RUBRIC SET - 1

CSE 202, Winter Session 2022

EndSem Exam

Marks: 35 **Time: 2 hrs**

Note: This exam has two sections. The Section-A carries 15 marks and it has 20 multiple choice questions. The first 10 questions are of 1 mark each and the next 10 questions 0.5 marks each. Section-B has 9 questions and the marks for each question are indicated.

Section-A

HariNagar Stock Exchange has decided to trade in international stocks/shares and achieve a target of 400 million transactions a month. You, as a database designer, need to make some choices of how you are going to handle this requirement. There are three tables that we consider – STOCK, TRADEREADY, and STOCK-TRADE-DEAL.

- Each STOCK will have StockID, Type, FirmName, QuantityAvailable, BasePrice, Country.
- Each TRADEREADY will have StockIDforTrade, TraderID, BuySell, QuantityForTrade, PriceQuoted.
- Each STOCK-TRADE-DEAL will have DealID, StockIDbeingTraded, SellingTraderID, BuyingTraderID, QuantityTraded, PriceDeal.

Based on this scenario answer the following questions by choosing the right option (only one answer correct)

Q1: If the following queriesare deemed to be frequent,

- SELECT DISTINCT SellingTraderID FROM STOCK-TRADE-DEAL WHERE StockIDbeingTraded = xyz
- SELECT SellingTraderID, QuantityTraded, PriceDeal FROM STOCK-TRADE-DEAL WHERE PriceDeal BETWEEN q AND r.

then on which attributes you will create the primary and secondary index of the data.

[✓ a.] Primary index on StockIDbeingTraded and Secondary index on PriceDeal
b. Primary index on StockIDbeingTraded and Secondary index on SellingTraderID
c. Primary index on SellingTraderID and Secondary index on PriceDeal
d. Primary index on PriceDeal and Secondary index on StockIDbeingTraded

Q2. Assume now that the HariNagar Stock Exchange has been implemented and during runtime many transactions are active. In this scenario answer the following questions on serializability and rollback. Here, the notation $tr_x(A)$ means transaction$_x$ reads item A, $tw_x(A)$ means transaction$_x$ writes item A, tc_x means transaction$_x$ commits.

For the schedule given below determine which all properties does this schedule have:

Schedule S = tr_1(PriceQuoted), tw_2(PriceQuoted), tr_1(QuantityForTrade), tc_1, tw_3(QuantityForTrade), tr_3(QuantityForTrade), tw_3(PriceQuoted), tc_3, tr_2(BuySell), tc_2

a. S is recoverable and conflict-serializable
b. S is recoverable and cascade-less
[✓ c.] S is recoverable, cascade-less and conflict-serializable
d. S is neither recoverable, nor cascade-less, nor conflict-serializable

Q3: For the above scenario if Schedule S1 was as follows then which all properties does this schedule have:

Schedule S1 = tr_1(PriceQuoted), tw_2(QuantityForTrade), tr_1(QuantityForTrade), tc_1, tc_2

a. S1 is only recoverable and cascade-less
[✓ b.] S1 is only conflict-serializable
c. S1 is recoverable and conflict-serializable
d. S1 is recoverable, cascade-less and conflict-serializable

Q4: In this given schedule, if Transaction 1 aborts during write of PriceDeal, then which other transactions need to be rolled back?

Schedule S2 = tw_1(PriceQuoted), tr_2(PriceQuoted), tw_1(QuantityForTrade), tr_3(QuantityForTrade), tr_4(BuySell), tw_1(PriceDeal)

[✓ a.] T_2 and T_3 rolls back
b. T_3 and T_4 rolls back
c. T_2, T_3, and T_4 rolls back
d. None of the above

Q5: In this given schedule, if Transaction 1 aborts during write of PriceDeal, then which other transactions need to be rolled back?

