Name: ___________________________________

Class Account:_____________________________

UNIVERSITY OF CALIFORNIA
Department of EECS, Computer Science Division

CS186 Hellerstein

Fall 2007 Midterm Exam

Midterm Exam: Introduction to Database Systems

This exam has four problems and one extra credit question, worth different amounts of points each. Each problem is made up of multiple questions. You should read through the exam quickly and plan your time-management accordingly. Before beginning to answer a problem, be sure to read it carefully and to *answer all parts of every problem!*

You **must** write your answers on the exam. Extra answer space has been provided at the back in case you run out of space while answering. If you run out of space, be sure to make a "forward reference" to the page number where your answer continues. ***Do not tear pages off of your exam!***

Good luck!

1. Sorting

Zombo.com has begun advertising "sorting as a service" for their website. Users can upload a table of numbers to Zombo.com, and the site will sort it for them and email it back.

Unfortunately, after launching their intergalactic advertising campaign, the executives at Zombo.com realize that they have no idea how to deal with large tables. They hire you as a consultant to work this out for them.

a) Your plan is for each file to be uploaded to a random server in their data center, sorted on that machine, and emailed from that machine – without any other tasks intervening. You promise the Zombo.com executives that the I/O count for sorting the file (including reading it from disk and writing the result to disk) will be approximately 4 times that of simply reading it. To guarantee this, you suggest introducing a limit on the size of file uploads to some number **maxsize** bytes per file. You are told that Zombo.com's favorite operating system is AmigaOS with 8K disk blocks, and that after booting the OS, the Zombo.com servers have approximately 128Kbytes of memory remaining for sorting. What value (in bytes!) should you recommend for **maxsize**?

b) The executives are unhappy with the idea of size limits. They remind you that “the unattainable is unknown at Zombo.com!” They ask you how many disk I/Os it would take to sort a file that is one Yottabyte big. Please state your answer as a function of the value Y (where Y = 1 Yottabyte).

c) The biggest user of the service keeps uploading files that are 256KB big and already sorted. Assuming Zombo.com uses quicksort in memory, fill in the following table for sorting one of those files:

	Random I/Os	**Sequential I/Os**
Pass 0 Read		
Pass 0 Write		
Pass 1 Read		
Pass 1 Write		

2. Query Languages
Consider the following schema of a library. Primary keys are underlined.

```
Author(name, citizenship, birthYear, birthPlace)
Book (isbn, title ,author)
Library (lname, city)
Bindex (isbn, subject)
In_stock (isbn, lib_name, edition, quantity)
```

Based on the above schema, try to answer the following questions:

a) Consider the following SQL query:

```
SELECT a.name
FROM author a
WHERE NOT EXISTS (SELECT b.isbn
                  FROM book b
                  WHERE b.author = a.name AND NOT EXISTS (
                     SELECT *
                     FROM in_stock i
                     WHERE i.isbn = b.isbn AND i.lib_name = 'Evans' AND
                            i.quantity > 2)) ;
```

Fill in the blanks in the following relational calculus query that makes it equivalent to the SQL above,. Note: the correct answer may not require all blanks to be filled in!

{R | __A∊Authors _ A.name = R.name ___ B∊Book ____ B.author = A.name

_______ I∊InStock ____ I.isbn = B.isbn ______ I.lib_name='Evans'
_____quantity > 2 ____ }

b) The following SQL query is given:

```
SELECT b.title, s.edition
FROM book b, in_stock s
WHERE b.isbn = s.isbn and s.lib_name = 'Cao Library' and
        NOT EXISTS (SELECT *
                    FROM in_stock
                    WHERE lib_name = 'Evans Library'
                          and quantity >= s.quantity );
```

Which of the following queries will produce a **different** result set?

i)
```
SELECT b.title, s.edition
  FROM book b, in_stock s
  WHERE b.isbn = s.isbn and s.lib_name = 'Cao Library' and
        s.quantity NOT IN (SELECT distinct quantity
                           FROM in_stock
                           WHERE lib_name = 'Evans Library');
```

ii)
```
SELECT b.title, s.edition
  FROM book b, in_stock s
  WHERE b.isbn = s.isbn and s.lib_name = 'Cao Library' and
        s.quantity > (SELECT MAX(quantity)
                      FROM in_stock
                      WHERE lib_name = 'Evans Library');
```

iii)
```
SELECT b.title, s.edition
  FROM book b, in_stock s
  WHERE b.isbn = s.isbn and s.lib_name = 'Cao Library' and
        s.quantity > ALL (SELECT quantity
                          FROM in_stock
                          WHERE lib_name = 'Evans Library');
```

iv) None of the above.

c) Given the following Relational Calculus statement:

$$\{B \mid \exists B \in Books(\exists A1 \in Authors(A1.birthYear > 1920) \wedge$$
$$(\exists A2 \in Authors(A2.birthYear > 1920 \wedge$$
$$A2.birthYear > A1.birthYear \wedge$$
$$B.author = A2.name))\}$$

circle the Relational Algebra expressions that compute the same result.

