

Short Questions

1. What is schema refinement in the context of database design?
2. What problems are caused by data redundancy in a database?
3. Define decomposition as it relates to database design.
4. What problems can arise from decomposition in database design?
5. What are functional dependencies in the context of a relational database?
6. Explain the concept of the First Normal Form (1NF) in database design.
7. What is the purpose of the Second Normal Form (2NF) in database normalization?
8. Describe the Third Normal Form (3NF) in the context of database normalization.
9. What is Boyce-Codd Normal Form (BCNF), and when is it achieved?
10. What is lossless join decomposition in the context of database normalization?
11. What are multi-valued dependencies (MVDs) in database design?
12. What is the Fourth Normal Form (4NF) in database normalization?
13. Explain the concept of the Fifth Normal Form (5NF) in the context of database design.
14. How can you identify functional dependencies in a given set of attributes?
15. What is the primary goal of normalization in database design?
16. When should you consider denormalization as an alternative in database design?
17. How does normalization contribute to efficient data storage in a database?
18. What is a superkey, and how does it relate to candidate keys and primary keys?
19. What is functional dependency preservation, and why is it important in schema refinement?
20. Why is maintaining referential integrity crucial in database design?

21. What role do keys play in functional dependency analysis?
22. What are some common examples of functional dependencies in real-world scenarios?
23. How does the process of normalization simplify database maintenance?
24. What is the difference between a strong and weak entity in a relational database?
25. Explain the concept of transitive dependency and how it relates to normalization.
26. What is a database transaction, and why is it important in a database management system?
27. Explain the ACID properties of a transaction.
28. Define the concept of a "transaction state" in the context of database transactions.
29. What is the difference between a committed state and an active state in a transaction?
30. Describe the role of the transaction manager in a database system.
31. How does a database system ensure the atomicity of a transaction?
32. Discuss the concept of "rollback" in transaction management.
33. What is the purpose of the transaction log in ensuring durability?
34. Explain the implementation of atomicity in a database system.
35. How does a database system recover from a system crash while ensuring atomicity?
36. Define the term "concurrent execution" in the context of multiple transactions.
37. What are the challenges associated with concurrent execution of transactions in a database system?
38. What is serializability, and why is it important in concurrent transaction processing?
39. Explain the concept of conflict-serializability in transaction scheduling.
40. What is a precedence graph, and how is it used in testing for serializability?

41. Describe the recoverability property of a schedule in database concurrency control.
42. What is the difference between strict schedules and non-strict schedules?
43. How does a database system ensure recoverability while allowing concurrency?
44. Explain the implementation of isolation levels in a database system.
45. What are the common isolation levels defined by SQL standards?
46. Describe the issues related to dirty reads, non-repeatable reads, and phantom reads in isolation levels.
47. What is the purpose of locks in database concurrency control?
48. Differentiate between shared locks and exclusive locks in a lock-based protocol.
49. Explain the concepts of deadlock and deadlock detection in lock-based protocols.
50. How does two-phase locking ensure serializability in concurrent transactions?
51. What is a timestamp-based protocol, and how does it use timestamps to control concurrency?
52. Describe the concept of validation-based concurrency control in a database system.
53. What is a conflict-serializable schedule, and how is it determined?
54. Explain the concept of multiple granularity locking in concurrency control.
55. How does hierarchical locking help in managing locks efficiently?
56. Discuss the trade-offs between strict two-phase locking and deadlock avoidance.
57. Explain the concept of log-based recovery in a database system.
58. What is the purpose of the write-ahead logging protocol?
59. Describe the steps involved in the recovery process after a system crash.
60. How does the redo phase of recovery work in log-based recovery?
61. What is the undo phase of recovery, and when is it necessary?

62. Define the term "checkpoint" in the context of log-based recovery.
63. How does the ARIES algorithm ensure the recovery of a database system?
64. Explain the concepts of deferred and immediate database modification in recovery.
65. What is the significance of stable storage in log-based recovery?
66. Describe the implementation of savepoints in database transactions.
67. What is the purpose of a recovery manager in a database system?
68. Explain the concept of media recovery in database management.
69. What are the challenges of recovering from media failures?
70. Describe the role of transaction identifiers (TIDs) in recovery.
71. How does recovery with concurrent transactions differ?
72. Discuss the importance of maintaining the durability property.
73. Explain the concept of crash recovery in database transactions.
74. How does the recovery process ensure that committed transactions are not lost?
75. Describe the steps involved in recovering from a disk failure.
76. What is external storage in the context of databases?
77. Why is file organization important in database systems?
78. Explain the concept of indexing in databases.
79. What are cluster indexes, and how do they differ from other types of indexes?
80. What is the purpose of primary indexes in a database?
81. How are secondary indexes different from primary indexes?
82. Name a common index data structure used in databases.
83. What is hash-based indexing, and when is it used?
84. What are tree-based indexing structures, and why are they important?
85. Compare and contrast hash-based and tree-based indexing.

86. What is the significance of choosing the right file organization for a database?
87. How can indexing improve the performance of database queries?
88. What is the intuition behind the structure of tree indexes?
89. Explain the concept of Indexed Sequential Access Methods (ISAM).
90. What is a B+ tree, and how does it differ from a regular binary tree?
91. How do B+ trees maintain data order?
92. What is the primary advantage of using B+ trees for indexing?
93. What is an overflow block in B+ trees?
94. What is a leaf node in a B+ tree?
95. How does a B+ tree support range queries efficiently?
96. Name a common database management system that uses B+ trees for indexing.
97. What is a sparse index?
98. How does clustering factor affect the performance of an index?
99. What is the purpose of a root node in an index structure?
100. How does indexing impact insertion and deletion operations in a database?
101. What is a composite index, and when is it used?
102. Explain the concept of multi-level indexing.
103. What is the difference between a dense index and a sparse index?
104. How does indexing impact the storage requirements of a database?
105. What is the role of a data dictionary in database management?
106. Why is data compression sometimes used in index structures?
107. What is the goal of performance tuning in database systems?
108. How can an index reduce disk I/O during query processing?
109. What is a non-clustered index, and how does it differ from a clustered index?

110. Explain the concept of prefix compression in indexing.
111. How does a B+ tree handle duplicate keys?
112. What is a covering index, and why is it useful?
113. What is a bitmap index, and in what scenarios is it effective?
114. How does the choice of index affect the execution plan of a query?
115. What is a fill factor in the context of index pages?
116. How can indexing improve the performance of joins in a database?
117. What is a heap file organization, and when is it suitable?
118. What is a sequential file organization, and how is it different from an indexed file?
119. Explain the concept of indexing by hashing.
120. What is the role of a search key in an index structure?
121. What is the primary purpose of a secondary index?
122. How can indexing help in avoiding full table scans in a database?
123. What is an index seek operation in query processing?
124. What is an index scan operation in query processing?
125. How does the choice of indexing method impact database write operations?