Department of Electrical Engineering and Computer Science

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

6.830 Database Systems: Fall 2008 Quiz I

There are 17 questions and 10 pages in this quiz booklet. To receive credit for a question, answer it according to the instructions given. *You can receive partial credit on questions.* You have **80 minutes** to answer the questions.

Write your name on this cover sheet AND at the bottom of each page of this booklet.

Some questions may be harder than others. Attack them in the order that allows you to make the most progress. If you find a question ambiguous, be sure to write down any assumptions you make. Be neat. If we can't understand your answer, we can't give you credit!

THIS IS AN OPEN BOOK, OPEN NOTES QUIZ.
NO PHONES, NO LAPTOPS, NO PDAS, ETC.

Do not write in the boxes below

1-4 (xx/20)	5-7 (xx/16)	8-10 (xx/20)	11-13 (xx/21)	14-17 (xx/23)	Total (xx/100)

Name:

I Short Answer

1. [6 points]: To reduce the number of plans the query optimizer must consider, the Selinger Optimizer employs a number of heuristics to reduce the search space. List three:
(Fill in the blanks below.)

A.

B.

C.

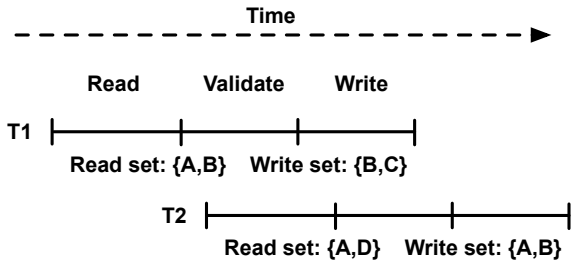
2. [4 points]: Give one reason why the REDO pass of ARIES *must* use physical logging.
(Write your answer in the space below.)

Name:

II Optimistic Concurrency Control

For the following transaction schedules, indicate which transactions would commit and which would abort when run using the parallel validation scheme described in the Kung and Robinson paper on Optimistic Concurrency Control. Also give a brief justification for your answer. You may assume that if a transaction aborts, it does not execute its write phase, rolling back instantly at the end of the validation phase.

3. [4 points]:



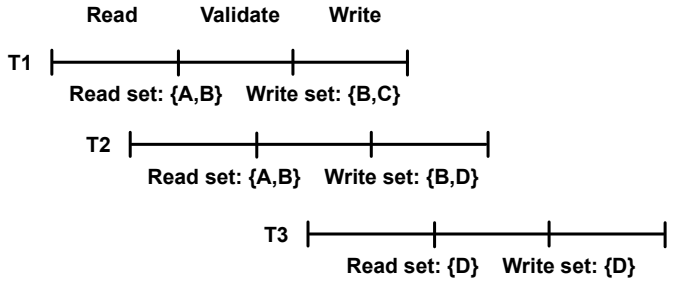
(Write your answer in the spaces below.)

Transactions that commit: _____

Transactions that abort: _____

Justification:

4. [6 points]:



(Write your answer in the spaces below.)

Transactions that commit: _____

Transactions that abort:_____

Justification:

Name:

III Schema Design

Consider a relational table:

```
Professor(
   professor_name, professor_id,
   professor_office_id, student_id,
   student_name, student_office_id,
   student_designated_refrigerator_id, refrigerator_owner_id,
   refrigerator_id, refrigerator_size, secretary_name,
   secretary_id, secretary_office )
```

Suppose the data has the following properties:

A. Professors and secretaries have individual offices, students share offices.

B. Students can work for multiple professors.

C. Refrigerators are owned by one professor.

D. Professors can own multiple refrigerators.

E. Students can only use one refrigerator.

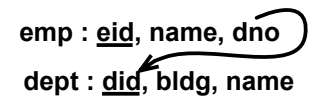
F. The refrigerator the student uses must be owned by one of the professors they work for.

G. Secretaries can work for multiple professors.

H. Professors only have a single secretary.

5. [10 points]: Put this table into 3rd normal form by writing out the decomposed tables; designate keys in your tables by underlining them. Designate foreign keys by drawing an arrow from a foreign key to the primary key it refers to (see example below.) Note that some of the properties listed above may not be enforced (i.e., guaranteed to be true) by a 3NF decomposition.

(Write your answer in the space below.)



Example decomposition of emp/dept tables showing primary and foreign keys.

Name:

6. [3 points]: Which of the eight properties (A–H) of the data are enforced (i.e., guaranteed to be true) by the 3NF decomposition and primary and foreign keys you gave above?

(Write your answer in the space below.)

7. [3 points]: What could a database administrator do to make sure the properties not explicitly enforced by your schema are enforced by the database?

(Write your answer in the space below.)

IV External Aggregation

Suppose you are implementing an aggregation operator for a database system, and you need to support aggregation over very large data sets with more groups than can fit into memory.

