Deloitte.

Data Engineer

Round -2

Experience: 0-4 Year

1. End-to-End Architecture Design

What they asked me:

"How would you architect a data platform for a fintech application supporting batch + real-time processing?"

What I said:

"I follow a Lakehouse architecture.

Real-time (Kafka/Event Hub → Databricks Streaming → Delta Bronze → Silver → Gold).
- Use watermarking, deduplication, and schema evolution.
- Write data into partitioned Delta tables.

Batch (ADF → ADLS → Databricks → Delta).
- Use incremental loads based on last_modified_date or surrogate keys.

Governance:
- Unity Catalog for data masking, lineage, and access control.
- Key Vault for credentials.

Consumption:
- Power BI on Gold tables."

Tips:

- Always say "Lakehouse architecture".
- Add governance details (Unity Catalog).

2. SQL: Remove duplicates but keep latest record

What they asked me:

"Write a query to remove duplicates keeping the latest row."

What I said:

```
WITH Dedup AS (
  SELECT *,
    ROW_NUMBER() OVER(PARTITION BY customerId ORDER BY updatedAt DESC) AS
rn
  FROM Orders
)
SELECT *
FROM Dedup
WHERE rn = 1;
```

Tips:

- Use ROW_NUMBER.
- Keep it simple and clean.

3. Pipeline success but wrong data (RCA process)

What they asked me:

"How will you find root cause if pipeline is successful but data is wrong?"

What I said:

"I check in this order:

1. Compare row counts between source and target.
2. Validate schema drift.
3. Inspect timestamp or incremental logic.
4. Check joins (null explosion or unintended inner joins).
5. Review business logic layer.
6. Check partition overwrite issues.
7. Validate whether stale cache or old checkpoint was used."

Tips:

- Always mention "row count audit".
- Mention "checkpoint corruption".

4. Data Quality Framework Design

What they asked me:

"How do you automate data quality checks?"

What I said:

"I build a metadata-driven framework.
For each table, I store rules:

- Null checks
- Data type checks
- Referential integrity
- Duplicate checks
- Threshold validation (min/max)
- Business rules (e.g., amount > 0)

I run these rules in Databricks before loading Silver.
Failed records go to a quarantine zone."

Tips:

- Say "metadata-driven".
- Mention "quarantine zone".

5. SCD Type 2 with multiple changes in a day

What they asked me:

"If customer updates phone number twice in a day, what happens?"

What I said:

"Two new SCD2 records are created.
Each one gets its own startDate, endDate and version.
The latest record has endDate = NULL and isCurrent = 1."

Tips:

- Mention "historical tracking".

6. Spark job optimization (40 mins → under 10 mins)

What they asked me:

"How will you reduce execution time of a slow Spark job?"

What I said:

"I optimize in this order:

1. Reduce shuffle: broadcast joins, bucketing.
2. Correct partitioning: repartition by business key.
3. Enable AQE (Adaptive Query Execution).
4. Cache only reused data.
5. Avoid wide transformations inside loops.
6. Use Delta instead of CSV/Parquet.
7. Tune cluster: more executors, correct memory overhead."

Tips:

- Mention “AQE reduces skew”.

7. Lakehouse Migration

What they asked me:

“How do you migrate from SQL DW to Lakehouse?”

What I said:

“Steps:

1. Identify fact & dimension tables.
2. Export to ADLS staged Parquet.
3. Create Bronze/Silver/Gold Delta layers.
4. Copy data using ADF bulk copy.
5. Rebuild SCD logic using MERGE.
6. Apply governance with Unity Catalog.
7. Redirect BI dashboards to Gold.”

Tips:

- Add “backfill + incremental sync”.

8. Incremental load without last_modified_date

What they asked me:

“How do you design incremental load when no timestamp is available?”

