

Question 1: Index Concurrency Control [290 points]

- (i) [10 points] **Index Data Structures:**
Distinguish between these data structures using illustrative queries on which they work well: (1) Hash Table, (2) B+Tree, and (3) Trie.
- (ii) [10 points] **Concurrency Control:**
List two reasons why concurrent data structures are more performant.
- (iii) [10 points] **Concurrency Control:**
Define a concurrency control protocol.
- (iv) [10 points] **Concurrency Control:**
Distinguish between logical correctness and physical correctness. How are these correctness criteria enforced?
- (v) [20 points] **Latches Overview:**
List four differences between locks and latches.
- (vi) [10 points] **Latches Overview:**
Are latches in DBMSs equivalent to locks in C++? Explain the source of this confusion.
- (vii) [10 points] **Latches Overview:**
List the latch modes using a compatibility matrix.
- (viii) [10 points] **Latches Overview:**
How is a blocking OS mutex used? How is it implemented in Linux?
- (ix) [10 points] **Latches Overview:**
How is a test-and-set spin latch used? How is it implemented in Linux?
- (x) [10 points] **Latches Overview:**
How is a reader-writer latch used? How is it implemented in Linux?
- (xi) [10 points] **Latches Overview:**
Define these terms:
- Starvation
 - Deadlock
 - Livelock
- (xii) [10 points] **Latches Overview:**
Distinguish between these latch implementations: (1) blocking OS mutex, (2) test-and-set spin latch, and (3) reader-writer latch.
- (xiii) [10 points] **Hash Table Latching:**
Distinguish between page latches and slot latches.
- (xiv) [10 points] **Hash Table Latching:**
Are deadlocks possible with latching in hash table? Justify your answer.
- (xv) [20 points] **B+Tree Latching:**
Explain the basic latch coupling protocol for FIND operation. Explain why this protocol is needed with an example.

- (xvi) **[10 points] B+Tree Latching:**
Explain the basic latch coupling protocol for INSERT operation.
- (xvii) **[20 points] B+Tree Latching:**
Explain the optimistic latch coupling protocol for INSERT operation. Explain why this protocol is better than the basic protocol with an example.
- (xviii) **[20 points] B+Tree Latching:**
Distinguish between optimistic and pessimistic latch coupling protocols. Which protocol is more scalable on a workload with low contention? Which protocol is more scalable on a workload with high contention?
- (xix) **[10 points] Leaf Node Scans:**
Explain the need for hint keys in leaf node scans.
- (xx) **[10 points] Leaf Node Scans:**
How are deadlocks avoided in a B+Tree latching protocol?
- (xxi) **[10 points] Leaf Node Scans:**
Explain the no-wait mode for avoiding deadlocks.
- (xxii) **[10 points] Leaf Node Scans:**
Explain why latches do not support deadlock detection or avoidance.
- (xxiii) **[10 points] B-link Trees:**
Explain the optimization underlying B-link trees.
- (xxiv) **[10 points] B-link Trees:**
Justify the name: "B-link" Trees.
- (xxv) **[10 points] B-link Trees:**
Distinguish between a B+tree and a B-link Tree.