Quiz 11: Tree Indexs & Index Concurrency Control

Started: Apr 7 at 1:20pm

Quiz Instructions

Question 1 1 pts

Both the hash index and the B+Tree index are helpful for range queries.

○ True

● False

Question 2 1 pts

In a B+Tree index, where is the actual data stored?

☑ Leaf nodes

□ Internal nodes

□ Root node

□ None of the above

Question 3 1 pts

What are clustered indexes in SQL?

☑ Indexes that store the data rows of the table in sorted order based on the index key

□ Indexes that store pointers to the data rows of the table in sorted order based on the index key

□ Indexes that store the data rows of the table in a random order

□ Indexes that store metadata information about the table's structure

Question 4 1 pts

What is the time complexity of searching for a record using a B+Tree index?

☑ O(log n)

□ O(1)

□ O(n)

□ O(n log n)

Question 5 1 pts

Which statement about B+Tree indexes is true?

☑ They improve performance for both equality and range queries.

□ They can only be created on numeric data types.

□ They are limited to indexing only single columns.

□ They are useful only for small-sized databases.

Question 6 1 pts

Which of the following are the possible number(s) of keys a non-root node can have in a 10-way B+Tree? (Select all that apply)

□ 2

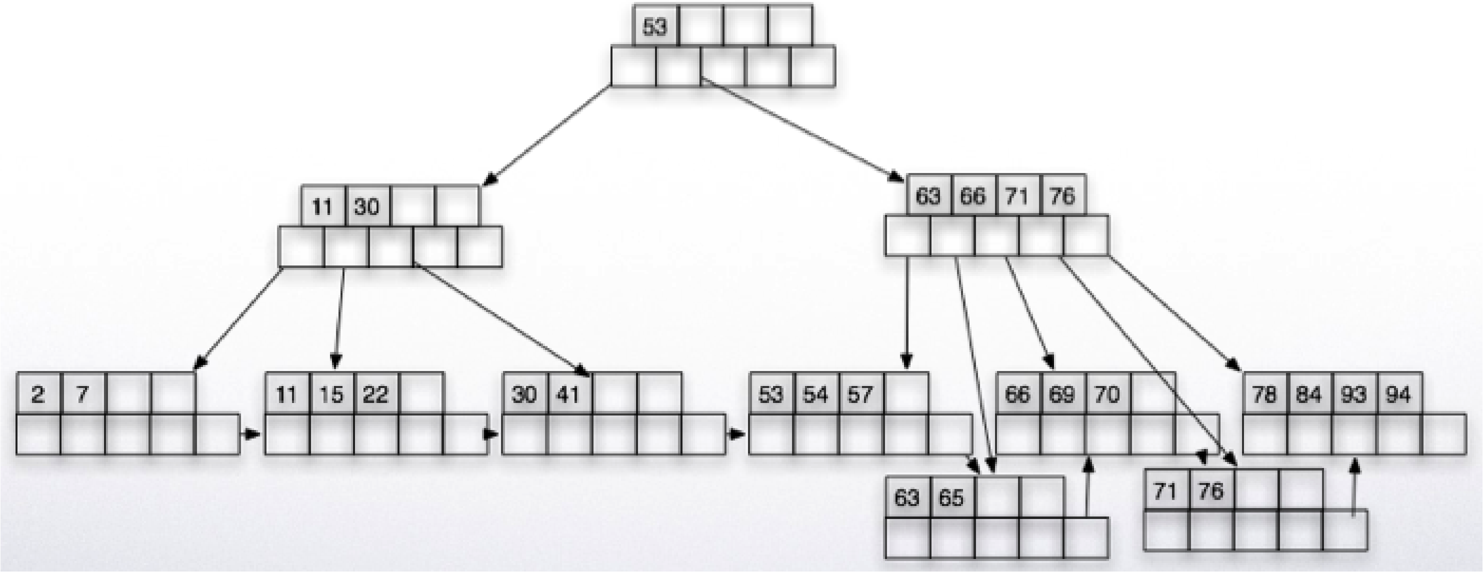
□ 3

☑ 5

☑ 9

Question 7 1 pts

What happens when we try to insert 90 in this B+tree? (Select all that apply)



