CSE 202, Winter Session 2023

End-Sem Exam

Marks: 40 **Time: 2 hrs**

Note: There are 40 questions in this paper. Each question carries 1 mark.

- A question can have multiple correct answers. You need to write all correct answers to get 1 mark, else zero marks will be given. There is no negative marking.
- No cutting or overwriting is allowed. If found, zero marks will be awarded. No discussion will be entertained in this regard.

Q1: Consider the schedules S1 below.

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

This schedule is conflict serializable. What would be its equivalent serial schedule?

a. T1 → T2 → T3
b. T2 → T3 →T1
c. T3 → T2 → T1
d. S1 is NOT conflict serializable

Q2: Consider the following transactions with data items P and Q initialized to zero:

T_1 : read (P) ;

read (Q) ;

if P = 0 then Q : = Q + 1 ;

write (Q).

T_2 : 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. A serial schedule
b. A schedule that is not conflict serializable
c. A conflict serializable schedule

d. A schedule for which precedence graph cannot be drawn

Q3: 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. 720
b. 1440
c. 2880
d. 5760

Q4: Which of the following RAID level provides the highest Data Transfer Rate(Read/Write)

a. RAID 1
b. RAID 2
c. RAID 3
d. RAID 4

Ans. (A) Allows both Read and write at the highest transfer rate. Other levels support read at the high rate but would not support write due to parity disk issue.

Q5: Specify the correct choice(s).

a. When an insertion into a B+ tree causes a node to split, the height of the tree always increases.
b. As the number of keys in a B+ tree increases, IO cost for searches grows logarithmically with the number of keys.
c. For a B+ Tree, during insertion, when we split a node, then the middle value of the node is retained in the right half node, and also goes up to the parent.
d. Bulkloading for B+ trees is efficient because we only keep the left most branch in memory for insertions and can disregard the rest of the tree.
   Ans. B. (Option C is incorrect as the statement is true only for the leaf nodes)

Q6: 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. How many number of bytes can be stored in this disk?

a. 2^{14} bytes
b. 2^{28} bytes

c. 2^{29} bytes
d. 2^{30} bytes

Ans (B) or (C)

Q7: Consider the following two transactions with lock and unlock instructions obeying two-phase locking protocol.

```
T1: lock-S(A)
     read(A);
    lock-X(B)
    read(B);
    If A = 0, then B = B+1;
    Write(B);
    unlock(A)
    unlock(B)

T2: lock-S(B)
     read(B);
    lock-X(A)
    read(A);
    If B = 0, then A = A+1;
    Write(A);
    unlock(B)
    unlock(A)
```

Can there be a situation where the execution of these transactions result in a deadlock?

(a) No
(b) Yes
(c) Sometimes
(d) Never

Ans. This question is dropped due to ambiguity in the interpretation of options.

Q8: Which statement is (are) true?

(a) I/O time for data transfer from disk to main memory is the summation of (seek time + rotational time + transfer time)
(b) To minimize I/O time, it is necessary to store and locate data strategically
(c) The only advantage of using RAID is to achieve good performance
(d) In RAID 5, the fault tolerance is achieved through distributed parity and 100% storage is used for data storage.
(e) In RAID 1, data is partially replicated across the disks so less than 100% storage is used for data storage.

Ans. (A,B)

Q9: Which statement is (are) false w.r.t. the B+ tree?

(a) It supports both equality and range queries.
(b) Its internal and leaf nodes direct the search.
(c) Leaf nodes are organized into doubly linked list.
(d) All paths from the root to leaf nodes are of varying length.

