DATA ENGINEERING AND ANALYTICS

Data Engineering with Databricks

Level: Practitioner • 2 days (expandable to 3) • Virtual, In-person

Overview

Databricks has become the default answer to a hard question: how do you build data pipelines that handle real volume, mixed batch and streaming sources, and constant schema change without collapsing into a pile of brittle scripts? The platform's answer, the lakehouse, is genuinely good, but it comes with its own vocabulary (Delta Lake, medallion layers, Unity Catalog, declarative pipelines) and it is easy to use the tools without understanding the architecture they serve.

This is a hands-on, practitioner course. It builds understanding in the order the platform itself is layered: Spark and the workspace first, then Delta Lake as the storage foundation, then the medallion architecture that organizes pipelines, then incremental ingestion, orchestration, and governance on top. In keeping with a less-but-deeper philosophy, we go deep on the core engineering path rather than surveying every corner of the platform; analytics and BI on Databricks get their own course in *Modern Analytics with Azure Databricks*. Every module ends with a lab in a live workspace, and each module builds on the one before.

Who Should Attend

- Data engineers building or migrating pipelines onto Databricks
- ETL and database developers moving from tools like SSIS or Data Factory data flows into code-first pipelines
- Analytics engineers and architects who need to understand how a lakehouse is actually built

Prerequisites

- Working Python: functions, data structures, and reading unfamiliar code
- Solid SQL, including joins and aggregation
- Familiarity with core data engineering concepts (batch loads, schemas, file formats); *Building ETL Pipelines with Azure Data Factory* is a good companion but not required

What You Will Learn

- Explain the lakehouse architecture and navigate the Databricks workspace, clusters, and notebooks
- Transform data at scale with Spark DataFrames and Spark SQL
- Use Delta Lake for reliable storage: ACID transactions, schema enforcement, and time travel
- Design pipelines around the medallion architecture (bronze, silver, gold)
- Build incremental ingestion with Auto Loader and orchestrate pipelines with Databricks Workflows
- Govern data with Unity Catalog and apply production practices: testing, monitoring, and cost control

Course Outline

Day one: the platform, Spark, and Delta Lake

- The Lakehouse and the Workspace
  - Why the lakehouse: what it keeps from warehouses and what it keeps from lakes

  - Workspaces, clusters, and notebooks: where code runs and what it costs
  - How Spark executes work: enough internals to reason about performance
  - Lab: set up a workspace, attach a cluster, and explore data in a notebook
- Transforming Data with Spark
  - DataFrames: reading, filtering, joining, and aggregating at scale
  - Spark SQL and when to prefer it over the DataFrame API
  - Handling messy data: schemas, corrupt records, and nulls
  - Lab: clean and reshape a raw dataset with DataFrames and Spark SQL
- Delta Lake
  - What Delta adds to Parquet: ACID transactions, schema enforcement, and time travel
  - Updates, deletes, and MERGE: doing warehouse things on a lake
  - Table maintenance: OPTIMIZE, VACUUM, and file layout
  - Lab: convert raw files into Delta tables and use MERGE and time travel

Day two: pipelines, orchestration, and production

- The Medallion Architecture
  - Bronze, silver, and gold: what belongs in each layer and why
  - Designing table layouts and contracts between layers
  - Lab: structure the day-one dataset into bronze, silver, and gold tables
- Incremental Ingestion and Orchestration
  - Auto Loader: ingesting new files continuously without reprocessing everything
  - Batch versus streaming in Databricks, and the shared API between them
  - Databricks Workflows: jobs, tasks, dependencies, and retries
  - Lab: build an incremental pipeline with Auto Loader and schedule it as a workflow
- Governance and Production Readiness
  - Unity Catalog: catalogs, schemas, permissions, and lineage
  - Testing pipeline code and monitoring pipeline health
  - Cost awareness: cluster sizing, autoscaling, and the habits that keep bills sane
  - Lab: apply Unity Catalog permissions and add monitoring to the course pipeline

Extended Version

The three-day version keeps the same gradient and adds depth on the platform's higher-level tooling:

- Declarative pipelines with Delta Live Tables, including data quality expectations
- Spark Structured Streaming beyond Auto Loader: stateful processing and watermarks
- Performance tuning: partitioning, liquid clustering, and diagnosing slow jobs
- A capstone that builds a complete medallion pipeline from raw landing to governed gold tables, scheduled and monitored