

# DP-750T00-A: Implement data engineering solutions using Azure Databricks

This intermediate-level course is designed for **Data Engineers** who want to master end-to-end data engineering solutions using **Azure Databricks** and **Unity Catalog**. Over four days, learners will progress from foundational setup to production deployment, gaining practical skills to implement, secure, and maintain scalable **lakehouse architectures** that meet enterprise requirements.

**Duration** : 4 days(32 hours)

## Learning Objectives

- Provision and configure Azure Databricks environments.
- Organize and govern data assets with Unity Catalog.
- Ingest, transform, and validate data for analytics and AI workloads.
- Secure data estates with enterprise-grade governance.
- Design, deploy, and optimize production-ready pipelines.
- Apply DevOps practices with Git and Databricks Asset Bundles.
- Monitor and troubleshoot workloads for efficiency and reliability.

## Prerequisites

- Fundamental knowledge of data analytics concepts
- Basic understanding of cloud storage
- Familiarity with data organization principles
- Comfortable working with SQL and Python (including notebooks)
- Understanding of Azure Databricks workspaces and Unity Catalog
- Familiarity with data access patterns and core data engineering and data warehouse concepts

- Foundational knowledge of Azure security, including Microsoft Entra ID
- Familiarity with Git version control fundamentals

## **Content Coverage**

### **✓ Set up and configure an Azure Databricks environment**

#### **Module 1: Explore Azure Databricks**

- Get started with Azure Databricks
- Identify Azure Databricks workloads
- Understand key concepts
- Data governance using Unity Catalog and Microsoft Purview

#### **Module 2: Understand Azure Databricks architecture**

- Understand Azure Databricks architecture
- Understand Unity Catalog managed storage
- Understand external storage
- Understand default storage (serverless compute)

#### **Module 3: Understand Azure Databricks Integrations**

- Understand integration with Microsoft Fabric
- Understand integration with Power BI
- Understand integration with VS Code
- Understand integration with Power Platform
- Understand integration with Copilot Studio
- Understand integration with Microsoft Purview
- Understand integration with Microsoft Foundry

#### **Module 4: Select and Configure Compute in Azure Databricks**

- Choose an appropriate compute type
- Configure compute performance
- Configure compute features
- Install libraries for compute
- Configure compute access

#### **Module 5: Create and organize objects in Unity Catalog**

- Apply naming conventions
- Create catalog
- Create schema
- Create tables and views
- Create volumes
- Implement DDL operations
- Implement foreign catalog
- Configure AI/BI Genie instructions
- Knowledge check

#### ✓ **Secure and govern Unity Catalog objects in Azure Databricks**

##### **Module 6: Secure Unity Catalog objects**

- Understand query lifecycle
- Implement access control strategies
- Understand fine-grained access control
- Implement row filtering and column masking
- Access Azure Key Vault secrets
- Authenticate data access with service principals

- Authenticate resource access with managed identities

### **Module 7: Govern Unity Catalog objects**

- Create and preserve table definitions
- Configure ABAC with tags and policies
- Apply data retention policies
- Set up and manage data lineage
- Configure audit logging
- Design secure Delta Sharing strategy

### ✓ **Prepare and process data with Azure Databricks**

#### **Module 8: Design and implement data modeling with Azure Databricks**

- Design ingestion logic and data source configuration
- Choose a data ingestion tool
- Choose a data table format
- Design and implement a data partitioning scheme
- Choose a slowly changing dimension (SCD) type
- Implement a slowly changing dimension (SCD) type 2
- Design and implement a temporal (history) table to record changes over time
- Choose granularity on a column or table based on requirements
- Choose managed vs unmanaged tables
- Design and implement a clustering strategy
- Knowledge check

#### **Module 9: Ingest data into Unity Catalog**

- Ingest data with Lakeflow Connect

- Ingest data with notebooks
- Ingest data with SQL methods
- Ingest data with CDC feed
- Ingest data with Spark Structured Streaming
- Ingest data with Auto Loader
- Ingest data with Lakeflow Spark Declarative Pipelines

### **Module 10: Cleanse, transform, and load data into Unity Catalog**

- Profile data
- Choose column data types
- Resolve duplicates and nulls
- Transform data with filters and aggregations
- Transform data with joins and set operators
- Transform data with denormalization and pivots
- Load data with merge, insert, and append

### **Module 11: Implement and manage data quality constraints with Azure Databricks**

- Implement validation checks
- Implement data type checks
- Detect and manage schema drift
- Manage data quality with pipeline expectations

### **✓ Deploy and maintain data pipelines and workloads with Azure Databricks**

### **Module 12: Design and implement data pipelines with Azure Databricks**

- Design order of operations for a pipeline
- Choose notebook vs Lakeflow Pipelines

- Design Lakeflow job logic
- Design error handling in pipelines and jobs
- Create pipeline with notebook
- Create pipeline with Lakeflow Spark Declarative Pipelines

### **Module 13: Implement Lakeflow Jobs with Azure Databricks**

- Create job setup and configuration
- Configure job triggers
- Schedule a job
- Configure job alerts
- Configure automatic restarts

### **Module 14: Implement development lifecycle processes in Azure Databricks**

- Apply Git version control best practices
- Manage branching and pull requests
- Implement testing strategy
- Configure and package DABs
- Deploy bundle with Databricks CLI

### **Module 15: Monitor, troubleshoot and optimize workloads in Azure Databricks**

- Monitor and manage cluster consumption
- Troubleshoot and repair Lakeflow Jobs
- Troubleshoot Spark jobs and notebooks
- Investigate caching, skewing, spilling, shuffle
- Implement log streaming with Azure Log Analytics

