Big Data

1. Define the term Big Data. Discuss its key characteristics using the 5V model, supported by real-world examples.
2. Compare traditional relational database management systems (RDBMS) with Big Data frameworks in terms of scalability, performance, and data structure handling.
3. Describe how Big Data architecture supports scalability, high availability, and fault tolerance.
4. Define and differentiate between structured, semi-structured, and unstructured data. Provide relevant examples to illustrate each category.
5. Discuss the role and significance of Big Data in contemporary industries such as healthcare, banking, logistics, and e-commerce.
6. What are the ethical concerns associated with Big Data collection and usage?
7. What is data cleaning? Discuss its importance in the data analysis pipeline and the key challenges involved in the process.
8. Identify and explain the major challenges associated with Big Data storage, processing, and analysis.
9. Define data cleaning in the context of data preprocessing. Why is it considered foundational to data quality and analytics integrity?
10. What is Apache Hadoop? Describe its architectural components and ecosystem.
11. Explain the architecture and working mechanism of the Hadoop Distributed File System (HDFS).
12. Define data blocks in HDFS. Discuss how block size and replication contribute to fault tolerance and reliability.
13. Elaborate on the roles and responsibilities of the NameNode, DataNode, and Secondary NameNode in HDFS.
14. Describe the mechanism by which Hadoop handles DataNode failures and maintains data availability.
15. Explain the limitations of Hadoop 1.x and describe how Hadoop 2.x addressed these issues. Highlight major architectural changes.
16. Explain the procedure for reading and writing data in HDFS from a client's perspective.
17. Discuss the functionality of JobTracker and TaskTracker in the legacy Hadoop 1.x framework.
18. Describe the architectural components of YARN, including the ResourceManager, NodeManager, and ApplicationMaster.
19. Describe the MapReduce programming model. Illustrate its working with a relevant example.

20. Define batch processing and real-time processing. Contrast their core principles and operational differences.
21. Discuss the shuffle and sort phase in MapReduce. Why is it considered critical for data aggregation?
22. How does MapReduce ensure fault tolerance during job execution and task failures?
23. Elaborate on the process of input data splitting and task assignment to mapper instances.
24. What is Apache Spark? Compare its architecture and capabilities with Hadoop MapReduce.
25. Discuss the concept of in-memory computation in Spark. How does it enhance performance?
26. Compare batch processing and stream processing in the context of Spark and Hadoop.
27. Compare NoSQL systems with traditional relational databases in terms of scalability, schema design, and consistency.
28. Compare column-oriented and document-oriented NoSQL databases with suitable use cases.
29. Define NoSQL. Explain the various types of NoSQL databases with relevant examples.
30. Describe the differences between *vertical scaling* in RDBMS and *horizontal scaling* in HDFS.