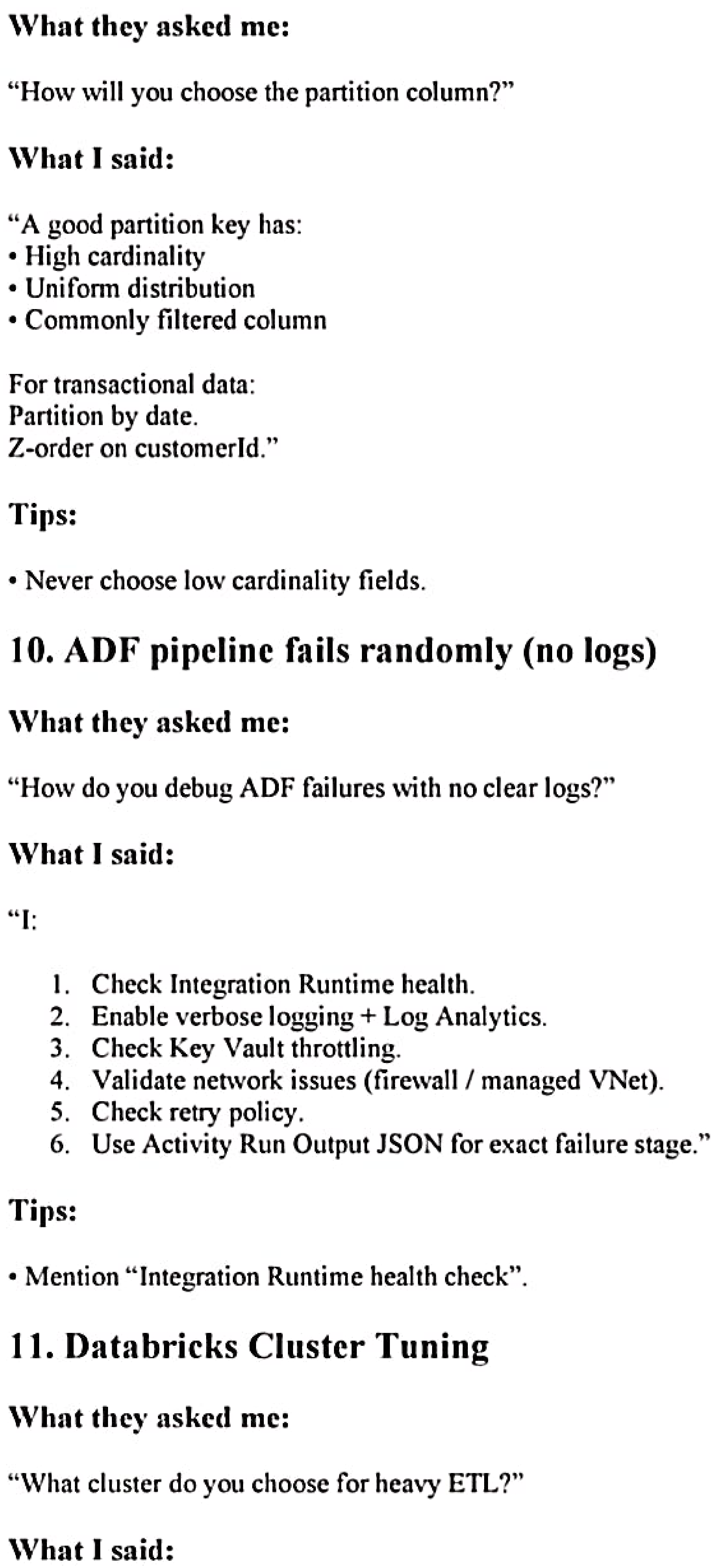
What I said:

“I use one of these:
- Hash comparison (MD5 of all columns).
- Surrogate primary key tracking.
- Snapshot comparison using Delta Change Data Feed (CDF).
- Watermarking if streaming.”

Tips:

- Mention “hash-based incremental loads”.

9. Partition Strategy for 200M row table



What they asked me:

“How will you choose the partition column?”

What I said:

“A good partition key has:
- High cardinality
- Uniform distribution
- Commonly filtered column

For transactional data:
Partition by date.
Z-order on customerId.”

Tips:

- Never choose low cardinality fields.

10. ADF pipeline fails randomly (no logs)

What they asked me:

“How do you debug ADF failures with no clear logs?”

What I said:

“I:

1. Check Integration Runtime health.
2. Enable verbose logging + Log Analytics.
3. Check Key Vault throttling.
4. Validate network issues (firewall / managed VNet).
5. Check retry policy.
6. Use Activity Run Output JSON for exact failure stage.”

Tips:

- Mention “Integration Runtime health check”.

11. Databricks Cluster Tuning

What they asked me:

“What cluster do you choose for heavy ETL?”

What I said:

"I choose a worker-heavy cluster:
- 1 driver, multiple worker nodes
- High memory per executor
- Autoscaling enabled
- Photon runtime for SQL workloads
- Spot instances for cost optimization"

Tips:

- Mention Photon runtime.

12. Kafka Duplicate Events (Streaming)

What they asked me:

"How do you remove duplicates in streaming?"

What I said:

"I use:
- Deduplication with watermarking

```
df.dropDuplicates(["eventId", "customerId"])
```

- Idempotent writes using MERGE
- Using Delta Lake constraints"

Tips:

- Mention event-id deduplication.

