

Name:

Problem 1

Consider the following two transactions:

T0:

read(A);

read(B);

if (A = 0) then B = B + 1;

write(B);

T1:

read(B);

read(A);

if (B = 0) then A = A + 1;

write(A);

Let the consistency requirement be (A = 0) OR (B = 0), with A = B = 0 the initial values

- (a) Show that every serial execution involving these two transactions preserves the consistency of the database.
- (b) Show a concurrent execution of T0 and T1 which produces a nonserializable schedule
- (c) Is there a concurrent execution of T0 and T1 which produces a serializable schedule? Prove why or why not.

Problem 2

Indicate whether each of the following is true or false. Explain why (unless it is already explicitly explained in the notes.)

1. Consider the merge join of R and S. The minimum memory requirement for the whole process is the square root of $\max(|R|, |S|)$ blocks - ignoring constant factors that do not depend on the size of R and S. $|R|$ and $|S|$ stand for the sizes of the relations in blocks.
2. Consider the hash join of R and S. The minimum memory requirement for the whole process is the square root of $\max(|R|, |S|)$ blocks - ignoring constant factors that do not depend on the size of R and S.
3. Checkpointing can significantly reduce recovery time when using UNDO logging.
4. If a schedule is not conflict serializable then it will definitely violate the consistency constraints of the database.
5. There are view serializable schedules that are not conflict serializable.
6. Two schedules with identical precedence graphs must be conflict equivalent.

7. In query optimization it is always better to perform projections as early as possible.

8. Sequential IOs are more expensive than random IOs.

9. When the sizes of join relations are big and we have indexes on the join attributes, the join index technique will outperform every other join technique we have seen.

10. A second level index for an indexed relation is usually dense.

11. Natural join is associative and commutative.

12. For queries of the form "SELECT_{A=a}R" extensible hashing is better than B+ tree.

13. For queries of the form "SELECT_{A>a}R" extensible hashing is better than B+ tree.

Problem 3

For each schedule below indicate whether the schedule is conflict serializable or not. If yes, provide the equivalent serial schedule.

1. $w1(A), w2(B), w3(C), r3(A), r3(B)$
2. $w1(A), w2(B), w3(C), r3(A), r3(B)$
3. $r1(A), w2(A), w2(B), r1(B)$
4. $w1(A), w1(B), r2(A), w2(B), w1(C), r3(B), w3(C)$

Problem 4

Consider a hash join with the following parameters:

- k : the number of buckets into which you will partition the relations
- m : the total amount of memory, in blocks. Remember that one of these blocks is used for reading the relations.
- s : the number of blocks for each relation (equal size)

- (a) How large does m need to be so that the first phase of the algorithm can run?
- (b) How large does m need to be so that the second phase can run?
- (c) Hybrid hash join is an optimization of hash join that keeps some of the buckets in memory during the "bucketizing" phase. Derive an equation that gives the maximum number of buckets that can be kept in memory when joining two relations. Introduce additional parameters to the ones above only if you feel it is absolutely necessary.

Problem 5

Consider the following schedule. Is it view serializable? Is it conflict serializable? For conflict serializability prove why or why not.

$w_3(B), w_3(A), w_1(B), r_2(B), r_2(A), w_1(A)$

Problem 6

Let $R(A,B,C)$ and $S(C,D,E,F)$ be two relations. Transform the following relational algebra expression into an equivalent one in which selections and projections are performed as early as possible. Assume sets, not bags.) Please provide the answer in tree format.

$PROJECT_{B, F} SELECT_{R.A=1 \text{ AND } S.D=2 \text{ AND } R.C=S.C} R \times S$

Problem 7

Consider B+ trees where the maximum number of values in a node is 2 (and the minimum is 1). Give an example of 3-level B+ tree that can be reorganized into a 2-level tree with exactly the same values. Display both the 3-level and the 2-level tree. Use letters or numbers for the values.

Problem 8

Construct the hypergraph for the following expression.

*SELECT*_{R.A=W.A} ((((((R(A,B) JOIN S(B,C,D)) JOIN T(B,E,F)) JOIN U(F,G,H)) JOIN V(G,I)) JOIN W(I,J,A)))

Provide 4 left-deep algebraic expressions that do not involve cartesian products and compute the above.

Problem 9

Suppose we have relations $R(A,B)$, with 1000000 tuples, and $S(B,C)$, with 100000 tuples. Also assume that 20 of R 's tuples fit in one block and 100 records of S fit in one block. Assume that there are 500 possible values for B . Each of the 500 values appears with equal probability in R and with equal probability in S . Assume the main memory is 1500 blocks. Compute the cost of the following, first assuming pipelining and then without assuming pipelining.

- (a) $\text{SELECT}_{A=a}(R \times S)$. Do not optimize the expression (say by pushing the selection down. Just estimate the cost of computing it.)
- (b) Sort-merge join for $\text{SELECT}_{A=a}(R \text{ JOIN } S)$ assuming that the relations are not sorted.
- (c) Assuming indexes on $R.B$ and $S.B$ estimate the cost of join index. State assumptions you have made (other than the ones given by the problem). You do NOT have to provide answers for multiple sets of assumptions.

Problem 10

Is the following equation correct for sets? For sets with duplicates? Prove why or why not for each case.

$$\text{PROJECT}_{\text{attr}}(\text{R}-\text{S}) = (\text{PROJECT}_{\text{attr}}\text{R}) - (\text{PROJECT}_{\text{attr}}\text{S})$$