i) $\rho(A1, \sigma_{birthYear > 1920}\ Author)$
$\rho(A2, \sigma_{birthYear > 1920}\ Author)$
$Books \bowtie \sigma_{A1.birthYear > A2.birthYear}(A1 \times A2)$

ii) $Books - ((Books \bowtie \sigma_{birthYear < 1921}\ Author) \cup (Books \bowtie \sigma_{birthYear < 1920}\ Author))$

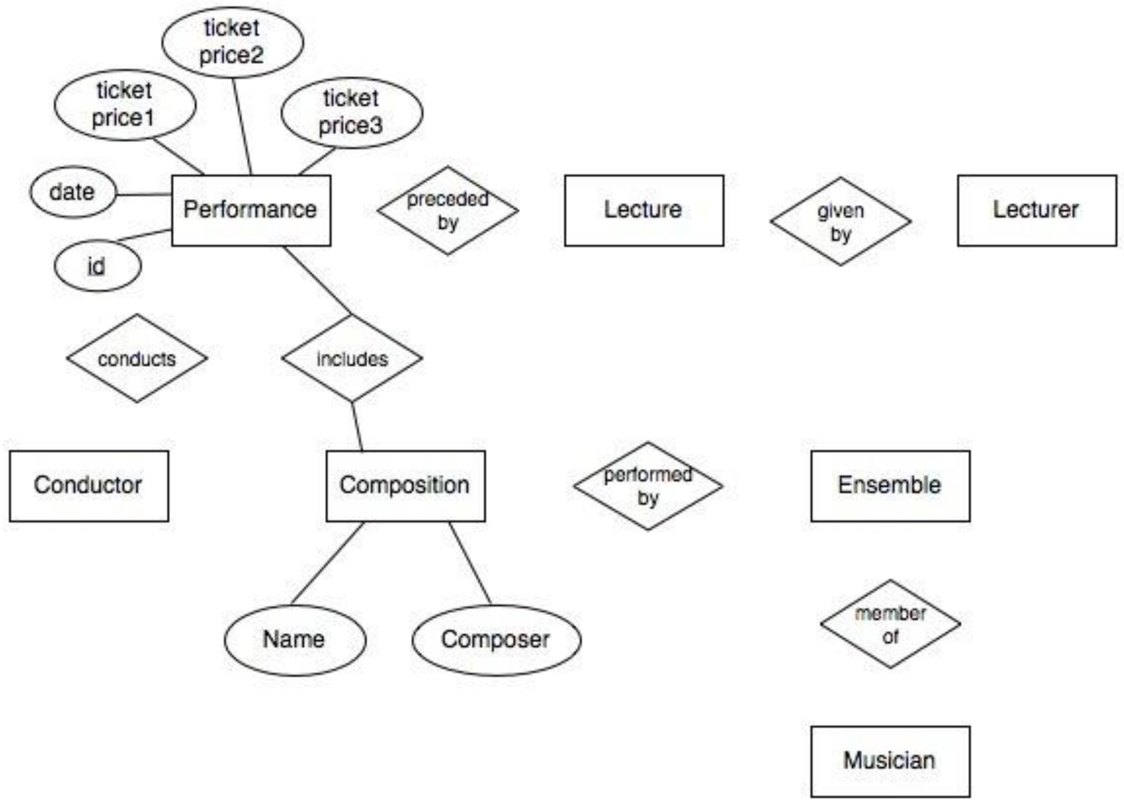
iii) All the above

iv) None of the above

Name____

3. Entity-Relationship Modeling

a. **[x points]** The following ER diagram models a database that is used by Zellerbach Hall to store information about the performances it hosts. The diagram contains all the entity sets, their attributes, and relationship sets you need to consider, but is missing lines connecting the relationships with the corresponding entities, and is missing key constraints. Also note that some entities *might be* weak entities, although if there are any, they are not (yet) drawn accordingly.



Express the following constraints, by drawing on the above picture:

i) [x points] A composition is comprised of a name (e.g. Symphony in C major) and a composer (e.g. Mozart). Be sure to label the key of this entity set!

ii) [x points] A *performance* may be preceded by a *lecture* analyzing the works that are going to be performed. A lecture must be given by one *lecturer*.

iii) [x points] A *performance* is always conducted by one *conductor*.

iv) [x points] A *performance* includes a number of *compositions*. For a particular *performance*, each *composition* is performed by one *ensemble* (orchestra). An *ensemble* is comprised of a number of *musicians*. Every *musician* may participate in one *ensemble*.

