CompSci 516: Database Systems

Midterm

Fall 2019

Question Booklet

Total : 100 points

- You should receive one question booklet (this one) and one answer booklet. Return both booklets at the end of the exam.
- Write all your solutions in the other (answer) booklet. This (question) booklet will not be graded.
- This question booklet is printed single-sided. You may use the back of the pages as scratch space, but what you wrote on this booklet will not be graded.
- No external help (books, notes, laptops, tablets, phones, etc.) or collaboration is allowed.
- You have 75 minutes to answer questions that add up to 100 points. i.e. you have about 7.5 mins for 10 points, and about 15 mins for 20 points.
- If you cannot solve a problem fully, write partial solution for partial credit.
- Do not spend too much time on a problem that you find difficult to solve - move on to other problems.
- The problems are organized in no particular order, easier problems may appear later.
- There is no penalty for guessing answers to questions. However, for short-answer questions, simplicity and clarity of solutions will count. You may get as few as 0 points for a problem if your solution is far more complicated than necessary.
- There are 9 pages (1 for scratch paper) in total in this question booklet, please raise your hand should your copy be incomplete.

Problem 1: True/False (12 points)

For each of the following questions, please answer “T” (for “true”) or “F” (for “false”).

Problem 1a. (2 points)

Consider relation R(A, B, C). {A} and {A, C} **CANNOT** both be keys of R at the same time.

Problem 1b. (2 points)

Consider relation R(A, B, C). {A, B} and {A, C} **CANNOT** both be keys of R at the same time.

Problem 1c. (2 points)

Consider relations R(A), S(A), T(A). $(R \bowtie S) - (R \bowtie T)$ is **ALWAYS** the same as $R \bowtie (S - T)$ under set semantics.

Problem 1d. (2 points)

A hash index will **ALWAYS** have better performance than tree index if we have equality conditions in the query.

Problem 1e. (2 points)

In R(A, B, C, D), if we have a **hash index** on (A, B, C), it can be used to answer the query

SELECT * FROM R WHERE A = 10 AND B = 7

Problem 1f. (2 points)

In R(A, B, C, D), if we have a **tree index** on (A, B, C), it can be used to answer the query

SELECT * FROM R WHERE A = 10 AND B = 7

Problem 2: Short Q/A (15 points)

Problem 2a. (4 points)

Consider tables R(A,B) and S(B,C). Would the two SQL queries below always return the same bag of rows (disregarding ordering) regardless of the contents of R and S?

- SELECT S.B, S.C FROM R, S WHERE R.B = S.B;
- SELECT * FROM S WHERE B IN (SELECT B FROM R);

Write yes/no, and briefly explain why or why not.

Problem 2b. (3 points)

Suppose we have relations R1(A, B), R2(B, C), R3(C, A) each with 10 distinct tuples. Can the size of the natural join of R1, R2, R3 can be as big as 1000 (= 10 x 10 x 10)?

Write yes/no, and briefly explain why or why not.

Problem 2c. (3 points)

Consider the following relation R.

A	B	C
10	null	7
null	7	null

SELECT *

FROM R

WHERE ((A >= 5) OR (B = 7)) AND (C <= 9)

What is the output of this query? 1^{st} tuple, 2^{nd} tuple, or both? Briefly explain.

Problem 2d. (5 points)

Consider external merge sort on R with 100 pages and B = 6.

What is the total cost of external merge sort? Explain showing your solution in Pass 0, Pass 1, etc. by mentioning the number of runs in each pass, and their size.

Problem 3: SQL (10 points)

Consider the following schema. Keys are underlined.

Likes(<u>drinker</u>, <u>beer</u>)

BeerInfo(<u>beer</u>, bottle-color, price) – color of bottle and price ($) of each beer

Write a “**single-block**” SQL query to find all drinkers who like at least 10 beers and all beers they like have price <= $4.

Note that your query can have exactly one SELECT and FROM clauses, so you cannot use any subquery or WITH clauses.

Problem 4. BCNF/FD (19 points)

Consider a relation R (A,B,C,D,E,F) with functional dependencies:

- AB → C
- AB → D
- BD → C
- F → A

Problem 4a. (4 points)

What are all the keys of R? List them all if there is more than one. Explain your answer briefly.

Problem 4b. (4 points)

Which one of the given dependencies is redundant, i.e., it follows from the other three dependencies? Explain briefly.

(i) AB → C
(ii) AB → D
(iii) BD → C
(iv) F → A
(v) None of the above

Problem 4c. (6 points)

Decompose R into BCNF. Show how you obtained the answer.

Problem 4d. (5 points)

Suppose we decompose R into R1(A, B, F, E), R2(A, C, D, F).

Is this decomposition lossless?

- If yes, give a brief explanation.
- If no, give an example why not.

Problem 5. Query evaluation (14 points)

Consider schema

B(bid, title, author, price) -- Books

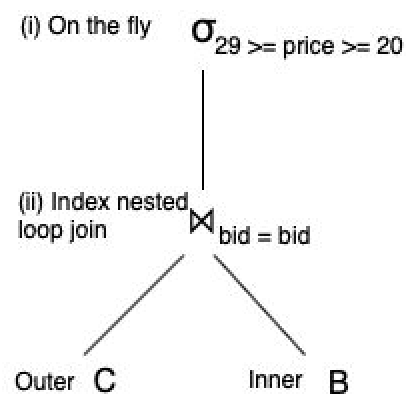
C(bid, sid, date) -- Checkout, bid, sid are foreign keys to B and S respectively.

S(sid, name) -- Students

Assume:

- B has T_B = 100 tuples, B_B = 20 pages, the range of price is [10, 99]
- C has T_C = 900 tuples, B_C = 10 pages
- All index pages are in memory
- B has a clustered index on B.bid
- Independence and uniformity as standard

Consider the plan:



Problem 5a. (6 points)

Fill out the following table with the cost (page I/O) and output size (number of tuples)

Step	Cost	Size
(i) Index nested loop join		
(ii) selection on price		

Problem 5b. (3 points)

Show your calculations briefly for the above question.

Problem 5c. (2 points)

Would the cost change if the index is “unclustered” in (i)? Briefly explain.

Problem 5d. (3 = 2 + 1 points)

- What would be the cost if B is outer and C is inner relation in the join for clustered index on C.bid? No explanations needed.
- Would the cost change from the above cost if the index on C.bid is unclustered? Explain in words.

Problem 6. RA/SQL/RC (10 + 10 + 10 = 30 points)

Consider the following schema. Keys are underlined.

Likes (<u>drinker, beer</u>)

BeerInfo (<u>beer</u>, bottle-color, price) – color of bottle and price ($) of each beer

Friend (<u>d1, d2</u>) – whether drinker d1 and drinker d2 are friends. The relation contains both pairs, say ('Alex', 'Bob') and ('Bob', 'Alex').

Answer the following query:

Find drinkers such that the <u>only</u> beers they like are in "red" bottles <u>and</u> are also liked by all their friends.

(If the drinker in your answer likes a beer, it must be in red bottle and must be liked by all of his/her friends)

(If there are no friends, then only the first condition on color is needed to hold.)

(If the drinker does not like any beer, s/he should be in the answer.)

(Likes.beer is a foreign key to BeerInfo.beer).

Problem 6a. (10 points).

Write the query in **Relational Calculus.**

Problem 6b. (10 points).

Write the query in **SQL**

Problem 6c. (10 points).

Write the query in **Relational Algebra** (you can write RA expression or a logical query plan tree).