UCLA
Computer Science Department
Winter 2021

Instructor: J. Cho

CS143 Final: Closed Book, 2.5 hours

Student Name: ____

Student ID: ____

(IMPORTANT PLEASE READ **):**

- The exam is *closed book* and *closed notes*. You may use *two double-sided cheat-sheets (4 pages in total)*. You can use a calculator.
- *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.
- Please, write your answers neatly.

Problem	**Score**	
1	10	
2	20	
3	10	
4	15	
5	10	
6	25	
7	10	
8	10	
Total	110	

Problem 1 (Normalization): 10 points

Consider the relation $R(A, B, C, D, E, F)$ and the following function dependencies:

$AC \rightarrow B$
$F \rightarrow C$
$BD \rightarrow F$

Is this relation in BCNF? If yes, please explain why it is. If not, decompose R into a set of tables that are in BCNF using the algorithm learned in the class. Explain your decomposition process and the final set of tables that R is decomposed into.

Problem 2 (Functional Dependency): 20 points

1. A set of attributes X is said to be closed if $X+ = X$; that is, the closure of X adds nothing to X.

   Consider the relation $R(A, B, C, D, E, F, G, H)$ with functional dependencies $A \rightarrow B$ and $B \rightarrow CD$. Among the 256 subsets of $\{A, B, C, D, E, F, G, H\}$, how many of them are closed? For example, the subset $\{G, H\}$ is a closed set, while the subset $\{B\}$ is not. Explain your answer. (10 points)

2. Relation $R(A, B, C)$ satisfies a set of functional dependencies F, where F is a subset of the following set G:

$$G = \{A \to B, \quad B \to A, \quad A \to C, \quad C \to A, \quad B \to C, \quad C \to B\}.$$

Among the 64 possibilities of F (note that G has 64 subsets), how many times does F logically imply (either directly or indirectly) $A \to B$? That is, how many subsets of G logically imply $A \to B$? (10 points)

Problem 3 (Referential Integrity): 10 points

Suppose we have the following table declarations:

```
CREATE TABLE A(w INT PRIMARY KEY);
CREATE TABLE B(x INT PRIMARY KEY REFERENCES A(w) ON DELETE SET NULL);
CREATE TABLE C(y INT REFERENCES A(w));
CREATE TABLE D(z1 INT REFERENCES B(x) ON DELETE SET NULL,
z2 INT REFERENCES A(w) ON UPDATE CASCADE);
```

Consider the following scripts:

I. `DELETE FROM C; DELETE FROM B; DELETE FROM A; DELETE FROM D;`
II. `DELETE FROM C; DELETE FROM D; DELETE FROM A; DELETE FROM B;`
III. `DELETE FROM B; DELETE FROM C; DELETE FROM D; DELETE FROM A;`

Which of the above scripts are guaranteed to empty all four tables, without error? Circle the script number(s) that will empty all four tables. Briefly explain your answer. An answer without any explanation may not get any point.

Problem 4 (Constraints): 15 points

Assume the following database schema:

`Executive(id, name, div_in_charge, salary)`
`Employee(id, name, division, salary)`

The `div_in_charge` attribute of the `Executive` table indicates the division that the executive is in charge. A division may have multiple executives. An executive is also an emplyee and belongs to the same division that they are in charge, so they appear in both `Executive` and `Employee` tables with the same `id`. Every division name is unique. No attribute can have the NULL value.

Now we want to enforce the following constraint on this database:

An executive's salary cannot be higher than 10 times the lowest salary of the employees in the division that they are in charge.

1. What database modifications can potentially violate the constraint? In the following table write YES for the operations that can potentially violate the constraint. For example, if the deletion of a tuple from the `Executive` table may violate the constraint, write YES in the cell corresponding to the `Executive` row and the `DELETE` column. (5 points)

	INSERT	DELETE	UPDATE
`Executive`			
`Employee`			

2. Assume the SQL92 CHECK constraint that supports subqueries. Is it possible to write one or more CHECK constraint(s) and reject any modifications that will violate the constraint? If yes, write the CHECK constraint(s) portion of the `CREATE TABLE` statement(s) for the relevant table(s). For example, if your answer is adding `CHECK(salary > 10000)` to the `Executive` table, your answer should look like

```
CREATE TABLE Executive(
    ...
    CHECK(salary > 100000),
    ...
);
```

