Seekho Bigdata
Data is the New Oil

2026 – Before You Start Interviews

Every PySpark Developer MUST Know These Interview Questions

Karthik Kondpak
9989454737

2026 — Before You Start Interviews

Every PySpark Developer MUST Know These Interview Questions

Spark Core & Architecture

1. How does Spark execute a job from Action → DAG → Stage → Task?
2. What exactly triggers a shuffle, and why is it expensive?
3. How does Spark decide the number of tasks per stage?
4. What happens internally when an executor is lost? 5. Explain lazy evaluation with a real production example. 6. How does Spark handle fault tolerance without data replication?
7. Difference between Job, Stage, and Task with a real scenario.
8. What information does the DAG Scheduler vs Task Scheduler manage?

DataFrames, RDDs & Spark SQL

9. Why are DataFrames faster than RDDs internally?
10. When would you still prefer RDDs over DataFrames?
11. How does Catalyst Optimizer transform a query?
12. What is logical plan vs physical plan?
13. How does Tungsten improve Spark performance?
14. What happens when you use a UDF in Spark SQL?

15. Why are Python UDFs slower than Spark SQL expressions?

Joins (VERY IMPORTANT)

16. How does Spark choose between Broadcast, Sort-Merge, and Shuffle Hash Join?
17. When does a broadcast join fail in production?
18. How do you handle data skew in joins?
19. What is salting, and when does it backfire?
20. How does AQE change join strategies at runtime?
21. Why can a join be slow even when both tables are small?

Partitioning & Shuffle

22. What happens if you have too many small partitions?
23. Repartition vs Coalesce — when does each hurt performance?
24. How does spark.sql.shuffle.partitions really work?
25. Why does repartition cause a full shuffle?
26. How do you decide optimal partition size in production?
27. What is map-side vs reduce-side processing?

Memory Management & Performance

28. How is executor memory divided internally?

29. Why can a job be slow even with high executor memory?
30. What causes GC overhead exceeded errors?
31. How does Spark spill data to disk?
32. Difference between storage memory and execution memory.
33. What happens when cache doesn't fit in memory?
34. Why does increasing executors sometimes slow down the job?

Caching & Persistence

35. Cache vs Persist — what's the real difference?
36. When should you never cache a DataFrame?
37. What happens if cached data is evicted?
38. How do different Storage Levels impact performance?

File Formats & Storage

39. Why is Parquet preferred over CSV/JSON? What is
40. predicate pushdown? How does column pruning
41. reduce IO? What causes the small files problem?
42. How do you fix small files in Spark? Difference
43. between append vs overwrite modes.
44.

Delta Lake / ACID (Common in 2026 Interviews)

45. How does Delta Lake provide ACID transactions?
46. What is Delta log, and why is it important?
47. Difference between MERGE INTO vs INSERT OVERWRITE.
48. How does time travel work internally?
49. What causes Delta table performance degradation?
50. How do you optimize a Delta table?

Streaming (Batch + Real-Time)

51. Difference between DStreams and Structured Streaming.
52. What is exactly-once processing in Spark?
53. How does checkpointing work in streaming?
54. What happens if a streaming job restarts?
55. Watermark vs Window — explain with a late-arriving data case.

Debugging & Monitoring

56. How do you debug a slow Spark job using the Spark UI?
57. Which metrics indicate a shuffle bottleneck?
58. How do you identify data skew from Spark UI?
59. What logs do you check when a job fails randomly?
60. How do you troubleshoot executor OOM issues?

Real Production Scenarios (INTERVIEW FAVORITES)

61. How would you process 1 billion records daily efficiently?
62. How do you join a huge fact table with multiple dimensions?
63. How do you design Spark jobs for incremental data loads?
64. How do you handle late-arriving data?
65. How do you optimize a job running fine in DEV but slow in PROD?
66. What Spark optimizations did you apply that gave the maximum impact?

PySpark Coding & Best Practices

67. Why should you avoid .collect() in production?
68. Difference between .count() and .isEmpty() performance-wise?
69. How do you handle nulls safely in Spark?
70. Why are wide transformations more expensive than narrow ones?
71. How do you write Spark code that is scalable and fault-tolerant?

Advanced / 2026-Level Questions

72. How does Adaptive Query Execution work internally?

73. What changes did Spark 3+ introduce that improved performance?
74. How does Spark optimize skewed aggregations?
75. How does Spark decide broadcast threshold dynamically?
76. How do you tune Spark for cloud environments (EMR / Databricks)?

Seekho Bigdata

Data is the New Oil

Let's build your Data Engineering journey together!

Call us directly at: 9989454737

https://seekhobigdata.com/