Home Grades View submissions View feedback Logout

jmajikes

Assignment was due by 2021-12-07 15:15:00

COMP 421 Final Exam Fall 2021

Note: There are 110 total points on this exam. You can get a maximum score of 100 points on this exams.

Don't panic!

You have 120 minutes to finish the exam.

- You *must* stay in full screen mode. Points removed for leaving full screen mode
- You must hand this final in on time.
  - Points removed for late submissions.
  - Only your **first** submission will be accepted.
  - Avoid accidental submissions. Fill in your name when you are ready to submit.
- Points removed for accidental submissions.

I recommend that you have several pieces of scrap paper to doodle notes on during the exam.

Consider this final **closed book**.
You MAY use your hand written notes. They MUST be on paper as you may not switch screens after starting the exam.

You **MAY NOT** Google or use other external websites for **answers** or copy from a friend. Do not paste information into your exam unless it was copied from your exam. You **MUST NOT** receive help from anyone.

If you do not know the origin of material you should not paste it into this exam. All material pasted into this exam must originate from this exam. This implies, but is not limited to, copying from previous assignments, copying from text messages, or copying from **any** website.

You **MUST** use the Google Chrome browser.

The browser will change input box color green to indicate correctness. A black or red box indicates an incorrect answer.

Note that HTML select statements with drop-downs are simple multiple choice questions. No highlighting of correct answers are done for select questions.

Green highlight should just assist you. If you believe your answer is correct and the input box did not turn green, continue on. Per the syllabus, highlighting is simply an aide not a guarantee. **Note:** For database queries that are applied to two databases, two green lights are required to get any credit for the question.

Questions Total of 100 points out of a possible 110 points.

Book SQL Queries 70 points

In this section, you will write SQL queries for book schema used in homework A5 and described below. Your queries will be tested immediately against two different databases. If your queries output matches the expected output, the displayed answers will be outlined in green. You need green, highlighted output from the query of both databases to get any credit. Your actual score will be determined when your query is tested against a different database but **two** green feedbacks should mean that you are on track to receive full credit.

The schema for the book database is below:

```
CREATE TABLE Authors (
   login text primary key,
   email text,
   first_name text,
   last_name text
)

CREATE TABLE Books (
   id integer primary key autoincrement,
   title text,
   author text references authors(login),
   language text,
   created date
)

CREATE TABLE Photos (
   id integer primary key autoincrement,
   url text unique,
   width integer,
   height integer
)

CREATE TABLE Pages (
   book integer references books(id),
   number integer,
   photo integer references photos(id),
   caption text,
   primary key (book, number)
)

CREATE TABLE Categories (

   category text,
   book integer references books(id),
   unique (book, category)
)

CREATE VIEW PageCounts as
   select book, count(*) as count from pages group by boc
```

Following is a scratch space which can be used to help you develop and test queries using **only the smaller** books database.

Execute

Simple.List.1: List all the authors' first names and last names in alphabetical order by last name.

Execute 2.5 points Minimize Output

Page.Pages.1: List all the authors' first names, last names, book titles, and number of pages in order by book title.
NOTE: If there are multiple editions of a book, possibly created at different times, they should be counted as separate, distinct books.

Execute 2.5 points Minimize Output

Average.Pages.1: List all the authors' first names, last names, and their average page count per book in inverse alphabetical order by their login.

Execute 5 points Minimize Output

Two.O.Author.List.1: List the first and last name of authors who have two case-insensitive O's anywhere in their last name.
NOTE: SQLite uses case insensitive matching for selection criteria using the LIKE operator and case sensitive in the selection criteria equal operator.

Execute 5 points Minimize Output

Categories

Category.1: List all the categories and the number of books that are in each category.
If the number of books is less than 2 display a null. If the number of books is 2 or more, display the actual count.

Execute 15 points Minimize Output

Max.Categories.1: List the book title of the book(s) with the most number of categories along with the number of categories for those book(s).

Execute 15 points Minimize Output

Author.Category.1: For all authors who have had more than one book in the category of `Animals and Nature` list the author's first name and last name along with the number of the author's books in the category `Animals and Nature`.

Execute 5 points Minimize Output

Photos

Book.Photo.1: For books that have photos, list the book's title along with the number of photos in that book.
NOTE$_1$: If a book has the same photo on two pages, count it as two.
NOTE$_2$: If there are multiple editions of a book with the same title, possibly created at different times, they should be counted as separate, distinct books.

Execute 5 points Minimize Output

Max.Book.Photo.1: For the book(s) that has the most photos, list the book's title along with the number of photos in that book.
NOTE: If a book has the same photo on two pages, count it as two.