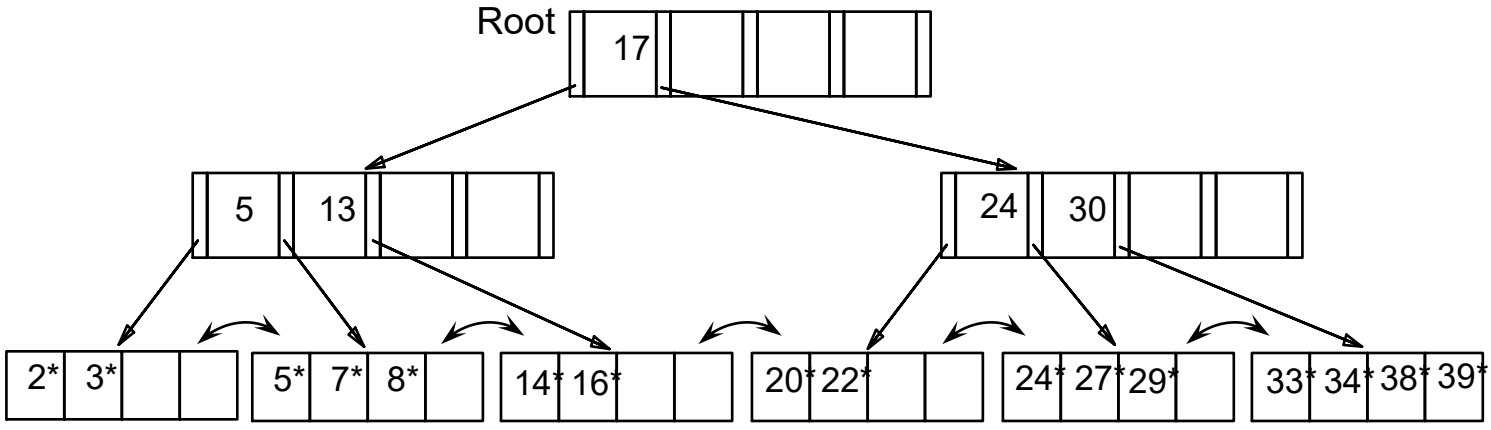
Ans. B, D (Option B is false as leaf nodes do not have any branching for the search purpose. Rather it directly gives the address of the record if key is present.)

Q10: Which statement is (are) true w.e.t. “When a new page is to be placed in the main memory, a resident page should be evicted first (assuming there is not enough space in the memory)”.

(a) The buffer manager will always determine the optimum replacement policy.
(b) The buffer manager may not determine the optimum uniform replacement policy.
(c) The DBMS administrator lets the buffer manager know to use the replacement policy.
(d) The buffer manager capitalizes the different criterion to finalize the replacement policy.

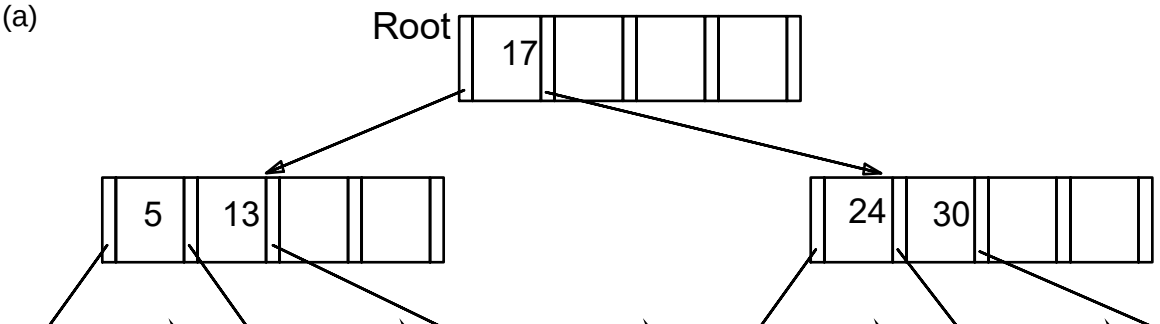
Ans. (A,C,D)

Q11: Consider the below B+ tree.

