 3</th>
<th>Site 4</th>
<th>Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>ABORT</td>
</tr>
<tr>
<td>W</td>
<td>C</td>
<td>W</td>
<td>W</td>
<td>IMPOSSIBLE</td>
</tr>
<tr>
<td>W</td>
<td>W</td>
<td>A</td>
<td>W</td>
<td>ABORT</td>
</tr>
<tr>
<td>P</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>COMMIT</td>
</tr>
<tr>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>COMMIT</td>
</tr>
<tr>
<td>P</td>
<td>P</td>
<td>P</td>
<td>C</td>
<td>COMMIT</td>
</tr>
<tr>
<td>P</td>
<td>P</td>
<td>A</td>
<td>P</td>
<td>IMPOSSIBLE</td>
</tr>
</tbody>
</table>

(b) Now assume that the two phase commit protocol is used. Indicate the outcome for each scenario below, using “COMMIT,” “ABORT,” “BLOCK,” or “IMPOSSIBLE.”

<table>
<thead>
<tr>
<th>Node 1</th>
<th>Node 2</th>
<th>Node 3</th>
<th>Node 4</th>
<th>Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>W</td>
<td>W</td>
<td>W</td>
<td>W</td>
<td>BLOCK</td>
</tr>
<tr>
<td>W</td>
<td>C</td>
<td>W</td>
<td>W</td>
<td>COMMIT</td>
</tr>
<tr>
<td>W</td>
<td>W</td>
<td>A</td>
<td>W</td>
<td>ABORT</td>
</tr>
<tr>
<td>C</td>
<td>W</td>
<td>A</td>
<td>W</td>
<td>IMPOSSIBLE</td>
</tr>
</tbody>
</table>

5
Problem 4 (10 points)

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

- $R_1 = \sigma_{(A<20)\land(B>40)}(\pi_{A,B}R)$
- $R_2 = \sigma_{(A\geq20)\land(B>40)}(\pi_{A,B}R)$
- $R_3 = \pi_{A,C}R$
- $R_4 = \sigma_{B\leq40}(\pi_{A,B}R)$

and

- $S_1 = \pi_{A,D}S$
- $S_2 = \pi_{A,E}S$

Perform decomposition but not localization to transform the following query into an operator tree. Optimize the query as much as possible.

(a) \text{SELECT B} \\
\text{FROM R} \\
\text{WHERE B > 40 AND A = 20}

\text{ANSWER:}

$$\pi_B \left( \sigma_{(A=20)\land(B>40)}(\pi_{A,B}R) \right)$$

Perform decomposition and localization to transform the following queries into an optimized operator tree on fragments. Optimize each query as much as possible. When ordering joins, you should perform a join on fragments of the same relation before a join on different relations.

(b) \text{SELECT B} \\
\text{FROM R} \\
\text{WHERE B > 40 AND A = 20}

\text{ANSWER:}

$$\pi_B \left( \sigma_{A=20}R_2 \right)$$
(c) SELECT A, D, E
FROM R, S
WHERE R.A = S.A AND B <= 40

ANSWER:

\[(S_1 \bowtie S_2) \bowtie (\pi_A R_4)\]

(d) SELECT C
FROM R, S
WHERE R.A = S.A AND E = e

ANSWER:

\[\pi_C (R_3 \bowtie (\pi_A (\sigma_{E=e} S_2)))\]

(e) SELECT C, E
FROM R, S
WHERE R.A = S.A AND E = e AND B <= 40

ANSWER:

\[\pi_{C,E} ((R_3 \bowtie \pi_A R_4) \bowtie (\sigma_{E=e} S_2))\]
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. The following statistics (histogram, minimum, maximum) are stored locally at $A$, $B$, and $C$, respectively:

<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>$K$</td>
<td>[6,10]</td>
<td>[11,15]</td>
</tr>
<tr>
<td></td>
<td># of tuples</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>$R_B$</td>
<td>$K$</td>
<td>[6,10]</td>
<td>[11,15]</td>
</tr>
<tr>
<td></td>
<td># of tuples</td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>$R_C$</td>
<td>$K$</td>
<td>[6,10]</td>
<td>[11,15]</td>
</tr>
<tr>
<td></td>
<td># of tuples</td>
<td>21</td>
<td>5</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 algorithm presented in class.

Partition $R_D$: $K \leq 17$

Partition $R_E$: $K > 17$

(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 $R_D$: $K \leq 13$

Partition $R_E$: $K > 13$
Problem 6 (10 points)

Consider a variant of the RPWP-LC replication scheme as follows. Read-only transactions may execute on any site by locking the local data items and committing locally. Transactions with writes are performed as per the RPWP-LC technique described in class. We will refer to this replication scheme as read-lock one, write-lock primary with local commit, or ROWP-LC.

Our distributed database system consists of three nodes A, B, and C. For replication schemes with a primary node, A acts as the primary; backups B and C cannot take over the role of primary in case A is unavailable.

Data items X and Y both have the initial value $X = Y = 12$ at all three nodes. The following two transactions are executed concurrently by the system:

$T_1 : r_1(X), Y ← X + 2, w_1(Y)$

$T_2 : r_2(X), r_2(Y)$

As the execution progresses, an observer notes that operation $w_1(Y)$ was performed before $r_2(Y)$.

Given the above, consider a set of scenarios involving different node availabilities:

- All nodes are up
- Only A and B are up
- Only B and C are up

and different replication schemes:

- ROWA
- RPWP-LC
- ROWP-LC

For each scenario, indicate whether transactions $T_1$ and $T_2$ can commit or not by writing “YES” or “NO”, respectively, in the corresponding cells in the $c_1?$ and $c_2?$ columns of the table below. For each scenario in which $T_2$ can commit, list all possible values of $Y$ that the transaction could return as a result of $r_2(Y)$ as well.

<table>
<thead>
<tr>
<th>All nodes are up</th>
<th>Only A and B are up</th>
<th>Only B and C are up</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c_1?$</td>
<td>$c_2?$</td>
<td>$r_2(Y)$</td>
</tr>
<tr>
<td><strong>ROWA</strong></td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td><strong>RPWP-LC</strong></td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td><strong>ROWP-LC</strong></td>
<td>YES</td>
<td>YES</td>
</tr>
</tbody>
</table>