

Ananya Rao

Data Engineer

Acme Analytics Pvt Ltd

Good

Email	Phone	Interview Date	Interviewer
ananya.rao@example.com	+91 98765 43210	12-11-2026	Vikram Nair

Watch Recording

Overall Summary

Ananya showed strong data engineering fundamentals and explained her approach in a structured, practical way, backed by examples from her own projects. She was confident with SQL, PySpark, pipeline design and lakehouse architecture, and handled follow-up questions well. Communication was clear throughout. Python depth and formal data modelling could be checked in the next round. Overall a good fit for the role. Recommended for the next round.



Overall Rating

Strengths

- Explained a Microsoft Fabric project end to end: Lakehouse for raw and cleaned data, Warehouse for SQL reporting and Dataflows for light transformations.
- Described incremental, idempotent ETL/ELT pipelines with watermarks, retries and upserts, so reruns never lose or duplicate data.
- Showed practical PySpark tuning: broadcast joins, partition sizing, handling skew and reading the Spark UI to find slow stages.
- Wrote a correct SQL solution for deduplication, aggregation and top-N ranking, and explained each step clearly.
- Gave a clear picture of the medallion (bronze, silver, gold) lakehouse design and of Delta Lake's ACID guarantees.

Areas of Improvement

- Core Python beyond PySpark (packaging, testing, typing) was not tested directly; a short Python exercise would confirm depth.
- Formal data modelling (star schemas, slowly changing dimensions) was covered only briefly and could be probed further.

Data quality checks and governance were mentioned, but without a concrete framework or example from production.

Technical Skills



Microsoft Fabric: 8

Databricks: 7

SQL - Advanced queries, performance tuning, data transformation (7/10 minimum): 9

PySpark - Data transformation, joins, aggregations, optimization (7/10 minimum): 8

Python: 8

ETL/ELT Pipeline Development: 9

Data Modeling & Schema Design: 7

Azure Data Factory (ADF): 8

Data Lake / Lakehouse Architecture: 9

8.0 / 10 Microsoft Fabric

CANDIDATE BACKGROUND

YEARS	TYPE	PROFICIENCY
2y	Hands-on	Advanced

Expected 7/10 · Exceeds +1.0

Comment

Described a Fabric project with Lakehouse, Warehouse and Dataflows, including when to use each, and how a medallion layout kept reporting data consistent. Also covered a real problem with incremental loads and how it was solved.

7.0 / 10 Databricks

CANDIDATE BACKGROUND

YEARS	TYPE	PROFICIENCY
1y	Hands-on	Intermediate

Expected 7/10 · Met +0.0

Comment

Outlined how she would set up a new Databricks workspace: Unity Catalog, Auto Loader for incremental ingestion and bronze/silver/gold jobs. Answers were correct but more general than specific, so rated at the expected level rather than above it.

9.0 / 10 SQL - Advanced queries, performance tuning, data transformation (7/10 minimum)

CANDIDATE BACKGROUND

YEARS	TYPE	PROFICIENCY		
5y	Hands-on	Advanced	Expected 7/10	Exceeds +2.0

Comment

Wrote a complete query that filters successful transactions, removes duplicates with ROW_NUMBER(), totals amounts per customer and ranks the top three with DENSE_RANK(). Explained WHERE vs HAVING, UNION vs UNION ALL and how to read an execution plan.

8.0 / 10 PySpark - Data transformation, joins, aggregations, optimization (7/10 minimum)

CANDIDATE BACKGROUND

YEARS	TYPE	PROFICIENCY		
3y	Hands-on	Advanced	Expected 7/10	Exceeds +1.0

Comment

Discussed processing large datasets in PySpark: Parquet, partitioning, broadcast joins, the small-file problem and how to find skew in the Spark UI. Answers reflected hands-on tuning experience.

8.0 / 10 Python

CANDIDATE BACKGROUND

YEARS	TYPE	PROFICIENCY		
4y	Hands-on	Intermediate	Expected 7/10	Exceeds +1.0

Comment

Python was mostly seen through PySpark work. General Python skills (testing, packaging, libraries) were not asked about directly, so the rating is based on indirect evidence.

