CS 245
Final Exam – Winter 2004

This exam is open book and notes. You have 70 minutes to complete it.

Print your name:____

The Honor Code is an undertaking of the students, individually and collectively:

1. that they will not give or receive aid in examinations; that they will not give or receive unpermitted aid in class work, in the preparation of reports, or in any other work that is to be used by the instructor as the basis of grading;
2. that they will do their share and take an active part in seeing to it that others as well as themselves uphold the spirit and letter of the Honor Code.

The faculty on its part manifests its confidence in the honor of its students by refraining from proctoring examinations and from taking unusual and unreasonable precautions to prevent the forms of dishonesty mentioned above. The faculty will also avoid, as far as practicable, academic procedures that create temptations to violate the Honor Code.

While the faculty alone has the right and obligation to set academic requirements, the students and faculty will work together to establish optimal conditions for honorable academic work.

I acknowledge and accept the Honor Code.

Signed:____

Problem	Points	Maximum
1		10
2		10
3		10
4		10
5		10
6		10
7		10
8		10
Total		80

Problem 1 (10 points)

Consider the following market basket data, where interegrs 1, 2, ... represent particular items bought:

a	1	2	3	4
b	1	2	5	3
c	1	6	3	2
d	1	6	7	8
e	6	1	7	8

For example, the first line represent a trasnaction a where items 1, 2, 3 and 4 were bought.

(a) List all item pairs with support greater than or equal to 3.

ANSWER:

(b) List all item triples with support greater than or equal to 3.

ANSWER:

(c) What is the confidence for the association rule $\{1,2\} \Rightarrow 5$?

ANSWER: ________________

Problem 2 (10 points)

Consider the following containment hierarchy, to be used for multi-granularity two-phase locking:

- R: Root, with children X_1, X_2,... .
- X_1: Has chidren A_1, A_2, A_3, ...
- X_2: Has chidren B_1, B_2, ...

Assume the following trasaction run, in this order:

- T_1: Reads all X_1 children.
- T_2: Writes B_1
- T_3: Writes A_1
- T_4: Reads A_1
- T_5: Writes B_2
- T_6: Reads all X_2 children, writes B_2

Trasnaction T_1 starts, requests all the S, X, IX, IS, SIX locks it needs to access the data it needs, and then holds on to the locks. Next, T_2 starts and requests its locks. If a lock is not available, T_2 waits; else T_2 gets all its locks and does not release any. Then, T_3 runs in a similar fashion, and so on.

In the table below, show the sequence of locks executed by the trasnactions, as they execute in the order T_1, T_2, ...T_6. Remember that no locks are released in the scenario we are looking at. In the “Tr” position, put the trasnaction id; in the “Ob” write the name of the object (e.g., X_1, A_1,...); in the “Lk” position, put the type of lock requested (e.g., IS, S,...); in the “Gr” position write “Y” if the lock is granted, “F” otherwise.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Tr:															
Ob:															
Lk:															
Gr:															

Problem 3 (10 points)

The following table describes the undo/redo log found on a system after a failure.

LSN	Action type	Transaction id	Other info
1	trans start	T_1	
2	db write	T_1	
3	trans start	T_2	
4	db write	T_2	
5	start dump	System	T_1 active
6	trans start	T_3	
7	db write	T_3	
8	end dump	System	
9	db write	T_3	
10	trans start	T_4	
11	db write	T_4	
12	db write	T_3	
13	commit	T_4	
14	commit	T_2	
15	start checkpoint	System	T_1, T_3 active
16	db write	T_1	
17	db write	T_3	
18	trans start	T_5	
19	db write	T_5	
20	trans start	T_6	
21	db write	T_6	
22	end checkpoint	System	
23	db write	T_5	
24	db write	T_3	
25	db write	T_1	
26	commit	T_5	
27	db write	T_6	
28	commit	T_3	
29	end of log		

The database dump that occurs between actions 5 and 8 writes a complete copy of the database to tape, to be used in case of media failure.

(a) Assume that at action 29 the system crashes but the database is *not* lost. Assume we use the two-pass algorithm described in class notes # 8, slide 50. Each write action logs physical before and after values. In the two lines below indicate what actions are undone in the first pass, and what action are redone. Write the actions *in the order* they would be undone or redone (left to right). Identify each action by the LSN given in the first column in the table. Also list all new log records that are created during the recovery process (if any).

