Databricks

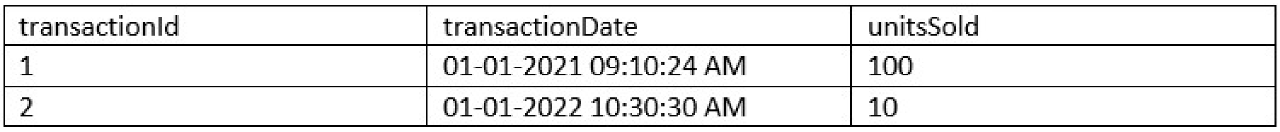
Databricks-Certified-Associate-Data-Engineer

Databricks Certified Data Engineer Associate

QUESTION & ANSWERS

Question: 1

You were asked to create a table that can store the below data, note that orderDate is the truncated date format of OrderTime, fill in the blank to complete the DDL.



transactionId	transactionDate	unitsSold
1	01-01-2021 09:10:24 AM	100
2	01-01-2022 10:30:30 AM	10

```
CREATE TABLE orders (
    orderId int,
    orderTime timestamp,
    orderdate date _______________________________________ ,
    units int)
```

A. AS DEFAULT (CAST(orderTime as DATE))

B. GENERATED ALWAYS AS (CAST(orderTime as DATE))

C. GENERATED DEFAULT AS (CAST(orderTime as DATE))

D. AS (CAST(orderTime as DATE))

E. Delta lake does not support calculated columns, value should be inserted into the table as part of the ingestion process

Answer: B

Explanation/Reference:

The answer is, GENERATED ALWAYS AS (CAST(orderTime as DATE))
https://docs.microsoft.com/en-us/azure/databricks/delta/delta-batch#--use-generated-columns
Delta Lake supports generated columns which are a special type of columns whose values are automatically generated based on a user-specified function over other columns in the Delta table. When you write to a table with generated columns and you do not explicitly provide values for them, Delta Lake automatically computes the values.
Note: Databricks also supports partitioning using generated column

Question: 2

The data engineering team noticed that one of the job fails randomly as a result of using spot instances, what feature in Jobs/Tasks can be used to address this issue so the job is more stable when using spot instances?

