databricks

Databricks Exam Guide

Databricks Certified Data Engineering Professional

BETA EXAM ONLY

Beta Note

The beta test of this exam is scheduled for August 28th -September 8th, 2025.

Purpose of this Exam Guide

This exam guide gives you an overview of the exam and what it covers to help you determine your exam readiness. **This version covers the Beta Exam only and is subject to change at any time.**

Audience Description

The Databricks Certified Data Engineer Professional certification exam assesses an individual's ability to use Databricks to perform advanced data engineering tasks. This includes an understanding of the Databricks platform and developer tools like Delta Lake, Unity Catalog , Auto Loader, DLT, Databricks Compute, Serverless, Workflows,Delta Sharing, Databricks Workspaces, Apache Spark, Delta Lake, the Medallion Architecture etc. It also assesses the ability to deploy using Databricks CLI, REST API, and Databricks Asset Bundles. Finally, ensuring individuals' ability to be able to use Python and SQL to perform data processing, and understanding CDC/SCD1/SCD2. Individuals who pass this certification exam can be expected to complete advanced data engineering tasks using Databricks and its associated tools.

About the Exam

- Number of items: **up to 180 multiple-choice or multiple-selection questions**, including standard items and extra beta-test items **for the beta test only**
- Time Limit: 180 minutes, **for the beta test only**
- Registration fee: The single exam attempt for the beta test is free of charge Delivery method: Online Proctored
- Test aides: None allowed
- Prerequisite: None required; course attendance and six months of hands-on experience in Databricks is highly recommended. Also, see Recommended Preparation in this document. Validity: 2 years. **Beta Note: Results will not be immediately available at exam attempt end but beta testers who are successful in their will receive the full credential at project end. Results can take 4-6 weeks.**
- Recertification: Recertification is required every two years to maintain your certified status. To recertify, you must take the full exam that is currently live. Please review the

"Getting Ready for the Exam" section on the exam webpage to prepare for taking the exam again.

Recommended Preparation

- Instructor-led: Advanced Data Engineering with Databrick
- Self-paced (available in Databricks Academy):Databricks Streaming and Lakeflow Declarative Pipeline and Automated Deployment with Databricks Asset Bundles
- Familiarity with the major topics in Data Engineering in Databricks documentation

Exam outline

Section 1: Developing Code for Data Processing using Python and SQL

1.a: Using Python and Tools for development

- Design and implement a scalable Python project structure optimized for Databricks Asset Bundles (DABs), enabling modular development, deployment automation, and CI/CD integration.
- Manage and troubleshoot external third-party library installations and dependencies in Databricks, including PyPI packages, local wheels, and source archives.
- Develop User-Defined Functions (UDFs) using Pandas/Python UDFs

1.b:Building and Testing an ETL pipeline with DLT, SQL, and Apache Spark on the Databricks platform.

- Build and manage reliable, production-ready data pipelines, for batch and streaming data using Lakeflow Declarative Pipelines and Autoloader
- Create and Automate ETL workloads using Jobs via UI/APIs/CLI
- Explain the advantages and disadvantages of streaming tables compared to materialized views.
- Use APPLY CHANGES APIs to simplify CDC in Lakeflow Declarative Pipelines
- Compare Spark Structured Streaming and Lakeflow Declarative Pipelines to determine the optimal approach for building scalable ETL pipelines.
- Create a pipeline component that uses control flow operators (e.g. if/else, foreach, etc.)
- Choose the appropriate configs for environments and dependencies, high memory for notebook tasks and auto-optimization to disallow retries
- Develop unit and integration tests using assertDataFrameEqual, assertSchemaEqual, DataFrame.transform, and testing frameworks, to ensure code correctness including built-in debugger

Section 2 :Data Ingestion & Acquisition

- Design and implement data ingestion pipelines to efficiently ingest a variety of data formats including Delta Lake, Parquet, ORC, AVRO, JSON, CSV, XML, Text and Binary from diverse sources such as message buses and cloud storage.

- Create an append-only data pipeline, capable of handling both batch and streaming data using Delta

Section 3 : Data Transformation, Cleansing and Quality

- Write efficient Spark SQL and PySpark code to apply advanced data transformations, including window functions, joins, and aggregations, to manipulate and analyze large datasets.
- Develop a quarantining process for bad data with Lakeflow Declarative Pipelines or autoloader in classic jobs

Section 4: Data Sharing and Federation

- Demonstrate delta sharing securely between Databricks deployments using Databricks to Databricks Sharing(D2D) or to external platforms using open sharing protocol(D2O)
- Configure Lakehouse Federation with proper governance across supported source systems.
- Use Delta Share to share live data from Lakehouse to any computing platform, securely

Section 5: Monitoring and Alerting

5.a: Monitoring

- Use system tables for observability over resource utilization, cost, auditing and workload monitoring.
- Use Query Profiler UI and Spark UI to monitor workloads.
- Use the Databricks REST APIs/Databricks CLI for monitoring jobs and pipelines inc
- Use Lakeflow Declarative Pipelines Event Logs to monitor pipelines

5.b: Alerting

- Use SQL Alerts to monitor data quality.
- Use the Workflows UI and Jobs API to set up job status and performance issue notifications.

Section 6 : Cost & Performance Optimisation

- Understand how / why using Unity Catalog managed tables reduces operational overhead and maintenance burden
- Understand delta optimization techniques, such as deletion vectors and liquid clustering.
- Understand the optimization techniques used by Databricks to ensure performance of queries on large datasets (data skipping, file pruning, etc)
- Apply Change Data Feed (CDF) to address specific limitations of streaming tables and enhance latency.
- Use query profile to analyze the query and identify bottlenecks such as bad data skipping, inefficient types of joins, data shuffling

Section 7: Ensuring Data Security and Compliance

7.a: Applying Data Security mechanisms

- Use ACLs to secure Workspace Objects, enforcing principles of least privilege including enforcing principles like least privilege, policy enforcement.
- Use row filters and column masks to filter and mask sensitive table data
- Apply anonymization and pseudonymization methods such as Hashing, Tokenization, Suppression, and Generalization to confidential data

7.b: Ensuring Compliance

- Implement a compliant batch & streaming pipeline that detects and applies masking of PII to ensure data privacy.
- Develop a data purging solution ensuring compliance with data retention policies.

Section 8: Data Governance

- Create and add descriptions/metadata about enterprise data to make it more discoverable
- Demonstrate understanding of Unity Catalog permission inheritance model

Section 9: Debugging and Deploying

9.1: Debugging and Troubleshooting

- Identify pertinent diagnostic information using Spark UI, cluster logs, system tables and query profiles to troubleshoot errors
- Analyze the errors and remediate the failed job runs with job repairs and parameter overrides
- Use Lakeflow Declarative Pipelines event logs & the Spark UI to debugLakeflow Declarative Pipelines and Spark pipelines

9.b: Deploying CI/CD

- Build and Deploy databricks resources using Databricks Asset Bundles
- Configure and integrate with Git-based CI/CD workflows using Databricks Git Folders for notebook and code deployment

Section 10: Data Modelling

- Design and implement scalable data models using Delta Lake to manage large datasets.
- Simplify data layout decisions and optimize query performance using Liquid Clustering
- Identify the benefits of using liquid Clustering over Partitioning and ZOrder
- Design Dimensional Models for analytical workloads, ensuring efficient querying and aggregation.