Name____

b. **[x points]** According to the above ER diagram, Zellerbach Hall is divided by default in three seating sections, each of which is associated with a particular price. The management decides to introduce a more flexible pricing policy, according to which the number of seating sections (and their pricing) may vary. What changes would you recommend to the above ER diagram to accommodate the management's wish to make more money?

c. **[x points]** The following DDL SQL statement creates the table to store the "includes" relationship from the original ER diagram. Fill in the missing details, so that it captures the constraints that the ER diagram represents.

```
CREATE TABLE includes (
   id INTEGER,
   name VARCHAR(20),



   PRIMARY KEY (                              ),



);
```

Name____

4. Buffer Management and Spatial Indexing

Suppose the following sequence of calls is presented to the Buffer Manager of a database:

1. get(1);
2. get(7);
3. pin(7);
4. get(3);
5. pin(3);
6. get(4);
7. get(5);
8. get(1);
9. get(4);
10. unpin(7);
11. get(3);
12. get(6);
13. pin(6);
14. get(2);
15. get(1);
16. get(1);
17. unpin(3);
18. get(2);
19. get(6);
20. get(2);
21. get(7);

The calls above have the following behavior:

- `get(RID)`: fetches the record identified by `RID` from the buffer, potentially retrieving it from disk as well if it is not already in the buffer.
- `pin(RID)`: ensures that the record `RID` stays in the buffer.
- `unpin(RID)`: permits the record `RID` to be evicted from the buffer.

Additionally, assume the following:

- The Buffer Manager has 4 buffers A, B, C and D and they are all initially empty.
- The calls `pin(RID)` and `unpin(RID)` cause the Buffer Manager to touch the buffer containing `RID`.
- If there are multiple free buffers for Buffer Manager to choose to assign to an `RID`, Buffer Manager chooses the first free buffer (for example, A before B if both are free).

For each of LRU, MRU and CLOCK, please indicate the **final buffer contents** and the **number of buffer misses** that occur in the table provided below:

	Final buffer contents (RIDs)				**# of buffer misses**
	A	B	C	D	
LRU					
	A	B	C	D	
MRU					
	A	B	C	D	
CLOCK					

Name____

Text Search
Suppose we were to build an inverted index on the following set of documents using a B+ tree:

ford.html
mustang
expedition
focus
fonefiftey

toyota.html
camry
prius
corolla
tacoma

honda.html
accord
civic

wanted.html
impala
corvette
mustang
civic

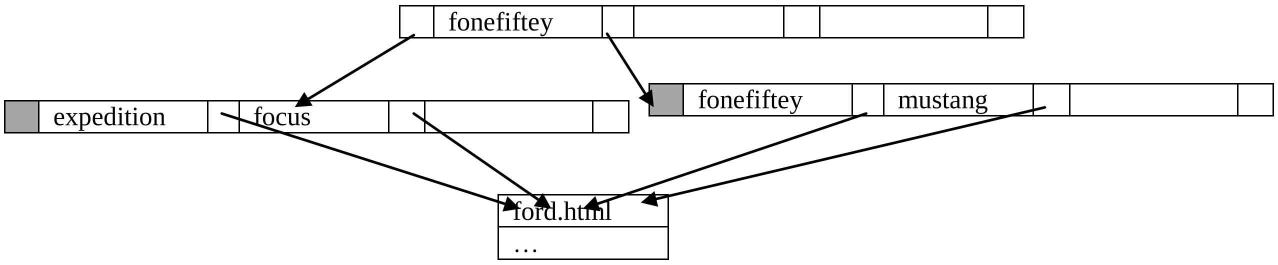
Additionally, assume that:

- The index is constructed by sequentially adding documents one at a time in this order (no bulk loading):
  (1)ford.html, (2) toyota.html, (3) honda.html , (4) wanted.html

- When considering x.html, its contents are considered sequentially. For example, for ford.html, the words are added one at a time in this order (no bulk loading):
  (a) mustang, (b) expedition, (c) focus, (d) fonefiftey

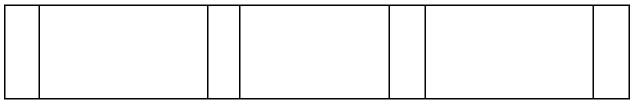
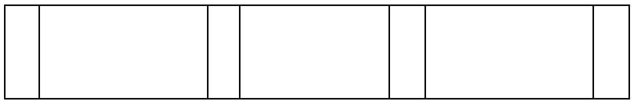
- The B+ Tree structure follows Alternative 2 where data entries are <key, rid> pairs. Note that records are webpages.

After ford.html's contents are added, the inverted index looks like this:



a) **[1 points]** What is the **order** of this B+ tree?

b) **[8 points]** Fill in the following table:

After processing html page:	**Contents of root index node**	**Maximum number of keys that can be inserted without splitting any nodes**
ford.html	fonefiftey	
toyota.html		
honda.html		
wanted.html		

c) **[4 points]** After **wanted.html** is processed, how many **index and data** nodes are accessed to answer each of the following queries? Assume a buffer size big enough to hold 1000 nodes.

Keywords in query	Number of nodes accessed
impala	
impala AND fonefiftey	
civic	
eclass	

Name____