Midterm 2

Name:

CS386D Database Systems
Date: 4/23/2020
Version: 1.0

There are 4 sections totaling 200 points. Point weighting is in parenthesis. Place your answers on the question sheet OR if this is not currently feasible for you, a) begin each of the 4 sections on a new page, being careful to clearly label the page with the section number, b) on the first page of the exam, be sure to write your name. You have one hour and fifteen minutes. Good luck. D.M.

I. **Definitions** – Define the following terms (60)

A) "System Catalog"

From course notes:

A database system catalog contains:

- *The schema in the database*
  - *relation names, their attributes, their types*
- *Statistics about the data*
  - *size of each field*
  - *number of rows in each table*
  - *statistical properties …*

Also acceptable, a summary,

System Catalog is a DBMS data structuring containing information about the database schema, and statistical details per the size of tables and statistical distribution of data in each column.

B) "View"

From the lecture notes:

A Database Mechanism

1. *For expressing a subroutine like mechanism*
2. *For capturing application semantics*
3. *For defining alternate data models of the actual*

Also acceptable
Text;

[virtual] views, which are relations that are defined by a query over other relations.

C) "Materialized View"

- *A view where the defining query is executed and stored*

D) “Join Selectivity” (preferred answer is expressed succinctly using mathematical notation)

$$\text{join selectivity} = \frac{|R \bowtie S|}{|R \text{ x } S|}$$

E) Define “logical query plan” and “physical query plan”, being careful to explain the differences.

(10) A logical plan, or logical query plan, is a relational expression, usually illustrated as a tree, that specifies the inputs and outputs of a collection of logical relational operators needed to implement a query.
(10) A physical plan is a logical plan annotated (or adorned) with implementation details such as, the specific physical operator (method) that will be used to execute the query, the schema of the rows that form the input and output of the operators and statistics, such as the number of rows, used by a query optimizer to evaluate the cost of a particular plan.

II. **Short Answer**(60)

A) (20) i)What advantages, if any, are gained by restricting query optimization to right or left deep query plans?

Grading: Each of the bullets bullets below may be an advantage or disadvantage depending on the Engish wording. Full credit, all five concepts, stated in the positve or negative 4 points each.

- *Enables pipelining of operators*
- *(which in turn) minimizes memory requirements*
- *Limits the search for a good plan (acceptable: faster)*

ii) What are the disadvantages, if any?

- *May miss a much better plan, in general, per cost, relative to allowing any tree topology*
- *Does not enable subqueries to be processed in parallel – in the form of independent subtrees of a larger plan to be executed independently.*

B) (20) Given a relation R(a,b,c), and its size in blocks B(R) = 10,000 and column cardinalities V(R,a) = 10,000, V(R,b) = 2000, V(R,c) = 11 and the number of memory buffers, B(M) = 1000. Per the text book's linear cost model (omit the final write),

i. How long will a two phase sort take when sorting on column a?

*3 * B(R)*

ii. How long will a two phase sort take when sorting on column b?

same

iii. Can we use two phase sort when sorting on column c? (yes/no and explain) (Hint: Consider each step and/or phase of the algorithm.)

Yes .Because B(R)/B(M)=10,000/1000=10< B(M)-1

Explanation in English:

- *In phase 1, a set of sorted temporary files will be created. They will have many duplicate values, but that is not a challenge.*
- *In phase 2, the number of temporary files is less than the number of available buffers, so the algorithm will not fail due to memory requirements. Although the final merge step will have to consider equal values appearing in more than one temporary file, this too is properly handled by the algorithm, (and it must do so, even when the cardinatliy of the column is very high).*

C) (20) In some RDBMSs foreign-key constraints are restricted to reference only the primary key of the parent table.

   i. Explain the advantage(s) of such a restriction, if any.

An operation that causes the foreign-key constraint to be checked will be supported [made faster] by a B+ tree

   ii. Explain the disadvantage(s) of such a restriction, if any.

May not be able to directly express the intended semantics. Consider ON UPDATE CASCADE, *an updated value in the parent table can not be propagated to the child table. So, as a feature, this restriction is a real limitation.*

III. Query Optimization (50)

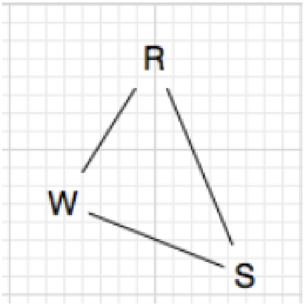
Consider the following SQL create table commands, including the consistency constraints, the query and values from the data catalog.

