Computer Engineering Department
Faculty of Engineering - Cairo University

# Advanced Database Systems (CMP 401)
## Final Exam– 4th Year - Fall 2010
## (2 Hours) - *Total Marks: 70*

| Question 1 (25) | Question 2 (8) | Question 3 (14) | Question 4 (15) | Question 5 (8) | Total (70) |
|---|---|---|---|---|---|
| | | | | | |

**Question 1: [ 18 points]**

1. **[10 pts] Indicate whether the following questions are (True) or (False)**

   a. Real DMBSs continuously test for serializability of schedules to ensure DB consistency. ( )

   b. In the shared mode for Read/Write Locking, more than one Transaction can apply a shared lock on a data item for accessing its value. ( )

   c. In the exclusive mode for Read/Write Locking, only one write lock on a data item can exist at any point of time. ( )

   d. Using Binary or Read/Write lock guarantees the serializability of generated schedules. ( )

   e. A schedule is said to follow Two-phase locking protocol if all the locking operations of transactions involved in the schedule precede the first unlock operation in any of the transactions. ( )

   f. The Wait/Die scheme is a non-preemptive deadlock prevention scheme. ( )

   g. The read operation in Multi-version concurrency control protocol is never rejected. ( )

   h. The WAL protocol ensures that modified data items are force written to the DB before it is written in the Transaction Log. ( )

   i. The Cache Manager sets the Pin-bit to enable the deferred data update mechanism. ( )

   j. To minimize the recovery task in the deferred data update mechanism, checkpointing regularly force-writes all modified data buffers for committed and active transactions in the DB. ( )

2. **[15 pts] Choose the correct answer for the following questions.**

1. [1 pt] A transaction is said to follow the **Two-Phase Locking Protocol** if:
   a. All read_lock operations precede all the unlock operations
   b. All the read_lock and write_lock operations precede the first unlock operation
   c. All read_lock operations precede all the write_lock operations
   d. All write_lock operations precede all the read_lock operations

2. [1 pt] In Deadlock avoidance using **Wound-Wait** scheme:
   a. Older transaction is allowed to wait on younger transaction.
   b. Younger transaction requesting an item held by older transaction is aborted and restarted.
   c. Younger transaction is allowed to wait on older transaction.
   d. None of the above.

3. [1 pt] Shadow Update is defined as:
   a. As soon as a data item is modified in cache, the disk copy is updated.
   b. The modified version of a data item does not overwrite its disk copy but is written at a separate disk location.
   c. The disk version of the data item is overwritten by the cache version.
   d. All modified data items in the cache is written either after a transaction ends its execution or after a fixed number of transactions have completed their execution.

4. [1 pt] The No-Steal/No-Force method for flushing the database cache to disk implies the recovery method:
   a. No-Steal/Force (No-undo/No-redo)
   b. No-Steal/No-Force (No-undo/Redo)
   c. Steal/No-Force (Undo/Redo)
   d. Steal/Force (Undo/No-redo)

5. [1 pt] The Deferred Update recovery protocol performs
   a. Undo/Redo
   b. No-Undo/Redo
   c. Undo/No-Redo
   d. No-Undo/No-Redo

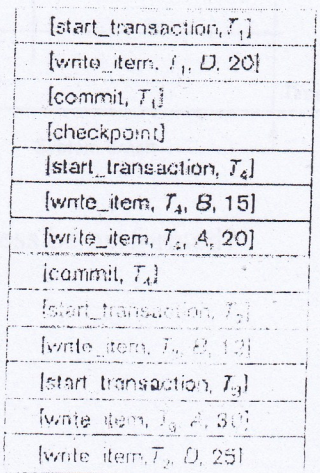
6. [4 pt] In the Multi-version concurrency control protocol, assume that item Y has 3 versions Y0, Y1, and Y2. Assume that there are 4 transactions and transaction T4 wants to write item Y. Given the following:
   RTS(Y0) = 1, WTS(Y0)=0, RTS(Y1) =2, WTS(Y1)=1, RTS(Y2)=7, WTS(Y2)=3, and TS(T4)=2

a. Transaction T4 is aborted
b. A new version of Y3 is created with RTS(Y3)=2 and WTS(Y3)=2
c. A new version Y3 is created with RTS(Y3)=2 and WTS(Y3)=3
d. Nothing happens

