Name: ____________________

Student ID: ____________________

Department of Computer Science

Final Exam, CS 411a/538a — Databases II

Prof. S. Osborn
Dec. 10, 2001
3 Hours
One sheet of notes allowed

Answer all questions on the exam page

Question	Maximum	Your Mark
1	16	
2	25	
3	43	
4	25	
5	18	
6	14	
7	12	
Total	153	

1. (16 marks) For each of the following, state whether the term applies to one, two or all three of: centralized relational databases, distributed relational databases or object-oriented databases. Give a **BRIEF** reason for your answer.

   (a) identity comparisons

   (b) the semi-join algorithm for computing joins

   (c) path expressions

   (d) query language calls embedded in programs

   (e) indexes on OIDs

   (f) discretionary access control

   (g) ACID properties

   (h) two-phase locking

2. (25 marks) For each of the following, state whether the statement is **true** or **false**. If it is false, correct the statement without changing the underlined text.

   (a) The uncertainty period occurs when we do not know what is the most efficient query execution plan.

   (b) Pointer swizzling is necessary to convert strings into the format used by the current compiler.

   (c) The waits-for graph is used with locking-based concurrency control to avoid deadlocks.

   (d) For locking-based concurrency control in a distributed database, it is considered better to use deadlock avoidance.

   (e) Closure of a query language refers to the successful completion of a query execution plan.

(f) With shadow paging, the data is written to the log buffers and forced to disk before the commit point.

(g) Write-ahead logging is commonly used in centralized database systems for handling commit, restart and recovery.

(h) Two-phase commit is commonly used in centralized database systems for handling commit, restart and recovery.

(i) To execute the join using an index algorithm, the join attribute needs to be hashed with the same hash function in both relations.

(j) Polyinstantiation can occur in the Seaview model but not in the Smith and Winslett model.

3. Consider the following O_2 database schema:

```
class Person type
  tuple (name : string,
         address : Addr) end;
class Addr type
  tuple (street : string,
         city : string) end;
class Student inherit Person type
  tuple (studID : string,
         year : integer,
         courses : set(Course) ) end;
class Course type
  tuple (cName : string,
         dept : string,
         registered : set(Student) ) end;
class GradStudent inherit Student type
  tuple (supervisor : Person
         gradcourses : set(Course)  ) end;
class GradCourse inherit Course type
  tuple (times : set( tuple (beginTime : string,
                              endingTime : string)) ) end;
name AllStudents : set(Student);
name Courses : set(Course);
method init (N : string, A : Addr) in class Person;
method init (Street : string, City : string) in class Addr;
method init (SID : string, Year : integer) in class Student;
method init (CNAME : string, Dept : string) in class Course;
```

(a) (6 marks) Write an init method for class Course, which takes a course name and department as parameters, and inserts the new course into Courses.

(b) (10 marks) Write a run body which creates a grad course, whose name is Advanced Databases, offered by the computer science department. Prompt the user for and read in a start time and an end time, and make this pair the only time recorded for the "times" attribute. Insert the course into the Courses set.

(c) (4 marks) Write a query in OQL which finds, in AllStudents, the students whose name contains "Do".

(d) (6 marks) Write a query in OQL which finds, in Courses, the courses whose registrants are ALL in year 4.

(e) (Query Execution)

i. (4 marks) Assume that we have a persistent set of GradStudents called GS. First, express the following query in OQL: from GS, find all GradStudent objects such that the name of the grad student is “Smith” and the name of the supervisor is “Smith”.

ii. (5 marks) Draw the query graph for your query in part i.

iii. (5 marks) Briefly describe the **top-down** execution of this query.

iv. (3 marks) For **bottom-up** execution of this query to work, what index(es) is(are) essential?

4. Consider the following **relations** for a **relational database**:

```
Student (name, STID, address) key: STID
Course (cName, dept, deptAddress) key: {cName}
Registered (STID, cName, mark) key: {STID, cName}
```

and the following statistics for relations Course and Registered:

tuples in relation Course = 200

bytes in attribute cName = 20	distinct values of cName = 200
bytes in attribute dept = 20	distinct values of dept = 25
bytes in attribute deptAddress = 25	distinct values of deptAddress = 30

tuples in relation Registered = 1000

bytes in attribute STID = 8	distinct values of STID = 400
bytes in attribute cName = 200	distinct values of cName = 200
bytes in attribute mark = 2	distinct values of mark = 51

(a) (1 mark) How many bytes per tuple are there in relation Course?

(b) (1 mark) How many bytes per tuple are there in relation Registered?

(c) (2 marks) How many tuples are estimated to be in the fragment:
F1 = $\sigma_{dept=\text{"}CS\text{"}}$ (Course)

(d) (2 marks) How many tuples are estimated to be in the fragment:
F2 = $\sigma_{dept\neq\text{"}CS\text{"}}$ (Course)

(e) (2 marks) How many bytes per tuple are in Course $\bowtie$ Registered?

(f) (2 marks) How many bytes per tuple are in Course $\rtimes$ Registered?

