WISSEN

Data Engineering Interview Questions

Ankita Gulati

Shubh Goyal

Job Details

- **Position:** Senior Data Engineer
- **Experience:** 5+ years
- **Location:** Bangalore / Pune
- **Work mode:** Hybrid
- **Compensation:** ₹20-24 LPA
- **Total Rounds:** 2
- **Top Required Skills:**

1. SQL
2. PySpark / Python
3. Cloud Data Engineering
4. ETL / Data Modeling
5. Big Data & Streaming

Ankita Gulati

Shubh Goyal

Round 1
Data Engineering & Cloud

1. Have you worked on Apache Hive? What role did it play in your projects?
2. Explain the concept of partitioning and bucketing in Hive. When would you use each?
3. How do you read Hive data using Python or PySpark?
4. What are the different types of partitions in Hive?
5. Explain User Defined Functions (UDFs) in Hive.
6. When you create UDFs in PySpark and compile them, what happens in the background? For example, if you create get_user_id vs get_user_id_upper, how would execution differ?
7. What are the different types of tables in Hive?
8. Suppose you have an external Hive table with millions of records in production. The user_id column is of type STRING, but you need to change it to BIGINT for arithmetic operations. How would you approach this safely in production?

Ankita Gulati

Shubh Goyal

9. How do you optimize joins in Hive?
10. What are the different types of serializations in Spark?
11. What does dynamic executor allocation in Spark Submit mean? How does it work?
12. What is the use of Accumulators in Spark?
13. What are the different partitioning techniques applied in an RDD?
14. What does fault tolerance mean in RDDs? How does Spark achieve it?
15. When submitting a job using spark-submit, how do you decide executor cores, memory, and driver memory values?
16. Suppose resources are unlimited. How would you decide whether to allocate 100 cores, 8GB executor memory, and 2GB driver memory?
17. Explain the major types of Spark transformations. What is the difference between narrow and wide transformations?
18. What is the difference between groupByKey and reduceByKey in Spark? Why can't we always use reduceByKey?

Ankita Gulati Shubh Goyal

Round 2
Applied Problem Solving

1. Nth Highest Salary (SQL) – Write a query to get the 3rd highest salary from an employee table without using TOP or LIMIT.
2. Duplicate Detection (SQL) – Find all customers having duplicate email IDs in a customer table.
3. Running Total (SQL) – Write a query to calculate a running total of daily sales per month.
4. Top-N Customers (SQL) – Find the Top 5 customers who spent the most in the last 30 days.
5. Gaps & Islands (SQL) – Given login/logout timestamps, identify continuous login streaks per user.
6. Pivoting (SQL) – Convert rows (month, revenue) into columns (Jan, Feb, Mar, … Dec).
7. Order Chunking (SQL) – If you have 1000 orders in a table, write logic to generate files with max 100 rows each, dynamically.

Ankita Gulati

Shubh Goyal

8. Anagram Check (Python) – Write Python code to check if two words (silent, listen) are anagrams.
9. Moving Average (Python) – Given stock price data, compute the 3-day moving average for each stock.
10. Sessionization (PySpark) – Given web logs (user_id, timestamp, url), group them into sessions with a 30-min inactivity window.
11. Top-N Words per User (PySpark) – From text logs (user_id, text), find the Top 3 most frequent words per user.
12. Frequency Sort (PySpark) – Given a list ["ABC", "ABC", "DEF", "XYZ", "XYZ", "XYZ"], output values sorted by frequency ascending.
13. Duplicate Removal (PySpark) – Deduplicate records in a DataFrame on (user_id, event_time) keeping only the latest record.
14. Transaction Aggregation (PySpark) – Compute daily transaction totals per user and save into Delta Lake with time-travel enabled.
15. Flatten Nested JSON (PySpark) – Given nested JSON logs in Hive, write PySpark code to flatten the structure and store as partitioned Parquet.

Thank You

Best of luck with your upcoming interviews — you've got this!

Ankita Gulati

Shubh Goyal