databricks

Databricks Exam Guide

Databricks Certified Data Engineer Associate

Provide Exam Guide Feedback

Note: A new version of this exam will go live on May 4, 2026. Please see below for the exam guide applicable to your exam date.

CURRENT EXAM GUIDE

Purpose of this Exam Guide

The purpose of this exam guide is to give you an overview of the exam and what is covered on the exam to help you determine your exam readiness. This document will be updated whenever there are changes to an exam (and when those changes take effect), so you are prepared. **This covers the current version of the exam that will remain live until May 3, 2026.**

Audience Description

The Databricks Certified Data Engineer Associate certification exam assesses an individual's ability to use the Databricks Data Intelligence Platform to complete introductory data engineering tasks. This includes an understanding of the Data Intelligence Platform, and its workspace, its architecture, and its capabilities. It also assesses the ability to perform ETL tasks using Apache Spark SQL or PySpark, covering extraction, complex data handling, and user-defined functions. Finally, the exam assesses the tester's ability to deploy and orchestrate workloads using Databricks workflows, and to configure and schedule jobs effectively. Individuals who pass this certification exam can be expected to complete basic data engineering tasks using Databricks and its associated tools.

About the Exam

- Number of scored items: 45 scored multiple-choice questions.
- Time limit: 90 minutes.
- Registration fee: USD 200, plus applicable taxes as required per local law.
- Delivery method: Online or test center
- Test aides: none allowed.
- Prerequisite: None required; course attendance and six months of hands-on experience in Databricks are highly recommended.
- Validity: 2 years.

- 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.
- Unscored Content: Exams may include unscored items to gather statistical information for future use. These items are not identified on the form and do not impact your score. Additional time is factored into account for this content.

Recommended Training

- Instructor-led: Data Engineering with Databricks
- Self-paced (available in Databricks Academy):
    - Data Ingestion with Lakeflow Connect
    - Deploy Workloads with Lakeflow Jobs
    - Build Data Pipelines with Lakeflow Spark Declarative pipeline.
    - Data Management and Governance with Unity Catalog.

Exam outline

Section 1: Databricks Intelligence Platform

- Enable features that simplify data layout decisions and optimize query performance.
- Explain the value of the Data Intelligence Platform.
- Identify the applicable compute to use for a specific use case.

Section 2: Development and Ingestion

- Use Databricks Connect in a data engineering workflow.
- Determine the capabilities of Notebooks functionality.
- Classify valid Auto Loader sources and use cases.
- Demonstrate knowledge of Auto Loader syntax.
- Use Databricks' built-in debugging tools to troubleshoot a given issue.

Section 3: Data Processing & Transformations

- Describe the three layers of the Medallion Architecture and explain the purpose of each layer in a data processing pipeline.
- Classify the type of cluster and configuration for optimal performance based on the scenario in which the cluster is used.
- Emphasize the advantages of LDP (for ETL process in Databricks).
- Implement data pipelines using LDP.
- Identify DDL (Data Definition Language)/DML features.
- Compute complex aggregations and Metrics with PySpark Dataframes.

Section 4: Productionizing Data Pipelines

- Identify the difference between DAB and traditional deployment methods.
- Identify the structure of Asset Bundles.
- Deploy a workflow, repair, and rerun a task in case of failure.
- Use serverless for a hands-off, auto-optimized compute managed by Databricks.
- Analyzing the Spark UI to optimize the query.

Section 5: Data Governance & Quality

- Explain the difference between managed and external tables.
- Identify the grant of permissions to users and groups within UC.
- Identify key roles in UC.
- Identify how audit logs are stored.
- Use lineage features in Unity Catalog.
- Use the Delta Sharing feature available with Unity Catalog to share data.
- Identify the advantages and limitations of Delta sharing.
- Identify types of delta sharing- Databricks vs external system.
- Analyze the cost considerations of data sharing across clouds.
- Identify Use cases of Lakehouse Federation when connected to external sources.

