CS432 Final Exam. **NAME: ____**

December 14, 2001

Start Time: 6:00pm

End Time: 8:30pm **Cornell NETID: ____________________________________**

I herewith state that I will not share any information about this exam with anybody, and I state that I understand and will adhere to the Cornell Code of Academic Integrity.

--

Signature

Maximum number of points possible: 100. This exam counts for 15% of your overall grade. Questions vary in difficulty. Do not get stuck on one question. Good luck!

PART A. Queries (10 points total)

A.1) Consider the following schema (keys are underlined):

Product(<u>pid</u>, name, type, mfgr, price), Buys(<u>cid, pid</u>), Customer(<u>cid</u>, cname, age, gender)

a) Write the following query **in relational algebra**: "Find the names of all customers who have purchased all products manufactured by Sears." (5 points)

b) Write the following query **in SQL**: "Find the names of all customers who have not purchased the most expensive product." (5 points)

PART B. Indexing and Sorting (12 points total)

B.1) Indexing. Write down a sequence of five data insertions into an initially empty **linear** hashing index such that the final index structure has exactly one overflow bucket. You should also draw the final index structure (including the position of the next pointer). You can assume that a split occurs whenever an overflow bucket is created. Further, you can assume that each bucket can hold at most two data entries. (6 points)

B.2) External Sorting. Assume that you want to sort a file of size N pages. You have B buffer pages available. Assume that in the first pass, you read and write in blocks of B buffer pages. In subsequent passes, you read in blocks of k buffer pages and write out blocks of m buffer pages.

a) What is the cost (measured in number of I/Os) of sorting the file using the external merge sort algorithm? You just have to write down the formula. (1 point)

b) Explain the individual components of the formula as we did in class. (5 points)

PART C. Query Optimization and Normal Forms (18 points total)

C.1) Query Optimization. Consider the dynamic programming query-optimization algorithm that we discussed in class.

a) The algorithm only enumerates left-deep plans. Why? Give two different reasons. (4 points)

b) Consider a query that joins 4 relations. How many different plans will the dynamic programming algorithm enumerate in the worst case (including intermediate plans)? Your answer should be the sum of the number of 1-relation, 2-relation, 3-relation and 4-relation plans enumerated by the algorithm. Assume that there are no interesting sort orders, no indices on any of the relations, and that you always use the block nested loop join algorithm. Your answer should be a single number. (6 points)

C.2) Normalization. Consider a relation R with five attributes ABCDE. Now assume that R is decomposed into two smaller relations ABC and CDE. Define S to be the relation (ABC NaturalJoin CDE).

a) Assume that the above decomposition is lossless join, but not dependency preserving. You do not know any additional information about the decomposition. Which of the following statements can you infer to be *always* true: (1) R = S, (2) R $\subseteq$ S, (3) R $\subset$ S, (4) R $\supseteq$ S, (5) R $\supset$ S, (6) none of the above. List all true statements if more than one statement can be inferred to be true. (4 points)

b) Repeat the above question when the decomposition is dependency preserving, but not lossless join. (4 points)

PART D. Physical Database Design and Database Security (11 points total)

D.1) Illustrate with an example how views can be used to mask changes to the conceptual schema. Your example should include the original conceptual schema, the new conceptual schema, and a view definition that masks this change so that applications do not have to be rewritten. (6 points)

D.2) Illustrate with an example how views can be used for security. Your example should include a table, a view defined over the table, and a brief description of how the view “hides” some information from an unauthorized user. (5 points)

PART E: Concurrency Control (14 points total)

E.1) Consider the following transaction schedule, where time increases from left to right. (C stands for commit).

T1:			R(B)		R(A)			C	
T2:		R(A)		W(B)			C		
T3:	R(A)					W(A)			C

a) Draw the serializability graph for this schedule. Is this schedule conflict serializable? (4 points)

b) Is the above schedule possible using **two-phase locking**? If so, illustrate where in the schedule read/write locks can be acquired/released by T1, T2 and T3. If not, explain why a two-phase locking schedule is not possible. (5 points)

E.2) Consider the following locking protocol: Before a transaction T writes a data object A, T acquires an exclusive lock on A, and holds onto this lock till the end of the transaction. Before a transaction T reads a data object A, T acquires a shared lock on A, but releases the lock immediately after reading A. State which of the following properties are ensured by this locking protocol: (a) serializability, (b) conflict-serializability, (c) recoverability, (d) avoids cascading aborts, (e) avoids deadlock. (5 points)

Part F: Recovery (14 points total)

Consider the ARIES Recovery Algorithm as we discussed it in class.