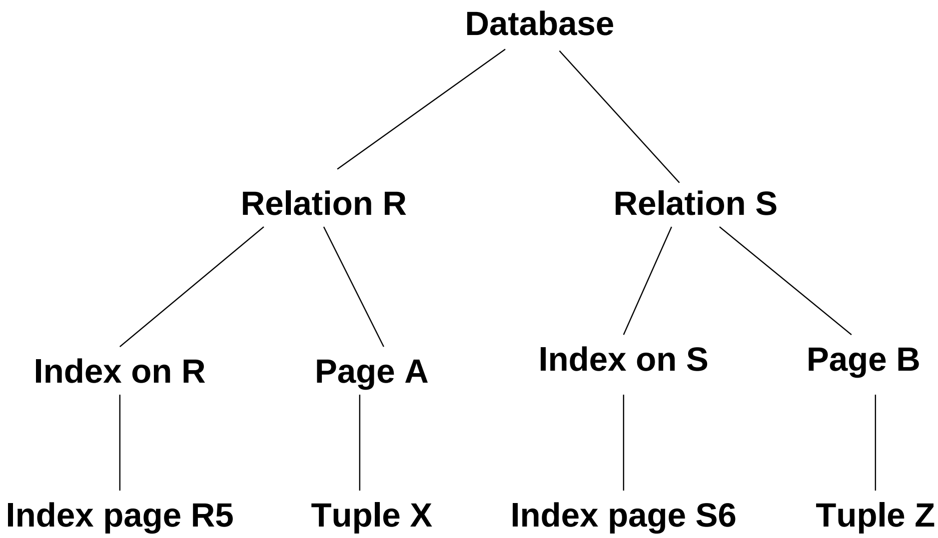
(g) (5 marks) What is the total cost of performing Registered $\ltimes$ F1, assuming they are at different sites of a network?

(h) (10 marks) Show how the algebraic query optimization technique would optimize the following query (you may just show the original tree and the final tree if you like):

π_{mark}
$(\sigma_{name=\text{"}Smith\text{"} \ \wedge \ STID=\text{"}11111\text{"} \ \wedge \ dept=\text{"}CS\text{"}}$
$(\pi_{Student.name,\ Student.STID,\ Course.cName,\ Course.dept,\ Registered.mark}$
$(\sigma_{Student.STID=Registered.STID \ \wedge \ Course.cName=Registered.cName}$
$(\text{Student} \ \times \ (\text{Registered} \ \times \ \text{Course})))))$

(for answer to 4(h))

5. (18 marks) Consider the following (simplified) tree of granules for a relational database:



(a) We have the following transaction T1: a new tuple, Y is to be inserted on page A of relation R. The transaction consists of first writing the new tuple to page A, followed by inserting a reference to it in the index, which involves changing page R5 in the index. Show a sequence of lock and unlock instructions which follow the rules for multiple granularity locking, and which allow this transaction to execute.

(b) The second transaction, T2, wants to read tuple X of relation R. In order to do this, it must first read the index for R, and the index page containing the relevant information is index page R5. Show a sequence of lock and unlock instructions which follow the rules for multiple granularity locking, and which allow this transaction to execute.

(c) Show one valid interleaving of T1 and T2 such that there are no deadlocks. Use the following lock compatibility matrix:

Lock Compatibility Matrix

	Already Granted					
Requested	None	IS	IX	S	SIX	X
IS	yes	yes	yes	yes	yes	-
IX	yes	yes	yes	-	-	-
S	yes	yes	-	yes	-	-
SIX	yes	yes	-	-	-	-
X	yes	-	-	-	-	-

(for answer to all of Question 5)

6. (14 marks) Suppose the following timestamps are recorded for the following data items:

data item	read TS	write TS
x	5	9
y	7	5
z	9	4

For each of the following operations, state whether or not it would be allowed using the timestamp ordering algorithms for concurrency control, and, if allowed, **whether or not any timestamps change**. Assume all these operations are independent of each other (i.e., they all refer to the original timestamps for x, y and z).

(a) read(x) on behalf of transaction T1 whose timestamp is 8.

(b) write(x) on behalf of transaction T1 whose timestamp is 8.

(c) read(y) on behalf of transaction T1 whose timestamp is 8.

(d) write(y) on behalf of transaction T1 whose timestamp is 8.

(e) write(z) on behalf of transaction T1 whose timestamp is 8.

(f) write(z) on behalf of transaction T1 whose timestamp is 8.

(g) do any of the above follow the Thomas write rule?

7. (12 marks) Consider the diagram for Two-Phase commit reproduced here from the Özsu and Valduriez book:

   (a) Which node(s) represent the uncertainty period?

   (b) Which node(s) represent the abort state for the coordinator?

(c) Which node(s) represent possible timeouts?

(d) Give the sequence of nodes the coordinator passes through for a successful commit.

(e) Give the sequence of nodes the participant passes through if it votes to commit but the coordinator decides to abort.

(page for rough work)