Sample Questions

These questions are retired from a previous version of the exam. The purpose is to show you the objectives as they are stated on the exam guide, and give you a sample question that aligns with the objective. The exam guide lists the objectives that could be covered on an exam. The best way to prepare for a certification exam is to review the exam outline in the exam guide.

Question 1

Objective: Compute complex aggregations and Metrics with PySpark Dataframes

A data engineer is curating data in the silver layer of a hospital management data warehouse system. The data engineer is trying to aggregate hospital billing data from a table `patient_billing` to generate a daily revenue fact table `daily_revenue`.

Assume this as a sample of dataframe `billing_df`**:**

billing_id	patient_id	department	billing_date	amount_billed	quantity
401	p001	Cardiology	2024-03-01	1500	1
402	p002	Radiology	2024-03-02	3000	1
403	p001	Cardiology	2024-03-01	6500	1

404	p003	Radiology	2024-03-03	500	1

Which code snippet aggregates the amount billed per day with the unique invoices from a Dataframe `billing_df`?

A.
```
daily_revenue_df = billing_df.groupBy("billing_date").agg(
    sum("amount_billed").alias("total_revenue"),
    sum("billing_id").alias("total_invoices")
)
```

B.
```
daily_revenue_df = billing_df.groupBy("billing_date").agg(
    col("amount_billed").alias("total_revenue"),
    count("billing_id").alias("total_invoices")
)
```

C.
```
daily_revenue_df = billing_df.groupBy("billing_date").agg(
    sum("amount_billed").alias("total_revenue"),
    count_distinct("patient_id").alias("total_invoices")
)
```

D.
```
daily_revenue_df = billing_df.groupBy("billing_date").agg(
    sum("amount_billed").alias("total_revenue"),
    count_distinct("billing_id").alias("total_invoices")
)
```

Question 2

Objective: Identify grant of permissions to users and groups within UC

A data engineer is working on a Databricks project with a schema named `sales_data` that stores transactional sales information. The analyst group, responsible for analyzing this data, needs read-only access to the `sales_data` schema.

Which SQL command should the data engineer use to grant the analyst group read-only access to the `sales_data` schema, assuming that the analyst group already has `USE CATALOG` and `USE SCHEMA` permissions?

A. `GRANT ALL PRIVILEGES ON SCHEMA sales_data TO analysts;`

B. `GRANT SELECT ON SCHEMA sales_data TO analysts;`

C. `GRANT INSERT ON SCHEMA sales_data TO analysts;`

D. `GRANT SELECT ON ALL TABLES IN SCHEMA sales_data TO analysts;`

Question 3

Objective: Use the Delta Sharing feature available with Unity Catalog to share data

A data engineer is configuring Delta Sharing for a multi-team project where teams from different departments will need to access shared data. The data engineer has successfully created a Unity Catalog metastore and is now setting up the Delta Share. The goal is to ensure that internal teams can access the data with full permissions, while external partners can only read the shared data.

Which action should the Data Engineer take to configure the sharing?

A. Grant READ permissions to external partners through the Delta Share and READ/WRITE permissions to internal teams on Unity Catalog.

B. Create a Delta Share, add the internal team's tables and views, and assign READ/WRITE permissions to both external partners and internal teams.

C. Create READ permissions to external partners and READ/WRITE permissions to internal teams through Delta Share.

D. Create a Delta Share, set up a secure access URL for internal teams and external partners, and distribute the URL to provide them access to the shared data.

Question 4

Objective: Identify DDL (Data Definition Language)/DML features

A data engineer has determined that a table of the following format is necessary:

employeeId	startDate	avgRating
a1	2009-01-06	5.5
a2	2018-11-21	7.1
...	...	...

Which code block is used by SQL DDL command to create an empty Delta table in the above format, regardless of whether a table already exists with this name?