ACTIONS TO BE UNDONE: ____________________

ACTIONS TO BE REDONE: ____________________

LOG RECORDS CREATED: ____________________

(b) Now assume that the database is lost at action 29. (All other assumptions from part (a) remain the same.) In this case we load the saved database from tape and bring it up to date. In the three lines below again indicate what actions are undone and redone to bring the dumped database up to date, as well as any log records generated during recovery.

ACTIONS TO BE UNDONE: ____________________

ACTIONS TO BE REDONE: ____________________

LOG RECORDS CREATED: ____________________

(c) Now assume that the database is *not* lost at action 29. But instead of the scenario of part (a), the log contains *logical* undo and redo actions. We now use the recovery algorithm of Notes #10, slides 51 and 52.

Assume that ALL actions with LSN less than or equal to 23 have been reflected to disk, and than NO writes with LSN greater than or equal to 24 have been reflected on disk.

In the three lines below indicate what actions are undone in the first pass, what action are redone, and any log records generated during recovery. Write the actions *in the order* they would be undone or redone (left to right). Identify each action by the LSN given in the first column in the table.

ACTIONS TO BE UNDONE: ____________________________________

ACTIONS TO BE REDONE: ____________________________________

LOG RECORDS CREATED: ____________________________________

Problem 4 (10 points)

Suppose that we run the following transactions using the validation protocol. Assume that the validation protocol has been extended, so that the condition $T_i \notin$ FIN includes the condition that the write sets involved do not intersect. (The version in the textbook already includes this optimization.) The following table lists the transactions involved, together with their read and write sets:

Transaction	Read Set	Write Set
T_1	$\{a, b\}$	$\{b, c\}$
T_2	$\{a, b\}$	$\{d\}$
T_3	$\{c, d\}$	$\{e\}$
T_4	$\{a\}$	$\{c, f\}$
T_5	$\{f\}$	$\{c\}$
T_6	$\{d\}$	$\{g\}$

The following sequence of events takes place. (No other transactions run before or concurrently with T_1 ... T_6.)

1. T_1, T_2, T_3, T_4 start (in this order)
2. T_1 initiates validation
3. T_2 initiates validation
4. T_3 initiates validation
5. T_5 starts
6. T_4 initiates validation
7. T_1, T_2 finish (if they were not aborted earlier)
8. T_5 initiates validation
9. T_6 starts
10. T_3, T_4 finish (if they were not aborted earlier)
11. T_6 initiates validation
12. T_5, T_6 finish (if they were not aborted earlier)

In the table on the next page, please write down the outcome of each transaction, either COMMITTED if it was successful at validation, or ABORTED if it was not.

Transaction	Outcome
T_1	
T_2	
T_3	
T_4	
T_5	
T_6	

Problem 5 (10 points)

Please answer the following questions:

(a) A particular $B+$ tree non-leaf node has 48 pointers. How many keys will that node contain?

ANSWER:________________

(b) A particular $B+$ tree leaf node has 48 pointers. How many keys will that node contain?

ANSWER:________________

(c) The "containment of value sets" rule is used to estimate what?

ANSWER:________________

(d) Consider an linear hashing structure with $i = 10$ and $m = 903$, and no overflow buckets. How many disk buckets are actually allocated at this point?

ANSWER:________________

(e) Consider an extensible hashing structure with $i = 4$ and 11 actual disk blocks allocated. (Assume directory is entirely in memory.) Within each of these 11 blocks, we have a counter i_j that records the "level" of the block, in general with values between 1 and $i = 4$. *At most* how many of the 11 blocks may have a counter i_j *different* from 4?

ANSWER:________________

(f) Consider a file with 2000 fixed-size records, each 300 bytes long. This file is to be stored on 2K disk blocks, with no spanning of records. How many blocks will be needed?

ANSWER:________________

(g) Consider a disk with a 5 inch diameter, 10,000 RPM speed, 32 surfaces, 8192 cylinders, average seek time of 10ms, 4KB blocks, 100GB usable capacity, and 10% overhead for gaps. How many tracks (total over the entire disk) does this disk have?

ANSWER:________________

(h) Consider the same disk (5 inch diameter, 10,000 RPM speed, 32 surfaces, 8192 cylinders, average seek time of 10ms, 4KB blocks, 100GB usable capacity, and 10% overhead for gaps). What is the average rotational delay to read a block, assuming we always need to find the start of the track?

ANSWER:________________

(i) What is the burst bandwidth (in GB/s) for the same disk (5 inch diameter, 10,000 RPM speed, 32 surfaces, 8192 cylinders, average seek time of 10ms, 4KB blocks, 100GB usable capacity, and 10% overhead for gaps)?