F.1) When does ARIES force the log to disk? State the precise point, such as “after XXX happens”, or “before XXX happens”. (Hint: There is more than one occasion.) (4 points)

F.2) ARIES maintains in the dirty page table a recLSN for the page. ARIES also maintains on each page a pageLSN. Explain the difference between the recLSN and the pageLSN, and explain why we need both. Use at most four sentences. (5 points)

F.3) Assume that the buffer manager implements the steal and force policies, i.e., the buffer manager allows pages to be written to disk before transactions that have written to the page have committed (steal), and the buffer manager flushes all updates by a transaction to disk before the transaction commits (force). How would this change the structure of an ARIES update log record? Why? (5 points)

Part G: Parallel and Distributed Databases (12 points total)

G.1) Briefly describe three different ways of horizontally partitioning a relation across several processors in a parallel database system that uses the shared-nothing architecture. Give one advantage of each scheme over the other two schemes. Be brief in your answer. (5 points)

G.2) Consider the following variant of the presumed abort two-phase commit protocol for distributed database systems. On receiving a prepare message from the coordinator, a subordinate that want to commit sends a yes reply, but *does not* force-write a prepare record. The rationale behind this modification is that a subordinate can always contact the coordinator to find out all necessary details about the status of the transaction, and thus each subordinate does not have to log anything other than the final commit record.

Barring the above modification, the new protocol is the same as the presumed abort two-phase commit protocol in all other respects. Does this modified protocol guarantee the atomicity of distributed transactions? If so, briefly describe why coordinator or subordinate crashes do not affect atomicity. If not, give a counter-example. In either case, use an example with a coordinator and a single subordinate to illustrate your answer. (7 points)

Part H: Decision Support (9 points total)

H.1) Consider the relation R(ProductId, LocationId, Sales), where ProductId and LocationId are dimension attributes, and Sales is a measure attribute. Further, consider the following tuples stored in the relation R: (1, 10, 15), (1, 20, 43), (1, 30, 19), (2, 10, 90), (2, 30, 17), (3, 20, 12).

a) Draw the two-dimensional data cube for R. (4 points)

b) Write down the set of SQL group-by queries on R that can be used to compute the above data cube. (5 points)

This page will be used for grading your prelim. Do not write anything on this page.

SECTION	**QUESTION**	**SCORE**	**SECTION TOTAL**
Part A Queries	**A.1** (Max: 5 points)		(Max: 10 points)
	A.2 (Max: 5 points)		
Part B Indexing and Sorting	**B.1** (Max: 6 points)		(Max: 12 points)
	B.2-a (Max: 1 point)		
	B.2-b (Max: 5 points)		
Part C Query Optimization and Normal Forms	**C.1-a** (Max: 4 points)		(Max: 18 points)
	C.1-b (Max: 6 points)		
	C.2-a (Max: 4 points)		
	C.2-b (Max: 4 points)		
Part D Database Design and Security	**D.1** (Max: 6 points)		(Max: 11 points)
	D.2 (Max: 5 points)		
Part E Concurrency Control	**E.1-a** (Max: 4 points)		(Max: 14 points)
	E.1-b (Max: 5 points)		
	E.2 (Max: 5 points)		
Part F Recovery	**F.1** (Max: 4 points)		(Max: 14 points)
	F.2 (Max: 5 points)		
	F.3 (Max: 5 points)		
Part G Parallel & Distributed Databases	**G.1** (Max: 5 points)		(Max: 12 points)
	G.2 (Max: 7 points)		
Part H Decision Support	**H.1-a** (Max: 4 points)		(Max: 9 points)
	H.1-b (Max: 5 points)		
Total (Max: 100 points)			