CSE 190D Spring 2017 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 one letter/A4-sized sheet of notes, formulae, 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. Please ensure that your writing is clear and legible!

<table>
<thead>
<tr>
<th>Points</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q 1</td>
<td>20</td>
</tr>
<tr>
<td>Q 2</td>
<td>10</td>
</tr>
<tr>
<td>Q 3</td>
<td>25</td>
</tr>
<tr>
<td>Q 4</td>
<td>21</td>
</tr>
<tr>
<td>Q 5</td>
<td>24</td>
</tr>
<tr>
<td>Total</td>
<td>100</td>
</tr>
<tr>
<td>Extra Credit</td>
<td>+5</td>
</tr>
</tbody>
</table>

Total 100
Mr. Jones lives 50 miles away from you. You both leave home at 5:00 and drive toward each other.

Mr. Jones travels at 35 mph, and you drive at 40 mph. At what time will you pass Mr. Jones on the road?

Given the traffic around here at 5:00, who knows?

I always catch these trick questions.
Q 1. [20pts] For the following questions, clearly circle True or False.

1. The hash join algorithm always has fewer page I/Os compared to the block nested loop join algorithm.
   True          False

2. Data shuffling among worker nodes is a key component of parallel query processing in MapReduce/Hadoop, Spark, and parallel DBMSs.
   True          False

3. Using the double buffering technique typically reduces the total number of passes needed for an external merge sort.
   True          False

4. All four SQL isolation levels guarantee that there will be no WW conflicts during a concurrent execution of transactions.
   True          False

5. It is typically possible to speed up the processing of the aggregation $\gamma_{\text{COUNT}(*)}(R)$ by using a B+ tree index on R.
   True          False

6. An avoiding-cascading-aborts (ACA) schedule is always guaranteed to be a recoverable schedule.
   True          False
7. Apart from schema information, the RDBMS catalog also stores statistics about both relation instances and indexes.
   
   True  False

8. A hash index is typically efficient for answering selection queries with \( \neq \) predicates (denoted <> in SQL).
   
   True  False

9. Spark’s API mostly subsumes the MapReduce programming abstraction.
   
   True  False

10. The clock algorithm is an approximation of the MRU buffer replacement policy.
    
   True  False
Q 2. [10pts] Hash Join with non-uniform partitioning. (This question is a small tweak on a quiz question!) We are joining two tables R and S, which have $4BN_R$ and $12BN_S$ pages respectively, using a hash join. We are given that $4BN_R \gg 12BN_S$ and that the number of available buffer pages is $4B + 1$. The buffer pool is initially empty. We are also given that $2FN_S = 4B - 1$, where $F$ is the hash table fudge factor.

The distribution of the join attribute values in R and S are such that after the first hash partitioning phase, we get exactly $4B$ partitions each of R and S. Each partition of R is $N_R$ pages long, but the partitions of S have multiple lengths. Suppose that S gets partitioned as follows: $B$ partitions have length $N_S$ pages each, $2B$ partitions of length $3N_S$ pages each, and $B$ partitions of length $5N_S$ pages each.

What is the I/O cost of the above join using the regular hash join algorithm discussed in class? Exclude the cost of writing the output. Assume that perfect uniform splitting occurs during the recursive repartitioning and that we do not need to recurse more than once. Briefly explain and show all of your calculations clearly.

(Hint: The answer is of the following form: $xBN_R + yBN_S$, where $x \in \{10, 12, 14, 16, 18, 20\}$ and $y \in \{50, 54, 58, 62, 66, 70\}$.)
Q 3. [25pts] Query processing and optimization. We are given a relational database schema with the following three relations, wherein $A$ and $B$ are discrete (string) attributes and $C, D,$ and $E$ are numeric attributes.

$$R(A, B, C), S(A, B), \text{ and } T(B, D, E)$$

Q 3.1 [10pts] For each of the following questions, clearly circle either Yes or No for whether the given queries $Q_1$ and $Q_2$ are equivalent.

