UCLA
Computer Science Department
Fall 2014

Instructor: J. Cho

Student Name and ID: _____

CS143 Final: Closed Book, 120 minutes

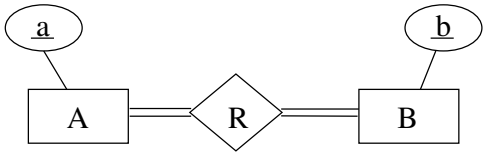
(IMPORTANT PLEASE READ **):**

- There are 5 problems on 9 pages to be completed in 120 minutes. *You should look through the entire exam before getting started, in order to plan your strategy.*
- Write down *only your answer* in each answer box. No other writing in answer boxes, please.
- To get partial credits, you may want to briefly write down explanations or justifications of your answer *below* each answer box.
- *Simplicity and clarity of your solutions will count.* You may get as few as 0 point for a problem if your solution is far more complicated than necessary, or if we cannot understand your solution.
- If you need to make any assumption to solve a question, please write down your assumptions.
- You may use two double-sided cheat sheets during exam. You are also allowed to use a calculator.
- Please, write your answers neatly. Attach extra pages as needed. Write your name and ID on the extra pages.

Problem	**Score**	
1	10	
2	25	
3	30	
4	20	
5	40	
Total	125	

Problem 1: 10 points

Assume that we convert the following E/R diagram into tables.



Assume that we have n entities in entity set A and m entities in entity set B. What is the minimum and maximum number of tuples that we will have in table R? Assume set semantics for the converted tables.

(*Note: You may get different answers depending on whether $m \leq n$ or $m > n$. If you do, please give the correct answer for each case.*)

ANSWER:
`min: max(m,n), max: m x n`

Problem 2: 25 points

Consider a table recording the number of hours an employee worked each day:

Work(employee,dept,date,hours)

1. (10 points) For now make the following two assumptions and no others:

   (A1) Functional dependency "employee,date → hours" holds.
   (A2) Functional dependency "employee → dept" holds.

   Based on assumptions (A1) and (A2) above, and no others, is Work in Boyce-Codd Normal Form (BCNF)? (4 points)

   ANSWER:
   `No`

   If you circled YES: In the space below, state the smallest possible change to the assumptions about the dependencies so that Work is no longer in BCNF. Your new assumptions should "make sense" in the real world, meaning it could be reasonable to require the data in Work to satisfy these assumptions.

   If you circled NO: In the space below, decompose Work into relations that are in BCNF, following the decomposition algorithm covered in class. Show the result of your decomposition only, not the process.

   ANSWER:
   `employee → dept violates BCNF. W1(employee,dept), W2(employee,date,hours)`

2. Now drop the two assumptions (A1) and (A2). Instead make the following assumption and no others:

   (A3) Multivalued dependency employee $\twoheadrightarrow$ dept holds.

   Under this assumption only, answer the following two questions.

   (a) (5 points) Is Work in Fourth Normal Form (4NF)?

   ANSWER:
   `NO. employee is not a key`

   (b) (10 points) Regardless of whether you circled YES or NO for part (a), is the instance of table Work shown below legal under assumption (A3)? (5 points)

employee	dept	date	hours
John	Accounting	6/1/06	8
John	Marketing	6/1/06	8
Susan	Accounting	6/1/06	8
Susan	HR	6/2/06	10

   ANSWER:
   `No`

   If you circled YES: In the space below, list the tuples that will make the table illegal under the assumption if deleted. List the minimum number of tuples to make the table illegal.

   If you circled NO: In the space below, list the the tuples that will make the table legal if inserted. List the minimum number of tuples to make the table legal.

   ANSWER:
   `(Susan, Accounting, 6/2/06, 10), (Susan, HR, 6/1/06, 8)`

Problem 3: 30 points

For this problem, assume the set semantic. We use the notation $|R|$ to denote the number of tuples in the relation R.

1. (10 points) Consider the relation $R(A, B, C, D, E, F)$ with functional dependencies:

   $$AEF \to C, \quad BF \to C, \quad EF \to D, \quad ACDE \to F$$

   List all keys for R.

   ANSWER:
   `ABEF and ABCDE. Since none of A, B, E are functionally dependent on any other attributes, they have to be in any key.`

2. (10 points) Consider the relation $R(A, B, C)$, on which the multivalued dependency $A \twoheadrightarrow B$ holds. If $|R| = 10$, $|\pi_A(R)| = 1$ and $|\pi_C(R)| = 5$, what are the possible value(s) of $|\pi_B(R)|$?

   ANSWER:
   `2`

3. (10 points) Consider the relation $R(A, B, C, D)$, on which the functional dependencies $B \to C$ and $C \to D$ hold. If $|R| = 10$, $|\pi_B(R)| = 6$, and $|\pi_D(R)| = 4$, what are the possible value(s) of $|\pi_C(R)|$?

   ANSWER:
   `4,5,6`

Problem 4: 20 points

1. (10 points) When all SQL transactions run at the isolation level SERIALIZABLE, the database system guarantees ACID properties. Briefly explain what ACID means.