9.0 / 10 ETL/ELT Pipeline Development

CANDIDATE BACKGROUND

YEARS	TYPE	PROFICIENCY		
5y	Hands-on	Advanced	Expected 7/10	Exceeds +2.0

Comment

Explained incremental pipelines with watermarks updated only after a successful load, retries and idempotent upserts. Clear grasp of orchestration and dependency handling.

7.0 / 10 Data Modeling & Schema Design

CANDIDATE BACKGROUND

YEARS	TYPE	PROFICIENCY		
3y	Hands-on	Intermediate	Expected 7/10	Met +0.0

Comment

Organised data well across bronze, silver and gold layers with sensible partitioning. Formal modelling techniques such as star schemas and SCDs were only touched on.

8.0 / 10 Azure Data Factory (ADF)

CANDIDATE BACKGROUND

YEARS	TYPE	PROFICIENCY		
3y	Hands-on	Advanced	Expected 7/10	Exceeds +1.0

Comment

Named and explained the main Azure Data Factory components (pipelines, activities, datasets, linked services, integration runtimes and triggers) and how they fit together.

9.0 / 10 Data Lake / Lakehouse Architecture

CANDIDATE BACKGROUND

YEARS	TYPE	PROFICIENCY		
3y	Hands-on	Advanced	Expected 7/10	Exceeds +2.0

Comment

Showed a clear understanding of lakehouse architecture: data flowing from bronze to gold, storage on ADLS Gen2 and Delta Lake for ACID transactions and time travel.

Behavioral Skills

9.0 / 10	Communication	
9.0 / 10	Industry Awareness	
9.0 / 10	Engineering Mindset	
9.0 / 10	Attitude	
10.0 / 10	Problem Solving	
9.0 / 10	Analytical Skill	

Overall Behavioral Comments

Ananya spoke clearly and kept her answers structured, even when the interviewer asked follow-ups mid-answer. She stayed calm, asked for clarification when a question was ambiguous and took feedback on the SQL exercise positively.

Candidate Interview Snapshot



Complete Interview Q&A

Q1. Can you walk us through a project where you used Microsoft Fabric, and the role of the Lakehouse, Warehouse and Dataflows in it?

A. Ananya described a retail analytics platform that brought sales and inventory data from several systems into Fabric. Raw files landed in the Lakehouse (bronze), were cleaned in silver and modelled in gold, with the Fabric Warehouse serving SQL reporting and Dataflows used for simple transformations.

Q2. What are the main differences between a Fabric Lakehouse and a Fabric Warehouse, and when would you choose each?

A. She explained that the Lakehouse suits large, varied data and Spark processing, while the Warehouse suits structured data and SQL analytics. She would use both together: Lakehouse for engineering layers, Warehouse for reporting.

Q3. Tell us about a challenge you faced on that project and how you solved it.

A. Daily incremental loads occasionally failed mid-run. She made the pipeline idempotent: the watermark is moved only after a successful load, and upserts (MERGE) prevent duplicates when a run is retried.

Q4. Where do Dataflows fit in your ingestion and transformation approach?

A. Dataflows (Power Query) handle light, low-code clean-up such as removing duplicates and fixing nulls. Heavier or high-volume transformations go to PySpark notebooks, orchestrated by pipelines.

Q5. Given a transactions table with duplicates, write a SQL query that keeps only successful transactions, removes duplicates keeping the latest record, totals the amount per customer and returns the top three customers.

A. She wrote a query using ROW_NUMBER() to keep the latest copy of each transaction, SUM() to total per customer and DENSE_RANK() to return the top three, filtering on status = 'SUCCESS' first. See the Coding Skills tab.

Q6. Why did you use ROW_NUMBER() there instead of DISTINCT?

A. Because the duplicate rows are not identical: they differ in updated_timestamp. ROW_NUMBER() partitioned by transaction_id and ordered by updated_timestamp descending keeps exactly the latest version.

Q7. How do you approach processing and transforming data in a distributed environment?