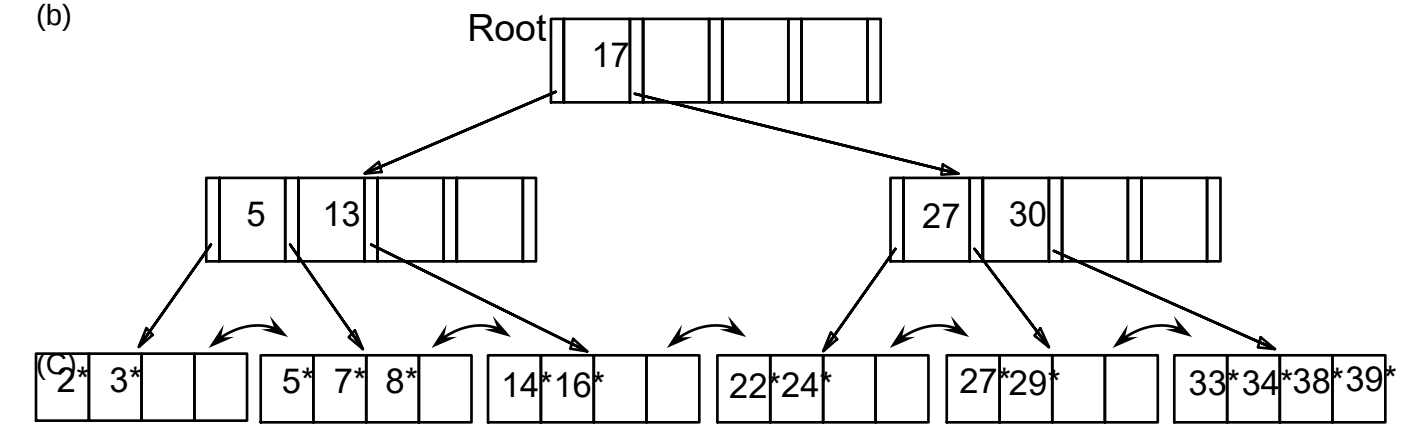


What would be the resultant B+ tree after the deletion of value 20?

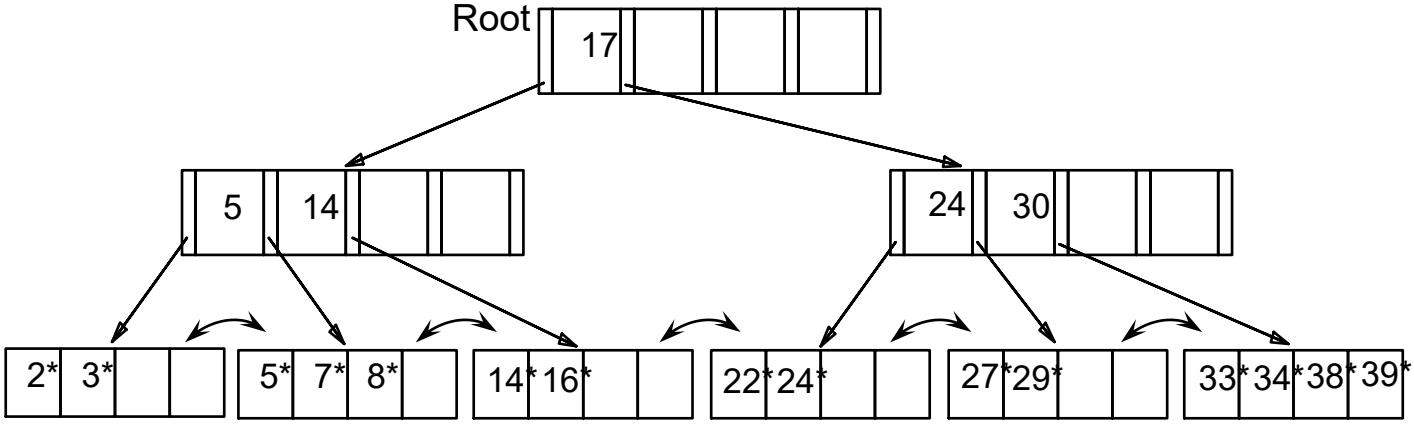
(a)



