MASTER OF COMPUTER SC. & ENGG. EXAM. - 2017
(1st Semester)
ADVANCED DATABASE SYSTEM CONCEPTS

Time: Three Hours **Full Marks: 100**

Answer any *five* questions.

1. (a) Why is an explicit transaction end statement needed in SQL but not an explicit begin statement? 2

(b) What is a recoverable schedule? Why is recoverability of schedules desirable? Are there any circumstances under which it would be desirable to allow nonrecoverable schedules? Explain your answer. 6

(c) Consider the following concurrent schedule involving the three transactions T1, T2 and T3.

Transaction 1	Transaction 2	Transaction 3
		read_item(Y);
		read_item(Z);
read_item(X);		
write_item(X);		
		write_item(Y);
		write_item(Z);
	read_item(Z);	
read_item(Y);		
write_item(Y);		
	read_item(Y);	
	write_item(Y);	
	read_item(X);	
	write_item(X);	

Construct the precedence graph for the above schedule and identify the equivalent serial schedule, if any. 6

(d) Explain why the read-committed isolation level ensures that schedules are cascade-free. 4

(e) Why is a serializable schedule considered correct? 2

2. (a) Most implementations of database systems use strict two-phase locking. Suggest three reasons for the popularity of this protocol. 5

(b) Under a modified version of the timestamp protocol, we require that a commit bit be tested to see whether a *read* request must wait. Explain how the commit bit can prevent cascading abort. Why is this test not necessary for *write* requests? 5

(c) Some index concurrency-control schemes use key-value locking on individual key-values, allowing other key-values to be inserted or deleted from the same leaf. Give example schedules to show that with *key-value* locking, if any of the lookup, insert or delete do not lock the next-key value, the phantom-phenomena could go undetected. 5

(d) When a transaction is rolled back under timestamp ordering, it is assigned a new timestamp. Why can it not simply keep its old timestamp? How multiversion timestamp ordering protocol ensures that a read request never fails and is never made to wait? 2 + 3

3. (a) Typical database systems employ steal and no-force approach. Explain why such database systems need to perform both *Undo* and *Redo* operations during the recovery process. In which order these two operations are performed and why? 5+2

(b) The idea behind deferred update is to defer or postpone any actual updates to the database on disk until the transaction completes its execution successfully and reaches its commit point. Explain why database system with deferred update need to perform *no Undo* and only *Redo* operations during the recovery process. 4

(c) Consider the following log records in the log file.

LSN	**prevLSN**	**transID**	**type**	**pageID**	**length**	**offset**	**before**	**after**
10	NULL	T1000	update	P500	3	21	ABC	DEF
20	NULL	T2000	update	P600	3	21	HIJ	KLM
30	20	T2000	update	P500	3	21	GDE	QRS
40	10	T1000	update	P505	3	21	TUV	WXY

Show the contents of the Transaction Table and the Dirty Page Table at the time when all the log records have been scanned after recovery. 6

(d) For each of the following requirements, identify the best choice of degree of durability in a remote backup system: 3

i. Data loss must be avoided but some loss of availability may be tolerated.

ii. Transaction commit must be accomplished quickly, even at the cost of loss of some committed transactions in a disaster.

iii. A high degree of availability and durability is required, but a longer running time for the transaction commit protocol is acceptable.

4. (a) Give an example where the *read one, write all available* approach leads to an erroneous state. 3

(b) Explain the difference between data replication in a distributed system and the maintenance of a remote backup site. 3

(c) Consider that a subtransaction in a global transaction does no updates. How the site in which the subtransaction is running will respond when the two phase commit protocol is executed by the transaction coordinator at the site where the global transaction was initiated? 5

(d) The persistent messaging scheme generally depends on timestamps combined with discarding of received messages if they are too old. Suggest an alternative scheme based on sequence numbers instead of timestamps. 5

(e) If a cloud data-storage service is used to store two relations r and s and we need to join r and s, why might it be useful, in terms of overall throughput, efficient use of space and response time to user queries, to maintain the join as a materialized view? 4

5. (a) Why is column-oriented storage potentially advantageous in a database system that supports a data warehouse? 5

(b) How the best split for an attribute is identified in the decision tree classifier? 5

(c) Apply the Apriori algorithm to the following data set to find all large itemsets containing three items:

Trans_id	Items_purchased
101	milk, bread, eggs
102	milk, juice
103	juice, butter
104	milk, bread, eggs
105	coffee, eggs
106	coffee
107	coffee, juice
108	milk, bread, cookies, eggs
109	cookies, butter
110	milk, bread

The set of items is {milk, bread, cookies, eggs, butter, coffee, juice}. Use 0.2 for the minimum support value. Show two rules that have a confidence of 0.7 or greater for an itemset containing three items. 10

6. (a) Describe the TF-IDF approach of measuring the relevance of a document to a query in an Information Retrieval system. 6

(b) Give the definition of *precision* and *recall* in a ranked list of results at position i. 4

(c) How is F-score defined as a metric of information retrieval? In what way does it account for both precision and recall? 4

(d) What is the basic idea behind the PageRank algorithm? Describe the algorithm. 6

7. (a) How does the concept of an object in object-oriented data model differ from the concept of an entity in the entity-relationship model? 4

(b) Consider a Library Information System with a relation *Books* with the following attributes: title, a list (array) of authors, Publisher with subfields *name* and *branch* and a set of keywords.

Define the above scheme in SQL with appropriate types for each attribute. 6

(c) What is nesting? Discuss with a suitable example. 5

(d) Define a type *Department* with a field *name* and a field *head* which is a reference of the type *Person*, with table *people* as scope. Now, insert a tuple in the *departments* table (type *Department*) with *name* ='CS' and a person with *name* 'BKS' as *head*. The table *people* has two attributes *name* and *person_id*. 5

8. (a) Suppose a system runs three types of transactions. Transactions of type A run at the rate of 50 per second, transactions of type B run at 100 per second, and transactions of type C run at 200 per second. Suppose the mix of transactions has 25% of type A, 25% of type B and 50% of type C. What is the average transaction throughput of the system, assuming there is no interference between the transactions? What factors may result in interference between the transactions of different types, leading to the calculated throughput being inaccurate? 4

(b) What are the three broad levels at which a database system can be tuned to improve performance? 4

(c) What is the motivation for splitting a long transaction into a series of small ones? What problems could arise as a result, and how can these problems be averted? 4

(d) Many database systems support built-in sequence counters that are not locked in two phase manner. Explain how such counters can improve concurrency. Explain why there may be gaps in the sequence numbers belonging to the final set of committed transactions. 4

(e) Discuss the role of materialized views in performance tuning of database systems. 4