Schedule S2 = tw_1(PriceQuoted), tw_2(PriceQuoted), tw_1(QuantityForTrade), tr_3(QuantityForTrade), tr_4(PriceQuoted), tr_4(BuySell), tw_1(PriceDeal)

a. T_2 rolls back
b. T_2 and T_3 rolls back
c. T_4 rolls back
[✓ d.] T_3 rolls back

Q6: Consider the following transactions with data items P and Q initialized to zero:

```
T1 : read (P) ;
        read (Q) ;
        if P = 0 then Q : = Q + 1 ;
        write (Q).
T2 : read (Q) ;
        read (P)
        if Q = 0 then P : = P + 1 ;
        write (P).
```

Any non-serial interleaving of T_1 and T_2 for concurrent execution leads to

a. Serializable schedule
[✓ b. A schedule that is not conflict serializable]
c. A conflict serializable schedule
d. A schedule for which precedence graph cannot be drawn

Q7: Which of the four ACID principles are violated by a non-serializable schedule?

a. Atomicity
b. Consistency
[✓ c. Isolation]
d. Durability

Q8: Consider the schedules S1 and S2 given below.

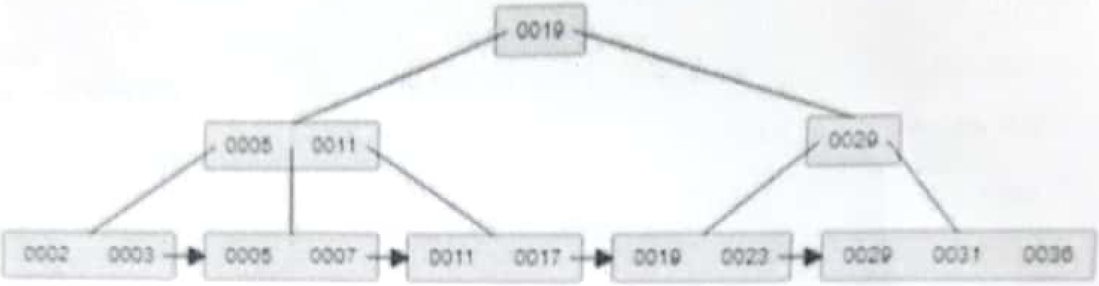
S1: r1(X); r3(Y); r3(X); r2(Y); r2(Z); w3(Y); w2(Z); r1(Z); w1(X); w1(Z)

S2: r1(X); r3(Y); r2(Y); r3(X); r1(Z); r2(Z); w3(Y); w1(X); w2(Z); w1(Z)

Which one of the following statements about the schedules is TRUE?

[✓ a. Only S1 is conflict serializable]
b. Only S2 is conflict serializable
c. Both S1 and S2 is conflict serializable
d. Neither S1 nor S2 is conflict serializable

Q9:



Consider the above B+ tree with n = 4. If we delete 11, the 2nd leaf node would look like:

a. 7, 17

b. 5, 7

[✓ c. 5, 7, 17]

d. 5

Q10: Consider the database with the below parameters:

Block Size = 500 bytes
Record Size = 50 bytes
Key Size = 5 Bytes
Pointer Size = 20 bytes

Given 28800 records, determine the number of dense index blocks?

[✓ a. 1440]

b. 2880

c. 720

d. None of the above

Q11: Which of the following is True.

[✓ a. Multi-level indexing reduces the block access from the disk.]

b. Multilevel indexing helps to fit the indexing tables in the main memory.

c. Multilevel indexing can access the index in one operation.

d. All of the above are true.

Q12: Which of these statements about recoverable schedules is true?

a. Every recoverable schedule is serializable

[✓ b. In a recoverable schedule, if a transaction T commits, then any other transaction that T read data from must also have committed]

c. In a recoverable schedule, no transaction will ever be aborted because a transaction that it read from aborted.
d. None of the above

Q13: Consider B+ tree in which the search key is 12 bytes long, block size is 1024 bytes, record pointer is 10 bytes long and block pointer is 8 bytes long. The maximum number of keys that can be accommodated in each non-leaf node of the tree is

a. 49
[✓ b. 50]
c. 51
d. 52

Q14: Which allows only committed data to be read and further requires that no other transaction is allowed to update it between two reads of a data item by a transaction.

a. Read committed
b. Serializable
[✓ c. Repeatable read]
d. Read uncommitted

Q15: Suppose a database system crashes again while recovering from a previous crash. Assume checkpoint is not done by the database either during transactions or during recovery. Which of the following is correct?