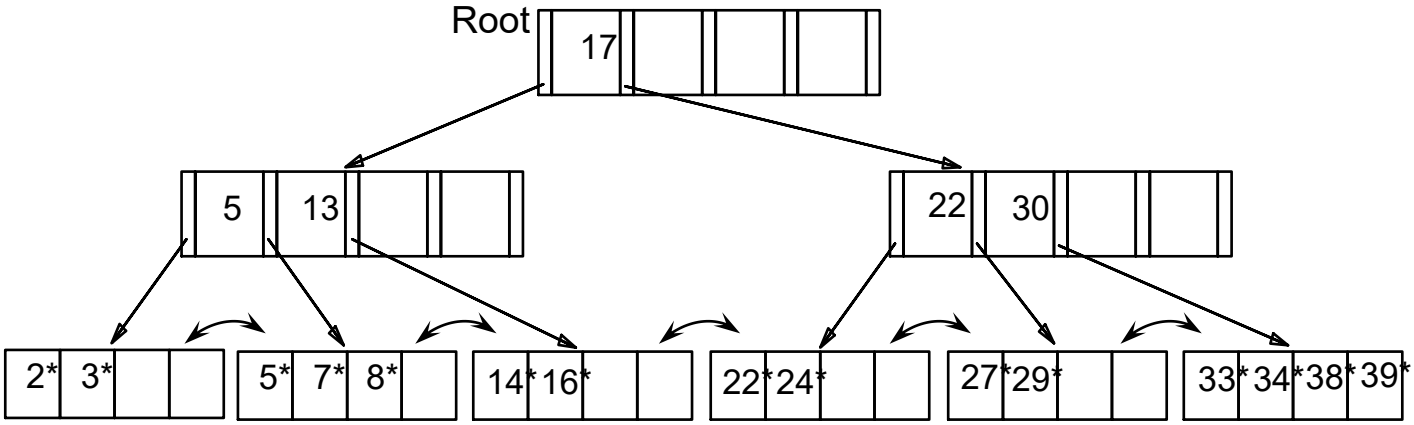
(b)


Root
17
5
13
27
30
2*
3*
5*
7*
8*
14*
16*
22*
24*
27*
29*
33*
34*
38*
39*

(C)