1. $Q_1 : \sigma_{A="a" \land B="b"}(R \bowtie S) \bowtie T$ and $Q_2 : (\sigma_{A="a" \land B="b"}(R \bowtie S)) \bowtie T$

   Yes \hspace{1cm} No

2. $Q_1 : \pi_E(\sigma_{D=1}(R \bowtie T))$ and $Q_2 : \pi_E(T) \bowtie \pi_{B,E}(\sigma_{D=1}(R \bowtie T))$

   Yes \hspace{1cm} No

3. $Q_1 : \pi_B(R \bowtie S)$ and $Q_2 : (\pi_B(R)) \bowtie (\pi_B(S))$

   Yes \hspace{1cm} No

4. $Q_1 : \pi_{B,C,E}(R \bowtie (\sigma_{E=1}(T)))$ and $Q_2 : \sigma_{E=1}(\pi_{B,C,E}(R \bowtie T))$

   Yes \hspace{1cm} No

5. $Q_1 : R \bowtie S \bowtie T$ and $Q_2 : (R \bowtie S) \bowtie (S \bowtie T) \bowtie (T \bowtie R)$

   Yes \hspace{1cm} No
Q 3.2 [9pts] We are given an instance of the relation $T$ whose heap file has a size of 10 million pages. We are also given a B+ tree index on $T$ with the index key $(B, D)$ that follows the alternative of storing RID lists in the leaf pages. The index has 5 million leaf pages. Assume that each $B$ value is 8 B in size and that there are a million unique values of $B$ in the given instance of $T$. A page is 8 KB in size.

Describe an efficient physical query plan with I/O cost less than 9 million pages for processing the following SQL query, given 4 GB of buffer memory and an initially empty buffer pool. Mention if each physical operator is pipelined or materialized and compute the total I/O cost of your plan (in number of pages) rounded to the nearest million.

$$\text{SELECT } B, \text{ AVG}(D) \text{ FROM } T \text{ GROUP BY } B$$
Q 3.3 [6pts] Name three indexes on \( R \) (including at least one hash index) that match the predicate in the following SQL query and briefly explain why each index matches.

```
SELECT * FROM R
WHERE (B = "b" OR A <> "a") AND NOT (C <= 20 OR A <> "a")
```
Q 4. **Transaction management and concurrency control**. We are given a database with three distinct data objects A, B, and C. We are also given the following three transactions that arrive concurrently.

\[ T1 : R(A), W(A), R(B), W(B), Commit \]
\[ T2 : R(B), W(B), R(C), W(C), Commit \]
\[ T3 : R(C), W(C), R(A), W(A), Commit \]

**Q 4.1 [2pts]** Give a clear example of a *serial* schedule of the three transactions.

**Q 4.2 [5pts]** Give a clear example of a *serializable* schedule of the three transactions that is not serial but is equivalent to the serial schedule in your above answer.
Q 4.3 [14pts] Consider the following interleaved schedule. For each of the questions that follow, clearly circle yes or no.

\[ R_{T_1}(A), R_{T_2}(B), R_{T_3}(C), W_{T_1}(A), W_{T_2}(B), W_{T_3}(C), \\
R_{T_1}(B), W_{T_1}(B), Commit_{T_1}, R_{T_2}(C), W_{T_2}(C), Commit_{T_2}, R_{T_3}(A), W_{T_3}(A), Commit_{T_3} \]

1. Is the schedule serializable?
   Yes
   No

2. Is the schedule recoverable?
   Yes
   No

3. Does the schedule have a WW conflict between any pair of transactions?
   Yes
   No

4. Does the schedule have a WR conflict between any pair of transactions?
   Yes
   No

5. Does the schedule have a RW(R) conflict between any pair of transactions?
   Yes
   No

6. Suppose we use the READ UNCOMMITTED isolation level of SQL. Will it lead to a deadlock with the given schedule?
   Yes
   No

7. Suppose we use the SERIALIZABLE isolation level of SQL. Will it lead to a deadlock with the given schedule?
   Yes
   No
Q 5. [24pts] For the following questions, **clearly circle** the right answer (only one option is correct).