A. `CREATE OR REPLACE TABLE table_name ( employeeId STRING, startDate DATE, avgRating FLOAT )`

B. `CREATE TABLE IF NOT EXISTS table_name ( employeeId STRING, startDate DATE, avgRating FLOAT )`

C. `CREATE TABLE table_name AS SELECT employeeId STRING, startDate DATE, avgRating FLOAT`

D. `CREATE OR REPLACE TABLE table_name WITH COLUMNS ( employeeId STRING, startDate DATE, avgRating FLOAT ) USING DELTA`

Question 5

Objective: Identify DDL (Data Definition Language)/DML features.

A data engineer has been given a new record of data:

```
id STRING = 'a1'

rank INTEGER = 6

rating FLOAT = 9.4
```

Which SQL command can be used to append the new record to an existing Delta table `my_table`?

A. `UPDATE VALUES ('a1', 6, 9.4) my_table`

B. `UPDATE my_table VALUES ('a1', 6, 9.4)`

C. `INSERT VALUES ('a1', 6, 9.4) INTO my_table`

D. `INSERT INTO my_table VALUES ('a1', 6, 9.4)`

Answers

Question 1: D
Question 2: B
Question 3: A
Question 4: A
Question 5: D

NEW EXAM GUIDE

Use this version of the exam guide if you are taking your exam ON or AFTER 4-May-2026

Purpose of this Exam Guide

The purpose of this exam guide is to give you an overview of the exam and what is covered on the exam to help you determine your exam readiness. This document will be updated whenever there are changes to an exam (and when those changes take effect), so you can be prepared. **This covers the exam version that will go live on May 4, 2026.**

Audience Description

The Databricks Certified Data Engineer Associate certification exam assesses an individual's ability to utilise the Databricks Data Intelligence Platform to execute foundational data engineering tasks. The exam assesses the test takers' knowledge on the Data Intelligence Platform, its workspace, architecture, and capabilities, the tasks related to Data Ingestion, Data Loading, Data Transformation and Modelling- such as the ability to perform Extract, Transform, Load (ETL) tasks using PySpark, working with Lakeflow Jobs, and CI/CD. Finally, the exam assesses the test takers' understanding of troubleshooting, monitoring, and optimization techniques, as well as their knowledge of achieving Governance and Security within the Databricks Platform.

About the Exam

- Number of scored items: 45 scored multiple-choice questions.
- Time limit: 90 minutes.
- Registration fee: USD 200, plus applicable taxes as required per local law.
- Delivery method: Online or test center
- Test aides: none allowed.
- Prerequisite: None required; course attendance and six months of hands-on experience in Databricks are highly recommended.
- Validity: 2 years.
- Recertification: Required every 2 years to maintain your certification. To recertify, you must take the current live exam. Please review the "Getting Ready for the Exam" section on the exam webpage to prepare for taking the exam again.
- Unscored Content: Exams may include unscored items to gather statistical information for future use. These items are not identified on the form and do not impact your score. Additional time is factored into account for this content.

Recommended Training

- Instructor-led: Data Engineering with Databricks

- Self-paced (available in Databricks Academy):
    - Data Ingestion with Lakeflow Connect
    - Deploy Workloads with Lakeflow Jobs
    - Build Data Pipelines with Lakeflow Spark Declarative pipeline.
    - Data Management and Governance with Unity Catalog
    - DevOps Essentials for Data Engineering
    - Data Interoperability with Unity Catalog

Exam outline

- Databricks Intelligence Platform
    - Understand the core components of the Databricks Data Intelligence Platform, such as its architecture, Delta Lake, and Unity Catalog.
    - Understand Databricks Data Intelligence Platform's compute services, including their characteristics, limitations, and cost models, and select the most suitable option for each workload use case.
- Data Ingestion and Loading
    - Enable and detail data ingestion patterns, including batch, streaming, and incremental loading, and import data from sources such as local files, Lakeflow Connect standard connectors, and Lakeflow Connect managed connectors.
    - Use the COPY INTO command to incrementally load files from cloud object storage (ADLS/S3/GCS) into Unity-Catalog–governed tables.
    - Use Auto Loader with schema enforcement and schema evolution in batch modes (for example, directory listing or file notification) to land data into Unity-Catalog–governed tables.
    - Configure Lakeflow Connect to reliably ingest data from diverse enterprise sources into Unity-Catalog–governed tables.
    - Use JDBC/ODBC or REST clients in notebooks to land data into cloud storage or directly into Unity-Catalog–governed tables, usually orchestrated and scheduled with Lakeflow Jobs.
    - Prioritize between Auto Loader, Lakeflow Connect (standard and managed connectors), partner connectors, and other ingestion methods based on technical requirements such as data volume, ingestion frequency, data types, and governance needs with Unity Catalog.
    - Ingest semi-structured and unstructured data (for example, JSON and nested data) via Lakeflow Connect and other managed connectors into Unity-Catalog–governed Delta tables.
- Data Transformation and Modeling
    - Implement data cleaning by reading bronze tables with PySpark/SQL, cleaning nulls, standardizing data types, and writing to new silver tables.

