UCLA
Computer Science Department
Fall 2019

Instructor: J. Cho

CS143 Final: Closed Book, 120 minutes

Student Name: _____

Student ID: _____

(IMPORTANT PLEASE READ **):**

- There are 5 problems on 9 pages. *You should look through the entire exam before getting started, in order to plan your strategy.*
- *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.
- To get partial credits, you may want to write down how you arrived at your answer step by step.
- 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	20	
2	20	
3	20	
4	20	
5	20	
Total	100	

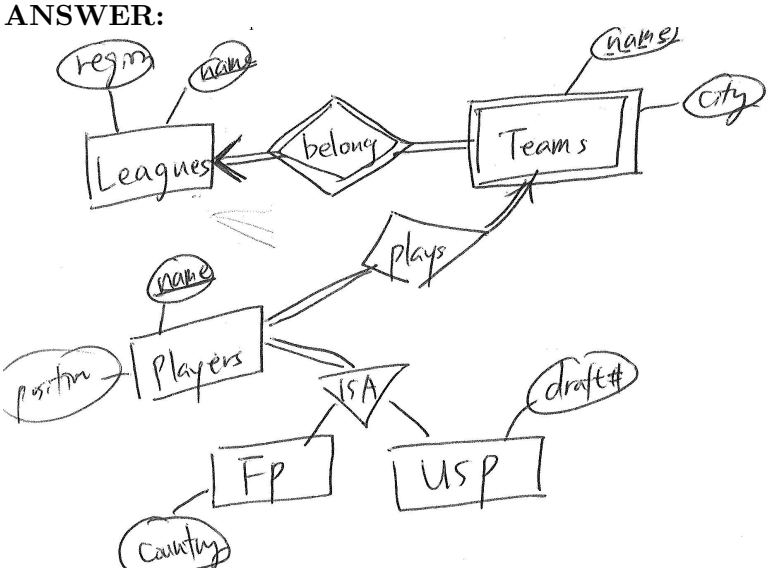
Problem 1 (ER model): 20 points

Draw an E/R diagram for the following situation.

- There are five entity sets *Leagues*, *Teams*, *Players*, *ForeignPlayers*, and *USPlayers*.
- Leagues have the attributes *name* and *region*. League names are unique.
- Teams have the attributes *name* and *city*. The name of a team is unique within a league, but teams in different leagues may have the same name.
- Every team belongs to one league, and every league has at least one team.
- Players have the attributes *name* and *position*. A player's name is unique.
- Every player plays for one team, and every team has at least one player.
- A player is either a foreign player or a US player.
- A foreign player has the additional attribute of the country of his/her origin.
- A US player has the additional attribute of the draft number.

1. Draw the E/R diagram that captures the above information. Pay special attention to 1) the cardinality and participation of relationship sets 2) whether an entity set is weak or strong and 3) the key or the discriminator of an entity set. (Remember that we underline key attributes and dash-underline discriminators.) You can use either the standard or the general notation to represent cardinalities. Use reasonable names for relationship sets and attributes. (11 points)

ANSWER:



2. Translate the "Leagues" entity set into a table. Underline the key of the table. If there is a name conflict for attributes, append the entity set name before the attributes. (3 points)

   ANSWER:
   `Leagues(name, region)`

3. Translate the "Teams" entity set into a table. Underline the key of the table. If there is a name conflict for attributes, append the entity set name before the attributes. (3 points)

   ANSWER:
   `Teams(leagues.name, teams.name, city)`

4. Translate the relationship set between "Players" and "Teams" into a table. Underline the key of the table. If there is a name conflict for attributes, append the entity set name before the attributes. (3 points)

   ANSWER:
   `Play(leagues.name, teams.name, players.name)`

Problem 2 (Database Design): 20 points

1. Suppose we want to build a database on a mother, her children and the cars that the mother owns.

   `R(mother-name, addr, child-name, birth-date, vin)`

   Here, mother-name is a mother's name, addr is the address of the mother, child-name is the name of her child, birth-date is the birth date of the child, and vin is the vehicle identification number of the mother's car. We make the following assumptions for this database:

   - A person's name is unique.
   - A mother has only one associated address.
   - A child has only one birth date.
   - A mother may have more than one child.
   - A mother may own more than one car.
   - A child has only one mother.
   - A car's vehicle identification number is unique.
   - A car is owned by only one person.
   - A car is not associated with a particular child. Any children may use any car that their mother owns.