Execute 15 points Minimize Output

B trees 5 points

In this section, you will show your knowledge of bulk loading a B$^+$ tree.

Calculator.B.Tree: You may use this box as a calculator. Just type in any expression that can be evaluated by a JavaScript eval. For example, `2 ** 0.5` will show you the result of the square root of 2 in the red box to the right. `log(10)` will show the natural log of 10 in the red box to the right. Lastly, you may want to utilize the log identity of $\log_x y = \log y / \log x$. Leaving the box empty or filled in will **NOT** affect your grade.

Enter an expression

B.Tree.1: Assume you have to bulk load a B$^+$ tree using the algorithm from the book. Each index page or node in the B$^+$ tree can contain 5 index entries. Each data page or leaf node can contain 3 tuples.
Further assume that the keys for the bulk data load are the following integers: 1, 12, 2, 11, 3, 10, 4, 9, 5, 8, 6, 7
In the table below, select the index entries in the root index node. If an index entry is unused, leave it blank.

Position	Index value
Index entry 1	
Index entry 2	
Index entry 3	
Index entry 4	
Index entry 5	

5 points

External Sort 20 points

In this section, you will show your knowledge of the external sort algorithm discussed in the textbook.

Calculator.External.Sort: You may use this box as a calculator. Just type in any expression that can be evaluated by a JavaScript eval. For example, `2 ** 0.5` will show you the result of the square root of 2 in the red box to the right. `log(10)` will show the natural log of 10 in the red box to the right.

Lastly, you may want to utilize the log identity of $\log_x y = \log y / \log x$. Leaving the box empty or filled in will **NOT** affect your grade.

Enter an expression

External.Sort.Table.1: Assume you have a relation with 30000 tuples, 100 tuples fit in a page, and there are 5 pages to use for the external sort algorithm (B=5). Fill in the table below for the number of runs and pages per run for each pass of the external sort.

Pass number	Number of runs	Pages per run
Pass 0		
Pass 1		
Pass 2		
Pass 3		

15 points

External.Sort.Two.Pass.1: Assume you want to external sort the same relation in two passes, what is the minimum number of buffers required to sort the relation in two passes.

5 points

Evaluations 15 points

In this section, you will show your knowledge of query evaluation.

Let relation Authors take 1000 pages and relation Books take 2000 pages. Assume both Authors and Books tuples are 64 bytes such that each 4K page contains 64 tuples. Neither relation has an index and neither is sorted.

Min.IO.1: Determine which of the following has the lowest cost in terms of I/Os to compute the cross product (not the natural join) of $Authors \bowtie Books$?
You may assume there is infinite memory, but if two of the following have the same lowest I/O cost, choose the one that requires the least number of memory pages.

A: Read in Authors tuple by tuple, perform a simple nested loop with an inner loop of Books
B: Read in Books tuple by tuple, perform a simple nested loop with an inner loop of Authors.
C: Read in Authors block by block, perform a block nested loop with an inner

loop of Books.
D: Read in Books block by block, perform a block nested loop with an inner loop of Authors.
E: Externally sort both relations in two passes, read in Authors block by block, perform a block nested loop with an inner loop of Blocks.
F: Externally sort both relations in two passes, read in Books block by block, perform a block nested loop with an inner loop of Authors.

Enter an expression 1 point

How.Many.IO.1: Again, assuming infinite memory, each tuple is 64 bytes, and the resulting cross product tuple is 128 bytes,
how many I/O operations does it require to read in both relations, complete the cross product using the least number of I/Os, and write out the result?

Enter an expression 7 points

How.Many.Pages.1: How many memory pages are required to complete the lowest-cost cross product?

Enter an expression 7 points

Pledge your exam

DO NOT PUT YOUR NAME ON THE EXAM UNTIL YOU ARE READY TO SUBMIT IT.
If you submit it and decide you would like to invalidate a submission, it will cost 10 percent of the points!

UNC Honor Pledge: I certify that no assistance has been received or given in the completion of this work. This unauthorized assistance includes, but is not limited to, using Google or other external websites, copying another student's answers, or giving another student answers.

Enter your full name

If other students helped in the completion of this exam, and you would like to avoid an Honor Court hearing, give attribution to them by entering their Onyen below. Of course, getting/giving assistance will result in your submission being invalidated but the attribution will avoid an Honor Court hearing.
If no other student helped with the completion of this exam, **None** must be entered below.

Enter the Onyen of any students who assisted in this exam's comple

Done!

Submit your work

Only your **first** submission will be accepted.

Did you **pledge your work** above?

Did you **acknowledge any student assistance** above?

Submit code from the front of class: Enter the submit code

Submit