13. Joining large tables without shuffle

What they asked me:

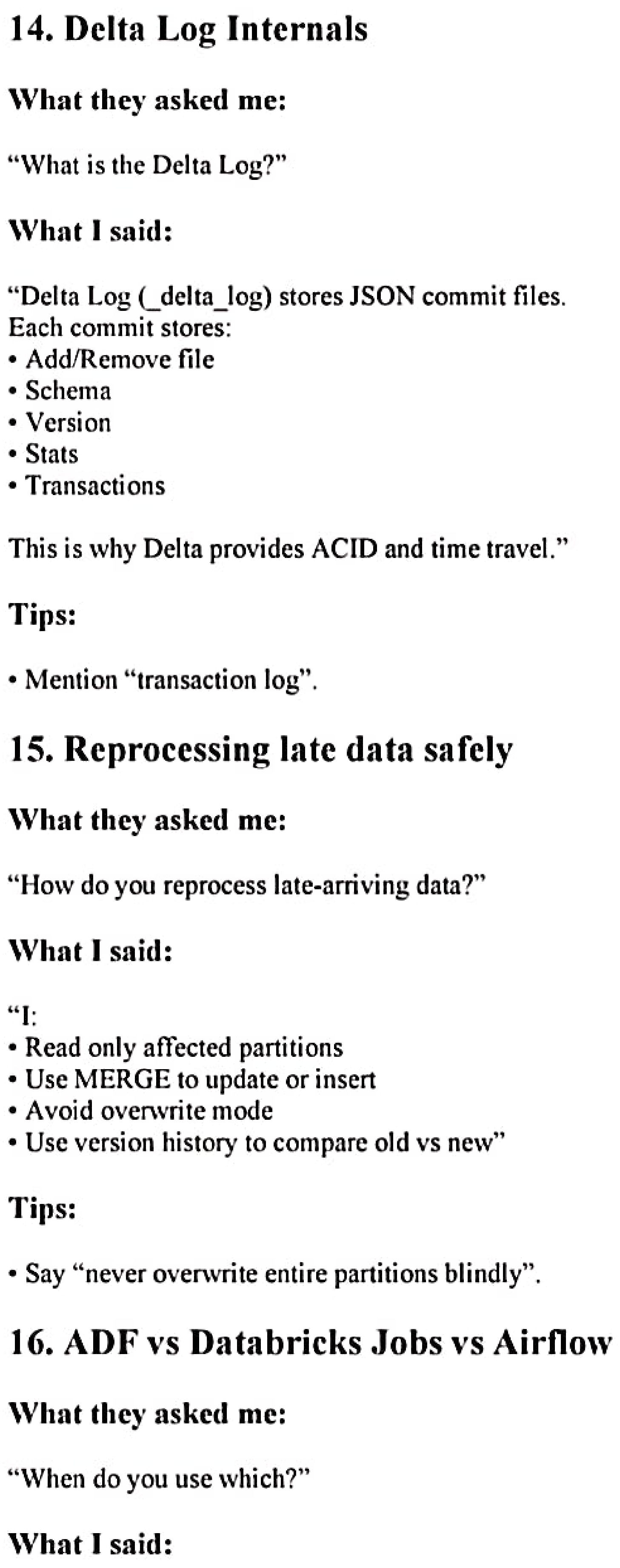
"How do you join two 500M+ row tables without shuffle?"

What I said:

"Use:
- Bucketing both tables on join key
- Broadcast join if one table is small
- Use partition pruning"

Tips:

- Say "shuffle is the biggest performance killer".



14. Delta Log Internals

What they asked me:

"What is the Delta Log?"

What I said:

"Delta Log (_delta_log) stores JSON commit files.
Each commit stores:

- Add/Remove file
- Schema
- Version
- Stats
- Transactions

This is why Delta provides ACID and time travel."

Tips:

- Mention "transaction log".

15. Reprocessing late data safely

What they asked me:

"How do you reprocess late-arriving data?"

What I said:

"I:

- Read only affected partitions
- Use MERGE to update or insert
- Avoid overwrite mode
- Use version history to compare old vs new"

Tips:

- Say "never overwrite entire partitions blindly".

16. ADF vs Databricks Jobs vs Airflow

What they asked me:

"When do you use which?"

What I said:

ADF → Orchestration, integrations, GUI pipelines
Databricks Jobs → Notebook scheduling, ML, ETL
Airflow → Complex DAGs, Python-heavy workflows

Tips:

- Show you know hybrid orchestration.

17. Securing PII (Aadhaar, PAN)

What they asked me:

"How do you secure sensitive data?"

What I said:

"I apply:
- Tokenization
- Column-level encryption
- Access-based RBAC via Unity Catalog
- Network isolation with Private Endpoints
- Secrets in Key Vault"

Tips:

- Always mention Key Vault.

18. Partition pruning not working

What they asked me:

"Query is scanning full data despite filtering partition column. Why?"