[✓ a. The same undo and redo list will be used while recovering again.]
b. The system cannot recover any further
c. All the transactions that are already undone and redone will be recovered again.
d. The database will become inconsistent.

Q16: Given the below schedule S,which among the options is not a possible serialized schedule for S

T1	T2	T3	T4
			R(A)
	R(A)		
		R(A)	
W(B)			
	W(A)		
		R(B)	
	W(B)		

a. $T_1 \rightarrow T_3 \rightarrow T_4 \rightarrow T_2$

b. $T_1 \rightarrow T_4 \rightarrow T_3 \rightarrow T_2$

[✓ c.] $T_4 \rightarrow T_3 \rightarrow T_1 \rightarrow T_2$

d. $T_4 \rightarrow T_1 \rightarrow T_3 \rightarrow T_2$

Q17: Suppose in the Hari Nagar Stock Exchange, any Seller can sell many different stocks/shares and any Buyer can buy/bid for many different stocks, and this data can be recorded at any time during the stock exchange operation. There are about 10 million sellers and buyers. You want to support the query for listing all stocks being traded by a Seller and listing all stocks being bought by a Buyer, such that the response is really fast. Which type of index will you create?

a. Hash index on SellingTraderID and another hash index on BuyingTraderID

b. Multi-level index on SellingTraderID and another multi-level index on BuyingTraderID

[✓ c.] Multi-level index on SellingTraderID with dense index for Stocks that seller is trading and another multi-level index on BuyingTraderID with dense index for Stocks that buyer is trading

d. Multi-level index on SellingTraderID with sparse index on Stocks that seller is trading and another multi-level index on BuyingTraderID with sparse index for Stocks that buyer is trading

Q18: Consider the following schedule:

T1	**T2**
R(X)	
R(Y)	
	W(Y)
?	

Here, R(X) stands for Read(X) and W(X) stands for Write(X). Replacing '?' by which of the following options will make the above schedule serializable?

[✓ a. R(X)]
b. R(Y)
c. W(Y)
d. None of the above

Q19: An index is clustered if

a. It is on a set of fields that forms candidate key
b. It is on a set of fields that includes primary key
[✓ c. The datarecords of the file are organized in the same order as the data entries of the index]
d. The datarecords of the file are not organized in the same order as the data entries of the index

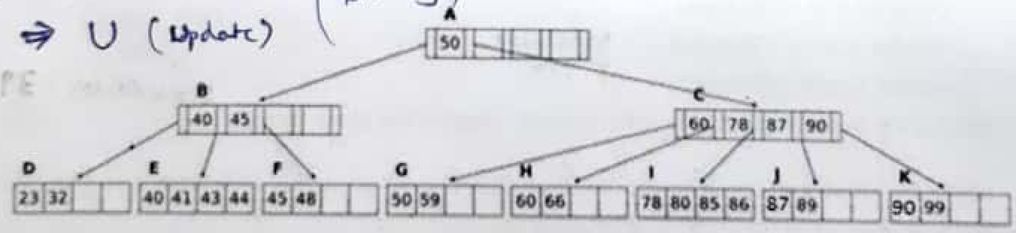
Q20: Which of the following about deadlocks are true?

a. If all transactions use two-phase locking, they cannot deadlock.
b. Systems that support update of locks cannot deadlock.
[✓ c. Once two transactions deadlock, one of them must be aborted to maintain consistency.]
d. None of the above.

