Database fundamentals

Hiring manager / Ownership / Deliver Results

Full guide: 11h 16m | Core route: 4h | Read this map first, then use the final recall strip.

BEFORE STUDY Grain and SQL → transactions/recovery → Oracle/storage → ingestion/publication → two real delivery stories.

01 / Define the answer before the system

Start with the customer decision, output grain, business key, unit, population, time zone and cutoff. Name freshness, correctness and availability targets.

Distinguish a natural key from a surrogate key. A declared constraint is not always enforced by the engine. Validate the actual business invariant.

OLTP favors transactional access; OLAP favors analytical scans and aggregates. Choose row/column layout, normalization and history from the workload.

02 / SQL and metric precision

State join cardinality. A left join can lose unmatched rows through a right-side WHERE filter; count a nonnull child key when counting matches.

NULL is not zero. NOT IN with nulls is dangerous; express an anti-match with NOT EXISTS. Use HAVING for aggregate thresholds.

Resolve identity before aggregation. SUM(DISTINCT amount) does not repair fanout. Ratios need defined denominators; distinct counts are not freely additive.

03 / Transactions and recovery

ACID: all-or-nothing transaction, invariant preservation, isolation and durable committed state. A consistent snapshot is not automatically serializable.

Trace two concurrent writers. Name the isolation level and anomaly; use the engine's constraints, locking or retry mechanism deliberately.

Separate replication from backup and recovery. Define RPO/RTO. A commit timeout has an unknown outcome; inspect stable identity before retrying a write.

04 / Oracle and access paths

Oracle: read consistency uses undo; redo supports recovery. Explain statement versus transaction visibility and the chosen isolation behavior.

Index access can reduce selected reads but adds write/storage cost. Composite key order, predicates, selectivity and table access determine usefulness.

Partition pruning, index lookup, sort order and distribution solve different problems. Do not transfer Oracle behavior blindly to Redshift.

05 / Diagnose before optimizing

Break latency into queue, scan, join/shuffle, sort/spill and output. Compare estimated and actual rows; inspect skew, statistics and predicate selectivity.

State one hypothesis, change one mechanism, prove equivalent keys/values/multiplicities, then compare representative p50/p95 runtime and cost.

Ask what becomes worse: writes, storage, concurrency, maintenance or freshness. Keep a rollback and a measured acceptance target.

06 / Ingest, correct and publish

CDC needs source ordering, stable event identity, deletes, late updates and schema-change policy. An exact retry is different from a conflicting payload.

Use replay-safe upserts or transactional publication. Checkpoint/sink ordering must survive a crash between the two; idempotent writes do not cover all side effects.

Gate schema, uniqueness, completeness, freshness and semantic totals. Publish a coherent version that readers actually honor; retain quarantine ownership and replay.

07 / Ownership and Deliver Results

Ownership: a gap you noticed, why it mattered, your decision beyond the immediate handoff, partners, durable operating mechanism and observed outcome.

Deliver Results: original commitment and deadline, obstacle, your prioritization or changed plan, escalation, accepted outcome and any remaining miss.

Use one primary and one backup real incident per principle. Separate your work from team results. Defend each metric's unit, period, denominator and attribution.

08 / Pressure-test the whole case

Sketch product usage analytics: raw events → validated canonical events → account/service/time facts → shared metrics → versioned consumer output.

Trace a duplicate, stale update, delete, missing partition, schema change and crash after commit. Explain detection, correction and the consumer consequence.

Close with one unresolved assumption and how to validate it. Ask which product decision is hardest to keep both trustworthy and timely.

FINAL 5-MINUTE RECALL 60s introduction • 90s architecture walkthrough • explain one plan defect • rehearse Ownership + Deliver Results • test a duplicate/delete/timeout • ask one role-specific question.

ELI5: Explain why the data is right, why the system works and how you finish the job.

Sources: Oracle concepts | Leadership Principles