K.L.E. Society's
B.V. Bhoomaraddi College of Engineering & Technology, Hubli
DEPARTMENT OF ISE

Minor: I **Semester: VI**
Course: Adv DBMS **Course Code: 08ISE624**
Date: 16/03/2011 **Time: 8:45 am- 10:00 am**

Answer any 2 full Questions. Each full question carries 20 marks

Q1.

a) Consider the three transactions T1, T2, &T3, and the schedules S1 & S2 given below
Draw the serializability (Precedence) graphs for S1 and S2, and state whether each schedule is serializable or not.
T1: r1(x); r1 (z); w1(x);
T2: r2 (z); r2(y); w2 (z); w2(y);
T3: r3(x); r3(y); w3(y);
S1: r1(x); r2 (z); r1 (z); r3(x); r3(y); w1(x); w3(y); r2(y); w2 (z); w2(y); **4marks**
S2: r1(x); r2 (z); r3(x); r1 (z); r2(y); r3(y); w1(x); w2 (z); w3(y); w2(y);

b) Briefly discuss the different variations of two-phase locking protocol? Why is rigorous two- phase locking often preferred.

8marks

c) What is the difference between the UNDO/REDO and UNDO/NO-REDO algorithms for recovery with immediate update? Write the procedures for recovery using immediate update in a single user environment **8marks**

Q2.

a) What are the problems that occur when concurrent transactions are executed in an uncontrolled manner? Give an example and explain.

9marks

b) Compare binary locks with exclusive/shared locks.

3marks

c) What are log sequence numbers (LSNs) in ARIES? What information does the dirty page table and transaction table contain? Describe how checkpointing is used in ARIES. **8marks**

Q3.

a) What is a recoverable schedule? Why is recoverability of schedules desirable? Are there any circumstances under which it would be desirable to allow non-recoverable schedules? Explain your answer **8marks**

b) Apply the time stamp ordering algorithm to the following schedule and determine whether the algorithm will allow the execution of the schedules? **6marks**

T1	T2	T3
		Read_item(Y) Read_item(Z)
Read_item (X) Write_item(X)		
		Write_item(Y) Write_item(Z)
	Read_item(Z)	
Read_itam (Y) Write_item(Y)		
	Read_item(Y) Write_item(Y) Read_item(X) Write_item(X)	

c) What is cascading rollback? Why is it avoided ? Give an example where cascading rollback is required**. 6marks**

K.L.E. Society's
B.V. Bhoomaraddi College of Engineering & Technology, Hubli
DEPARTMENT OF ISE

Minor: I **Semester: VI**
Course: Adv DBMS **Course Code: 08ISE624**
Date: 16/03/2011 **Time: 8:45 am- 10:00 am**

Scheme & Solution

Q1

a) S1: Serializable schedule

(2m)

S2: Nonserializable schedule

(2m)

(2+2=4m)

b) 4 variations of 2PL protocol
Basic: read and write lock precede the first unlock operation, need to explain two phases (expanding and shrinking)
Conservative: read-set and write-set
Strict: will not release any locks until T commits or aborts, its not deadlock free
Rigorous: guarantees strict schedule **(6m)**
Rigorous 2PL is preferred because its deadlock free, and guarantees strict schedule **(2m)**
(6+2=8m)

c) UNDO/NO-REDO
If all updates of transaction are recorded in the database on disk before T commits, then there is no need of REDO operation **(2m)**
UNDO/REDO
If T is allowed to commit before all its changes are written to disk
Procedures for REDO and UNDO **(6m)**
(6+2=8m)

Q2

a) 3 Problems occurs when transactions are executed in uncontrolled manner
i) Lost update problem explanation with example
ii) Temporary update problem explanation with example
iii) Incorrect summary problem explanation with example **(3*3=9m)**

b) Binary locks have 2 states locked & unlocked
Need to define lock and unlock operations
Shared/ Exclusive locks consists
Read_lock, Write_lock, Unlock operations **(3m)**

c) LSN indicates the address of the log record on disk each LSN correspond to specific change of some transaction and also data page stores the LSN of the latest log record**.** **(2m)**

Transaction table with explanation

T_id	Last_LSN	Status

(2m)

Dirty Page Table with explanation

Page_id	LSN

(2m)

Checkpointing: need to give explanation on
Begin_checkopoint, **(2m)**
End_checkpoint,
Special file **(2+2+2+2=8m)**

Q3

a) Recoverable schedule def: Consider a pair of transactions Ti and Tj such that Tj reads data items previously written by Ti, the commit operation of Ti appears before of commits operation of Tj. **(3m)**

Recoverable schedule are desirable because failure of transaction might bring the system into an irreversibly inconsistent state. **(2m)**

Nonrecoverable schedule needed because updates must be made visible early due to time constraints, even if they have not been committed, which is required for very long duration transactions. **(3m)**

(3+3+2=8m)

b) Initially TS(T1)=3, TS(T2)=7,TS(T3)=1
R(x)=0,R(y)=0,R(z)=0,W(x)=0,W(y)=0,W(z)=0;
T3: R(y)
R(x)=0,R(y)=1,R(z)=0,W(x)=0,W(y)=0,W(z)=0;
T3: R(z)
R(x)=0,R(y)=1,R(z)=1,W(x)=0,W(y)=0,W(z)=0;
T1: R(x)
R(x)=3,R(y)=1,R(z)=1,W(x)=0,W(y)=0,W(z)=0;
T1: W(x)
R(x)=3,R(y)=1,R(z)=1,W(x)=3,W(y)=0,W(z)=0;
T3: W(y)
R(x)=3,R(y)=1,R(z)=1,W(x)=3,W(y)=1,W(z)=0;
T3: W(z)
R(x)=3,R(y)=1,R(z)=1,W(x)=3,W(y)=1,W(z)=1;
T2: R(z)
R(x)=3,R(y)=1,R(z)=7,W(x)=3,W(y)=1,W(z)=1;
T1: R(y)
R(x)=3,R(y)=3,R(z)=7,W(x)=3,W(y)=1,W(z)=1;
T1: W(y)
R(x)=3,R(y)=3,R(z)=7,W(x)=3,W(y)=3,W(z)=1;
T2: R(y)
R(x)=3,R(y)=7,R(z)=7,W(x)=3,W(y)=3,W(z)=1;
T2: W(y)
R(x)=3,R(y)=7,R(z)=7,W(x)=3,W(y)=7,W(z)=1;
T2: R(x)

R(x)=7,R(y)=7,R(z)=7,W(x)=3,W(y)=7,W(z)=1;
T2: W(x)
R(x)=7,R(y)=7,R(z)=7,W(x)=7,W(y)=7,W(z)=1;
Result: Schedule executes successfully **(6m)**

c) Explanation of cascading rollback **(2m)**
Major Reasons why cascading rollback is avoided is time consuming and quite complex and does not ensure strict schedule **(4m)**
Example with explanation

(2+4=6m)