☑

A split is done at the leaf node to accommodate the new key

☑

Node with keys [63, 66, 71, 76] also splits

☑

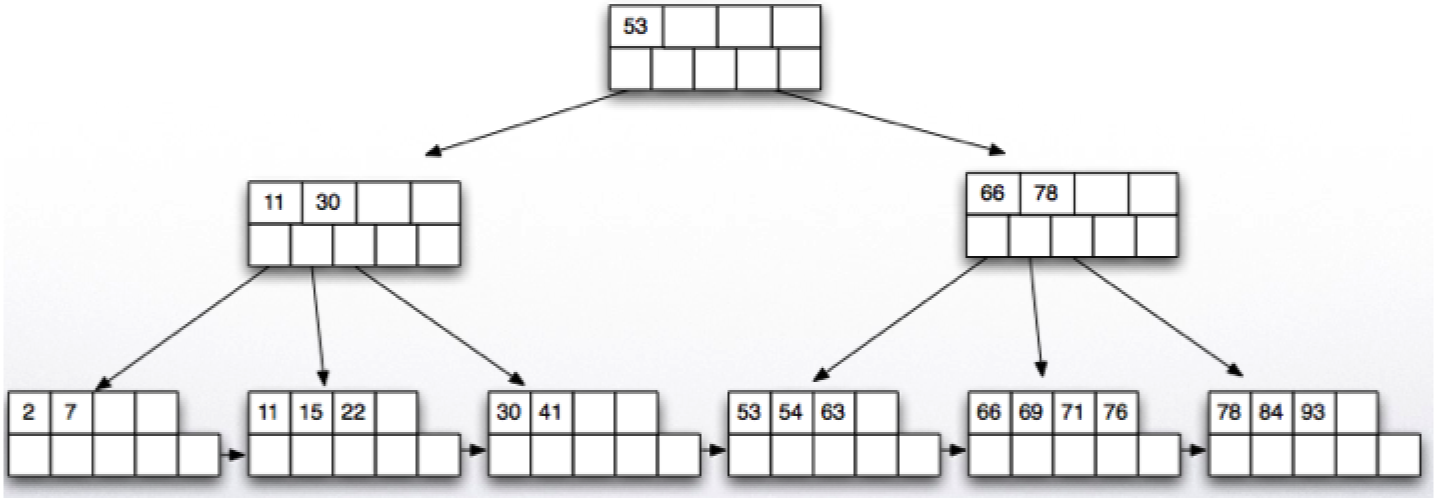
The root node of the B+Tree will be modified.

□

The value 90 will be inserted into a leaf node without any modifications to the tree structure.

Question 8 1 pts

What happens when we try to delete 30 in this B+tree? (Select all that apply)



□

The value 22 will be promoted to the parent node.

☑
The node containing 41 will be merged with a sibling node.

□
The deletion will cause no changes to the B+Tree structure.

□
Node with value 41 will borrow from its sibling.

Question 9 1 pts

For the composite index**, CREATE INDEX LFM_name ON artist (last_name, first_name, middle_names NULLS FIRST),** select all the statements that accurately describe the usage of this B+Tree index.

☑
The index can efficiently search for records matching all the key components in the specified order (e.g., last_name=X, first_name=Y, middle_names=Z).

☑
The index can efficiently search for records matching a prefix of the key components in the specified order (e.g., last_name=X, first_name=Y).

□
The index can efficiently search for records where any individual key component matches a specific value (e.g., first_name=Y) in most databases.

□
The index can efficiently search for records matching a suffix of the key components in the specified order (e.g., first_name=Y, middle_names=NULL) in most databases.

Question 10 1 pts

What is the name of the process to efficiently retrieve tuples from a database table based on a non-clustered index, by sorting the needed tuples based on their page ID so that the DBMS retrieves each page once?

☑
Index scan page sorting

□
Tuple indexing

□
Clustered order retrieval

□
Page-based data retrieval

Question 11 1 pts

Which factors are crucial design choices when implementing a B+ Tree?

☑ Merge Threshold

☑ Intra-Node Search Strategy

☑ Variable-Length Keys

☐ Language Support

Question 12 1 pts

B+Tree performance can be improved through various optimization techniques. Which of the following are valid B-Tree optimizations?(Select all that apply)

☑ Buffered Updates

☑ Pointer Swizzling

☑ Prefix Compression

☐ Index Fragmentation

Question 13 1 pts

In database terminology, locks are for deadlock detection and avoidance, while latches are for deadlock resolution.

○ True

● False

Question 14 1 pts

What are the implementation goals for latch mechanisms in concurrency control? (Select all that apply)

☐ Deschedule thread when it has been waiting for too long to avoid burning cycles

☐ Each latch should have its own queue to track waiting threads

☐ Fast execution path when no contention

☑ All the above

Question 15 1 pts

Which of the following implementations are examples of read-write aware latch mechanisms?

A) Test-and-Set Spinlock
B) Blocking OS Mutex
C) Reader-Writer Locks

☐ A and C

☐ C

☐ B and C

☑ A, B, and C

Question 16 1 pts

In hash table latching, only page-level latches can be implemented.

○ True

● False

Question 17 1 pts

To allow multiple threads to access/modify B+Tree at the same time, when is it safe to release the latch from the parent node?

☑ Child node will not split when updated during an insertion operation

☐ Child node is full

☐ Child node is half full

☑ Child node will not merge when updated during a deletion operation

Quiz saved at 11:29pm Submit Quiz