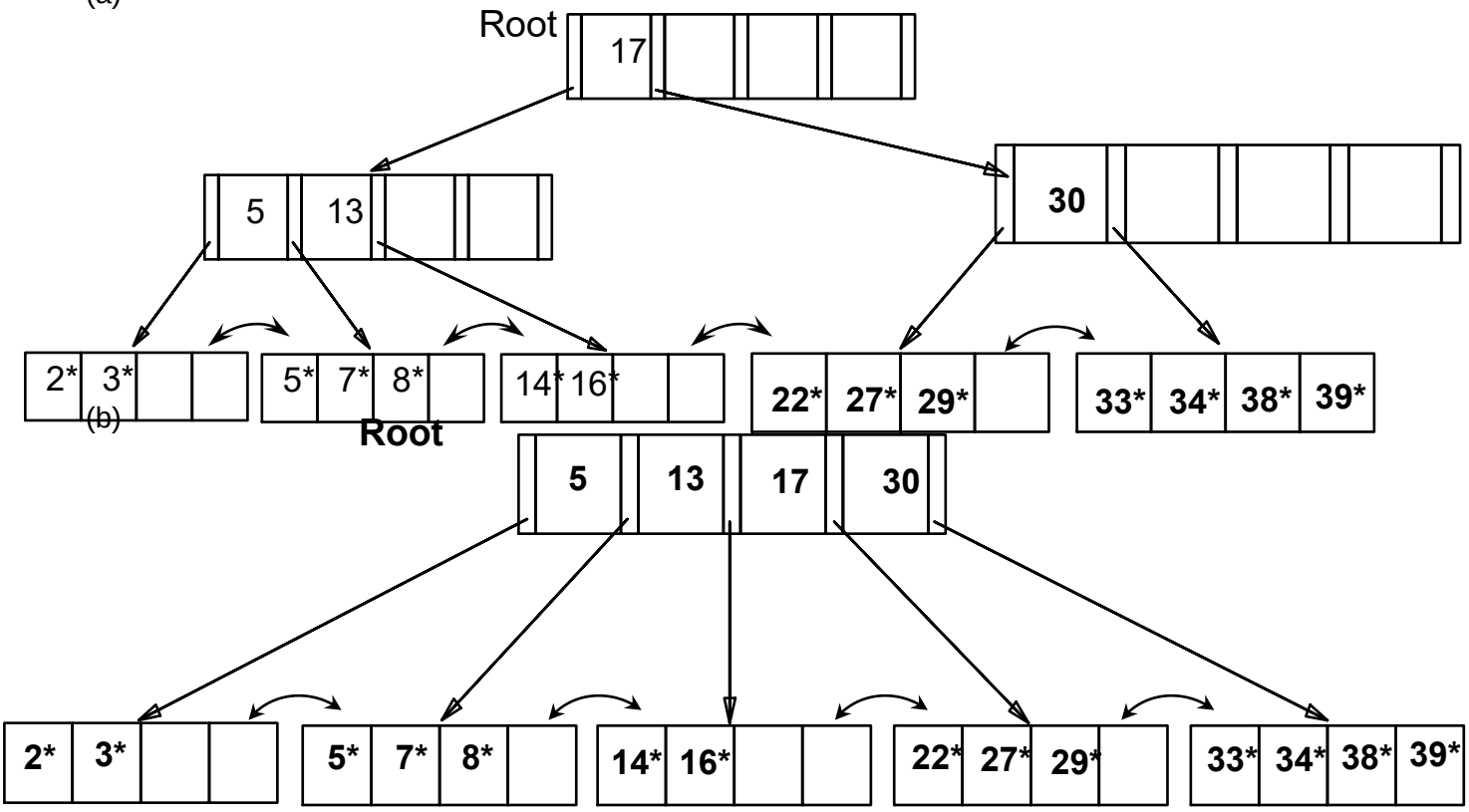

Root
17
5
14
24
30
2*
3*
5*
7*
8*
14*
16*
22*
24*
27*
29*
33*
34*
38*
39*

(d)


Root
17
5
13
22
30
2*
3*
5*
7*
8*
14*
16*
22*
24*
27*
29*
33*
34*
38*
39*

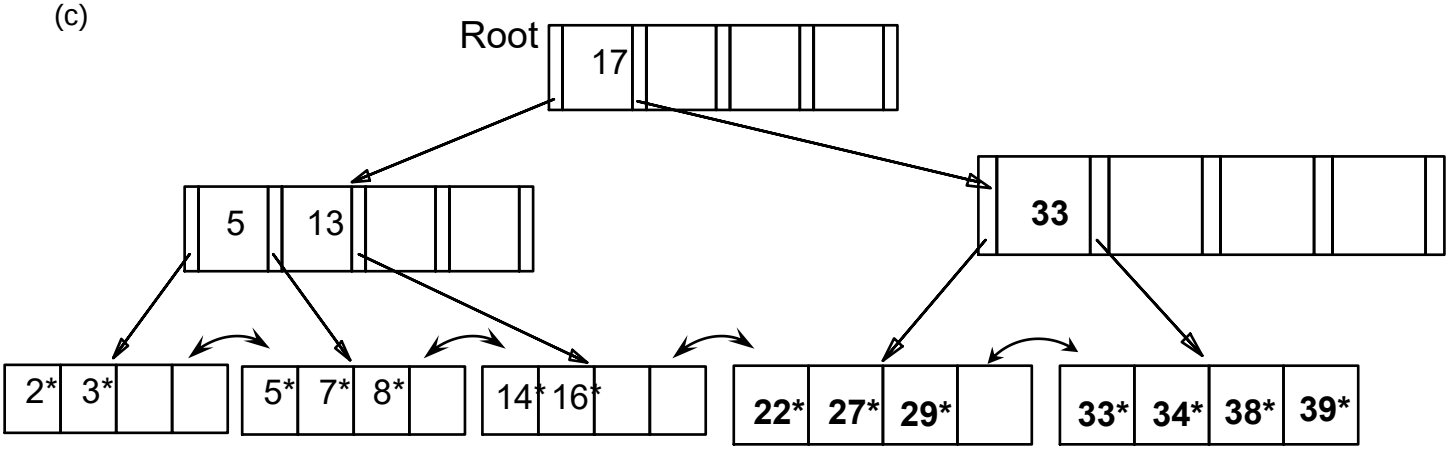
Q12: What would be the resultant B+tree if we delete the value 24 in the B+ tree ontained in the above question?