   (a) Find functional dependencies and *one* multivalued dependency given the above constraints and the schema. You do not need to list all functional dependencies. It is enough to list the set of functional dependencies that imply all functional dependencies satisfied by the relation. Also, the MVD that you list should not be implied by a functional dependency. (8 points)

   ANSWER:
   FD:
   `mother-name` $\rightarrow$ `addr`
   `child-name` $\rightarrow$ `mother-name, birth-date`
   `vin` $\rightarrow$ `mother-name`
   MVD:
   `mother-name` $\twoheadrightarrow$ `vin OR`
   `mother-name` $\twoheadrightarrow$ `child-name,birth-date OR`
   `mother-name,addr` $\twoheadrightarrow$ `vin OR`
   `mother-name,addr` $\twoheadrightarrow$ `child-name,birth-date`

   (b) Is the relation R in 4NF? If yes, explain why the functional dependencies and multivalued dependencies you identified do not violate the 4NF definition. If no, normalize it into 4NF. Underline key attributes in the normalized table. (6 points)

ANSWER:
No. R1(mother-name, addr), R2(child-name, birth-date, mother-name), R3(vin, mother-name)

2. Consider a relation $R(A, B, C, D, E)$ and a set of functional dependencies:
   $A \to B, \quad CD \to E, \quad E \to D.$

   Is the table in BCNF? If yes, explain why the functional dependencies do not violate the BCNF definition. If no, normalize it into BCNF. Underline the key in the normalized tables. (6 points)

   ANSWER:
   No. R1(A,B), R2(A,C,D), R3(C,E), R4(D,E)

Problem 3 (Dependencies): 20 points

For this problem, assume the set semantic. That is, no duplicates exist in relations and we eliminate all duplicates in the output of any relational operator. We use the notation $|S|$ to denote the number of tuples in the relation S.

1. 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`

2. Consider the relation $R(A, B, C, D)$, on which the functional dependencies $B \rightarrow C$ and $C \rightarrow D$ 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 (Index): 20 points

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

1. 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 nested-loop join in which we read each block of R and S only once (and do no other disk I/O other than writing the result). What is the minimum number of main-memory buffers (each the size of a disk block) required? (10 points)

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

2. We constructed a *dense nonclustering* B+tree on the attribute A of the table R. Assume that the B+tree has 3,000 nodes at the leaf level, 100 nodes at the non-leaf level directly above leaf and 1 node at the root level. Assume that the table R has 100,000 tuples stored in 10,000 blocks. The values of $R.A$ are uniformly distributed between 1 and 100,000. Now consider the following query:

   ```
   SELECT * from R WHERE A > 90000
   ```

   We use the B+tree index to identify the tuples that satisfy the condition $A > 90000$ and retrieve the tuples. How many disk IOs do we need to process this query? Ignore the IO cost for writing the final result. If necessary, you may assume that we have 3 main memory buffers to process this query. Is using the B+tree any better than scanning R linearly to process the query? Briefly explain how you obtained your answer. (10 points)

   ANSWER:
   Traversing the B+ tree requires 3 node accesses (1 root, 1 non-leaf, and 1 leaf). Roughly 10% of the keys will satisfy the condition, so the matching keys will be contained in 300 leaf nodes at the right end of B+treee (10% of the 3000 leaf nodes). Therefore, in order to retreive these keys in the leaf nodes, we will have to follow 299 more succeeding leaf nodes.

   For each matching tuple (a total of 10,000), we then access one block to fetch the actual record, so the total disk IO is $3 + 299 + 10000 = 10,302$ blocks. A linear search will only require 10,000 block reads.