ANSWER:
Atomicity: all or nothing. Consistency: If the database was in consistent state, it remains in consistent state. Isolation: The end result is the same as when transactions are run in isolation. Durability: Results from committed transactions are never lost.

2. (10 points) Consider the relation $R(A)$ where A is of integer type. The following three transactions are being executed:

   T_1:
   ```
   SELECT SUM(A) FROM R;
   COMMIT;
   ```

   T_2:
   ```
   INSERT INTO R VALUES(10);
   INSERT INTO R VALUES(20);
   INSERT INTO R VALUES(30);
   COMMIT;
   ```

   T_3:
   ```
   UPDATE R SET A = A + 40 WHERE A=20;
   UPDATE R SET A = A + 30 WHERE A=10;
   COMMIT;
   ```

   Before any of these transactions, the sum of the integers in R was 1000, and none of these integers were 10, 20, or 30. Assume that T_1 runs under isolation level READ UNCOMMITTED, T_2 under READ COMMITTED, and T_3 under SERIALIZABLE. List all possible values that can be returned by T_1.

   Possible values from T_1:

   ANSWER:
   1000, 1010, 1030, 1060, 1100, 1130.

   Since T_3 is serializable, it does not see any phantom. So T_3 sees either all insertions of T_2 or none of them. Therefore, possible ‘‘schedules’’ are T_2T_3 or T_3T_2. If it is T_2T_3, SUM(A) changes from 1000 to 1010, 1030, 1060, 1100, and 1130. If it is T_3T_2, SUM(A) changes from 1000 to 1010, 1030 and 1060.

Problem 5: 40 points

For all questions, assume the relation $R(\underline{A}, B, C)$ and the relation $S(\underline{C}, \underline{D}, E)$.

1. (10 points) Suppose relation R occupies n blocks and relation S occupies m blocks. We assume that $n \leq m$. Initially, both relations are on disk and we want to perform a block-nested-loop join. If we want to read each block of R and S only once during the join, what is the minimum number of main-memory buffers (each the size of a disk block) required, including the buffer(s) for output?

   ANSWER:
   $n + 2$. n to load the entire R, 1 for reading S, 1 for final output

2. (10 points) Assume that the table R has 100,000 tuples stored in 10,000 blocks. We construct a dense unclustered B+tree on the attribute A of the table R. Assume that the B+tree has 1,000 nodes at the leaf level, 10 nodes at the non-leaf level directly above leaf and 1 node at the root level. Each leaf node of the B+tree contains 100 key values. The values of $R.A$ are uniformly distributed between 1 and 100,000. Now consider the following query:

   ```
   SELECT * from R WHERE A > 99000
   ```

   We use the B+tree index to identify the tuples that satisfy the condition $A > 99000$ and retrieve the tuples. How many disk IOs do we need to process this query? Assume that we have 10 main memory buffers to process this query.

   ANSWER:
   1,012 block. Traversing the B+ tree requires 3 node accesses (1 root, 1 non-leaf, and 1 leaf). 1% of the keys will satisfy the condition, so the matching keys will be contained in 10 leaf nodes at the right end of B+tree (1% of the 1,000 leaf nodes). Therefore, in order to retrieve these keys in the leaf nodes, we will have to follow 9 more succeeding leaf nodes.

   For each matching tuple (a total of 1,000), we then access one block to fetch the actual record, so the total disk IO is $3+9+1,000 = 1,012$ blocks.

   Is using the B+tree better than scanning the entire table of R to process the query? Ignore potential performance difference due to random IO vs sequential IO.

   ANSWER:
   Yes. Scanning R requires 10,000 disk IOs.

3. Now, assume that we are sorting the table R using the merge-sort algorithm. Again, R is stored in 10,000 blocks, and we have 10 main memory buffers. We just finished the first sorting stage of the merge-sort algorithm and generated multiple sorted runs.

   (a) (5 points) How many sorted runs do we have now?

   ANSWER:
   `1,000 sorted runs`

   (b) (5 points) We started the second "merging stage," where we merge sorted runs. So far, you have read a total of 5 disk blocks during this stage. What is the minimum and the maximum number of disk blocks may have been written to the disk in this stage? Count the disk writes during the merging stage only. Assume that we immediately flush an output buffer when it is full. Again, each block has 10 tuples and we have 10 main memory buffer blocks.

   ANSWER:
   `min: 0, max: 0. When we have 10 main memory buffers, we can merge 9 sorted runs in each stage. In order to start ‘‘merging,’’ and producing an output, we need to read at least one block from each sorted run. Since we haven’t read 9 blocks yet, we cannot start producing output.`

   (c) (10 points) What if you have read a total of 15 disk blocks during the merging stage now (not 5 blocks as we previously assumed)? What is the minimum and the maximum number of disk blocks that you may have written to the disk so far?

   ANSWER:
   `min: 6, max: 14. In one extreme case, all main memory buffers will be full except the output buffer (because when the output buffer is full, it is immediately flushed). Therefore we can hold slightly more than 9 blocks in main memory in this case, which means that we must have written 15-9=6 blocks. In the opposite extreme case, each buffer will have only one tuple, so we hold 10 tuples (= 1 block) in main memory buffers. So we have written 14 blocks to the disk.`