DATA ENGINEERING | INTERMEDIATE | 3 DAYS

ETL/ELT with Apache Spark + Databricks Bootcamp

Build a strong foundation in building scalable ETL/ELT pipelines with Apache Spark and Databricks, from core transformations to Delta Lake lakehouse patterns. Learn how to implement incremental processing, optimize performance, enforce data quality, operationalize pipelines, and troubleshoot real production failures confidently.

Why Spark + Databricks?

Faster big data processing

Spark handles large-scale transformations efficiently across massive datasets.

Better performance optimization

Use partitioning, caching, and tuning to reduce runtime and cost.

Unified lakehouse approach

Combine data engineering, analytics, and ML on one platform with Databricks.

Future-proof data platform

Enable advanced analytics and AI workloads with scalable architectures.

Improved pipeline reliability

Delta Lake patterns support ACID transactions and safer incremental loads.

What you'll be able to do

- ✓ Understand Spark architecture and why it is a standard for large-scale ETL.
- ✓ Build ETL/ELT pipelines using Databricks notebooks and Spark DataFrames.
- ✓ Read and write data in CSV, JSON, Parquet, and Delta Lake formats.
- ✓ Apply production transformation patterns including cleansing, deduplication, joins, aggregations, and incremental logic.
- ✓ Implement scalable patterns using partitioning, file sizing, and performance tuning fundamentals.
- ✓ Build reliable pipelines using structured logging, validation checks, and error handling strategies.
- ✓ Understand Delta Lake and lakehouse patterns including ACID tables, time travel, and merge/upsert.
- ✓ Orchestrate pipelines using Databricks Jobs and understand Airflow integration approaches.
- ✓ Deliver an end-to-end ETL/ELT pipeline capstone project with production readiness practices.

Who this is for

- Data Engineers
- Cloud Data Engineers
- Analytics Engineers
- Data Platform Engineers
- Backend engineers moving into big data engineering
- BI engineers working with transformation pipelines at scale

Curriculum

01 Spark Architecture and Execution Model (Why Spark for Large-Scale ETL)

- Why Spark is used (distributed processing, scale-out workloads, fault tolerance)
- Spark components (driver, executors, cluster manager concepts)
- Lazy evaluation and the DAG execution model
- Transformations vs actions and what triggers compute
- **Hands-on:** Lab: Inspect a Spark job plan and identify stages/shuffles for a sample workload

02 Databricks Fundamentals (Workspace, Notebooks, Clusters)

- Databricks workspace components and notebook workflow
- Cluster types (interactive vs job clusters) and sizing basics
- Databricks filesystem concepts and data access patterns
- Development workflow (notebook, repos, job runs concept)
- **Hands-on:** Lab: Create a notebook, start a cluster, and validate Spark session and basic DataFrame operations

03 Reading and Writing Data (CSV, JSON, Parquet, Delta)

- Reading CSV (schema inference vs explicit schema) and common pitfalls
- Reading nested JSON and flattening selected fields
- Parquet fundamentals (columnar format, schema, performance benefits)
- Delta Lake basics (tables, transactions, metadata)
- **Hands-on:** Lab: Read CSV + JSON, transform to Parquet, and write curated output as a Delta table

04 Production Transformations with Spark DataFrames

- Cleansing patterns (null handling, type casting, standardizing dates)
- Deduplication patterns (keys, window functions concepts)
- Joins and join pitfalls (row explosion, skew awareness)
- Aggregations for analytics datasets (daily revenue, KPIs)
- **Hands-on:** Lab: Build a cleansing + dedup + join + aggregate pipeline for an orders dataset

05 Incremental Processing Logic (ELT at Scale)

- Full refresh vs incremental processing trade-offs
- Partition-based incremental loads (by date/region/source)
- Watermark strategy concepts (updated_at, ingestion timestamp)
- Handling late-arriving data concept and reprocessing windows
- **Hands-on:** Lab: Implement an incremental pipeline and validate correct row counts across multiple runs

06 Delta Lake and Lakehouse Patterns (ACID, Time Travel, Merge)

- Why Delta (ACID transactions, consistent writes, schema enforcement)
- Time travel and table history for auditability
- Merge/upsert patterns for CDC-like workflows
- Building curated dimensions and facts using MERGE
- **Hands-on:** Lab: Write Delta tables, query versions, and implement MERGE into a dimension table

07 Scalability and Performance Fundamentals (Partitioning, File Sizing, Tuning)

- Shuffles and why they slow down Spark
- Partitioning controls (repartition vs coalesce) and when to use each
- Small files problem and file sizing strategy
- Intro tuning patterns (caching, broadcast joins concept)
- **Hands-on:** Lab: Optimize a slow job by reducing shuffle and improving join strategy

08 Reliability and Quality in Data Pipelines (Logs, Validation, Errors)

- Structured logging patterns (run id, dataset, rows processed)
- Validation checks (schema, nulls, duplicates, ranges)
- Fail-fast vs warn-only strategies and quarantine patterns
- Debugging broken pipelines (schema mismatch, corrupt records, join explosion)
- **Hands-on:** Lab: Add validation stages, generate a quality report, and fix a broken pipeline scenario

09 Orchestration with Databricks Jobs (Airflow Integration Overview)

- Databricks Jobs fundamentals (scheduling, parameters, retries, notifications)
- Job cluster vs interactive cluster for production
- Parameterizing pipelines (run_date, env, paths)
- Airflow integration approaches (trigger job runs, dependency chaining concepts)
- **Hands-on:** Lab: Convert a notebook pipeline into a scheduled Databricks Job with retries and parameters

10 Capstone Project (End-to-End ETL/ELT Pipeline with Production Readiness)

- Capstone goal: Deliver an end-to-end lakehouse pipeline (raw → cleansed → curated)
- Ingest and transform using Spark DataFrames (CSV/JSON → Parquet/Delta)
- Implement incremental processing and Delta MERGE for curated tables
- Add validation checks, structured logging, and error handling
- Operationalize using Databricks Jobs and provide runbook notes
- **Hands-on:** Capstone Lab: Deliver working notebooks/jobs, Delta tables with history, and KPI query outputs