7. [4 pt] In the Timestamp based concurrency control protocol, assume that there are two transactions T1 and T2. Also assume that there is one data item (X) and transaction T2 wants to read it. Given the following:
TS(T1)=6, TS(T2)= 1, RTS(X)=6 and WTS(X)=6

a. RTS(X) is updated to be 1
b. RTS(X) and WTS(X) are updated to 6
c. T2 is aborted
d. T1 is aborted
e. Nothing happens

8. [2 pt] Using the following figure and assume a deferred update recovery protocol:
a. T1 should be redone
b. T2 should be redone
c. T3 should be redone
d. T4 should be redone



## Question 2: [ 8 points]

1) [8 pts] The following figures illustrate three transactions T1, T2 & T3, their System log at the point of crash and the timeline of Operations performed before the crash. Assume the generic case where updates may have been immediately recorded in the database.

1. (a)

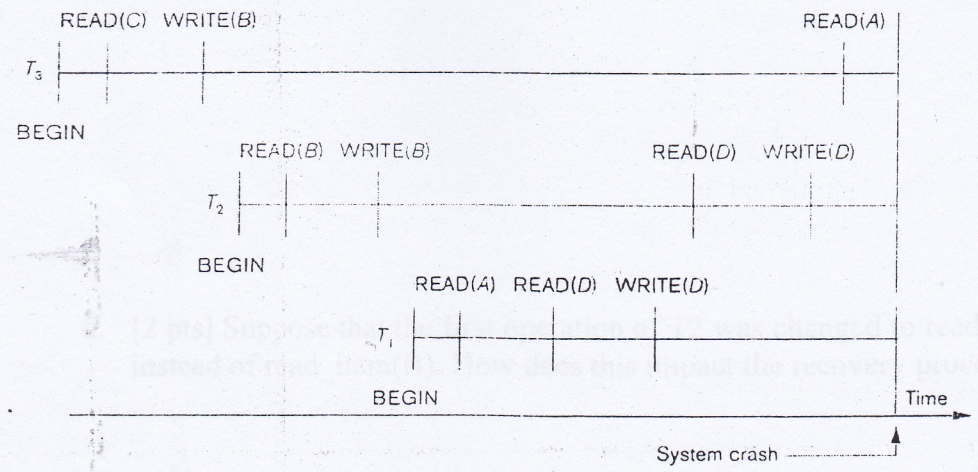
| $T_1$ | $T_2$ | $T_3$ |
|---|---|---|
| read_item(A) | read_item(B) | read_item(C) |
| read_item(D) | write_item(B) | write_item(B) |
| write_item(D) | read_item(D) | read_item(A) |
| | write_item(D) | write_item(A) |

(b)

| | A | B | C | D |
|---|---|---|---|---|
| | 30 | 15 | 40 | 20 |
| [start_transaction, $T_3$] | | | | |
| [read_item, $T_3$,C] | | | | |
| [write_item, $T_3$,B,15,12] | | 12 | | |
| [start_transaction, $T_2$] | | | | |
| [read_item, $T_2$,B] | | | | |
| ** [write_item, $T_2$,B,12,15] | | 15 | | |
| [start_transaction, $T_1$] | | | | |
| [read_item, $T_1$,A] | | | | |
| [read_item, $T_1$,D] | | | | |
| [write_item, $T_1$,D,20,25] | | | | 25 |
| [read_item, $T_2$,D] | | | | |
| ** [write_item, $T_2$,D,25,26] | | | | 26 |
| [read_item, $T_3$,A] | | | | |

System crash

(c)




a. [2 pts] Indicate how will the recovery process be performed?