A. She uses PySpark on a cluster, stores data as partitioned Parquet or Delta, broadcasts small lookup tables to avoid shuffles and checks partition sizes so work is spread evenly.

Q8. If you were starting a new project on Databricks, what would your first steps be?

A. Set up the workspace and Unity Catalog, connect the sources, use Auto Loader for incremental file ingestion and build bronze, silver and gold jobs with clear ownership and tests.

Q9. What other managed Spark or distributed platforms have you worked with?

A. Azure Synapse Spark pools and Microsoft Fabric Spark, plus some Databricks in a proof of concept. She compared their cluster management and pricing briefly.

Q10. What is the difference between WHERE and HAVING?

A. WHERE filters rows before aggregation; HAVING filters groups after aggregation. She gave an example of filtering customers whose total spend exceeds a threshold.

Q11. What is the difference between UNION and UNION ALL?

A. UNION removes duplicate rows and is slower; UNION ALL keeps all rows and is faster when duplicates are not a concern.

Q12. What is database tuning, and how would you approach it?

A. She gave a partial answer covering indexes and avoiding SELECT *, but did not discuss statistics, partitioning strategy or workload analysis.

Q13. What is a query execution plan and how do you use it?

A. The optimiser's plan shows how a query will run: scans, joins, index use and estimated costs. She uses it to spot full scans and expensive joins before rewriting a query.

Q14. How would you use Fabric Lakehouse and Warehouse together across the medallion layers?

A. Bronze and silver in the Lakehouse with Spark and Dataflows, gold tables exposed to the Warehouse for SQL and Power BI, and CDC or incremental pipelines to keep gold up to date.

Q15. What are the main components of Azure Data Factory?

A. Pipelines, activities, datasets, linked services, integration runtimes and triggers, with a short explanation of what each one does.

Q16. What is ADLS Gen2?

A. Azure's data-lake storage built on Blob Storage, with a hierarchical namespace and fine-grained ACLs, commonly used as the storage layer for lakehouse platforms.

Q17. What is the small file problem, and how do you fix it?

A. Many small files slow down reads and overload metadata. She would compact them with coalesce/repartition or Delta OPTIMIZE, and choose sensible partition columns.

Q18. What is Delta Lake?

A. An open table format on top of Parquet that adds ACID transactions, schema enforcement, MERGE/UPDATE support and time travel.

Q19. A Spark job that used to take 20 minutes now takes two hours. How would you troubleshoot it?

A. Compare with a good run, check the Spark UI for skewed or spilling stages, look for new large joins that should be broadcast, check input size and partition counts, and only then consider a bigger cluster.

Q20. How would you design a scalable Azure data platform from scratch?

A. Start from sources, volumes and reporting needs; land data in ADLS Gen2 or OneLake; build medallion layers with pipelines for orchestration and Spark for processing; serve gold data to Power BI and APIs, with security and monitoring designed in from the start.

Coding

Q1 You have a table called transactions: transaction_id customer_id transaction_date amount status updated_timestamp. A customer can have multiple transactions, and duplicate transaction records may exist. Task: Write a SQL query to: Consider only SUCCESS transactions. Remove duplicate transaction records, keeping the latest record based on updated_timestamp. Calculate the total transaction amount for each customer. Find the top 3 customers by total transaction amount.

Code

```
WITH latest AS (
    SELECT t.*,
           ROW_NUMBER() OVER (
               PARTITION BY transaction_id
               ORDER BY updated_timestamp DESC
           ) AS rn
    FROM transactions t
    WHERE status = 'SUCCESS'
),
customer_totals AS (
    SELECT customer_id,
           SUM(amount) AS total_amount
    FROM latest
    WHERE rn = 1
    GROUP BY customer_id
),
ranked AS (
    SELECT customer_id,
           total_amount,
           DENSE_RANK() OVER (ORDER BY total_amount DESC) AS rnk
    FROM customer_totals
)
SELECT customer_id, total_amount
FROM ranked
WHERE rnk <= 3
ORDER BY total_amount DESC;
```