QUESTION BANK – Module 5

(50 questions)

1. Explain the concept of concurrency control and its significance in transaction processing.
2. Define the transaction model and discuss its key components.
3. What are the desirable properties of transactions? Explain each property in detail.
4. Differentiate between serial schedules and concurrent schedules in transaction processing.
5. Define conflict equivalence and conflict serializability. How are they related to concurrent schedules?
6. Explain the concepts of recoverable schedules and cascade-less schedules in transaction processing.
7. Discuss the basics of locking in transaction processing. How does locking help ensure data consistency?
8. Describe the two-phase locking protocol and its variations. How do they contribute to concurrency control?
9. What is log-based recovery in transaction processing? Explain the role of the system log in recovery.
10. Discuss deferred database modification and its benefits in transaction processing.
11. Explain the concept of check-pointing and its role in ensuring database consistency.
12. What are NoSQL databases? How do they differ from traditional relational databases?
13. Discuss the main characteristics of key-value databases, with examples from Redis.
14. Explain the main characteristics of document databases, using examples from MongoDB.
15. Describe the main characteristics of column-family databases, with examples from Cassandra.
16. Discuss the main characteristics of graph databases, using examples from ArangoDB.
17. How does concurrency control affect the performance of transaction processing systems?
18. Compare and contrast optimistic and pessimistic concurrency control approaches.
19. Discuss the challenges and benefits of implementing distributed concurrency control.
20. Explain the concept of deadlock in transaction processing. How can it be prevented or resolved?
21. What is a serializable schedule? How can it be ensured in a concurrent transaction environment?
22. Describe the concept of multiversion concurrency control (MVCC) and its advantages.
23. Discuss the ACID properties of transactions and explain their significance in database systems.
24. Explain the concept of two-phase commit protocol and its role in distributed transaction processing.
25. What are the advantages and disadvantages of using log-based recovery mechanisms?
26. Discuss the concept of durability in transaction processing. How is it achieved?
27. How does recovery manager handle failures in a transaction processing system?
28. Explain the concept of write-ahead logging and its role in ensuring transaction durability.
29. Discuss the challenges and techniques for managing concurrency in distributed databases.
30. How does transaction isolation level affect the concurrency control mechanism?
31. Describe the concept of index locking and its impact on transaction processing.
32. Discuss the concept of snapshot isolation and its implications on concurrency control.
33. Explain the concept of lock escalation and its role in managing resource contention.

34. How can deadlock detection and resolution be implemented in a transaction processing system?
35. Discuss the role of timestamp ordering in concurrency control and transaction scheduling.
36. Describe the concept of data replication in distributed databases and its impact on concurrency.
37. Discuss the challenges and techniques for handling distributed deadlocks in transaction processing.
38. Explain the concept of optimistic concurrency control and its applications in distributed systems.
39. Discuss the impact of long-running transactions on concurrency control and system performance.
40. Explain the concept of data consistency and the role of concurrency control in maintaining

41. Consider a transaction T1 that reads data item A and transaction T2 that reads and writes data item A. Both transactions run concurrently in a system with a two-phase locking protocol. Draw a schedule that demonstrates a conflict serializable execution of these transactions. Also, explain how the two-phase locking protocol ensures serializability.

42. Given the following schedule of transactions:

    T1: R(A), W(B)

    T2: R(B), W(A)

    Determine if the schedule is conflict serializable. If it is not, demonstrate a conflict-serializable schedule that is equivalent to the given schedule.

43. Assume a transaction T1 acquires a shared lock on data item A, followed by a shared lock on data item B, and then requests an exclusive lock on data item C. Another transaction T2 holds an exclusive lock on data item C and requests a shared lock on data item B. Determine whether the system is in a deadlock state. If it is, explain the deadlock and suggest a strategy to resolve it.

44. Consider a distributed database system with two sites A and B. Each site has a copy of data item X, and both sites allow read-committed isolation level. If transaction T1 reads data item X from site A and transaction T2 updates data item X at site B, explain how the system ensures read-committed isolation and resolves any conflicts.

45. Suppose a system uses the write-ahead logging protocol for recovery. Describe the steps involved in recovering a transaction after a system failure, starting from the analysis phase to the redo and undo phases.

46. Assume a NoSQL database system based on a key-value store, such as Redis. Design a data model and propose an indexing strategy to efficiently retrieve all customer records with a specific age. Explain how your design supports efficient querying and scalability.

47. Consider a graph database system, such as ArangoDB, that stores social network data. Design a query to find the shortest path between two users based on their connections. Describe the graph traversal algorithm you would use and discuss its efficiency and scalability.

48. Suppose you are designing a distributed transaction processing system using the two-phase commit protocol. Outline the steps involved in executing a distributed transaction and handling failures at various stages. Discuss the challenges and strategies for ensuring atomicity and consistency in the system.
49. Assume you are working with a column-family database, such as Cassandra, for a large-scale e-commerce application. Design a data model that efficiently supports querying for products based on multiple attributes, such as category, price range, and availability. Explain how your design optimizes data storage and retrieval.
50. Given a scenario where multiple transactions are executing concurrently, explain how timestamp ordering can be used to ensure serializability and manage concurrency. Discuss the potential conflicts that may arise and how they can be resolved using timestamp-based protocols.