



Vendor: Microsoft

Exam Code: DP-750

Exam Name: Azure Databricks Data Engineer Associate

Version: DEMO

QUESTION 1

You have an Azure Databricks workspace that is enabled for Unity Catalog and contains a catalog named Catalog1. Catalog1 contains a table named Transactions. Transactions contains the following columns:

- transaction_id
- customer_name
- email_address
- credit_card_number
- transaction_amount

You need to ensure that business analysts can query all the rows in the Transactions table. The solution must meet the following requirements:

- Prevent the analysts from seeing the full values in the email_address and credit_card_number columns.
- Ensure that the analysts can see only the values after the @ character in each email address.
- Ensure that the analysts can see only the last four digits of each credit card number.
- Enable the analysts to query the table without errors.
- Follow the principle of least privilege.

What should you do?

- A. Grant the analysts the `SELECT` permission for the Transactions table and apply column-level encryption.
- B. Grant the analysts the `SELECT` permission for columns that do NOT contain sensitive data.
- C. Grant the analysts the `SELECT` permission for the Transactions table and implement row-level filters.
- D. Grant the analysts the `SELECT` permission for the Transactions table and apply column masks to email_address and credit_card_number.

Answer: D

Explanation:

To protect sensitive customer information while allowing a specific group to run queries, use Unity Catalog's Dynamic Column Masking. This grants the group table access while applying User-Defined Functions (UDFs) to redact the email addresses and credit card numbers dynamically.

Here is the exact SQL implementation to apply:

1. Create the masking SQL UDFs
Define functions to apply the exact partial masking rules requested.
2. Apply the column masks to the table
Alter the transaction table to attach these UDFs to the respective columns.
3. Grant least privilege permissions
Grant the group the `SELECT` permission on the table, along with the foundational permissions to access the catalog and schema.

QUESTION 2

Hotspot Question

You have an Azure Databricks workspace that is enabled for Unity Catalog and contains a

catalog named catalog1.

You have a group named group1.

You plan to create a schema named schema1 in catalog1.

You need to ensure that group1 meets the following requirements:

- Can create tables in schema1
- Can modify and query tables
- Cannot grant permissions for the schema and its objects

How should you complete the SQL statements? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area

```
CREATE SCHEMA catalog1.schema1;
```

GRANT	USAGE USAGE, CREATE TABLE USAGE, MODIFY	ON SCHEMA catalog1.schema1	TO CATALOG catalog1 TO ROLE group1 TO SCHEMA schema1
-------	--	----------------------------	---

Answer:

Answer Area

```
CREATE SCHEMA catalog1.schema1;
```

GRANT	USAGE USAGE, CREATE TABLE USAGE, MODIFY	ON SCHEMA catalog1.schema1	TO CATALOG catalog1 TO ROLE group1 TO SCHEMA schema1
-------	--	----------------------------	---

QUESTION 3

Hotspot Question

You have an Azure Databricks workspace that is enabled for Unity Catalog.

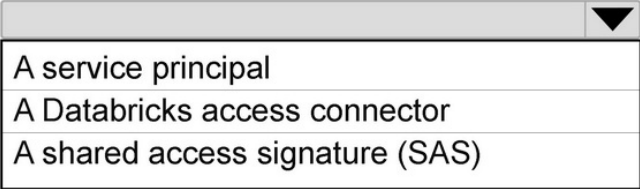
You need to create an external volume named Volume1 in an existing schema. Volume1 must expose files from an Azure Storage container. The solution must meet the following requirements:

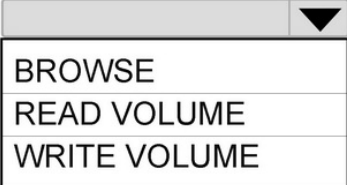
- Ensure that authentication does NOT require storing credentials in Databricks.
- Ensure that users can access the files, but NOT modify the files.
- Follow the principle of least privilege.