What I said:

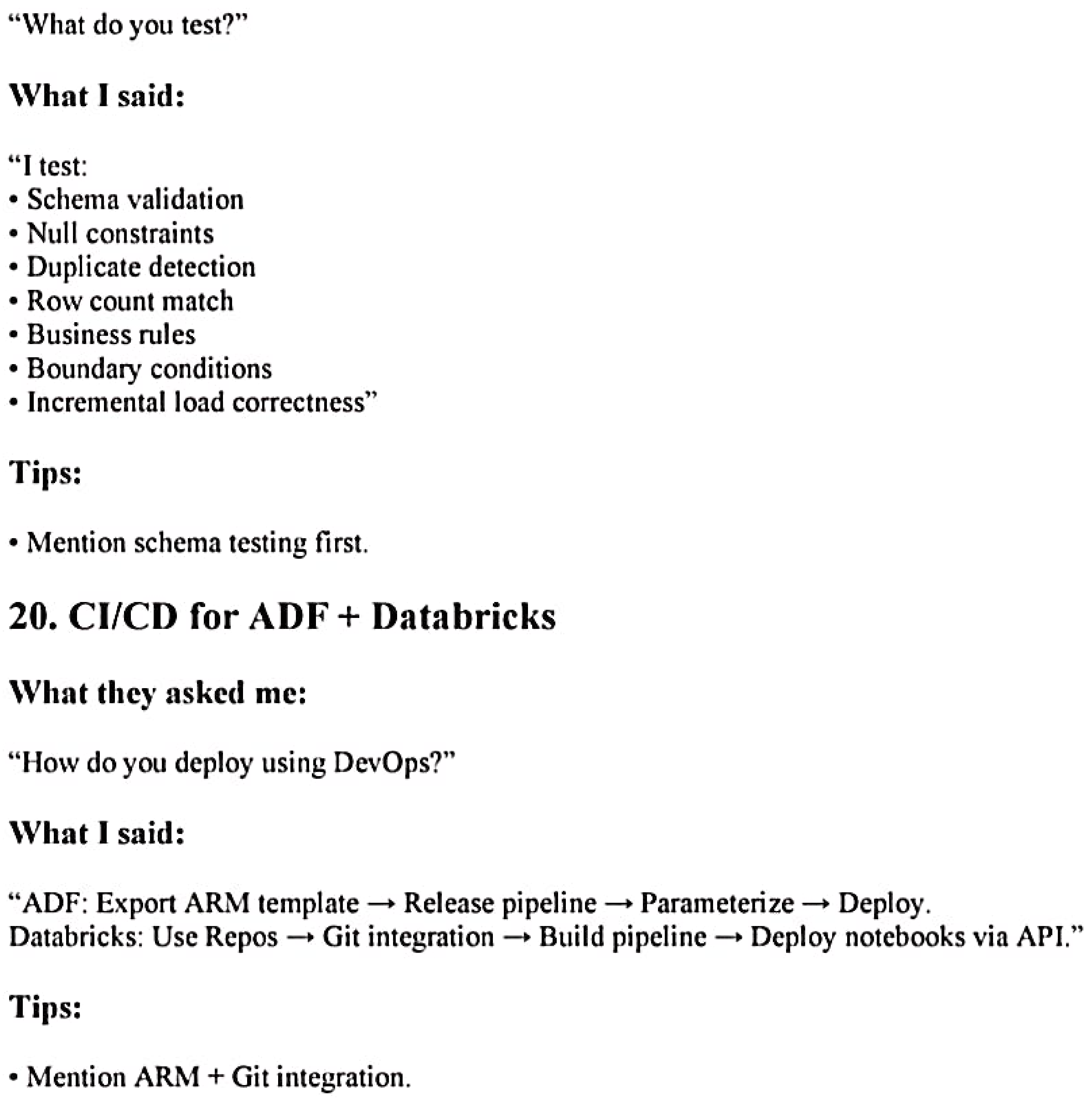
"Common reasons:
- Partition column is transformed (EXTRACT, CAST).
- Partition column missing in WHERE.
- Table not stored in Delta/Parquet correctly.
- Statistics not updated."

Tips:

- Mention "function on partition column disables pruning".

19. Unit testing for Data Pipelines

What they asked me:



"What do you test?"

What I said:

"I test:
- Schema validation
- Null constraints
- Duplicate detection
- Row count match
- Business rules
- Boundary conditions
- Incremental load correctness"

Tips:

- Mention schema testing first.

20. CI/CD for ADF + Databricks

What they asked me:

"How do you deploy using DevOps?"

What I said:

"ADF: Export ARM template → Release pipeline → Parameterize → Deploy.
Databricks: Use Repos → Git integration → Build pipeline → Deploy notebooks via API."

Tips:

- Mention ARM + Git integration.