1. Which of the following symbols does *not* represent a relational operator from the extended relational algebra?
   
   (a) $\gamma$  
   (b) $\cup$  
   (c) $\mu$  
   (d) $\pi$  
   (e) $\times$

2. Which of the following relational operators do not preserve the schema of (at least one of) their inputs?
   
   (a) Set union  
   (b) Set intersection  
   (c) Set difference  
   (d) Select  
   (e) Project

3. Which of the following relational operators can be processed using a regular (un-modified) hash join implementation?
   
   (a) Set union  
   (b) Set intersection  
   (c) Set difference  
   (d) Select  
   (e) Project

4. Which of the following SQL aggregates require a shuffle among worker nodes in a parallel DBMS when the **GROUP BY** list is empty?
   
   (a) SUM  
   (b) AVG  
   (c) VARIANCE  
   (d) MEDIAN  
   (e) MAX

5. Which file organization is typically the most efficient for inserting new records?
   
   (a) Heap file  
   (b) Sorted file  
   (c) B+ tree index with AltRecord

6. We are given this join query: $R \bowtie S \bowtie T \bowtie U$. Recall that some query optimizers only consider left-deep join trees for join order enumeration. How many different left-deep join trees exist for this query? (**Hint:** Swapping the left and right input of a $\bowtie$ in a given tree yields a different tree.)
   
   (a) 4  
   (b) 6  
   (c) 15  
   (d) 16  
   (e) 24
7. Which is the dominant parallelism paradigm that is used in parallel DBMSs, MapReduce/Hadoop, and Spark?

(a) Shared-nothing  
(b) Shared-memory  
(c) Shared-Disk

8. Given the following bags $A = \{a, b, c, a, a, x, y, x, z\}$ and $B = \{e, a, b, d, x, y, a, y\}$, what is the result of the bag intersection of $A$ and $B$ (INTERSECT ALL in SQL)?

(a) $\{a, b, x, y\}$  
(b) $\{b, a, y, x, a\}$  
(c) $\{x, b, a, x, y\}$  
(d) $\{a, b, c, x, y, z\}$

9. In a hard disk, which of the following components of the data access time accounts for the delay caused by the radial movement of the arm?

(a) Seek time  
(b) Rotational delay  
(c) Transfer time

10. Given 1 GB of buffer memory, what is roughly the largest file size (to the nearest order of magnitude) that can technically be sorted with only 2 passes?

(a) 10 GB  
(b) 1 TB  
(c) 100 TB  
(d) 10 PB  
(e) 100 PB

11. Suppose we are given two union-compatible relations $R$ and $S$ with sizes $N_R$ and $N_S$ ($\ll N_R$) pages respectively. Suppose we have $B = 2 + F \times N_S / 2$ pages of buffer memory ($F$ is the hash table fudge factor) and the buffer pool is initially empty. What is the minimum possible I/O cost of a UNION ALL of $R$ and $S$, excluding output write cost?

(a) $6N_R + 6N_S$  
(b) $4N_R + 4N_S$  
(c) $2N_R + 2N_S$  
(d) $N_R + 2N_S$  
(e) $N_R + N_S$

12. In a B+ tree index, which nodes are allowed to have duplicates of the index key?

(a) Leaf nodes  
(b) Root node  
(c) Non-root internal nodes
Extra Credit Q. [5pts]. Suppose you are working for Netflix and using a Hadoop cluster. You are given a dump of a large Ratings table with the same schema from class: Ratings (RID, MovieID, UserID, Stars, RateDate). The table is stored as a distributed CSV file on HDFS that is hash partitioned on RID. You are tasked with obtaining the number of five-star ratings for each movie in the given file. Unfortunately, your Hadoop cluster only has regular MapReduce (no Hive, Pig, Spark, or any SQL engine).

How will you implement this task in MapReduce? A proper explanation of the Map and Reduce phases are enough; no need to write code in Java, Python, etc.

Tired of writing low-level MapReduce programs, you decide to convince your boss that Netflix really needs to use Hive or SparkSQL. How can you use the above task to achieve your goal?