CSE 190D Spring 2019 Final Exam

Full Name:

Student ID:

Major:

INSTRUCTIONS

1. You have up to 2 hours and 59 minutes to complete this exam.

2. You can have up to 2 letter/A4-sized sheets of notes, formulas, etc. Apart from this, the exam is closed book/notes/electronics/peers.

3. Please wait until being told to start reading and working on the exam.

4. If you think a question is ambiguous, write down your assumptions, argue that they are reasonable, and then work on the problem using those assumptions.

5. Please ensure that your writing is clear and legible!

<table>
<thead>
<tr>
<th></th>
<th>Points</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q 1</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>Q 2</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>Q 3</td>
<td>36</td>
<td></td>
</tr>
<tr>
<td>Q 4</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>Extra Credit</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>
Consultants say three quintillion bytes of data are created every day.

It comes from everywhere. It knows all.

According to the book of Wikipedia, its name is "big data."

Big data lives in the cloud. It knows what we do.

In the past, our company did many evil things.

But if we accept big data in our servers, we will be saved from bankruptcy.

Let us pay.

Is it too late to side with evil? Shhh! It hears you.
Q 1. [20pts] For the following questions, clearly circle True or False.

1. All secondary indexes are unclustered indexes.
   
   True  False

2. Median is not an algebraic aggregate.
   
   True  False

3. Rename is the only operator in relational algebra whose runtime does not vary across different instances of the same relation schema.
   
   True  False

4. It is impossible for a parallel RDBMS to exhibit super-linear speedups.
   
   True  False

5. Natural joins are always equi joins.
   
   True  False

6. The blocked I/O improvement to the external merge sort may increase the I/O cost by increasing the number of merge passes.
   
   True  False

7. The iterator interface for physical operators is the main software abstraction used to implement pipelining.
   
   True  False
8. Sort-merge join is never amenable to pipelining and is always a blocking operator.

True  False

9. All recoverable schedules are Avoid-Cascading-Abort (ACA) schedules.

True  False

10. The union operator can always be executed using just an unmodified hash join implementation.

True  False
Q 2. [24pts] For the following questions, clearly circle the right answer. Only one option is correct.

1. Which relational operator implementation in a parallel RDBMS never needs on-the-fly re-partitioning of the data?
   (a) $\times$  (b) $\bowtie$  (c) $\pi$  (d) $\sigma$  (e) $\gamma$

2. Which index alternative stores the records themselves in the data entries?
   (a) AltRID  (b) AltRecord  (c) AltRIDList  (d) None of the above

3. Which ACID property states that the changes of committed transactions will be persisted in the database?
   (a) Atomicity  (b) Consistency  (c) Isolation  (d) Durability

4. Which of the following paradigms of parallelism is often called “symmetric multiprocessing” (SMP) database systems?
   (a) Shared-disk  (b) Shared-memory  (c) Shared-nothing

5. Which join order enumerations are the most amenable to pipelining if BNLJ was the only physical join operator allowed?
   (a) Left deep trees  (b) Right deep trees  (c) Bushy trees  (d) All of the above

6. Which of the following invariants/conditions always hold for an extendible hash index? GD and LD stand or global and local depths respectively.
   (a) All LD > GD  (b) All LD $\geq$ GD  (c) All LD < GD  (d) All LD $\leq$ GD

7. Which of these SQL capabilities has no counterpart in (extended) relational algebra?
   (a) WHERE  (b) GROUP BY  (c) LIMIT  (d) UNION  (e) SELECT DISTINCT
8. Which improvement to the external merge sort was designed to exploit the direct-memory access (DMA) capability of modern machines?

(a) Internal replacement sort  (b) Double buffering  (c) Blocked I/O

9. In which set operator’s hashing-based implementation does the in-memory hash table hold (a part of) the output?

(a) $\cup$  (b) $\cap$  (c) $-$  (d) $\times$

10. Which SQL isolation level ensures that the phantom problem will not arise?

(a) Read Uncommitted  (b) Read Committed  (c) Repeatable Read  (d) Serializable

11. Which component of the access latency of a magnetic hard disk is affected by how fast the head moves?

(a) Rotational delay  (b) Seek time  (c) Transfer time  (d) None of the above

12. Which is the most common data partitioning scheme for ETL in parallel RDBMSs?

(a) Round Robin  (b) Range-based  (c) Hashing-based  (d) Random
Q 3. [36pts] Consider the simplified sales database schema given to you in the separate sheet. The arrows represent key-foreign key dependencies.

1. (3pts) Write an SQL query to compute the total sales in California stores in 2019.

2. (3pts) Write a relational algebra expression to answer the same query as in the previous question. You can draw the logical query plan instead if you wish.
3. (3pts) Write a different relational algebra expression than your response to the above question that is logically equivalent and also answers the query in the first question. Once again, you can draw the logical query plan if you wish. (Hint: Use the algebraic rewrite rules covered in class.)

4. (4pts) Draw a physical query plan corresponding to one of your above two logical query plans. Mention which one it corresponds to. Clearly show which physical operator implementation was picked for each logical operator.
5. (3pts) Name 3 indexes, with at least one being a hash index, that match the selections in the query given in the first question (total sales in California stores in 2019).

6. (9pts) For the following pairs of queries, indicate if they are logically equivalent.