Problem 5 (Transactions): 20 points

1. Consider the relation `Employee(name, salary)` where `name` is the key. The following three transactions are being executed:

   T_1:
   ```
   SELECT SUM(salary) FROM Employee;
   COMMIT;
   ```

   T_2:
   ```
   UPDATE Employee SET salary = salary + 200;
   UPDATE Employee SET salary = salary + 1000 WHERE name = 'Tony';
   COMMIT;
   ```

   T_3:
   ```
   UPDATE Employee SET salary = salary + 100 WHERE name = 'James';
   UPDATE Employee SET salary = salary + 200 WHERE name = 'Tony';
   COMMIT;
   ```

   The table `Employee` originally has two tuples, ('Tony', 1000) and ('James', 1000).

   (a) (4 points) Assume that all three transactions run under the isolation level SERIALIZABLE. List all possible values that can be returned by T_1. Briefly explain your answer.

   ANSWER:
   2000, 2300, 3400, 3700.
   When all transactions run under SERIALIZABLE, any possible schedule is conflict equivalent to a serial schedule. Possible schedules for T_1, T_2 and T_3 are: $T_1T_2T_3$, $T_1T_3T_2$, $T_2T_1T_3$, $T_2T_3T_1$, $T_3T_1T_2$, and $T_3T_2T_1$. The outputs from T_1 are 2000, 2000, 3400, 3700, 2300, 3700, respectively.

   (b) (6 points) Assume that T_1 runs under the isolation level READ UNCOMMITTED and T_2 under REPEATABLE READ and T_3 under SERIALIZABLE. List all possible values that can be returned by T_1. Briefly explain your answer.

   ANSWER:
   2000, 2100, 2200, 2300, 2400, 2500, 2700, 3200, 3400, 3500, 3600, 3700.
   Only T_2 and T_3 are updating values, so let us focus on these two transactions first. Under REPEATABLE READ, the only exception to ACID is phantom, but because T_2 and T_3 do not insert any tuple, we do not need to worry about ACID exceptions for the two. So the possible schedules for T_2 and T_3 are equivalent to T_2T_3 or T_3T_2. Regarding T_1, since it is READ UNCOMMITTED, the two reads in T_1 (one read on 'Tony' and the other read on 'James') may appear anywhere between the actions of T_2 and T_3 in any order.

Now let us consider the schedule T_2T_3. Under this schedule, Tony's salary changes from 1000 to 1200 to 2200 to 2400. James' salary changes from 1000 to 1200 to 1300. Since T_1 may read any of these salary values, possible outputs from T_1 are 2000, 2200, 2300, 2400, 2500, 3200, 3400, 3500, 3600 and 3700.

For the schedule T_3T_2, Tony's salary changes from 1000 to 1200 to 1400 to 2400. James' salary changes from 1000 to 1100 to 1300. Since T_1 may read any of these salary values, possible outputs from T_1 are 2000, 2100, 2200, 2300, 2400, 2500, 2700, 3400, 3500, 3700.

2. Consider the following schedule:

$$r_1(X)\, w_1(X)\, r_2(X)\, c_2\, r_3(Y)\, w_3(Y)\, c_3\, r_1(Y)\, w_1(Y)\, c_1$$

Circle YES or NO for the questions below and provide brief explanation. Be careful with your answers for this problem. For every correct answer, you will get 2 points, but for every incorrect answer, you will get 1 point deducted.

(a) Is the schedule serial? YES / NO

ANSWER:
No

(b) Is the schedule conflict serializable? YES / NO

ANSWER:
Yes. The precedence graph is $T_2 \leftarrow T_1 \leftarrow T_3$ and has no cycle.

(c) Is the schedule recoverable? YES / NO

ANSWER:
No. c_2 appears before c_1 even though T_2 reads a value written by T_1.

(d) Should the above schedule be allowed when all three transactions run under the isolation level SERIALIZABLE? YES / NO

ANSWER:
No. T_2 does dirty read from T_1 and commits before T_1.

Depending on your answer, provide an answer to one of the following questions. (2 points)

- If your answer is yes, write down the serial schedule that is equivalent to the above schedule.

- If your answer is no, what is the the minimum relaxation necessary to the isolation level of each transaction to allow this schedule? The isolation levels of the three transactions may or may not be the same after the relaxation.

  T_1 isolation level: _____

  T_2 isolation level: _____

T_3 isolation level: _____

ANSWER:
No. Change the isolation level of T_2 to READ UNCOMMITTED.