Rapid Data Engineering System Design – Interview Cheat Guide

High-signal answers for system design, scalability, reliability, and fault tolerance

1. How I Design Any Data System (30■second answer)

- Clarify SLAs: latency, freshness, completeness, correctness.
- Choose architecture: Kappa-first with Medallion layers.
- Land immutable raw data for replay and audit.
- Apply deterministic transformations and idempotent writes.
- Monitor SLOs, not just job failures.

2. Backfill & Replay (Interview Critical)

- Raw data is immutable (Bronze).
- Pipelines are deterministic and re-runnable.
- Backfill by partition or event-time window, not full recompute.
- CDC replay supported via ordered change events.
- State reset via checkpoints/versioned tables.

3. Fault Tolerance & Failure Handling

- At-least-once ingestion + deduplication = effective exactly-once.
- Checkpointing for streaming state.
- Retry with exponential backoff; poison-pill isolation.
- Dead-letter queues/tables for bad data.
- Graceful degradation on partial failures.

4. Late Data & Watermarking

- Always process using event-time.
- Define late arrival tolerance per dataset (e.g., 48 hours).
- Watermarks bound state and memory usage.
- Late beyond watermark routed to backfill workflow.

5. CDC & SCD Handling

- Capture inserts, updates, deletes in Bronze.
- Merge in Silver using business key + sequence.
- SCD1 for corrections, SCD2 for historical tracking.
- Late arriving updates handled deterministically.

6. Scalability Levers (What Interviewers Look For)

- Parallelism: partitions, micro-batch sizing, consumer groups.
- Storage: partitioning + clustering to reduce scans.
- Compute: broadcast joins, pre-aggregation, skew handling.
- Avoid small files; enforce compaction.

7. Reliability Guarantees

- Idempotent writes via MERGE or upserts.
- Deterministic deduplication keys.
- Replay from raw with same results.

- Clear RPO/RTO definitions.

8. Metrics That Matter (Say These)

Freshness: data age vs SLA.
Completeness: expected vs actual volume.
Correctness: quality rule pass rate.
Throughput: rows/sec or MB/sec.
Lag: Kafka lag or file backlog.
Cost: $ per TB or per million events.

9. Monitoring & Alerts

- Alert on SLA breach, not just job failure.
- Separate data-late vs data-wrong vs pipeline-down.
- Trend-based alerts for drift and anomalies.

10. How I Explain Reliability in One Sentence

“I design pipelines to be replayable, idempotent, and observable, so failures result in delayed data—not incorrect data.”

11. Medallion One■Liner

Bronze = immutable truth.
Silver = business correctness.
Gold = consumer performance.

12. 60■Second System Design Script

“I ingest data immutably into Bronze with contracts and checkpoints. I process in Silver using event-time, watermarking, deduplication, and idempotent CDC merges so the system is replayable and fault-tolerant. I publish Gold tables optimized per consumer and monitor freshness, completeness, correctness, and cost against SLAs.”