Which type of authentication should you configure, and which permission should you grant to the users? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

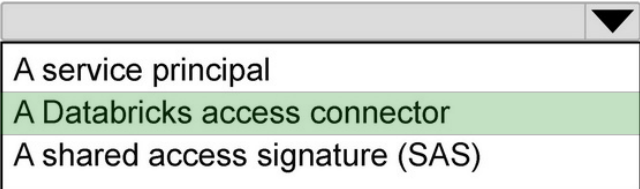
Answer Area

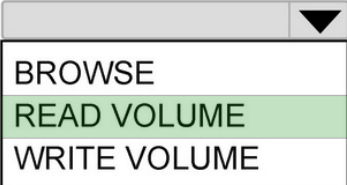
Authentication type: 

Permission: 

Answer:

Answer Area

Authentication type: 

Permission: 

QUESTION 4

Note: This section contains one or more sets of questions with the same scenario and problem. Each question presents a unique solution to the problem. You must determine whether the solution meets the stated goals. More than one solution in the set might solve the problem. It is also possible that none of the solutions in the set solve the problem.

After you answer a question in this section, you will NOT be able to return. As a result, these questions do not appear on the Review Screen.

You have an Azure Databricks workspace named Workspace1 that contains a lakehouse and is enabled for Unity Catalog.

You have a connection to a Microsoft SQL Server database named DB1.

You need to expose the schemas and tables of DB1 to meet the following requirements:

- The schemas and tables can be queried in Databricks.
- The schemas and tables appear alongside other Unity Catalog objects.
- The data is NOT copied into Databricks-managed storage.

Solution: You create a Lakeflow Connect pipeline and connect it to DB1.

Does this meet the goal?

- A. Yes
- B. No

Answer: B

Explanation:

Correct:

* You create a foreign catalog in Catalog Explorer.

You should create a Foreign Catalog using Lakehouse Federation.

Data Copying: Lakehouse Federation queries data directly in the source SQL Server without moving or copying it.

Seamless Integration: The database schemas and tables appear right inside Unity Catalog alongside your other data objects. Real-time Access: It provides immediate access to live SQL Server data.

Incorrect:

* You create a Databricks access connector.

* You create a Lakeflow Connect pipeline and connect it to DB1.

Data Copying: Lakeflow Connect is an ingestion tool that physically replicates and copies data into Databricks-managed storage (Delta tables).

Storage Costs: It violates your requirement to keep data out of Databricks storage.

* You create a new native catalog in Unity Catalog.

Note:

To expose the external SQL Server database in Unity Catalog without copying the data, you must use Lakehouse Federation.

Here are the step-by-step actions you need to take:

1. Create a Connection

Create a securable object in Unity Catalog that specifies the path and credentials to access the SQL Server database.

Go to Catalog Explorer or use SQL.

Select External Data > Connections.

Create a connection using the SQL Server connection details (URL, host, port, and database credentials).

*-> 2. Create a Foreign Catalog

Create a specific type of catalog in Unity Catalog that mirrors the external database.

Use the CREATE FOREIGN CATALOG SQL command or the Catalog Explorer UI.

Link this foreign catalog directly to the connection you created in step 1.

3. Query the Data Once the foreign catalog is created, Unity Catalog automatically syncs the

schemas and tables from SQL Server.

QUESTION 5

Note: This section contains one or more sets of questions with the same scenario and problem. Each question presents a unique solution to the problem. You must determine whether the solution meets the stated goals. More than one solution in the set might solve the problem. It is also possible that none of the solutions in the set solve the problem.

After you answer a question in this section, you will NOT be able to return. As a result, these questions do not appear on the Review Screen.

You have an Azure Databricks workspace named Workspace1 that contains a lakehouse and is enabled for Unity Catalog.

You have a connection to a Microsoft SQL Server database named DB1.

You need to expose the schemas and tables of DB1 to meet the following requirements:

- The schemas and tables can be queried in Databricks.
- The schemas and tables appear alongside other Unity Catalog objects.
- The data is NOT copied into Databricks-managed storage.

Solution: You create a Databricks access connector.

Does this meet the goal?

- A. Yes
- B. No

Answer: B

Explanation:

Correct:

* You create a foreign catalog in Catalog Explorer.

You should create a Foreign Catalog using Lakehouse Federation.

Data Copying: Lakehouse Federation queries data directly in the source SQL Server without moving or copying it.

Seamless Integration: The database schemas and tables appear right inside Unity Catalog alongside your other data objects. Real-time Access: It provides immediate access to live SQL Server data.

Incorrect:

* You create a Databricks access connector.