A. Use Databrick REST API to monitor and restart the job

B. Use Jobs runs, active runs UI section to monitor and restart the job

C. Add second task and add a check condition to rerun the first task if it fails

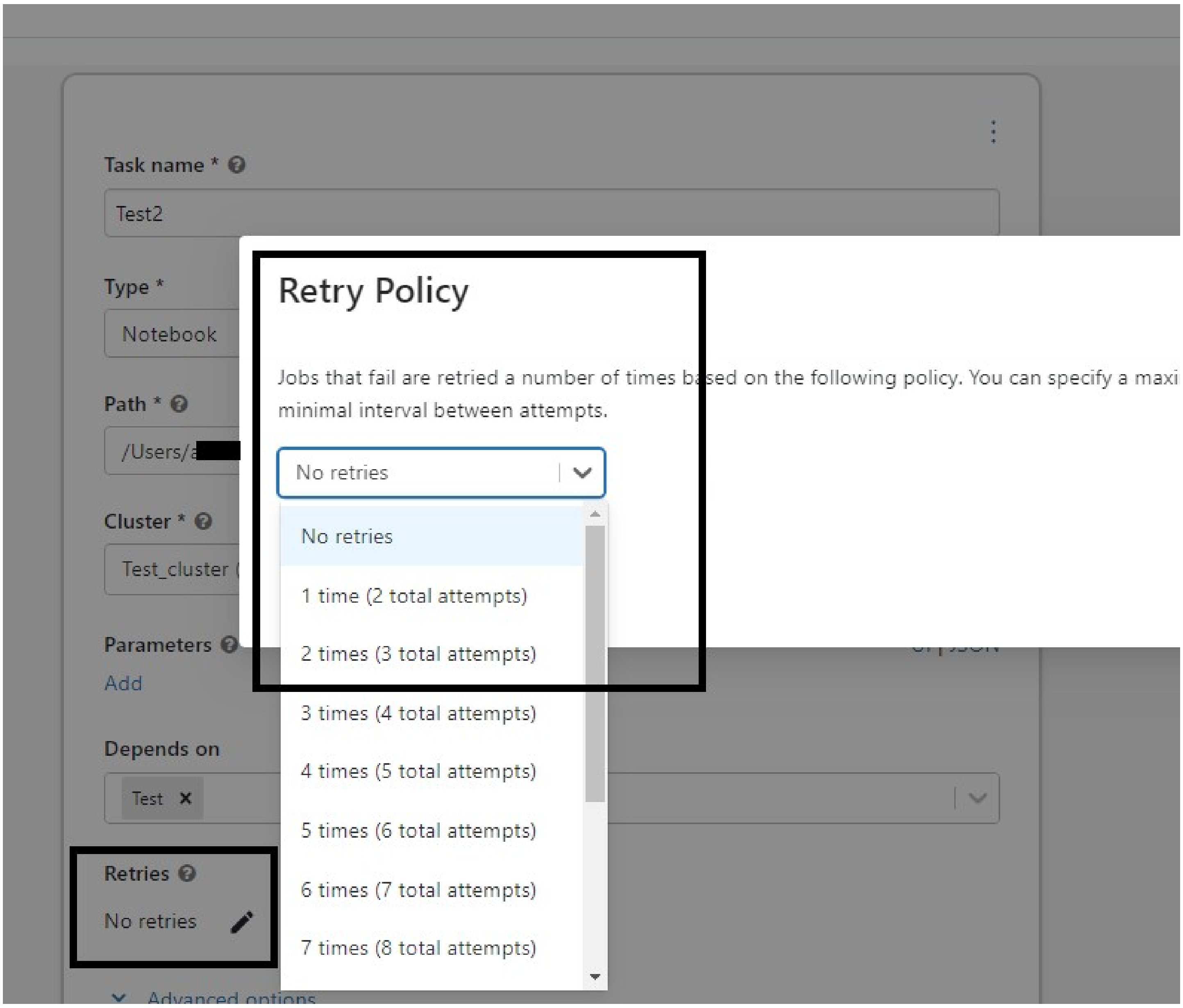
D. Restart the job cluster, job automatically restarts

E. Add a retry policy to the task

Answer: E

Explanation/Reference:

The answer is, Add a retry policy to the task
Tasks in Jobs support Retry Policy, which can be used to retry a failed tasks, especially when using spot instance it is common to have failed executors or driver.



Task name *

Test2

Type *

Notebook

Path *

/Users/a[redacted]

Cluster *

Test_cluster (

Parameters

Add

Depends on

Test ×

Retries

No retries

Advanced options

Retry Policy

Jobs that fail are retried a number of times based on the following policy. You can specify a maxi
minimal interval between attempts.

No retries

- No retries
- 1 time (2 total attempts)
- 2 times (3 total attempts)
- 3 times (4 total attempts)
- 4 times (5 total attempts)
- 5 times (6 total attempts)
- 6 times (7 total attempts)
- 7 times (8 total attempts)

Question: 3

What is the main difference between AUTO LOADER and COPY INTO?

