AR

Ananya Rao

Data Engineer

Acme Analytics Pvt Ltd

Good

Email	Phone	Interview Date	Interviewer
ananya.rao@example.com	+91 98765 43210	05-11-2026	AI

Overall Summary

Ananya communicates clearly and backs her answers with specific examples from her own projects. She explained a Microsoft Fabric lakehouse she built, how she made incremental loads idempotent, and how she tunes slow PySpark jobs, and she answered SQL follow-ups accurately. Python and data-quality practices were covered in less depth, and formal data modelling could be explored further. Overall she meets the expectations for the Data Engineer role and is a good fit to move forward.



Overall Rating

Recommendation

Move Ananya to the expert panel round. Use it to probe data modelling (star schemas, SCDs) and data-quality frameworks.

Strengths

- Explains real project work with specifics: sources, layers, tools and the reasons behind each choice.
- Strong grasp of incremental, idempotent pipelines (watermarks, MERGE upserts, safe reruns).
- Practical PySpark tuning experience: broadcast joins, partition sizing and reading the Spark UI.
- Accurate SQL knowledge, including window functions and execution plans.
- Clear, structured communication and good follow-up answers.

Areas of Improvement

- Python beyond PySpark (testing, packaging) was described only briefly.
- Formal data modelling (star schemas, slowly changing dimensions) needs a deeper check.
- Data-quality checks were mentioned without a named framework or production example.

Technical Skills



Microsoft Fabric (Lakehouse | Warehouse | Dataflows): 8

SQL | Window Functions | Query Tuning: 8

PySpark | Distributed Processing: 7

Azure Data Factory | Orchestration: 7

Python: 6

Data Modelling: 6

Git | CI/CD: 7

8.0 / 10 Microsoft Fabric (Lakehouse | Warehouse | Dataflows)

Comment

Described a Fabric lakehouse she built with bronze, silver and gold layers, the Warehouse for SQL reporting and Dataflows for light clean-up, with clear reasons for each.

8.0 / 10 SQL | Window Functions | Query Tuning

Comment

Explained ROW_NUMBER() for deduplication, DENSE_RANK() for top-N and how she reads execution plans to remove full scans.

7.0 / 10 PySpark | Distributed Processing

Comment

Gave concrete tuning steps from her own work: broadcast joins, repartitioning and skew handling, cutting a nightly job from 2 hours to 25 minutes.

7.0 / 10 Azure Data Factory | Orchestration

Comment

Named the main ADF components and how she used pipelines, triggers and integration runtimes for daily loads.

6.0 / 10 Python

Comment

Python came up mostly through PySpark notebooks; testing and packaging practices were described only briefly.

6.0 / 10 Data Modelling

Comment

Organised data well across medallion layers; star schemas and slowly changing dimensions were only touched on.

7.0 / 10 Git | CI/CD

Comment

Uses feature branches and pull requests in Azure DevOps, with deployment pipelines for notebooks and pipeline definitions.

Behavioral Skills

7.0 / 10	Industry Awareness	
8.0 / 10	Engineering Mindset	
8.0 / 10	Attitude	
7.0 / 10	Team Work	
8.0 / 10	Analytical Skill	

Overall Behavioral Comments

Industry Awareness: Referred to current tools such as Microsoft Fabric, Delta Lake and Unity Catalog, and when each fits. Engineering Mindset: Designs for reruns and failure: moved the watermark only after a successful load and used upserts to avoid duplicates. Attitude: Positive and open; acknowledged gaps honestly (for example, limited Databricks production use) and explained how she would close them. Team Work: Described agreeing data contracts with the reporting team and handing over runbooks to the support team. Analytical Skill: Broke down a slow Spark job step by step: compared with a good run, checked skew and joins, then tuned.

Complete Interview Q&A

Q1. How are you doing today?

A. I'm doing well, thank you. Looking forward to this.

Q2. Could you share a brief self-introduction?