ANSWER:________________

(j) Given the following schedule:
$S = r_1(A); r_2(B); r_3(C); w_2(C); r_3(B); r_1(C); r_3(A); w_1(A); w_3(D)$
If possible, give an equivalent serial schedule to schedule S. If not, state which actions are conflicting.

ANSWER:________________

Problem 6 (10 points)

In an attempt to thwart terrorism the Department of Homeland Security (DHS) acquires access to the database containing all sales of materials by the Russian government or army in that last 10 years. They wish to cross-check this database with their own database containing individuals and their suspected affiliations to see if anyone connected to Al-Qaeda has purchased materials for making Weapons of Mass Destruction (WMD).

The Russian database schema is as follows:

Sales(item, customer, price, WMD-related)

and has the following statistics

- *T(Sales)* = 10,000,000;
- *V(Sales,item)* = 1000;
- *V(Sales,customer)* = 6,000,000;
- *V(Sales,price)* = 40,000;
- *DOM(Sales,WMD-related)* = 2 (i.e. yes/no)

The DHS database schema is as follows:

Suspects(name, address, nationality, organization)

and has the following statistics

- *T(Suspects)* = 300,000,000;
- *V(Suspects,name)* = *V(Suspects,address)* = 200,000,000;
- *V(Suspects,nationality)* = 50;
- *V(Suspects,organization)* = 20

(Note: Only one organization is listed per tuple. If a suspect is a member of multiple organizations they will be listed multiple times in the database. This is the only reason for names to be duplicated in the database.)

(a) Give an efficient logical query plan (in tree form) for a query which would return the names and addresses of all people suspected of being affiliated with organization Al-Qaeda that were customers for WMD-related materials from Russia (name and customer fields must match).

(b) Assuming containment of value sets, how many tuples do you expect to be output by the query?

ANSWER:________________

(c) Assume all 10,000,000 items sold by Russia were WMD-related, but only 200,000 suspects have connections to Al-Qaeda. (These assumptions are only for this part of the problem.) How many I/Os (not including final output) do we expect to perform in answering our query, assuming the following:

- We use a hash-join,
- Sales records are 20 bytes each. Suspects records are 100 bytes each,
- each memory/disk block has 4000 available bytes (after removing headers and such),
- the database system can hold 101 blocks in memory?

ANSWER:________________

Problem 7 (10 points)

You have implemented an algorithm for association rule data mining, to find 2-sets. To find the high support 2-sets, you use a hash table with 100 counters. Each occurring pair is hashed, and the counter is incremented. As discussed in class, this scheme may produce false positives, so a second pass is required to discard the false positives.

On a particular run of your algorithm, you have a table with 900 transactions, each of the form (t, p_1, p_2). (Here t is the transaction id, and p_1 and p_2 are the two products bought together.) In the table there are 150 unique pairs occuring. In particular:

- There are 50 pairs that occur in 10 different transactions.
- There are 100 pairs that occur in 4 transactions.

The threshold is use is 9. Thus, the algorithm should eventually identify the 50 pairs that occur 10 times.

The hash function you use is such that:

- The probability that only one unique pair hashes to a given bucket (counter) is 0.5.
- The probability that 2 different pairs hash to the same bucket is 0.5.
- All other cases occur with negligible probability.

How many false positives will be generated by the algorithm in the first phase? Give your answer as the expected number of unique pairs that are false positives. Please explain your answer.

NUMBER OF FALSE POSITIVES: ____________________

Problem 8 (10 points)

For this problem, consider hash indexes with the following characteristics:

- A block can hold 50 value/pointer pairs. (The hash structure does not contain the records themselves, only pointers to them.)
- The file being indexed has 1000 records.
- The indexed field can contain duplicates.

(a) For a linear hash index with an average occupancy (utilization) of 50%, how many blocks will the index require in the worst-case?

ANSWER:________________

(b) In case (a), what is the worst-case number of I/Os to look up a value (including the record itself)?

ANSWER:________________

(c) Consider an extensible hash index with a hash function that produces $b = 10$ bits. What is the worst-case (largest possible) size of the directory? For parts (c) and (d), assume the following strategy: If we want to split a bucket but we have already used all $i = b$ bits produced by the hash function, and there is no way to split the bucket further, we use an overflow bucket. In all other cases we split the bucket, even if all keys in the original bucket remain in one of the two new buckets (and hence one of the two new buckets requires further splitting).

ANSWER:________________

(d) For an extensible hash index with $b = 10$, what is the best-case (smallest possible) size of the directory?

ANSWER:________________