EASTERN UNIVERSITY, SRI LANKA

DEPARTMENT OF MATHEMATICS

SPECIAL DEGREE EXAMINATION IN COMPUTER SCIENCE

2009/2010 (Sep./Oct., 2011)

CS 401: Advanced Database Management Systems

Answer all questions

This paper has 4 questions in a total of 5 pages

llowed: Three Hours

1 efficient transaction processing systems should provide high availability and fast response time
r hundreds of concurrent users.

a) *Describe* what is meant by *interleaved concurrency* using a suitable example. [10%]

b) *Describe* briefly the problems that could arise when the concurrency is uncontrolled. [10%]

c) *State* what is serialisable schedule and conflict serialisable schedule. [10%]

d) *State* the method to construct the precedence graph for a schedule and ***explain*** how a precedence graph can be used to analyse a schedule. [15%]

e) Consider the following schedule of four transactions T_1, T_2, T_3 and T_4:

Transaction T_1	Transaction T_2	Transaction T_3	Transaction T_4
read_item(X)			
			read_item(Y)
			Y:=Y+Y+100
			write_item(Y)
	read_item(Y)		
	Y:=Y+250		
	write_item(Y)		
		read_item(X)	
		X:=X+200	
		write_item(X)	
read_item(Y)			
Y:=Y-X			
write_item(Y)			
	read_item(Z)		
	Z:=Z-250		
	write_item(Z)		
		Z:=500	
		write_item(Z)	
			read_item(X)

i. ***Construct*** the precedence graph for the schedule given above. [15%]

ii. Considering only the order and type of operations, ***state*** whether the given schedule is

conflict serialisable or not. ***Justify*** your answer.

iii. Suppose the transaction T_4 is removed from the schedule discuss about the serialisa of the new schedule.

(f) ***Discuss*** about the relation between the serialisability of a schedule and concurrency.

2. The main challenge with concurrency and transactions is to preserve isolation *i.e.*, each transa operates as if it were the only one running on a database.

(a) ***State*** the basic two-phase locking(shared or Exclusive) protocol in detail.

(b) ***Explain*** how the two-phase locking protocol(shared or Exclusive) ensures isolation of a t action using a suitable example.

(c) Conversion of locks is one approach in which a read lock can be upgraded to a write lock a write lock can be down graded to a read lock.

i. ***Formulate*** rules that should be followed during a lock conversion.

ii. ***List*** the advantages and disadvantages of lock conversion.

(d) By considering the necessary conditions for a dead lock, ***show*** that the conservative 2P deadlock-free.

(e) ***State*** clearly the *Strict 2PL* and *Rigorous 2PL* protocols.

(f) Consider the following schedule:

Transaction T_1	Transaction T_2	Transaction T_3
read_item(X)		
read_item(Y)		
Y:=Y-X		
write_item(Y)		
Commit		
		read_item(X)
		read_item(Z)
		X:=X+Z
	read_item(Y)	
	Y:=Y+250	
	write_item(Y)	
	read_item(Z)	
	Z:=Z-250	
	write_item(Z)	
	Commit	
		write_item(X)
		Commit

i. ***Write*** the list of operations that would be performed and the outcome of the given sche under the strict 2PL.

ii. ***Write*** the list of operations that would be performed and the outcome of the given schedule under the rigorous 2PL. [10%]

iii. ***Discuss*** about the performance of the strict 2PL and rigorous 2PL protocols based on the list of operations obtained in parts (i) and (ii). [10%]

nestamp ordering is a popular protocol for concurrency control.

) ***Explain*** the *basic timestamp ordering protocol* in detail. [15%]

) Using a simple schedule,

show that the basic timestamp ordering can enforce concurrency control. [15%]

) *The basic timestamp ordering protocol cannot ensure recoverability.*

Do you agree with the above statement? ***Justify*** your answer using an example. [15%]

) One problem with basic timestamp ordering protocol is that *it may reject some operations unnecessarily*. Using a simple example ***show*** that this statement is true. [15%]

) *Wait-die* and *wound-wait* are two deadlock prevention schemes.

i. ***Write*** rules to implement the two schemes. [10%]

ii. Using the following schedule ***show*** that wait-die (or wound-wait) scheme prevents deadlock: [15%]

Transaction T_1	Transaction T_2	Transaction T_3
Begin		
read_item(X)		
read_item(Y)		
Y:=Y-X		
	Begin	
	read_item(Z)	
	Z:=Z-250	
		Begin
		read_item(X)
		read_item(Z)
		X:=X+Z
write_item(Y)		
Commit		
	write_item(Z)	
	read_item(X)	
	X:=X+250	
	write_item(X)	
		write_item(X)

iii. The schemes may cause some transactions to be aborted needlessly, even though they never cause a deadlock. ***Show*** that this statement is true by applying one of the sc to a suitable schedule.

4. A database system should keep information about the changes it makes, especially to recover a failure.

(a) ***Describe*** briefly the process of caching of disk pages in main memory when the DBMS req action on some data item.

(b) ***Explain*** briefly the deferred update and immediate update techniques.

(c) ***State*** what is meant by *steal/no-steal* approach.

(d) ***State*** what is meant by *force/no-force* approach.

(e) ***Describe*** the recovery technique based on deferred update. Your answer should includ idea, the relevant log entries and the recovery procedure.

(f) A DBMS uses immediate update technique for recovery. The DBMS is provided with a nu of slots in memory for caching where each slot can hold one disk page.

Suggest a suitable combination of disk management techniques (steal/no-steal and forc force) when the number of slots provided is low. ***Justify*** your answer.

(g) Consider the part of the schedule for the three transactions T_1, T_2 and T_3 given in pa Assuming that the DBMS uses a deferred update technique for recovery and strict two-p locking protocol for concurrency, answer the following regarding the schedule given above

i. ***Write*** the log entries that would be made during the execution of the transactions T and T_3. Note that the DBMS uses a strict 2PL protocol.

ii. Suppose the system crashes at the end of the schedule given above, suggest a suitable r ery procedure and ***write down*** the operations that would be performed during recov

iii. ***Discuss*** about the state of the database and propose measures to improve the effic of the DBMS.

Transaction T_1	Transaction T_2	Transaction T_3
Begin		
read_item(X)		
read_item(Y)		
Y:=Y-X		
		Begin
		read_item(X)
		read_item(Z)
	Begin	
	read_item(X)	
	X:=X-500	
	write_item(X)	
		Z:=X+Z
		write_item(Z)
		Commit
write_item(Y)		
	read_item(Y)	
	Y:=Y+250	
	write_item(Y)	
	Commit