A. COPY INTO supports schema evolution.

B. AUTO LOADER supports schema evolution.

C. COPY INTO supports file notification when performing incremental loads.

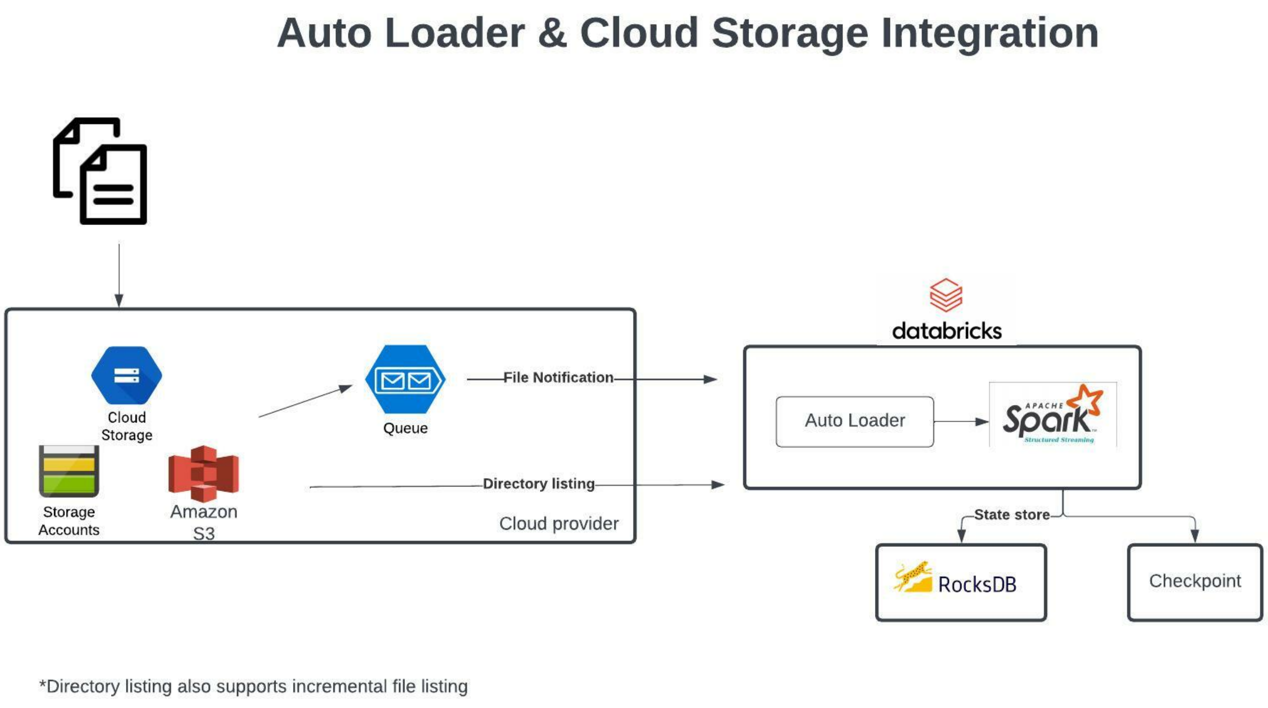
D. AUTO LOADER supports directory listing when performing incremental loads.

E. AUTO LOADER Supports file notification when performing incremental loads.

Answer: E

Explanation/Reference:

Auto loader supports both directory listing and file notification but COPY INTO only supports directory listing.
Auto loader file notification will automatically set up a notification service and queue service that subscribe to file events from the input directory in cloud object storage like Azure blob storage or S3. File notification mode is more performant and scalable for large input directories or a high volume of files.



Auto Loader and Cloud Storage Integration
Auto Loader supports a couple of ways to ingest data incrementally
Directory listing - List Directory and maintain the state in RocksDB, supports incremental file listing
File notification - Uses a trigger+queue to store the file notification which can be later used to retrieve the file, unlike Directory listing File notification can scale up to millions of files per day.
[OPTIONAL]
Auto Loader vs COPY INTO?
Auto Loader
Auto Loader incrementally and efficiently processes new data files as they arrive in cloud storage without any additional setup. Auto Loader provides a new Structured Streaming source called cloudFiles. Given an input directory path on the cloud file storage, the cloudFiles source automatically processes new files as they arrive, with the option of also processing existing files in that directory.
When to use Auto Loader instead of the COPY INTO?
You want to load data from a file location that contains files in the order of millions or higher. Auto Loader can discover files more efficiently than the COPY INTO SQL command and can split file processing into multiple batches.
You do not plan to load subsets of previously uploaded files. With Auto Loader, it can be more difficult to reprocess subsets of files. However, you can use the COPY INTO SQL command to reload subsets of files while an Auto Loader stream is simultaneously running.
Auto loader file notification will automatically set up a notification service and queue service that subscribe to file events from the input directory in cloud object storage like Azure blob storage or S3. File notification mode is more performant

