Modeling and performance

Data models / Tuning / Bias for Action

Full guide: 11h 02m | Core route: 4h | Read this map first, then use the final recall strip.

BEFORE STUDY Decision and grain → three model domains → history and corrections → physical plan → measured tuning → bounded action.

01 / Run the design conversation

Clarify consumer decision, queries, volume, freshness, retention and correctness. State the grain of each fact before drawing tables.

Declare business identity, surrogate keys, cardinalities, required/null fields, event time and source version. Show one representative query.

Explain load, correction, deletes, publication and operating ownership. A schema without a lifecycle is an incomplete design.

02 / Model measures and relationships

Separate events, periodic snapshots and accumulating snapshots. Classify additive, semi-additive and non-additive measures.

Ratios: retain numerator/denominator. Distinct users: deduplicate at the requested scope. Balances: do not sum blindly across time.

Use conformed dimensions for shared meaning. Many-valued attributes need an explicit bridge/allocation rule; avoid silent fanout.

Model facts at their own grain before joining. Header, line, payment and refund facts do not become one grain by sharing an order ID.

03 / History and late data

Type 1 overwrites; Type 2 preserves version intervals. Use stable entity identity plus version keys and nonoverlapping [start,end) intervals.

Distinguish event-valid time from when the system learned a fact. As-reported and restated answers need explicit cutoff/version semantics.

Late facts/dimensions, corrections and tombstones require deterministic resolution and downstream recomputation. Choose latest version before filtering deletes.

04 / Three model rehearsals

Product usage: account × service × event identity; derive daily usage/adoption with eligibility and observation cutoff.

Commerce: order line, capture and refund grains; avoid multiplying money; reconcile currency, status and partial refund rules.

Subscriptions: subscription events plus day/month snapshots; cohort entry, observable retention periods and reactivation policy.

05 / Read the physical evidence

Break elapsed time into queue, scan, shuffle/join, sort/spill and result transfer. Compare estimated and actual cardinality before adding an index.

Find skew, missing pruning, oversized intermediates, wrong statistics or unnecessary data movement. State the mechanism each change addresses.

Separate workload latency, throughput and cost. A faster single query may worsen concurrency or maintenance.

06 / Engine-specific choices

Redshift: columnar scan, sort keys/pruning, distribution and movement, skew, spill and queue/workload behavior. Check automatic design choices before overriding.

Oracle/row store: predicates, selectivity, composite indexes, join method, table lookups and statistics. Indexes add write and storage cost.

Spark: stage/task evidence, shuffle size, skewed partitions, broadcast eligibility, partition sizing and spill. The slowest partition can control completion.

Partitioning, distribution, sorting and indexing are distinct controls. A CTE is not a universal materialization or performance guarantee.

07 / Prove the optimization

Compare keys, values, multiplicities, nulls and rounding, including boundary cases. Matching grand totals is insufficient.

Use representative volume, skew, concurrency, cache state and repeated runs. Report p50/p95 and cost with a defined baseline.

Roll out narrowly with monitoring, stop conditions and rollback. Reconcile corrected partitions and publish a coherent consumer version.

08 / Bias for Action

State the deadline, uncertainty, minimum evidence and decision reversibility. Choose a bounded next step with a clear owner and acceptance test.

Prepare one real pilot that worked and one fast decision that exposed a bad assumption. Show what stopped, rolled back or changed.

Speed does not justify hiding data corruption. Know which decisions can be reversed cheaply and which require stronger evidence first.

FINAL 5-MINUTE RECALL Sketch one schema in 90s • name every grain/key • test a late change and delete • diagnose one slow operator • explain correctness + benchmark • rehearse a reversible action and rollback.

ELI5: Define what a row means, prove the answer, then make the right answer faster.

Sources: Redshift table design | PostgreSQL EXPLAIN