Your friend Johnny Join begins writing out how he would implement the algorithm. Unfortunately, he stayed up all night working on Lab 3 and so trails off in an incoherent stream of obscenities and drool before finishing his code. His algorithm begins as follows:

input : Table T, aggregation function $fname$, aggregation field $aggf$, group by field $gbyf$
`/* Assume` T `has` $|T|$ `pages and your system has` $|M|$ `pages of memory */`

`/* Partition phase */`
$n \leftarrow \left\lceil \frac{|T|}{|M|} \right\rceil$;
Allocate $bufs$, n pages of memory for output buffers ;
Allocate $files$, an array of n files for partitions ;
foreach record $r \in T$ **do**
  $h \leftarrow hash(r.gbyf)$; /* $h \in [1 \ldots n]$ */
  Write r into page $bufs[h]$;
  if page $bufs[h]$ is full **then**
    Add $bufs[h]$ to file $files[h]$, and set the contents of $bufs[h]$ to empty ;
  end
end

`/* Aggregate phase */`
foreach $f \in files$ **do**
  `/* Your code goes here. */`
end

Name:

8. [6 points]: Relative to $|T|$, how big must $|M|$ be for the algorithm to function correctly? Justify your answer with a sentence or brief analytical argument.

(Write your answer in the space below.)

9. [8 points]: Assuming the aggregate phase does only one pass over the data, and that *fname* = 'AVG', describe briefly what should go in the body of the for loop (in place of "Your code goes here"). You may write pseudocode or a few short sentences. Suppose the system provides a function `emit(`*group*`,`*val*`)` to output the value of a group.

(Write your answer in the space below.)

10. [6 points]: Describe an alternative to this hashing-based algorithm. Your answer shouldn't require more than a sentence or two.

(Write your answer in the space below.)

Name:

V Cost Estimation

Suppose you are given a database with the following tables and sizes, and that each data page holds 100 tuples, that both leaf and non-leaf B+Tree pages are dense-packed and hold 100 keys, and that you have 102 pages of memory. Assume that the buffer pool is managed as described in the DBMIN Paper ("An Evaluation of Buffer Management Strategies for Relational Database Systems.", VLDB 1985.)

Table	Size, in pages
T_1	100
T_2	1000
T_3	5000

11. [15 points]: Estimate the *minimum* number of I/Os required for the following join operations. Ignore the difference between random and sequential I/O. Assume that B+Trees store pointers to records in heap files in their leaves (i.e., B+Tree pages only store keys, not tuples.)

(Write your answers in the spaces below. Justify each answer with a short analytical argument.)

- Nested loops join between T_1 and T_2, no indices.

- Grace hash join between T_2 and T_3, no indices.

- Index nested loops join between a foreign key of T_2 and the primary key of T_3, with a B+Tree index on T_3 and no index on T_2.

Name:

VI ARIES with CLRs

Suppose you are given the following log file.

LSN	TID	PrevLsn	Type	Data
1	1	-	SOT	-
2	1	1	UP	A
3	1	2	UP	B
4	2	-	SOT	-
5	2	4	UP	C
6	-	-	CP	dirty, trans
7	3	-	SOT	-
8	2	5	UP	D
9	3	7	UP	E
10	1	3	COMMIT	-
11	2	8	UP	B
12	2	8	CLR	B
13	3	7	CLR	E

12. [2 points]: After recovery, which transactions will be committed and which will be aborted?
(Write your answers in the spaces below)

Committed:_____

Aborted:_____

13. [4 points]: Suppose the dirty page table in the CP record has only page A in it. At what LSN will the REDO pass of recovery begin?
(Write your answer in the space below)

LSN:_____

Name:

14. [4 points]: Again, suppose the dirty page table in the CP record has only page A in it. What pages may be written during the REDO pass of recovery?

(Write your answer in the space below)

Pages:_____

15. [4 points]: Once again, suppose the dirty page table in the CP record has only page A in it. What pages may be written during the UNDO pass of recovery?

(Write your answer in the space below)

Pages:_____

VII Snapshot Isolation

Oracle and Postgres both use a form of transaction isolation called snapshot isolation. One possible implementation of snapshot isolation is as follows:

- Every object (e.g., tuple or page) in the database has a timestamp; multiple copies ("versions") of objects with old timestamps are kept until no transaction needs to read them again. (For this question, you don't need to worry about how such old versions are maintained or discarded.)
- When a transaction begins, the system records the transaction's start time stamp, ts_s. Timestamps are monotonically increasing, such that no two transactions have the same timestamp value.
- When a transaction T writes an object O, it adds the new version of the O to T's *local write set*. Versions in the local write set are not read by other transactions until after T has committed.
- When a transaction T reads an object O, it reads the most recent committed version with timestamp $\leq ts_s$, reading O from T's own local write set if O has been previously written by T.
- When a transaction T commits, a new timestamp, ts_c is taken. For every object O in T's local write set, if the most recent version of O in the database has timestamp $\leq ts_s$, then O is written into the database with timestamp ts_c. Otherwise, T aborts. Only one transaction commits at a time.

For example, consider the following schedule:

Initial database: objects A and B, both version 0 (denoted A_0 and B_0)

$T1(ts_s = 1)$	$T2(ts_s = 2)$
$Read(A_0)$	
$Write(A)$	
	$Read(A_0)$
	$Write(A)$
commit ($ts_c = 3$)	
install A_3	
	attempt to commit with $ts_c = 4$,
	but abort, because last version of A (3) > $ts_s = 2$

Here, $T2$ aborts because it tried to write A concurrently with T1.

16. [10 points]: Is snapshot isolation conflict serializable? If yes, state briefly why. If not, give an example of a non-serializable schedule.

(Write your answer in the space below.)

17. [5 points]: Oracle claims that snapshot isolation is much faster than traditional concurrency control. Why?

(Write your answer in the space below.)

End of Quiz I

Name:

6.830 / 6.814 Database Systems

Fall 2010