CS 564 Midterm Exam

Open book, open notes, open internet – closed friends and neighbors

If you find a question ambiguous or difficult, please do not ask other students in the class. Instead, please post your questions on Piazza. Please do not post answers to other students' questions – please leave that to the instructor and the TAs. Feel free to post follow-up questions if you find an answer ambiguous. Before you post a question, please check whether a similar question is already posted and perhaps even answered.

Please submit your exam as a single pdf file by **Friday, November 1, 12:00PM noon** (not midnight) on Canvas. Make sure to submit before the deadline-- late submissions will not be accepted.

Unless you have very readable handwriting, please type your answers. If we can't read it, we can't credit it. For some questions, please draw by hand and label by hand, but make sure it's all readable – then scan it or photograph it and include it in your one pdf file.
For questions 8-11, please be concise and use short, succinct sentences. 30-100 words should suffice. Do not copy from the book, from Wikipedia, or from other internet sources.

Starting on the next page, answer each of the following questions or assignments:

1. Draw an ER diagram for a (much simplified) database for a lending library. Map the following concepts to entities, relationships, and attributes: book, title, author(s), borrower, loan date (when the book was borrowed from the library), due date, patron (a person who may borrow books), name, address, phone (some attributes apply to both the library and to each patron). You may add a few attributes but keep that to a small number. For better or worse, there may be books with the same title, but perhaps by different authors. A patron may borrow the same book multiple times.
2. Map your ER diagram to a schema (table names, column names, primary and foreign keys). You can use the TPC-H schema as an example, but make sure you clearly indicate the primary and foreign keys.
3. Write "create table" statements for these tables – include all integrity constraints that make sense. Try to include each kind of integrity constraint at least once in the database. Each table should have a primary key and each table should include a foreign key or be referenced by a foreign key.
4. Write the following queries over your tables. Feel free to use "with" clauses where convenient; use nested queries sparingly.
    a. List all book titles in ascending order – include the first author's name for each book, and include duplicate titles.

    b. Count the number of books that the library owns.

    c. For each book borrowed at least once, list the title, the number of times it has been borrowed, and the most recent due date (whether that due date is in the past or in the future).
5. For the last query in question 4, draw or write an expression in relational algebra. (As we did in class, assume the algebra includes a grouping/aggregation operation in addition to the usual select, project, and join operations).

6. Map your relational algebra expression to a query execution plan (draw or write).

7. Provide an alternative query execution plan (draw or write).

8. Decide which one of your two alternatives is likely more efficient – state your choice and give reasons for your choice, or state that you can't decide and give reasons why either plan might be more efficient.

9. Explain (in your own words) the differences between an inner join, a (left) semi join, and a (left) outer join. Your explanation should address the sets of columns and the sets of rows in each operation's output.

10. Explain (in your own words) the differences between index nested loops join, merge join, and hash join (all used to compute an inner join). Your explanation should include required properties of the join inputs and properties of the join output. For the hash join, you may assume that one join input fits in memory. For extra points, explain what happens if neither join input fits in memory.

11. For extra points, explain (in your own words) the differences and similarities between hash aggregation and hash join. For the hash join, you may assume that one join input fits in memory. For extra extra points, explain differences and similarities if neither input nor output fit in memory.