and scalable for large input directories or a high volume of files.
Here are some additional notes on when to use COPY INTO vs Auto Loader
When to use COPY INTO
https://docs.databricks.com/delta/delta-ingest.html#copy-into-sql-command
When to use Auto Loader
https://docs.databricks.com/delta/delta-ingest.html#auto-loader

Question: 4

Why does AUTO LOADER require schema location?

A. Schema location is used to store user provided schema

B. Schema location is used to identify the schema of target table

C. AUTO LOADER does not require schema location, because its supports Schema evolution

D. Schema location is used to store schema inferred by AUTO LOADER

E. Schema location is used to identify the schema of target table and source table

Answer: D

Explanation/Reference:

The answer is, Schema location is used to store schema inferred by AUTO LOADER
Auto Loader samples the first 50 GB or 1000 files that it discovers, whichever limit is crossed first. To avoid incurring this inference cost at every stream start up, and to be able to provide a stable schema across stream restarts, you must set the option cloudFiles.schemaLocation. Auto Loader creates a hidden directory _schemas at this location to track schema changes to the input data over time
The below link contains detailed documentation on different options
Auto Loader options | Databricks on AWS

Question: 5

Which of the following statements are incorrect about the lakehouse

A. Support end-to-end streaming and batch workloads

B. Supports ACID

C. Support for diverse data types that can store both structured and unstructured

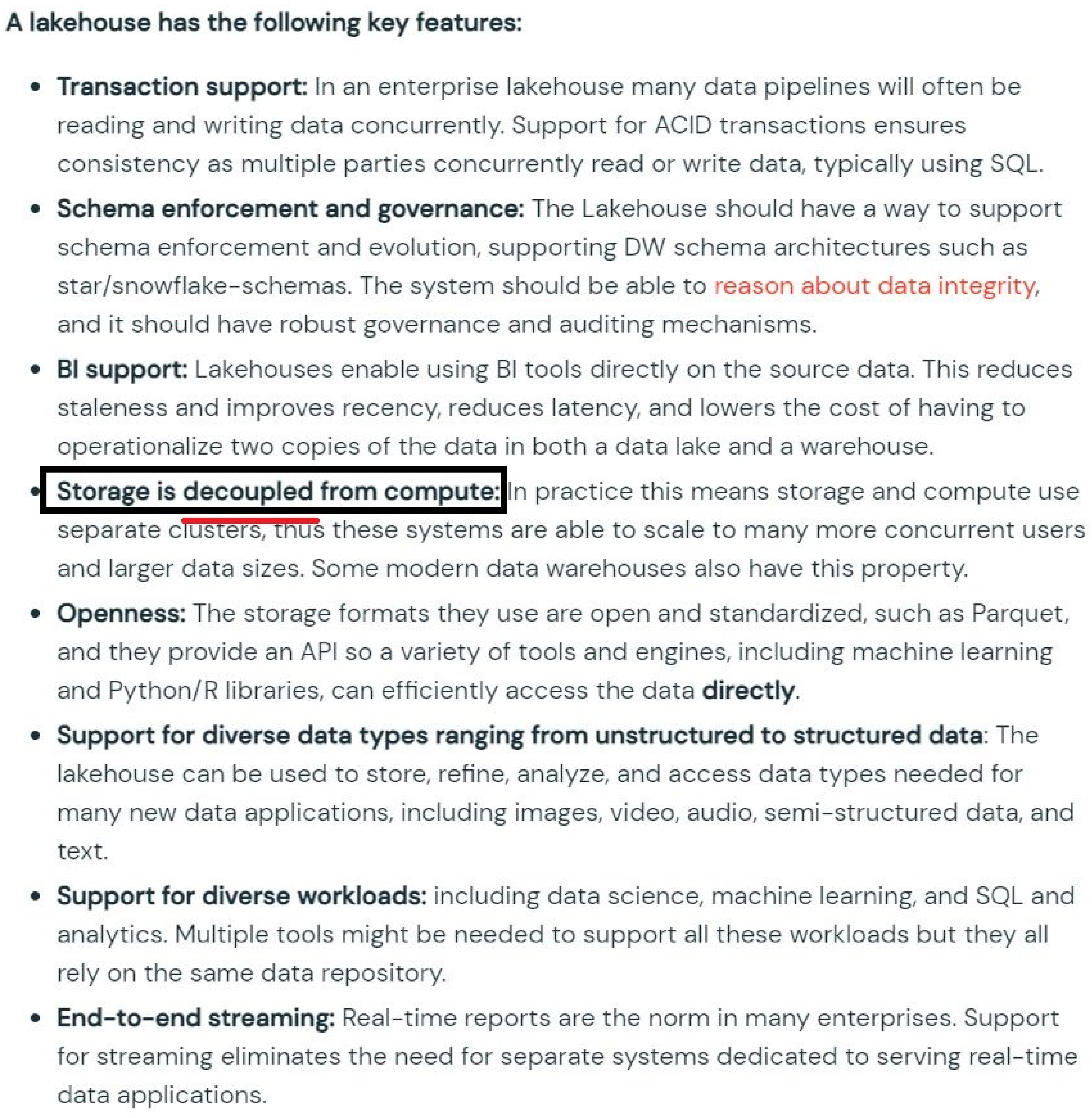
D. Supports BI and Machine learning

E. Storage is coupled with Compute

Answer: E

Explanation/Reference:

The answer is, Storage is coupled with Compute.
The question was asking what is the incorrect option, in Lakehouse Storage is decoupled with compute so both can scale independently.
What Is a Lakehouse? - The Databricks Blog



A lakehouse has the following key features:

- **Transaction support:** In an enterprise lakehouse many data pipelines will often be reading and writing data concurrently. Support for ACID transactions ensures consistency as multiple parties concurrently read or write data, typically using SQL.
