

DATA ENGINEERING

BEGINNER

3 DAYS

Data Engineering Fundamentals

Build a strong foundation in Data Engineering fundamentals, from ingestion and transformation to orchestration and analytics-ready serving. Learn how to design reliable pipelines using SQL, Python, dbt, and Airflow with production practices like incremental loads, data validation, and operational workflows.

Why Data Engineering Fundamentals?

Reliable decision-making

Build trusted datasets that power reporting and analytics.

Faster data access

Standardize ingestion and transformation to reduce time-to-insight.

Improved data quality

Implement validation, governance, and consistency across pipelines.

Scalable data operations

Design data systems that grow with volume and business complexity.

Better AI readiness

Strong foundations enable ML and GenAI initiatives to succeed.

What you'll be able to do

- ✓ Understand the role of a Data Engineer and the end-to-end data lifecycle (ingestion, storage, transformation, orchestration, serving).
- ✓ Explain core data architecture concepts including OLTP vs OLAP and warehouse vs lake vs lakehouse.
- ✓ Build foundational and intermediate SQL skills for analytics and pipeline logic (joins, aggregations, windows, incremental patterns).
- ✓ Understand batch vs streaming pipelines and where each is used in real systems.
- ✓ Learn data modeling fundamentals including star schema design and fact/dimension tables.
- ✓ Implement basic ingestion workflows for CSV/JSON into staging tables.
- ✓ Build a Python ETL script to clean, validate, and load data into PostgreSQL.
- ✓ Create dbt models for transformations and add dbt tests for quality validation.
- ✓ Orchestrate an end-to-end workflow using an Airflow DAG with retries and visibility.
- ✓ Deliver a mini capstone project that demonstrates a working modern data pipeline with quality checks.

Who this is for

- Beginner Data Engineers
- Software Engineers transitioning to Data Engineering
- Data Analysts moving into engineering
- DevOps engineers supporting data platforms
- BI professionals exploring modern data workflows
- Engineering students and fresh graduates

Curriculum

01 Data Engineering Foundations (Role and Data Lifecycle)

- What a Data Engineer does in modern organizations
- End-to-end data lifecycle (ingestion, storage, transformation, orchestration, serving)
- Key pipeline requirements (reliability, scalability, data quality, governance basics)
- Common data platform components and how they connect
- **Hands-on:** Activity: Map a real business use case into the data lifecycle stages

02 Core Data Architecture Concepts (OLTP vs OLAP, Warehouse vs Lake vs Lakehouse)

- OLTP vs OLAP and how workloads differ
- Warehouse vs data lake vs lakehouse (trade-offs and best-fit use cases)
- Batch vs ELT vs ETL patterns in practice
- Common reference architectures for modern analytics platforms
- **Hands-on:** Activity: Choose the right architecture for 3 scenarios (BI reporting, ML feature store, event analytics)

03 SQL Fundamentals for Analytics and Pipelines

- Core SQL querying (select, where, group by, order by)
- Joins (inner/left/right/full) and practical join patterns
- Aggregations and basic KPI computation
- CTEs for readable transformations and pipeline logic
- **Hands-on:** Lab: Write SQL queries to compute daily revenue and top customers from raw tables

04 Advanced SQL (Windows + Incremental Patterns)

- Window functions (row_number, rank, lag/lead)
- Deduplication patterns using windows
- Incremental logic patterns (max timestamp watermark, upsert concepts)
- Building repeatable analytical datasets using SQL
- **Hands-on:** Lab: Implement a deduped “latest record” table and an incremental load query using watermark logic

05 Batch vs Streaming Pipelines (Real-World Usage)

- Batch pipelines (scheduled processing, backfills)
- Streaming pipelines (near real-time analytics, event processing)
- Choosing between batch and streaming (SLA, cost, complexity)
- Hybrid patterns (micro-batch, lambda/kappa concepts)
- **Hands-on:** Activity: Classify pipeline requirements and decide batch vs streaming for each

06 Data Modeling Fundamentals (Star Schema, Facts, Dimensions)

- Why data modeling matters for analytics performance and clarity
- Star schema basics (fact tables vs dimension tables)
- Grain definition and avoiding common modeling mistakes
- Designing facts/dims for e-commerce style analytics
- **Hands-on:** Workshop: Design a star schema (orders fact + customer/product dimensions)

07 Basic Ingestion Workflows (CSV/JSON to Staging)

- Staging layer purpose and raw-to-staging patterns
- Ingesting CSV and JSON (schema mapping and type casting basics)
- Handling bad records and corrupt rows patterns
- Loading data into PostgreSQL staging tables
- **Hands-on:** Lab: Ingest CSV and JSON files into staging tables with basic validation

08 Python ETL Script (Clean, Validate, Load into PostgreSQL)

- Python ETL structure (extract, transform, load) with reusable functions
- Data cleaning patterns (null handling, type casting, standardizing dates)
- Validation checks (schema, nulls, duplicates, range checks)
- PostgreSQL loading patterns (bulk load, transactions, retry basics)
- **Hands-on:** Lab: Build a Python ETL script that cleans and loads data into PostgreSQL with validations

09 Transformations with dbt (Models + Tests)

- dbt fundamentals (models, sources, refs)
- Building transformation layers (staging → intermediate → marts)
- dbt tests (not null, unique, relationships) for data quality
- Incremental dbt model concepts (overview)
- **Hands-on:** Lab: Create dbt models for curated tables and add dbt tests for quality validation

10 Orchestration with Airflow (DAGs + Retries + Visibility)

- Airflow concepts (DAGs, tasks, operators, scheduling)
- Retries, timeouts, and failure handling patterns
- Dependency chaining (ingestion → transform → validate → publish)
- Observability (task logs, retries, run history)
- **Hands-on:** Lab: Build an Airflow DAG that orchestrates ingestion, dbt transforms, and quality checks

11 Mini Capstone (Modern Data Pipeline with Quality Checks)

- Capstone goal: Deliver a working end-to-end pipeline
- Ingest CSV/JSON into PostgreSQL staging
- Transform using dbt models (facts/dims and reporting tables)
- Validate with dbt tests and additional Python checks (optional)
- Orchestrate using Airflow with retries and clear run visibility
- **Hands-on:** Capstone Lab: Deliver pipeline, demo DAG runs, and submit documentation + query outputs