# Enterprise Azure Databricks Data Engineering

**Duration: 5 Days**
**Mode: Instructor-led | Hands-on Lab | Qubit Test & Qubit Live Test**

---

**Day 1 – Databricks, PySpark, and Lakehouse Foundations**

Platform & Environment Setup

- Databricks Intelligence Platform overview
- Lakehouse Architecture
    - Lakehouse vs Data Lake vs Data Warehouse
- Databricks Workspace tour
    - Clusters, Notebooks, Jobs, Data, Compute
- All-purpose vs Job clusters
- Multi-language support (SQL, Python, Scala)

**Spark Architecture Overview**

- Distributed processing model of Apache Spark
- Role of Driver, Executors, and Cluster Manager
- Spark architecture in Azure Databricks context

**PySpark Fundamentals on Azure Databricks**

- Apache Spark & PySpark overview
- Why PySpark on Azure Databricks
- Creating and configuring clusters
- Creating first PySpark notebook
- Running basic PySpark commands

**DataFrames & Spark SQL**

- Reading data (CSV, JSON, Parquet)
- Creating DataFrames
- DataFrame operations
    - select, filter, withColumn, drop, distinct
- Spark SQL usage
    - Temp views
    - SQL queries on DataFrames

**Transformations & Core Processing**

- Joins (inner, left, right, full)
- GroupBy & aggregations
- Handling nulls and missing data
- Schema inference & data types
- UDFs in PySpark

**Writing & Optimization Basics**

- Writing data (CSV, Parquet, Delta)
- Introduction to Delta Lake
- Partitioning & bucketing basics
- Caching & persisting
- Sample ETL pipeline using PySpark

---

**Day 2 – ELT Development with Spark SQL & PySpark**

Spark SQL & DataFrames for ELT

- Spark SQL vs DataFrames
- Loading datasets from Data Catalog
- Creating tables using Spark SQL & PySpark

**Table Management & Schema Handling**

- Temporary vs permanent tables
- Custom schema definition
- Column selection, slicing, indexing

**ELT Pipeline Development**

- ELT pipeline design patterns
- Data cleaning & preprocessing
- Data wrangling using filters & sorting
- Writing transformed data to Delta

---

**Day 3 – Incremental Loads, Streaming, and Scalable Architecture**

**Incremental & Streaming Processing**

- Structured Streaming fundamentals
- Auto Loader for real-time ingestion
- Watermarking & late-arriving data
- Checkpointing strategies
- Deduplication using MERGE

**Medallion Architecture**

- Bronze, Silver, Gold design
- Modular and scalable pipeline patterns

## Introduction to Databricks Lakeflow

- Why Lakeflow was introduced
- Lakeflow vs traditional Spark pipelines
- Components of Lakeflow:
- Lakeflow Connect (ingestion)
- Lakeflow Declarative Pipelines (DLT)
- Lakeflow Orchestration (Jobs)

**Lakeflow Declarative Pipelines (Delta Live Tables)**

- DLT concepts and pipeline creation
- Triggered vs continuous pipelines
- Data quality expectations (intro)

---

**Day 4 – Performance Optimization, Data Skew, and Monitoring**

**Delta Lake Performance Optimization**

- OPTIMIZE, Z-ORDER, VACUUM
- File compaction strategies
- Schema evolution & constraints

**Data Skew & Spark Performance**

- What is data skew
- Identifying skew using Spark UI
- Skew mitigation techniques
    - Salting
    - Broadcast joins

- Adaptive Query Execution (AQE)

**Workflow Orchestration**

- Databricks Jobs & multi-task workflows
- Dependencies, retries, scheduling

**Monitoring & Cost Optimization**

- Job & pipeline monitoring
- Logs, alerts, execution history
- Cluster tuning & scaling
- Storage & checkpoint optimization

---

**Day 5 – Governance, Sharing, BI, and Deployment**

**Unity Catalog & Governance**

- Unity Catalog architecture
- Metastore concepts (creation explained conceptually)
- Creation Catalogs, schemas, tables
- RBAC permissions (USAGE, SELECT)
- Row-level & column-level security

**ADLS Gen2 Integration**

- Access tokens (PAT) with Azure DataFactory & Databrick
- Connecting ADLS Gen2 securely
- Storage credentials & external locations

**Delta Sharing**

- Delta Sharing architecture
- Provider vs recipient
- Secure cross-org data sharing

**Databricks SQL & BI**

- Analytical SQL queries
- Dashboards & visualizations
- Secure sharing & refresh scheduling