    - Combine DataFrames with operations such as Inner join, left join, broadcast join, multiple keys, cross join, union, and union all.
    - Manipulate columns, rows, and table structures by adding, dropping, splitting, renaming column names, applying filters, and exploding arrays.
    - Perform data deduplication operations and aggregate operations on DataFrames, such as count, approximate count distinct, and mean, summary.
    - Understand the basic tuning parameters (spark.sql.shuffle.partitions:, spark.default.parallelism, spark.executor/driver.memory, spark.sql.autoBroadcastJoinThreshold) and re-measure the performance.
    - Understand the difference between, and how to build, Gold layer objects such as materialized views, views, streaming tables, and tables for BI and analytics teams in Unity Catalog.
    - Apply data quality checks and validation rules to ensure reliable Silver and Gold datasets.
- Working with Lakeflow Jobs
    - Implement control flows (retries and conditional tasks such as branching and looping) using Lakeflow Jobs for pipeline orchestration
    - Configure common tasks (notebook, SQL query, dashboard, and pipeline tasks) and their dependencies using Lakeflow Jobs and its DAG-based task graph
    - Implement job schedules using Lakeflow Jobs with an understanding of trigger types (scheduled, file arrival, and table update)
    - Choose between time-based and data-driven triggers based on data availability and pipeline dependencies.
- Implementing CI/CD
    - Manage your code development workflow within the Databricks workspace UI, including creating and switching between branches in Databricks Repos, committing and pushing changes, and creating pull requests using Databricks Git integration.
    - Understand environment-specific configuration using Automation Bundle (formerly Databricks Asset Bundles) variables and overrides while promoting the same codebase across dev, test, and prod targets.
    - Deploy Declarative Automation Bundles (formerly Databricks Asset Bundles) to package, configure, and promote Lakeflow Jobs, Lakeflow Spark Declarative Pipelines, and other workspace assets across dev, test, and prod environments.
    - Understand the Databricks CLI to validate, deploy, and manage Declarative Automation Bundles (formerly Databricks Asset Bundles) and other workspace assets in automated CI/CD workflows.
- Troubleshooting, Monitoring, and Optimization
    - Identify trends in job performance using the Lakeflow Jobs run history view to compare current execution times against historical baselines.
    - Use the Lakeflow Jobs UI to monitor pipeline health by interpreting job statuses, viewing DAG-based task graphs to spot upstream blockers, and tracking pipeline run

times and failure rates.
    - Identify common performance bottlenecks such as data skew, shuffling, and disk spilling by interpreting stage-level metrics in the Spark UI.
    - Understand the features of Liquid Clustering and predictive optimization.
    - Diagnose cluster startup failures, library conflicts, and out-of-memory issues.
- Governance and Security
    - Differentiate between managed and external tables in Unity Catalog and perform basic operations (create, modify, delete, and convert between managed and external tables) on them.
    - Configure access controls using the UI and SQL by applying GRANT, REVOKE, and DENY privileges to principals (users, groups, and service principals) at appropriate levels of the security hierarchy.
    - Understand column-level masking and row-level security to restrict data visibility based on user groups.
    - Understand Unity Catalog ABAC policies to centrally control row-level filtering and column masking for sensitive data.