**Question 3: [ 14 points]**

Consider a disk with block size B=512 bytes. A block pointer is P=6 bytes long, and a record pointer is PR =7 bytes long. A file has r=30,000 EMPLOYEE records of fixed-length. Each record has the following fields:

NAME (30 bytes),
SSN (9 bytes),
DEPARTMENTCODE (9 bytes),
ADDRESS (40 bytes),
PHONE (9 bytes),
BIRTHDATE (8 bytes),
SEX (1 byte),
JOBCODE (4 bytes),
SALARY (4 bytes).

An additional byte is used as a deletion marker.

(a) [2 pts] Calculate the record size R in bytes.

(b) [2 pts] Calculate the blocking factor bfr and the number of file blocks b assuming an unspanned organization.

(c) Suppose the file is ordered by the key field SSN and we want to construct a primary index on SSN. Calculate

i. [2 pts] the index blocking factor bfr i (which is also the index fan-out fo);

ii. [2 pts] the number of first-level index entries and the number of first-level index blocks;

iii. [2 pts] the number of levels needed if we make it into a multi-level index;

iv. [2 pts] the total number of blocks required by the multi-level index; and

v. [2 pts] the number of block accesses needed to search for and retrieve a record from the file--given its SSN value--using the primary index.

**Question 4: [15 points]**

1) [3 pts.] Differentiate between the Atomicity Concept in Operating Systems Concurrency Control versus Database Transactions.

2) [6 pts.] Explain clearly what problems arise with Uncontrolled Concurrent Execution of Transactions. Support your answer with examples.

3) [6 pts.] Consider the three transactions T1, T2, and T3, and the Schedule E given below. Draw the serializibility (precedence) graph for Schedule E. State whether this schedule is serializable or not. If a schedule is serializable, write down the equivalent serial schedule. If the schedule is not serializable then clearly indicate the cycles in the graph.

(a)

| Transaction $T_1$ |
|---|
| read_item($X$); |
| write_item($X$); |
| read_item($Y$); |
| write_item($Y$); |

| Transaction $T_2$ |
|---|
| read_item($Z$); |
| read_item($Y$); |
| write_item($Y$); |
| read_item($X$); |
| write_item($X$); |

| Transaction $T_3$ |
|---|
| read_item($Y$); |
| read_item($Z$); |
| write_item($Y$); |
| write_item($Z$); |

b)

Time

| Transaction $T_1$ | Transaction $T_2$ | Transaction $T_3$ |
|---|---|---|
| | read_item($Z$); | |
| | read_item($Y$); | |
| | write_item($Y$); | |
| | | read_item($Y$); |
| read_item($X$); | | read_item($Z$); |
| write_item($X$); | | |
| | | write_item($Y$); |
| | | write_item($Z$); |
| | read_item($X$); | |
| read_item($Y$); | | |
| write_item($Y$); | | |
| | write_item($X$); | |

Schedule E

**Question 5: [8 points]**

1) Consider the following two Transactions T1 & T2 with Timestamps of 10 & 20 respectively. The initial values of read and write Timestamps for all data items are assumed to be equal to zero (i.e. RTS = 0 & WTS = 0).

T1 (TS = 10)

```
A1 <- Read(X)
A1 <- A1 – k
Write(X, A1)
A2 <- Read(Y)
A2 <- A2 + k
Write(Y, A2)
```

T2 (TS = 20)

```
A1 <- Read(X)
A1 <- A1* 1.01
Write(X, A1)
A2 <- Read(Y)
A2 <- A2 * 1.01
Write(Y, A2)
```

Use the Basic Timestamp Ordering Algorithm to show whether the following schedule is going to be executed or aborted.

T1 (TS = 10)

```
A1 <- Read(X)
A1 <- A1 – k
Write(X, A1)
```

T2 (TS = 20)

```
A1 <- Read(X)
A1 <- A1* 1.01
Write(X, A1)
A2 <- Read(Y)
A2 <- A2 * 1.01
Write(Y, A2)
```

```
A2 <- Read(Y)
A2 <- A2 + k
Write(Y, A2)
```