Deloitte.

Data Engineer Interview Questions (3 YOE)

1. Design an ingestion pipeline in Azure Data Factory for batch and streaming data

What they asked me

"Design an ingestion pipeline in ADF for both batch and streaming data into ADLS and Synapse."

What I said

I designed a hybrid ingestion architecture:

Batch ingestion:

- Source systems: SQL Server, Oracle, flat files, REST APIs
- ADF Copy Activity used for batch ingestion
- Data landed into ADLS Gen2 Bronze layer
- Folder structure partitioned by source and date
- Schema stored as-is (schema-on-read)

Streaming ingestion:

- Real-time events ingested via Event Hub
- Azure Databricks Structured Streaming reads from Event Hub
- Data written to ADLS Delta tables (Bronze)
- Checkpointing enabled for fault tolerance

Downstream:

- Databricks transforms data into Silver and Gold layers

- Synapse Serverless used for analytics
- Power BI consumes curated Gold data

Tips

- Deloitte expects clarity between batch vs streaming design
- Mention fault tolerance, checkpointing, and scalability

---

2. Incremental loading (CDC) using watermark or change tracking

What they asked me

"How would you implement incremental loading in ADF?"

What I said

I implemented a watermark-based CDC approach:

- Metadata table stores last successful load timestamp
- ADF Lookup fetches watermark value
- Source query filtered using watermark column
  `WHERE modified_date > @watermark`
- Incremental data loaded into Bronze layer
- Delta MERGE used in Databricks for upserts
- Watermark updated only after successful load

For SQL Server sources:

- Use Change Tracking or CDC tables
- Capture INSERT, UPDATE, DELETE operations
- Process changes incrementally

Tips

- Deloitte prefers metadata-driven CDC
- Always mention idempotency and rollback safety

---

3. Optimize a Spark job with skew or memory issues

What they asked me

"How would you optimize a Databricks Spark job suffering from skew or memory issues?"

What I said

I followed a structured optimization approach:

- Identify skewed keys using Spark UI
- Apply broadcast join for small dimension tables
- Use salting technique for heavily skewed keys
- Repartition data by join key
- Filter early and select only required columns
- Enable Adaptive Query Execution
- Increase executor memory if required
- Avoid unnecessary caching

Tips

- Deloitte values reasoning over listing techniques
- Always mention Spark UI and metrics

---

4. Small Parquet files problem and solution

What they asked me

"You see too many small Parquet files in the lake. Why is it a problem and how do you fix it?"

What I said

Problem:

- Small files increase metadata overhead
- Slow query performance
- Inefficient Spark job planning
- High cost in cloud storage operations

Solution:

- Use repartition or coalesce before writing
- Schedule compaction jobs
- Use Delta OPTIMIZE where applicable

- Control micro-batch sizes in streaming
- Avoid frequent small writes

Tips

- This is a very common Deloitte question
- Always connect it to performance and cost

---

5. SQL: 2nd highest salary using window function

What they asked me

"Write a SQL query to find the 2nd highest salary."

What I said

```
SELECT salary
FROM (
    SELECT salary,
           DENSE_RANK() OVER (ORDER BY salary DESC) AS rnk
    FROM employee
) t
WHERE rnk = 2;
```

Tips

- Deloitte expects window functions, not subquery hacks

---

6. Design a modern Azure data warehouse (batch + real-time)

What they asked me

"How would you design a modern Azure data warehouse for multiple clients?"

What I said

I designed a scalable, multi-tenant architecture:

- Ingestion: ADF (batch), Event Hub (streaming)

- Storage: ADLS Gen2 with Bronze/Silver/Gold
- Processing: Databricks with PySpark
- Storage format: Delta Lake
- Serving: Synapse Serverless / Dedicated Pool
- Security: RBAC, data masking, client isolation
- Monitoring: Log Analytics, pipeline metrics

Tips

- Deloitte focuses on enterprise-grade design
- Mention multi-client isolation and governance

---

7. Heavy shuffle during Spark join

What they asked me

"You see heavy shuffle during a Spark join. What do you do?"

What I said

- Check Spark UI for shuffle stages
- Broadcast small tables
- Use salting for skewed keys
- Repartition by join key
- Avoid unnecessary joins
- Enable AQE
- Cache only reused datasets

Tips

- Mention shuffle size and task skew explicitly

---

8. Explain end-to-end data flow (ADF → ADLS → Databricks → Synapse → Power BI)

What they asked me

"Explain the complete data flow."

What I said

- ADF ingests data from source to ADLS Bronze
- Databricks cleans and standardizes data (Silver)
- Business transformations applied in Gold
- Delta Lake ensures ACID and versioning
- Synapse queries curated data
- Power BI builds dashboards on Gold layer

Tips

- Deloitte wants clean logical flow, not buzzwords

---

9. Partial ETL failure recovery without full reload

What they asked me

"A nightly ETL failed after partial load. How do you recover?"

What I said

- Use batch ID and watermark to identify failed data
- Roll back partial writes using Delta transactions
- Reprocess only failed partitions
- Ensure idempotent MERGE logic
- Update audit table post recovery

Tips

- Never say "full reload" unless forced

---

10. Design CDC using SQL Server + ADF + Event Hub/Kafka

What they asked me

"How would you design CDC using SQL Server and Event Hub?"

What I said

- Enable CDC or Change Tracking in SQL Server
- Capture changes into CDC tables
- ADF pulls changes incrementally
- Optionally push changes to Event Hub
- Databricks consumes events for near real-time processing
- Store data in Delta Lake

Tips

- Deloitte likes hybrid batch + streaming CDC answers

---

11. Batch runtime increased from 1 hour to 3 hours

What they asked me

"What steps do you take for RCA?"

What I said

- Compare historical pipeline metrics
- Check data volume growth
- Analyze Spark UI for shuffle/skew
- Validate source system slowness
- Check small file explosion
- Review recent code or schema changes
- Optimize or scale resources accordingly

Tips

- Root cause analysis is a key Deloitte skill

---

12. Improving pipeline performance or cost

What they asked me

"Tell me about a project where you improved performance or cost."

What I said

- Reduced pipeline runtime by optimizing joins

- Implemented incremental loads
- Reduced storage cost via compaction
- Tracked metrics: runtime, cost, SLA adherence

Tips

- Always quantify impact

---

13. Handling conflicts with business teams

What they asked me

"How do you handle priority conflicts?"

What I said

- Understand business urgency
- Present technical constraints clearly
- Offer short-term workaround and long-term fix
- Align on impact and delivery timeline

Tips

- Deloitte evaluates communication maturity here

---

14. Delivering under tight deadlines

What they asked me

"Describe a critical delivery under pressure."

What I said

- Broke work into milestones
- Prioritized critical data paths
- Automated validations
- Communicated progress daily
- Delivered MVP first, enhancements later

Tips

- Focus on structure, not hero stories

---

15. Ensuring data quality, monitoring, and observability

What they asked me

"How do you ensure production-grade quality?"

What I said

- Data quality checks at Silver layer
- Row count reconciliation
- Schema validation
- Error handling with audit tables
- Alerts for failures or anomalies
- Dashboards for pipeline health

Tips

- Deloitte expects enterprise observability thinking