(a)

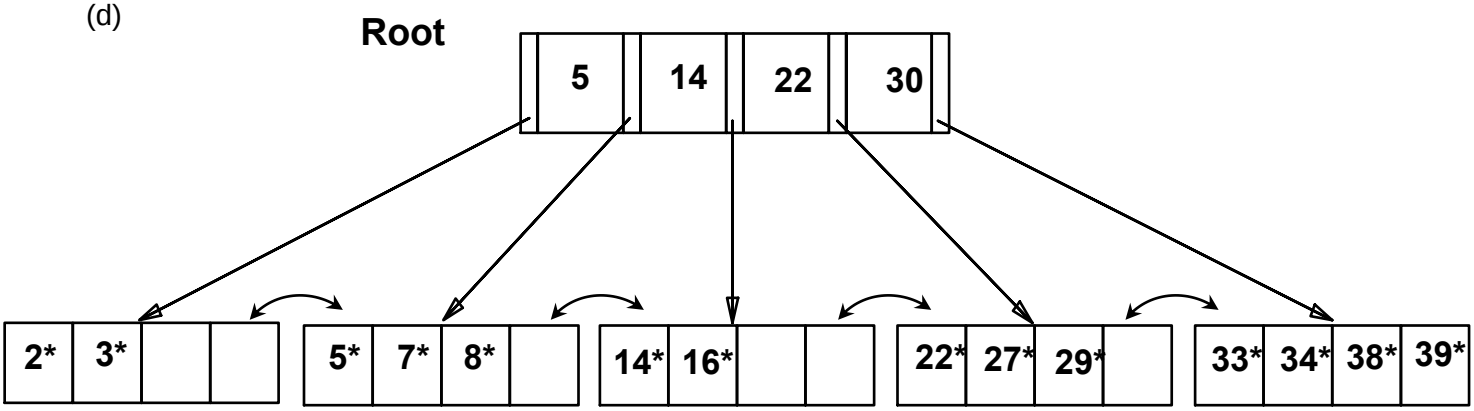


(b)

(c)



(d)



Q13: In extendable hashing, when does the pointer directory gets doubled, when

(a) all data buckets get full.
(b) When any of the associated data bucket gets full and that to leading to split the bucket
(c) The data bucket local depth equals to the global depth of the directory
(d) The data bucket local depth is less than to the global depth of the directory

Ans. None of the option is correct in this question. Therefore this question is dropped. Directory gets doubled only if the data bucket local depth equals to the global depth of the directory and insertion of record results into the data buckets result into splitting of the bucket.

Q14: Reading a page into memory and then sorting it and finally writing it back to the disk, how many buffer pages are needed?

(a) 1
(b) 2
(c) 3
(d) 0

Ans. (A) we can perform inplace sorting.

Q15: Given a STUDENT table having attributes STD_ID, STD_Name, STD_Dept, STD-Hostel with a constraint that each student belongs to one dept only and an Index table on STD_ID plus a pointer to the tuple position. If we need to answer the query “How many students are there in all dept?”, this query can be answered from

(a) STUDENT Table
(b) Index Table and STUDENT Table
(c) Index Table
(d) Need more data tables to answer this query

Ans. (A, B, C) or (C)

Q16: Which statements are true?
(a) In Index Nested Join, the inner relation is chosen that has an index table.
(b) Sort-Merge Join algorithm gives better performance for the range queries.
(c) The Block Nested Loops Join is more sensitive to the buffer size, i.e. the I/O cost reduces as we increase the buffer size.
(d) Hash join is sensitive to data skew while sort-merge join is not.

