DATA ENGINEERING | ADVANCED | 2 DAYS

SQL for Data Engineering (Advanced)

Build a strong foundation in advanced SQL for real-world Data Engineering, from scalable transformations to incremental modeling patterns. Learn how to implement deduplication, upserts, SCD design, and performance optimization techniques to build reliable analytics datasets with production-quality validation.

Why Advanced SQL?

Faster analytics delivery

Build complex transformations and reporting logic efficiently.

Stronger troubleshooting skills

Engineers can debug pipelines and data issues quickly using SQL.

Improved performance

Optimize queries to reduce compute cost and dashboard latency.

More self-sufficient teams

Reduce dependency on specialized roles for day-to-day data tasks.

Better data accuracy

Advanced joins, window functions, and aggregations reduce logic errors.

What you'll be able to do

- ✓ Write complex SQL queries used in real-world data engineering pipelines.
- ✓ Build reliable transformations for ELT pipelines using advanced SQL patterns.
- ✓ Use window functions for analytics engineering and incremental model building.
- ✓ Perform deduplication, SCD logic, and CDC-style merges.
- ✓ Implement incremental load logic using watermark strategies and upsert/merge patterns.
- ✓ Optimize queries for performance using explain plans, indexes, and partitioning concepts.
- ✓ Validate data quality using SQL checks including uniqueness, null validation, and referential integrity.
- ✓ Design analytics-layer transformations including star schema modeling with fact and dimension tables.
- ✓ Build reusable SQL patterns suitable for production pipelines.

Who this is for

- Data Engineers (beginner to intermediate moving advanced)
- Analytics Engineers
- BI Engineers working with pipelines
- Backend engineers moving into data engineering
- Data Analysts transitioning to engineering workflows

Curriculum

01 Advanced SQL Foundations for Data Engineering Pipelines

- How SQL is used in ELT pipelines (staging → cleansed → curated)
- Query readability and maintainability (style, naming, CTE structure)
- Joins in production (types, pitfalls, row explosion awareness)
- Robust filtering patterns (date ranges, late data windows concept)
- **Hands-on:** Lab: Refactor a messy query into a clean, production-style CTE pipeline

02 Advanced Query Patterns (CTEs, Subqueries, Set Operations)

- CTEs for stepwise transformations and debugging
- Correlated subqueries and when to avoid them
- Set operations (UNION/UNION ALL/EXCEPT/INTERSECT) for pipeline logic
- Reusable transformation patterns for common datasets
- **Hands-on:** Lab: Build a multi-step ELT transformation using CTE stages and set operations

03 Window Functions for Analytics Engineering and Incremental Models

- Window function fundamentals (PARTITION BY, ORDER BY, frames concept)
- Ranking and dedup logic (row_number, dense_rank)
- Running totals and moving windows (rolling KPIs concepts)
- Latest record and snapshot-building patterns using windows
- **Hands-on:** Lab: Build an incremental-friendly model using windows for latest-state and KPIs

04 Deduplication and Data Cleaning Patterns in SQL

- Dedup strategies (business keys, last-write-wins concepts)
- Handling nulls and standardizing types
- Detecting duplicates and producing quarantine outputs concepts
- Data standardization patterns (dates, enums, trimming, casing)
- **Hands-on:** Lab: Deduplicate an events dataset and produce a clean “latest state” table

05 SCD and CDC-Style Merges (Slowly Changing Dimensions)

- SCD Type 1 vs Type 2 patterns (overwrite vs history tracking)
- Building dimension tables with effective_start/effective_end concepts
- CDC merge concepts (insert/update handling)
- Conflict handling and late-arriving updates concepts
- **Hands-on:** Lab: Implement SCD logic for a customer dimension using SQL patterns

06 Incremental Loads (Watermarks, Upserts, Merge Patterns)

- Full refresh vs incremental and when to choose each
- Watermark strategies (updated_at, ingestion timestamp) and storing state concepts
- Upsert patterns (merge/insert-on-conflict) across common SQL engines
- Reprocessing windows for late data and backfills
- **Hands-on:** Lab: Build an incremental pipeline query using a watermark and validate multiple runs

07 Performance Optimization (Explain Plans, Indexes, Partitioning)

- Reading explain plans and identifying bottlenecks
- Join optimization patterns (filter early, reduce dataset size)
- Index concepts and when they help (OLTP vs analytics awareness)
- Partitioning concepts (date partitions, pruning, clustering awareness)
- **Hands-on:** Lab: Optimize a slow query using explain plan insights and improved filtering/join strategy

08 Data Quality Validation Using SQL Checks

- Uniqueness checks (primary key expectations and duplicates)
- Null validation and required fields
- Referential integrity checks between fact and dimension tables
- Range and anomaly checks (negative values, impossible timestamps)
- **Hands-on:** Lab: Build a SQL quality gate that fails the pipeline if checks do not pass

09 Analytics Modeling (Star Schema and Curated Layer Design)

- Fact and dimension modeling fundamentals
- Designing a star schema for a real dataset (orders, customers, payments)
- Building analytics-ready transformations (grain definition, surrogate keys concepts)
- KPI tables and reporting views (daily revenue, cohorts concepts)
- **Hands-on:** Workshop: Design a star schema and implement fact + dimension build queries

10 Reusable SQL Patterns for Production Pipelines

- Parameterization patterns (run_date, env separation concepts)
- Standard CTE layouts for staging/cleansing/curation steps
- Idempotent query design (safe re-runs, deterministic outputs)
- Documentation and testing approach for SQL transformations
- **Hands-on:** Capstone Workshop: Package a reusable SQL transformation set with checks and runbook notes