* You create a Lakeflow Connect pipeline and connect it to DB1.

Data Copying: Lakeflow Connect is an ingestion tool that physically replicates and copies data into Databricks-managed storage (Delta tables).

Storage Costs: It violates your requirement to keep data out of Databricks storage.

* You create a new native catalog in Unity Catalog.

Note:

To expose the external SQL Server database in Unity Catalog without copying the data, you must use Lakehouse Federation.

Here are the step-by-step actions you need to take:

1. Create a Connection

Create a securable object in Unity Catalog that specifies the path and credentials to access the SQL Server database.

Go to Catalog Explorer or use SQL.

Select External Data > Connections.

Create a connection using the SQL Server connection details (URL, host, port, and database credentials).

*-> 2. Create a Foreign Catalog

Create a specific type of catalog in Unity Catalog that mirrors the external database.

Use the CREATE FOREIGN CATALOG SQL command or the Catalog Explorer UI.

Link this foreign catalog directly to the connection you created in step 1.

3. Query the Data Once the foreign catalog is created, Unity Catalog automatically syncs the schemas and tables from SQL Server.

QUESTION 6

Note: This section contains one or more sets of questions with the same scenario and problem. Each question presents a unique solution to the problem. You must determine whether the solution meets the stated goals. More than one solution in the set might solve the problem. It is also possible that none of the solutions in the set solve the problem.

After you answer a question in this section, you will NOT be able to return. As a result, these questions do not appear on the Review Screen.

You have an Azure Databricks workspace named Workspace1 that contains a lakehouse and is enabled for Unity Catalog.

You have a connection to a Microsoft SQL Server database named DB1.

You need to expose the schemas and tables of DB1 to meet the following requirements:

- The schemas and tables can be queried in Databricks.
- The schemas and tables appear alongside other Unity Catalog objects.
- The data is NOT copied into Databricks-managed storage.

Solution: You create a foreign catalog in Catalog Explorer.

Does this meet the goal?

- A. Yes
- B. No

Answer: A

Explanation:

Correct:

* You create a foreign catalog in Catalog Explorer.

You should create a Foreign Catalog using Lakehouse Federation.

Data Copying: Lakehouse Federation queries data directly in the source SQL Server without moving or copying it.

Seamless Integration: The database schemas and tables appear right inside Unity Catalog alongside your other data objects. Real-time Access: It provides immediate access to live SQL

Server data.

Incorrect:

* You create a Databricks access connector.

* You create a Lakeflow Connect pipeline and connect it to DB1.

Data Copying: Lakeflow Connect is an ingestion tool that physically replicates and copies data into Databricks-managed storage (Delta tables).

Storage Costs: It violates your requirement to keep data out of Databricks storage.

* You create a new native catalog in Unity Catalog.

Note:

To expose the external SQL Server database in Unity Catalog without copying the data, you must use Lakehouse Federation.

Here are the step-by-step actions you need to take:

1. Create a Connection

Create a securable object in Unity Catalog that specifies the path and credentials to access the SQL Server database.

Go to Catalog Explorer or use SQL.

Select External Data > Connections.

Create a connection using the SQL Server connection details (URL, host, port, and database credentials).

*-> 2. Create a Foreign Catalog

Create a specific type of catalog in Unity Catalog that mirrors the external database.

Use the CREATE FOREIGN CATALOG SQL command or the Catalog Explorer UI.

Link this foreign catalog directly to the connection you created in step 1.

3. Query the Data Once the foreign catalog is created, Unity Catalog automatically syncs the schemas and tables from SQL Server.

QUESTION 7

Note: This section contains one or more sets of questions with the same scenario and problem. Each question presents a unique solution to the problem. You must determine whether the solution meets the stated goals. More than one solution in the set might solve the problem. It is also possible that none of the solutions in the set solve the problem.

After you answer a question in this section, you will NOT be able to return. As a result, these questions do not appear on the Review Screen.

You have an Azure Databricks workspace named Workspace1 that contains a lakehouse and is enabled for Unity Catalog.

You have a connection to a Microsoft SQL Server database named DB1.

You need to expose the schemas and tables of DB1 to meet the following requirements:

- The schemas and tables can be queried in Databricks.
- The schemas and tables appear alongside other Unity Catalog objects.
- The data is NOT copied into Databricks-managed storage.

