CS 186, Fall 93
Final

Professor Unknown

There are 10 questions to this exam. Credits for each question is indicated in brackets. There are a total of 100 points.

Queries expressed in relational algebra and relational calculus must follow syntax used in class. Queries expressed in SQL must follow the syntax used in class or in the INGRES Manual. All SQL queries should contain NO duplicates in their outputs. It is not necessary to sort the outputs in any order.

You are not allowed to use views for any questions.

Problem #1

Consider the following relational database schema:

STUDENT(name, regno, gpa, level, dept)
COURSE(cno, cname, dept)
TAKE(regno, cno)

(1a)[5] Express in relational algebra the names of students who are taking a course offered by the EECS department.
(1b)[5] Express in relational calculus the names of the students who are not taking any courses offered by the EECS department.

Problem #2

Consider the following relational database schema:

DEPT(dname, location)
STUDENT(name, regno, gpa, level, dept)
COURSE(cno, cname, dept)
TAKE(regno, cno)

(2a)[5] Express in SQL the names f the courses taken by any student who is from a department located in 'Evans Hall'.
(2b)[5] Express in SQL the names of the departments which have no students with gpa better than 3.5 taking less than 4 courses.

Problem #3

[10] Consider the following relational database schema:

DEPT(dname, location)
STUDENT(name, regno, gpa, level, dept)

Express in SQL the names of the departments that satisfy the following condition: the average gpa of all the students in the department is better than the average gpa of all the students from departments located in the same building as the department.

Problem #4

[10] Consider the following relational database schema:

DEPT(dname, location)
STUDENT(name, regno, gpa, level, dept)
COURSE(cno, cname, dept)
TAKE(regno, cno)

Write the rule(s) and the associated stored procedure(s) to make sure no student takes more than 5 courses.

Problem #5

[10] Consider a table with data records stored in a heap structure Suppose there are 1,048,576 fixed length records stored in this table, each record has 64 bytes. Now we want to build a B+ tree secondary index on it. The key field on which the secondary index will be built has a fixed length of 12 bytes. Each record id (*rid*) is 6 bytes. A page number is 4 bytes long. Each page is 1024 bytes(of which 1000 bytes can be used, the rest is used to store page header information).

(5a)[5] How many levels (including the root and leaf level) at most is necessary in the B+ tree index? Please state your reasoning in support of your conclusion.
(5b)[5] At most how many page splits will occur in the process of building this index? Please state your reasoning in support of your conclusion.

Problem #6

Suppose there are two tables: EMP and DEPT. EMP is stored in a B+-tree, the key is its dept field. The DEPT table is stored in a heap structure. EMP has 50,000 records, occupying 25,000 data pages. DEPT has 500 records, occupying 500 pages. The *dname* attribute in DEPT has unique values. Assume uniform distribution. Consider the query:

select e.name from EMP e, DEPT d where e.dept = d.dname

Suppose the DBMS has only implemented iterative substitution and hash join. Neither the EMP table nor the DEPT table fits into memory. However, the number of memory pages available for processing the query is 26. Assume the parameter $w = 20$.

(6a)[10] Choose an optimal strategy to evaluate the above query and give the cost for this strategy. Please state your reasoning in support of this conclusion.
(6b)[5] If the above query is processed using the hash join algorithm, what is the number of hash buckets that should be used? Please state your reasoning in support of your conclusion.

Problem #7

Consider the following two transactions:

T1:

begin xact
write C
read B
write C
commit xact

T2:

begin xact
write B
read C
read C
commit xact

(7a)[10] In a DBMS using the two-phase locking algorithm, whether transactions will cause deadlocks depends on how they are executed. If the above two transactions are executed concurrently, under what situations can a deadlock occur?
(7b)[5] In a DBMS that has not implemented any concurrency control algorithms, can non-repeatable reads occur if the above two transactions are executed concurrently? State your reasoning in support of your conclusion.

Problem #8

[5] In a DBMS using the write-ahead log technique, after a soft crash, the transaction log is processed to redo the changes of all committed transactions and to undo the changes of all uncommitted transactions. In the redo process, the transaction log is scanned forward and log records are redone in the order that they were written into the log. In the undo process, the transaction log is scanned backward and log records are undone in the reverse order that they were written into the log. If in the undo process, the transaction log was scanned forward and the log records were undone in the order that they were written into the log, give an example of a transaction that would be undone incorrectly.

Problem #9

We have discussed two techniques that have the phrase "two-phase" in them: two-phase locking and two-phase commit.

(9a)[5] What are the similarities and differences between two-phase locking and two-phase commit?

(9b)[5] What are the two phases in each of the two techniques?

Problem #10

[5] In a parallel DBMS running on a multiprocessor system, when the number of processors increases, the performances of the DBMS should also improve. The performance improvement is usually measured by "speed up" and "scale up". What is the difference between the two measurements?