If a SQL92 CHECK constraint cannot be used for this purpose, explain why. (10 points)

Problem 5 (NoSQL): 10 points

Assume that information on books and authors is stored in the `authors` collection of MongoDB database as documents of the following format:

```
{
    authorName,
    birthYear,
    books: [{
        title,
        price,
        publisher
    }]
}
```

Your colleague wrote the following MongoDB query to obtain the information that your boss requested:

```
db.authors.aggregate([
    {$match: {birthYear: {$gt: 1950}}},
    {$unwind: "$books"},
    {$group: {_id: "$books.publisher" a1: {$sum: "$books.price" }}},
    {$match: {a1: {$gt: 10000} }}
]);
```

Assume that the same book-author information is also stored in an RDBMS table with the following schema:

Books(authorName, birthYear, title, price, publisher)

In the space provided below, write a SQL query that returns "equivalent" information to the previous MongoDB query. Do not worry about column names in your SQL query results. As long as your query returns the same "values" as the MongoDB query, you will get full credit.

Problem 6 (Disk and Index): 25 points

Assume a disk of the following characteristics for this problem:

- 1TB total disk capacity
- 10 surfaces
- 1000 tracks per surface
- 6000 RPM
- 10ms average seek time
- 4KB block size (1KB is 1024 bytes)

1. Suppose that we are reading a file F that occupies exactly one entire track, and we want to estimate how long it takes to read the whole file sequentially. Are there any key parameters that are missing for doing this? Select one from the following choices:

   (a) the diameter of the disk surface — d

   (b) the average rotational latency — r

   (c) the transfer rate of the disk — t

   (d) none

   Based on your choice above, how long does it take to read the file on average? You may use the symbol (after each choice) to represent the parameter that you think is useful in the estimation result. Assume that the disk head may not initially be on top of the track where the file is located. (5 points)

2. You need to store the following table in the provided disk:

   `Student(id LONG, name CHAR(32))`

   LONGs are 8 bytes and CHAR(32)'s are 32 bytes. Assume that tuples are not spanned across blocks. The table has 1 million tuples and is sequenced by the `id` attribute. What is the minimum number of blocks needed to store this table? (5 points)

3. We decide to create a B+tree index on the attribute `id` of the `Student` table. The pointers in the B+tree are of size 8 bytes. One node of the B+tree corresponds to one disk block. What is the minimum number of nodes that are needed for the B+tree? (10 points)

4. Using the constructed B+tree, how many disk IOs are needed to execute the following query?

   ```
   SELECT * FROM Student WHERE id = 334234
   ```

   Note that the student id is a key of the table. Assume that the root node of the B+tree is always cached in main memory and does not require a disk IO to read it. No other B+Tree nodes or the `Student` table blocks are cached in main memory. (5 points)

Problem 7 (B+Tree): 10 points

Assume a B+tree of $n = 3$ (i.e. a node has space for 3 pointers and 2 keys). Give an example of B+tree of height 3 that could be reorganized into a tree of height 2 with exactly the same set of keys at the leaf level.

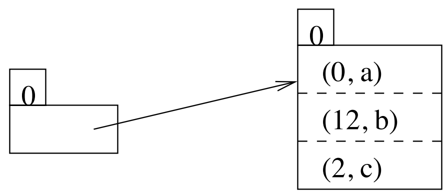
Your example should consist of two trees: (1) a tree of height 3, and (2) an equivalent tree of height 2. Your examples should show all the keys and pointers in the leaf and non-leaf nodes of the tree. Use letters A, B, C, ... to represent the keys. (Assume A $\neq$ B $\neq$ C ...).

Height-3 Tree:

Height-2 Tree:

Problem 8 (Extendable Hash): 10 points

Consider a simple extendable hash index below. Assume that we use the hash function $h(k) = (k \bmod 16)$ that produces 4 bit hash values. For example, $h(20) = (20 \bmod 16) = 4$, which is 0100 in binary representation. Each hash bucket has room for 3 records. The record (0,a) means that 0 is the search key and a is the remaining attributes of the record. Consider the following extendable hash index with $i = 0$.



Suppose that we insert two records (3,e), (4,f) into this hash index in the given order. Draw the index after each insertion.

After inserting (3,e):

(continued on the next page)

After inserting (3,e) and (4,f):