   (a) \( Q_1 : \gamma_{\text{Supplier, AVG(Price)}}(\text{Items}) \quad Q_2 : \gamma_{\text{Supplier, AVG(Price)}}(\pi_{\text{Supplier, Price}}(\text{Items})) \)

      Yes \quad No

   (b) \( Q_1 : (\text{Sales} \bowtie \text{Stores}) \bowtie \text{Customers} \quad Q_2 : (\text{Stores} \times \text{Customers}) \bowtie \text{Sales} \)

      Yes \quad No

   (c) \( Q_1 : \sigma_{\text{Price} \geq 100}(\text{Items}) \bowtie \text{Sales} - \sigma_{\text{Price} \geq 100 \land \text{Month='December'}}(\text{Sales} \bowtie \text{Items}) \)

      Yes \quad No
7. (11pts) For the following questions, use these pieces of information about the given sales database and the machine environment:

\[ N_{Sales} = 10^8, \quad N_{Stores} = 10^5, \quad N_{Items} = 10^6, \quad \text{and} \quad N_{Customers} = 10^7 \] (respective number of pages of each table in row store format)

Page size is 8 KB. Available buffer memory is 32 GB.
The fudge factor for hash tables is 1.4.
The size of each attribute is 8 B, except for S_State, I_Name, Supplier, Category, C_Name, and C_State, each which is 40 B.

Assume there are no indexes and the buffer pool is empty to start with. Exclude the cost of writing the output of a given query in your I/O cost calculations.

(a) (3pts) Which of the following joins cannot be executed with an I/O cost that involves just one read of each base table?

(i) Sales \bowtie\leftarrow Customers 
(ii) Sales \bowtie\leftarrow Stores 
(iii) Sales \bowtie\leftarrow Items

(b) (3pts) What is the I/O cost of the fastest possible physical query plan for the following query?

\[ \sigma_{CustomerID=123}(Customers \bowtie Sales)? \]

(i) \(10^7\) (ii) \(10^8\) (iii) \(11 \cdot 10^7\) (iv) \(12 \cdot 10^7\) (v) \(33 \cdot 10^7\)

(c) (5pts) What is the I/O cost of the fastest possible physical query plan for the following query?

\[ \gamma COUNT(\text{DISTINCT CustomerID})\big(\sigma_{C\_State\neq S\_State}\big((Customers \bowtie Sales) \bowtie\leftarrow Stores)\big)\)?

(i) \(1101 \cdot 10^5\) (ii) \(1303 \cdot 10^5\) (iii) \(3003 \cdot 10^5\) (iv) \(3301 \cdot 10^5\)

(v) \(3303 \cdot 10^5\) (vi) \(1505 \cdot 10^5\) (vii) \(5105 \cdot 10^5\) (viii) \(5501 \cdot 10^5\)
Q 4. [20pts] You are given a database with three distinct data objects A, B, and C. You are also given the following three transactions that arrive concurrently.

\[ T_1 : R(A), W(B), \text{Commit} \]
\[ T_2 : R(B), W(A), \text{Commit} \]
\[ T_3 : W(A), W(B), \text{Commit} \]

Consider the following three interleaved schedules.

\[ S_1 : R_{T1}(A), W_{T1}(B), \text{Commit}_{T1}, R_{T2}(B), W_{T2}(A), \text{Commit}_{T2}, W_{T3}(A), W_{T3}(B), \text{Commit}_{T3} \]
\[ S_2 : R_{T1}(A), R_{T2}(B), W_{T1}(B), W_{T2}(A), \text{Commit}_{T1}, \text{Commit}_{T2}, W_{T3}(A), W_{T3}(B), \text{Commit}_{T3} \]
\[ S_3 : W_{T3}(A), R_{T1}(A), W_{T1}(B), R_{T2}(B), W_{T2}(A), W_{T3}(B), \text{Commit}_{T1}, \text{Commit}_{T2}, \text{Commit}_{T3} \]
\[ S_4 : R_{T1}(A), W_{T3}(A), W_{T1}(B), \text{Commit}_{T1}, W_{T3}(B), \text{Commit}_{T3}, R_{T2}(B), W_{T2}(A), \text{Commit}_{T2} \]

Answer the following questions by clearly circling all correct options and only the correct options. Multiple options may be correct. If none of the given options are correct, clearly circle only the “None” option.

If multiple options are correct but you circle only a subset of the correct options, you will get points proportionally. But if you circle any wrong options, you will lose points proportionally! But the score for each question is lower bounded by zero.

1. (2pts) Which schedules are serial?

   S1   S2   S3   S4   None

2. (4pts) Which schedules are serializable?

   S1   S2   S3   S4   None

3. (4pts) Which schedules are recoverable?

   S1   S2   S3   S4   None
4. (3pts) Which schedules have no conflicts?

S1  S2  S3  S4  None

5. (3pts) Which schedules have a WW conflict?

S1  S2  S3  S4  None

6. (4pts) Suppose we use the Repeatable Read isolation level of SQL. Which schedules will lead to a deadlock?

S1  S2  S3  S4  None
Extra Credit Q. [5pts]. Consider the same sales database used for Q3. Use the same database statistics and machine setup as Q3.7. What is the rough I/O cost (in number of pages) of the following query? Ignore record-level metadata overheads. Explain your calculation briefly.

\( \pi_{CustomerID}(Sales) \)

Now suppose we use a column store format instead of row store. What is the rough I/O cost (in number of pages) of the above query now? Explain your calculation briefly. One again, ignore record-level metadata overheads. Assume no compression is used.