ONID:

CS540 Database Management Systems

Spring 2023

School of Electrical Engineering & Computer Science

Oregon State University

Final Examination

Time Limit: 120 minutes

- Including this cover page, this exam booklet contains 8 pages. Check if you have missing pages.
- The exam is closed book and closed notes. You are allowed to use scratch papers.
- Please write your solutions in the spaces provided on the exam.
- Please make your answers clear and succinct; you will lose credit for verbose, convoluted, or confusing answers. *Simplicity does count!*

Question:	1	2	3	4	5	6	7	Total
Points:	2	4	4	2	2	2	4	20
Score:								

1. **Query Processing: Join Algorithms**
   Consider relations *Student(StudentID, Name)* and *Enrollment(CourseID, StudentID)*. The size of relation *Student* is 40,000 pages and the size of *Enrollment* is 400 pages. Assume that the query processor has to choose between the in-memory (internal memory) join algorithms, page-oriented and block nested loop join algorithms, and the sort-merge join with two-pass multi-way merge-sort algorithm for sorting to perform the join of $Student \bowtie_{StudentID=StudentID} Enrollment$.

   (a) (2 points) Given that we have 405 pages available in main memory, i.e., $M = 405$, which one of the aforementioned algorithms is the fastest one to perform the join? You should also provide the cost, i.e., number of I/O accesses, of the join using your proposed algorithm.

2. **Query Processing Algorithms**
The *left anti-join* operator is a variant of the relational join. Given two input relations, it outputs only those tuples in the left relation that do *not* join with any tuple in the right relation. For example, consider two relations R(A, B) and S(B, C). Assume that R has the following tuples: (1, 10), (1, 20), (2, 30), (2, 40) and S has the following tuples: (10, 75), (10, 85), (30, 95). The left anti-join of R and S on attribute B will produce the following tuples: (1, 20) and (2, 40). Given the above description of left anti-joins, answer the following question.

   (a) (4 points) From the set of join algorithms: block nested loop, sort-merge, and hash join algorithms, pick the one that computes the results of the left anti-join for two relations with the fewest I/O accesses in average. You must explain the changes to the main steps of the algorithm in your extension of the selected algorithm and justify why it is faster than extensions of other algorithms.

3. **Query Processing Algorithms: Join Algorithms**
Users may like to see **only a small subset of query results**. For example, a user might like to see only a subset of 10 output tuples of a join between two large relations to get some high level understanding of the complete join.

   (a) (2 points) Assume that a user joins two relations R and S where each relation has one million pages. The number of available buffers in main memory, M, is 2000. The user would like to see **only 10 output tuples** of the join between R and S. From block nested loop and sort-merge join algorithms, which one is able to return these 10 tuples using the fewest I/O accesses in average? You should justify your answer briefly.

   (b) (2 points) Consider again the join of R and S with $M = 2000$ in the preceding question. Suggest a change to the slower algorithm in the preceding question to improve its number of I/O accesses for returning 10 output tuples from the join of R and S. If your change improves the running time of the algorithm *only in one special case*, your solution is *still acceptable*. You should explain one case (example) in which your change improves the number of I/O accesses of the algorithm.

4. **Query Optimization: Logical Plans**

Consider the following relations. Primary keys are underlined.

$Student(\underline{StudentId}, SName, DeptId)$
$Department(\underline{DeptId}, DName)$
$Faculty(\underline{FacultyId}, FName, DeptId)$

where the attribute *DeptId* in all relations refers to the IDs of the departments. Consider the following query.

```
SELECT SName, FName
FROM Student, Department, Faculty
WHERE Student.DeptId = Department.DeptId and
      Department.DeptId = Faculty.DeptId
```

(a) (2 points) Provide the list of logical query plans, i.e., query trees, that are considered by query optimizer that implements System-R optimization method.

5. **Query Optimization: Cost Estimation** Consider relations R(A, B) with 20,000 tuples. Use Selinger-style, i.e., System-R style, cost estimation formulas to answer the following questions.

   (a) (2 points) We want to compute U = $\sigma_{(A=20)and(1<B<10)}$ R. We know that V(R, A) = 500. We do **not** have any information about the number of distinct values or range of of values in B. What is a reasonable estimate on the size, i.e., number of tuples, of U?

6. **Concurrency Control: Serializability**

   For each following schedule, draw its serialization graph and state if it is serializable.

   (a) (2 points) T1:R(X), T2:R(Y), T3:W(X), T2:R(X), T1:Commit, T2:Commit, T3.Commit.

7. **Concurrency Control: Degrees of consistency** Determine the maximum possible degrees of consistency for T2 in the following schedule. You must find the maximum degree of consistency for T2 that makes this schedule possible. You may assume that each transaction in this schedule has at least degree consistency 0. The actions are listed in the order they are scheduled and prefixed with their transaction names.

   (a) (4 points) T1:W(X), T2:W(X), T1:Commit, T2:R(X), T2:Commit.