Ans. (A,C,D)

Q17: A phantom read is a situation where a transaction reads a set of records that satisfy a certain condition, but then another transaction inserts or deletes a record that also satisfies that condition, causing the first transaction to see a phantom record. What techniques can be used to prevent Phantom reads?
(a) Locking
(b) Multi-version concurrency control
(c) Serializability
(d) Undo and Redo

Ans. (A,B)

Q18: A dirty read is a situation where a transaction reads data that has been modified by another transaction that has not yet been committed. It can be avoided by using techniques
(a) Locking
(b) Multi-version concurrency control
(c) Serializability
(d) Undo and Redo

Ans. (A,B)

Q19: Read-only transactions are used for making
(a) Database updates
(b) Reporting purpose
(c) Data replication
(d) Analysis applications

Q20: Which statement(s) is (are) true?
(a) Covering Index table has multiple keys and pointers to the data record.

(b) Covering index table includes all the columns required by a query and pointers to the data record.
(c) There is an additional overhead for maintaining these required columns whenever the corresponding attributes values are updated.
(d) It significantly improves query performance by reducing the number of I/O operations required to execute the query.

Q21: Is it possible to create indexes on a subset of the rows in a table?
(a) Yes
(b) No
(c) Depends on data values
(d) Yes, but the query performance in all scenarios will be very poor.

Q22: Which statement(s) is (are) true w.r.t. query processing and optimization?
(a) Query optimizer always generate one plan for execution.
(b) Query optimizer generates multiple plans, but ten chooses the one which has minimal cost.
(c) Query optimizer uses the data dictionary and statistics for computing the cost.
(d) Query optimizer also tells which algorithm should be used for each relational operator based on the data distribution and index tables.

Q23 WR conflicts arise when transaction T2 reads a data object A that has been modified by another transaction T1, *which has not yet committed*. This read is
(a) Unrepeatable Read
(b) Repeatable Read
(c) Dirty Read
(d) Clean Read

Q24: In the below schedule, what problem(s) do you see?

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)	

(a) Dirty Read
(b) Repeatable Read
(c) Deadlock
(d) Starvation

Q25: The starvation of a transaction T can be avoided by

(a) granting the lock on data item Q so that there is no other transaction holding a lock on Q in a mode that conflicts with the requested lock by T AND
There is no other transaction that is waiting for a lock on Q and that made its lock request before T

(b) releasing the lock on data item Q so that there is no other transaction holding a lock on Q in a mode that conflicts with the requested lock by T AND
There is no other transaction that is waiting for a lock on Q and that made its lock request before T

(c) granting the lock on data item Q so that there is no other transaction holding a lock on Q in a mode that conflicts with the requested lock by T AND
There is no other transaction that is holding a lock on Q and that made its lock request before T

(d) releasing the lock on data item Q so that there is no other transaction holding a lock on Q in a mode that conflicts with the requested lock by T AND
There is no other transaction that is holding a lock on Q and that made its lock request before T

Ans. Dropping this questions as there can be multiple interpretation due to lack of policy context.

Q26: Which statement(s) is (are) true?

(a) Strict 2PL does not limit the concurrency
(b) Strict 2PL avoids the ‘dirty reads’
(c) Strict 2PL avoids the ‘deadlocks’
(d) Strict 2PL may lead to ‘deadlocks’

Q27: Which statement(s) is (are) true?

(a) For a schedule to be *recoverable*, transactions should commit only after all transactions whose changes they read commit.
(b) *Recoverable schedules* avoid *cascading aborts*
(c) For a schedule to be *recoverable*, transactions should abort only after all transactions whose changes they read commit.
(d) *Recoverable schedules* do not avoid *cascading aborts*
(e) Strict 2PL ensures the schedule *recoverable*.