T(R) = 1000	V(R,b) = 1000	V(R,c) = 50	
T(S) = 500	V(S,a) = 50	V(S,b) = 50	
T(W) = 1500	V(W,c) = 100	V(W,d) = 50	V(W,e) = 15

CREATE TABLE R (b float NOT NULL, c int);
CREATE TABLE S (a float, b float NOT NULL REFERENCES R)
CREATE TABLE W (c int, d float, e varchar(64));

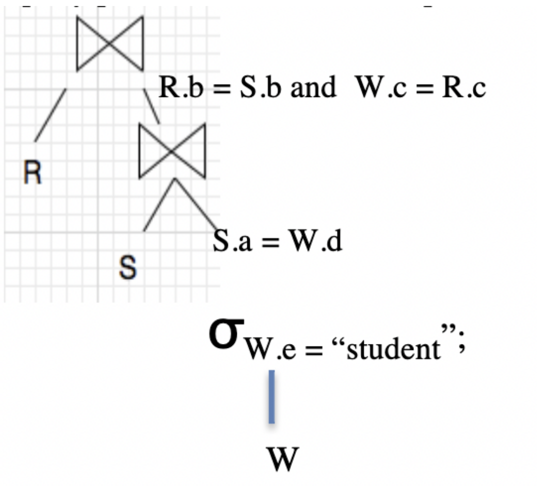
SELECT *
FROM R, S, W
WHERE R.b = S.b and S.a = W.d and W.c = R.c and W.e = “student”;

a) Draw the query graph. (10)



b) Illustrate a right deep logical query plan with the leaves in alphabetical order.(10)

c) Annotate your illustration with the detail (predicate for) each join operator, the size of the intermediate results and the size of the final result.(30)



$$\text{Size} = \frac{T(R) * 1000}{max(V(R,b),\, V(S,b)) * max(V(R,c),V(W,c))} = \frac{1000*1000}{1000*100} = 10$$

$$\text{Size} = \frac{T(S) * 100}{max(V(s,a),\, V(W,d))} = \frac{500*100}{50} = 1000$$

$$\text{Size} = \frac{T(W)}{V(W,e)} = \frac{1500}{15} = 100$$

IV. **Semantics of foreign key constraints.** (Multiple choice, choose the best answer) (30)

1) Consider this snippet of code declaring a table in SQL that contains the information of all UT student employees, including your TAs:

```
Create Table UT_StudentEmployees(
        Name char(200),
        SSN char(11) UNIQUE,
        EID char(7) Primary Key,
        Phone Integer);
```

Consider two additional tables, UT_Employees and UT_Students. Assume both tables also contain (at minimum) name and EID fields and that EID is primary key in both. Which of the options below BEST describes the most appropriate foreign key constraints for this database?

A. UT_Employees.EID references UT_StudentEmployees(EID);
UT_Students.EID references UT_StudentEmployees(EID)

B. UT_Employees.name references UT_StudentEmployees(name);
UT_Students.name references UT_ Employees(name)

[✓ C. UT_StudentEmployees.EID references UT_Students (EID);
UT_StudentEmployees.EID references UT_ Employees(EID)]

D. UT_StudentEmployees.name references UT_Students(name);
UT_StudentEmployees.name references UT_ Employees(name)

E. Foreign key constraints are not suitable for this problem

Consider the following SQL fragment
"(T1.attribute1, T1.attribute2) references T2(attribute1,attribute2) ON UPDATE SET NULL"

2) Suppose a tuple in T2 is updated from (a,1) to (a,5). The modification succeeds. What happens next?

A. Nothing.
[✓ B. All occurrences of (a,1) in T1 would be set to (NULL,NULL)]
C. All occurrences of (a,1) in T1 would be set to (a, NULL)
D. All occurrences of (a,5) in T1 would be set to (a, NULL)
E. All occurrences of (a,5) in T1 would be set to (NULL,NULL)

3) Now I change (b,3) in T1 to (m,3). The modification succeeds. What happens next?

[✓ **A.** Nothing]
B. All occurrences of (b,3) in T2 would be set to (NULL,NULL)
C. All occurrences of (m,3) in T2 would be set to (NULL, NULL)
D. All occurrences of (b,3) in T2 would be set to (NULL,3)
E. All occurrences of (m,3) in T2 would be set to (NULL,3)