Section-B

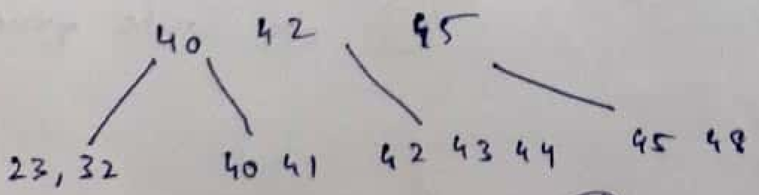
Question 1: Consider the following B+ tree. (1+1+1)

0.5 A, C, K ⇒ R (Read) (Binary)
0.5 = K ⇒ U (Update)



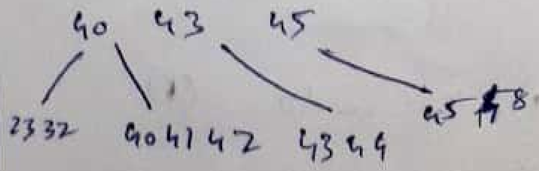
a. Suppose a data entry with key 98 is inserted to the B+ tree. Write down all the nodes that need to be read and all the nodes that need to be updated. You can mark "R" (for "read") and "U" (for "Update") on the nodes in the tree above directly.

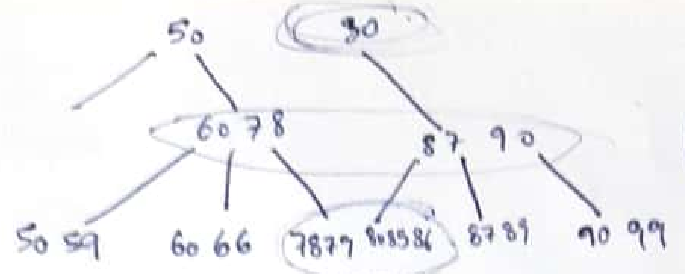
b. Consider the original B+ tree 1 above. Draw the new B+ tree after inserting a data entry with key 42. If there is no ambiguity, you can just draw the part that is changed.



1

Alternate Drawing



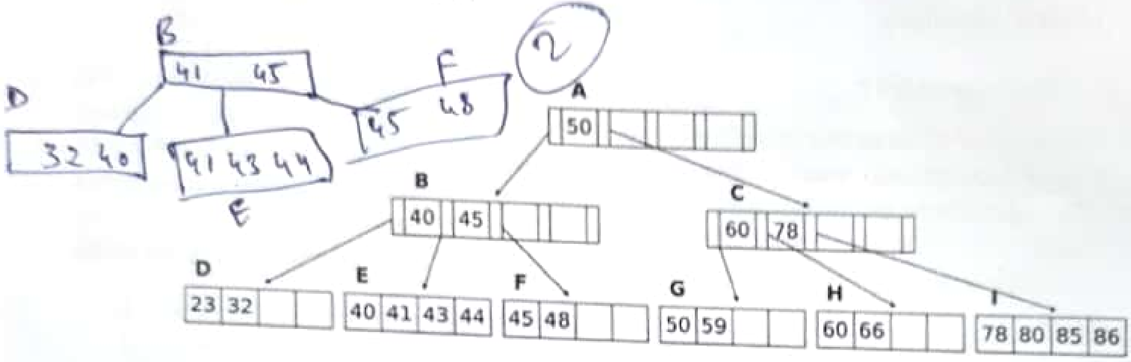


1

c. Consider the original B+ tree given in question 1 above. Draw the new B+ tree after inserting a data entry with key 79. If there is no ambiguity, you can just draw the part that is changed.

Question 2: Consider the following B+ tree. Draw the new B+ tree after deleting the data entry with key 23. Assume we need to redistribute data entries if possible. If there is no ambiguity, you can just draw the part that is changed. (2)

2



Question 3: A magnetic disk with 32 surfaces having 512 tracks per platter with 128 sectors per track is provided to you. You also know that this particular disk can only store 256 bytes in a sector. The disk rotates at a speed of 2400 rotations per minute and the average seek time is 5 milliseconds. (1+1)

(i) Calculate the total number of bytes that can be stored in this disk.Give your answers in MB.

