Handing In

Upload your homework to Gradescope.
Please write your Banner ID on your submission. Do not write your name on the submission.

Warmup #1

Assume that the following 2 plans answer the same query:
Plan A:
\[(\sigma_{\text{att}=3} A) \bowtie B\]
Plan B:
\[\sigma_{\text{att}=3} (A \bowtie B)\]

Answer the following questions:
1. If you had no knowledge of the data which one would you pick? Why?
2. Is there a situation in which you would prefer the other plan? Why?

Warmup #2

Consider the relations \(R(A, B), S(B, C)\) and \(T(C, D)\), such that their sizes in tuples are 3 \(\times 10^4\), 2 \(\times 10^5\) and \(10^4\), respectively.
Consider the query:
\[\sigma_{R.A<40}(R \bowtie S \bowtie T)\]

Estimate the size of the size of the resulting relation given the following 3 pieces of information:
1. Selectivity of \(R.B = S.B\) is \(\frac{1}{3}\)
2. Selectivity of \(S.C = T.C\) is \(\frac{1}{10}\)
3. Selectivity of \(R.A < 40\) is \(\frac{1}{2}\)

Warmup #3

List the ACID properties. Explain the usefulness of each.

Warmup #4

Assume there is a database with 2 objects \(A\) and \(B\) and consider two transactions \(T1\) and \(T2\).
Give an example an example of a schedule composed of reads and writes of \(T1\) and \(T2\) that result in a read-write conflict.
Problem 5 (To Be Graded)

Consider the following (incomplete) schedule S:

<table>
<thead>
<tr>
<th></th>
<th>T1</th>
<th>T2</th>
<th>T3</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>R(X)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>R(Y)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>W(X)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>R(Y)</td>
<td>W(Y)</td>
</tr>
<tr>
<td></td>
<td>W(X)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>R(Y)</td>
<td></td>
</tr>
</tbody>
</table>

1. Can you determine the serializability graph for this schedule? Assuming that all three transactions eventually commit, show the serializability graph.

2. For each of the following, modify S to create a complete schedule that satisfies the stated condition. If a modification is not possible, briefly explain why. If it is possible, use the smallest possible number of actions (read, write, commit, or abort). You are free to add new actions anywhere in the schedule S, including in the middle.

   (a) Resulting schedule avoids cascading aborts but is not recoverable
   (b) Resulting schedule is recoverable
   (c) Resulting schedule is conflict-serializable

Bonus Problem 6

So far we have looked at the impact of read and write and how they pertain to a database transaction. Suppose that a new database finds that most of their work is incrementing and decrementing integer records. In this new DBMS they want to introduce a new operation called increment, which increases an integer valued object by 1, and decrement (opposite of increment). A transaction that increments an object need not know the value of the object; increment and decrement are versions of blind writes. In addition to the shared and exclusive locks, two special locks are supported: an object must be locked in I mode before increment and locked in D mode before decrementing it. An I lock is compatible with another I or D lock on the same object, but not with S and X locks.

1. Illustrate how the use of I and D locks can increase concurrency. (Show a schedule allowed by strict 2PL that only uses S and X locks. Explain how the use of I and D locks can allow more actions to be interleaved, while continuing to follow Strict 2PL.)

2. Informally explain how Strict 2PL guarantees serializability even in the presence of I and D locks. (Identify which pairs of actions conflict, in the sense that their relative order can affect the result, and show that the use of S, X, I and D locks according to Strict 2PL orders all conflicting pairs of actions to be the same as the order in some serial schedule.)