Q28: The magnetic disk rotates at a speed of 2400 rotations per minute and the average seek time is 5ms. What would be the average access time?

(a) 30ms
(b) 25ms
(c) 17.5ms
(d) 12.0ms

Q29: Consider the database with the below parameters:

block size is 4096 bytes

header size is 100 bytes
record size is 100 bytes

where, a file storing variable length records. What would be the value of the block factor?

(a) 41
(b) 40
(c) 39
(d) 32

Answer the following questions based on this description.

Hari Nagar 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.

Q30: If the following queries are 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 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

Q31. Assume now that the Hari Nagar 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

Q32: 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

Q33: 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

Q34: 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

Q35: 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

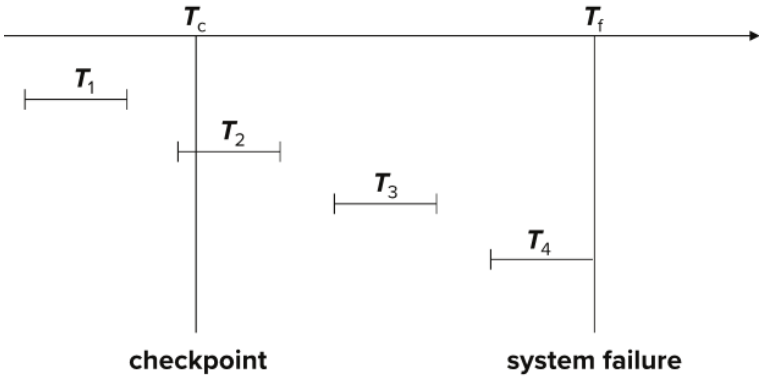
Q36: Let's consider the log as it appears at three instances of time

(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>

The recovery actions would be

(a) In instance a: T_0: undo,
(b) In instance b: T_0: redo, T_1: undo
(c) In instance c: T_0: redo, T_1: redo
(d) In instance c: T_0: undo, T_1: undo

Q37: Consider the below log with checkpoints.



What would be the recovery actions?

(a) T_1: redo, T_2: redo, T_3: redo, T_4: undo
(b) T_1: Ignore, T_2: redo, T_3: redo, T_4: undo
(c) T_1: Ignore, T_2: undo, T_3: undo, T_4: redo
(d) T_1: redo, T_2: undo, T_3: undo, T_4: redo

Q38: Jai (T1) and Viru (T2) are browsing items A and B in an online store. Suppose that T1 and T2 actions are interleaved as follows:

- T1 reads A
- T2 reads A, decrements A and commits
- T1 tries to decrement A

The schedule for these actions is:
S: R1(A), R2(A), W2(A), T2(Commit), W1(A), T1(Commit)

Which anomaly this schedule suffers from:

(a) RW
(b) WR
(c) WW
(d) RR

Ans. (A) T1 performs a write operation on A which is read by T2. Locking protocol can not allow it. Therefore the anomaly is RW.

Q39: In the above question, how can the anomaly be resolved?

(a) T1 acquires the Exclusive lock on A, T2 requests the lock on A, but has to wait till T1 releases the lock before it commits.
(b) T1 requests the Exclusive lock on A, T2 acquires the lock on A, but T1 has to wait till T2 releases the lock before it commits.
(c) T1 acquires the Exclusive lock on A, T2 also acquires the lock on A. Both of them continue processing till they commit.
(d) T1 acquires the Exclusive lock on A, T2 requests the lock on A, but has to wait till T1 releases the lock after it commits.

Ans. (A) as per Strict2PL, (D) as per 2PL, (A,D)

Q40: Two schedules are said to be *equivalent* if for any database state, the effect of executing the 1st schedule is identical to the effect of executing the 2nd schedule. Here the 1st and 2nd schedule could be:

(a) Conflict serializable schedule and serial schedule
(b) Serial schedule and serial schedule
(c) Conflict serializable schedule and conflict serializable schedule
(d) Serial schedule and conflict serializable schedule