(1) 256 MB (assumption - double surface)
OR 512 MB (assumption - single surface)

(ii) Calculate the average access time. 17.5 ms (1)

Question 4: Consider the database with the below parameters:

block size is 4096 bytes
header size is 100 bytes
record size is 100 bytes

$\frac{4096 - 100}{100}$

Consider a file storing variable length records. What is the block factor? (2)

Block Factor = 39 (2 if correct
1 if partially correct)

Question 5: What are the 3 conditions that determine if an action (A1) from one transaction takes precedence over an action (A2) in a different transaction? (1.5)

(0.5) a) A1 is ahead of A2 in schedule

(0.5) b) Both A1 and A2 access the same data item

(0.5) c) One of them has to be write operation

Question 6: Let's consider the log as it appears at three instances of time

(a)	(b)	(c)
$<T_0$ start>	$<T_0$ start>	$<T_0$ start>
$<T_0$, A, 1000, 950>	$<T_0$, A, 1000, 950>	$<T_0$, A, 1000, 950>
$<T_0$, B, 2000, 2050>	$<T_0$, B, 2000, 2050>	$<T_0$, B, 2000, 2050>
	$<T_0$ commit>	$<T_0$ commit>
	$<T_1$ start>	$<T_1$ start>
	$<T_1$, C, 700, 600>	$<T_1$, C, 700, 600>
		$<T_1$ commit>

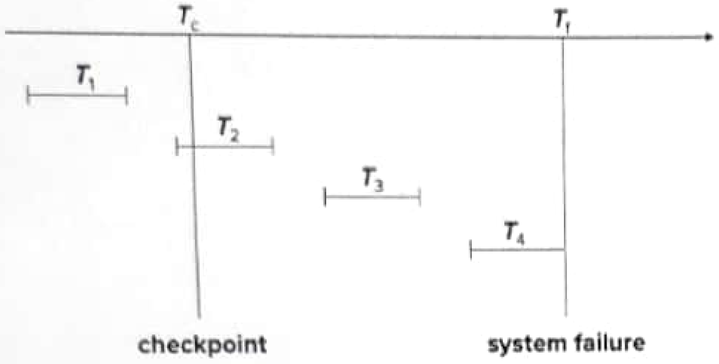
a) T_0 : undo (1)

b) T_0 : redo (0.5)
T_1 : undo (0.5)

c) T_0 : redo (0.5)
T_1 : redo (0.5)

What would be the recovery actions in each case? (1*3)

Question 7: Consider the below log with checkpoints.



T_1 : ignore (0.5)

T_2 : redo (0.5)

T_3 : redo (0.5)

T_4 : undo (0.5)

What would be the recovery actions? (2)

Question 8: Explain the problem in the following partial schedule, if any? (2)

T_3	T_4
lock-X(B)	
read(B)	
$B := B - 50$	
write(B)	
	lock-S(A)
	read(A)
	lock-S(B)
lock-X(A)	

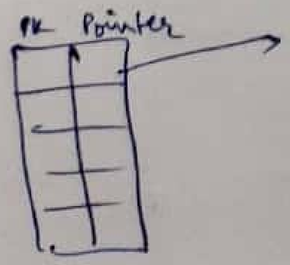
Problem: Deadlock (1)

Explanation: T_3 waits for T_4 (0.5)

T_4 waits for T_3 (0.5)

Question 9: Consider 'Hari Nagar Stock exchange scenario' given in Section-A. Assume the primary index is created on the search key (StockID, which is a primary key) on STOCK table. Can we answer any sub-class of SQL query using ONLY Index table? If yes, write at least 2 sub-class of such queries. (0.5+(1*2))

Can we : YES (0.5)



Sub-Class of such queries:

- SELECT
- COUNT
- EXISTS / NOT EXISTS

(1 each max 2 marks)