A. I'm Ananya, a data engineer with about five years of experience. I currently work on a retail analytics platform, mainly with Microsoft Fabric, Azure Data Factory and PySpark. Before that I worked on transaction analytics for a payments company.

Q3. May I have your permission to begin the main round?

A. Yes, please go ahead.

Q4. What made you choose data engineering as a career?

A. I enjoy building the foundations that other teams depend on. When a pipeline is reliable, analysts and business teams can trust the numbers, and I like solving the problems that make that possible.

Q5. Can you walk me through a project where you used Microsoft Fabric?

A. On my current project we bring sales, inventory and returns data from five systems into Fabric. Raw files land in the Lakehouse as bronze, we clean and deduplicate them into silver, and build gold tables for reporting. The Fabric Warehouse sits on top of gold for SQL and Power BI.

Q6. Why did you use both a Lakehouse and a Warehouse, rather than just one?

A. The Lakehouse suits large, mixed data and Spark processing, so we use it for the engineering layers. The Warehouse is better when people mainly query structured data with SQL, so the reporting team works there. Using both kept each team in the tool that suits them.

Q7. Tell me about a problem you faced on that project and how you solved it.

A. Our daily incremental load sometimes failed halfway. If the watermark had already moved, the next run skipped records. I changed it so the watermark only moves after a successful load, and switched the writes to MERGE, so a rerun can't create duplicates. Since then we can rerun any day safely.

Q8. How would you remove duplicate records and keep only the latest version of each one in SQL?

A. I'd use ROW_NUMBER() partitioned by the record's ID and ordered by the updated timestamp descending, then keep rows where the number is one. DISTINCT doesn't work when duplicates differ in the timestamp.

Q9. And how would you find the top three customers by total spend?

A. Sum the amount per customer from the deduplicated rows, then apply DENSE_RANK() over the total in descending order and keep ranks one to three. DENSE_RANK keeps ties without skipping ranks.

Q10. How do you use a query execution plan?

A. I check it for full table scans, expensive joins and bad row estimates. Then I add or fix indexes, filter earlier, or rewrite the join, and compare the plan again.

Q11. Tell me about a time you made a slow Spark job faster.

A. A nightly job at my previous company took about two hours. In the Spark UI one stage was heavily skewed, and a large table was joined to a small lookup table. I broadcast the lookup table and repartitioned on a better key, and the job came down to about 25 minutes.

Q12. Which Azure Data Factory components have you used, and how?

A. Pipelines with copy and data flow activities, datasets and linked services for the sources, self-hosted integration runtimes for on-premises SQL Server, and schedule triggers for the daily runs.

Q13. How do you use Python outside of PySpark?

A. Mostly for utility scripts, like validating files before ingestion and calling APIs. I use pytest for a few of those scripts, but I'd like to do more structured testing.

Q14. How do you design your data model for reporting?

A. I keep the gold layer close to what reports need: fact tables for sales and returns and dimension tables for stores and products. I have handled slowly changing store details with a simple history table, though I haven't built a full SCD Type 2 framework.

Q15. How do you check data quality in your pipelines?

A. We have row-count and null checks after each load, and alerts if the counts differ too much from the previous day. I've read about Great Expectations but haven't used it in production yet.

Q16. How do you manage code and deployments for your data pipelines?

A. Everything is in Git in Azure DevOps. We work on feature branches with pull requests and reviews, and deployment pipelines promote notebooks and pipeline definitions from dev to test to production.

Q17. Tell me about a time you worked closely with another team.

A. The reporting team kept getting surprised by schema changes. We agreed a simple data contract for the gold tables and a change notice before any column changes. After that, report breakages mostly stopped.

Q18. Do you have any questions for me about the role or the team?

A. Yes. What does the team's current data stack look like, and is there a plan to move any workloads to Databricks?

Q19. Is there anything else you'd like to ask?

A. No, that's all. Thank you for your time.