Solution: You create a new native catalog in Unity Catalog.

Does this meet the goal?

- A. Yes
- B. No

Answer: B

Explanation:

Correct:

* You create a foreign catalog in Catalog Explorer.

You should create a Foreign Catalog using Lakehouse Federation.

Data Copying: Lakehouse Federation queries data directly in the source SQL Server without moving or copying it.

Seamless Integration: The database schemas and tables appear right inside Unity Catalog alongside your other data objects. Real-time Access: It provides immediate access to live SQL Server data.

Incorrect:

* You create a Databricks access connector.

* You create a Lakeflow Connect pipeline and connect it to DB1.

Data Copying: Lakeflow Connect is an ingestion tool that physically replicates and copies data into Databricks-managed storage (Delta tables).

Storage Costs: It violates your requirement to keep data out of Databricks storage.

* You create a new native catalog in Unity Catalog.

Note:

To expose the external SQL Server database in Unity Catalog without copying the data, you must use Lakehouse Federation.

Here are the step-by-step actions you need to take:

1. Create a Connection

Create a securable object in Unity Catalog that specifies the path and credentials to access the SQL Server database.

Go to Catalog Explorer or use SQL.

Select External Data > Connections.

Create a connection using the SQL Server connection details (URL, host, port, and database credentials).

*-> 2. Create a Foreign Catalog

Create a specific type of catalog in Unity Catalog that mirrors the external database.

Use the CREATE FOREIGN CATALOG SQL command or the Catalog Explorer UI.

Link this foreign catalog directly to the connection you created in step 1.

3. Query the Data Once the foreign catalog is created, Unity Catalog automatically syncs the schemas and tables from SQL Server.

QUESTION 8

You have an Azure Databricks workspace that uses serverless compute.

You need to ingest data by using Lakeflow Jobs. New records must be processed as soon as

they become available.

Which type of job trigger should you use for the ingestion?

- A. manual
- B. file arrival
- C. scheduled
- D. continuous

Answer: D

Explanation:

The best trigger type for this scenario is the Continuous trigger.

Immediate Processing: The Continuous trigger mode processes new records as soon as they arrive at the configured data sources. This matches your requirement to ingest and process records without waiting for an artificial time interval.

Native Serverless Support: When paired with serverless compute, Lakeflow Jobs efficiently manage resources by automatically scaling up or down according to the real-time stream volume.

Built-in Fault Tolerance: Continuous pipelines on Databricks automatically handle failures by retrying with an exponential backoff policy, keeping your automated ingestion operational 24/7 without manual intervention.

QUESTION 9

You have an Azure Databricks workspace that contains multiple all-purpose clusters.

You discover that some clusters remain idle for long periods after users finish their work.

You need to reduce compute costs without affecting active workloads.

What should you do?

- A. Enable autoscaling.
- B. Convert the clusters into job clusters.
- C. Use spot instances.
- D. Configure automatic termination.

Answer: D

Explanation:

To reduce compute costs from idle clusters without impacting active workloads, you must configure Auto-Termination and use Cluster Policies.

Core Remedies

*-> Auto-Termination: Set a strict inactivity timeout (e.g., 20–30 minutes) on all-purpose clusters to automatically shut them down when idle.

Cluster Policies: Enforce maximum auto-termination limits across the workspace so users cannot disable or set excessively long idle timeouts.

Single User Access Mode: Use this mode where possible, as it tracks idleness more accurately than Shared mode by monitoring the specific user's activity.

Thank You for Trying Our Product

Lead2pass Certification Exam Features:

- ★ More than **99,900** Satisfied Customers Worldwide.
- ★ Average **99.9%** Success Rate.
- ★ **Free Update** to match latest and real exam scenarios.
- ★ **Instant Download** Access! No Setup required.
- ★ Questions & Answers are downloadable in **PDF** format and **VCE** test engine format.
- ★ Multi-Platform capabilities - **Windows, Laptop, Mac, Android, iPhone, iPod, iPad**.
- ★ **100%** Guaranteed Success or **100%** Money Back Guarantee.
- ★ **Fast**, helpful support **24x7**.



View list of all certification exams: <http://www.lead2pass.com/all-products.html>



10% Discount Coupon Code: ASTR14