- **Schema enforcement and governance:** The Lakehouse should have a way to support schema enforcement and evolution, supporting DW schema architectures such as star/snowflake-schemas. The system should be able to reason about data integrity, and it should have robust governance and auditing mechanisms.
- **BI support:** Lakehouses enable using BI tools directly on the source data. This reduces staleness and improves recency, reduces latency, and lowers the cost of having to operationalize two copies of the data in both a data lake and a warehouse.
- **Storage is decoupled from compute:** In practice this means storage and compute use separate clusters, thus these systems are able to scale to many more concurrent users and larger data sizes. Some modern data warehouses also have this property.
- **Openness:** The storage formats they use are open and standardized, such as Parquet, and they provide an API so a variety of tools and engines, including machine learning and Python/R libraries, can efficiently access the data **directly**.
- **Support for diverse data types ranging from unstructured to structured data**: The lakehouse can be used to store, refine, analyze, and access data types needed for many new data applications, including images, video, audio, semi-structured data, and text.
- **Support for diverse workloads:** including data science, machine learning, and SQL and analytics. Multiple tools might be needed to support all these workloads but they all rely on the same data repository.
- **End-to-end streaming:** Real-time reports are the norm in many enterprises. Support for streaming eliminates the need for separate systems dedicated to serving real-time data applications.

Question: 6

You are designing a data model that works for both machine learning using images and Batch ETL/ELT workloads. Which of the following features of data lakehouse can help you meet the needs of both workloads?

A. Data lakehouse requires very little data modeling.

B. Data lakehouse combines compute and storage for simple governance

C. Data lakehouse provides autoscaling for compute clusters.

D. Data lakehouse can store unstructured data and support ACID transactions.