

# Databricks.Databricks-Certified-Data-Engineer-Associate-JPN.v2025-12-26.q71

試験コード : Databricks-Certified-Data-Engineer-Associate-JPN  
試験名称 : Databricks Certified Data Engineer Associate Exam (Databricks-Certified-Data-Engineer-Associate日本語版)  
認証ベンダー : Databricks  
無料問題の数 : 71  
バージョン : v2025-12-26  
ページの閲覧量 : 103  
問題集の閲覧量 : 745  
<https://www.jpnsiken.com/shiken/Databricks.Databricks-Certified-Data-Engineer-Associate-JPN.v2025-12-26.q71.html>

## 質問: 1

データエンジニアリングプロジェクトでは、ETLを用いて毎日大量のデータを処理することが求められます。これらのジョブはリソースを大量に消費し、規模も変動するため、ワークロードに応じて自動的にスケーリングできる、スケーラブルでコスト効率の高いコンピューティングソリューションが必要です。

説明したニーズを満たすコンピューティング アプローチはどれですか？

- A. 専用クラスター
- B. ジョブクラスター
- C. 多目的クラスター
- D. Databricks SQL サーバーレス

正解: ([正解を表示します](#))

## 質問: 2

データエンジニアは、テーブルを毎日更新する際にミスをしたことに気づきました。デルタタイムトラベルを使用して、テーブルを3日前のバージョンに復元する必要があります。しかし、データエンジニアが古いバージョンにタイムトラベルしようとしたところ、データファイルが削除されているため、データを復元できませんでした。

データ ファイルが存在しなくなった理由を説明するのは次のどれですか。

- A. テーブルに対してVACUUMコマンドが実行されました
- B. TIME TRAVELコマンドがテーブル上で実行されました
- C. テーブルに対してDELETE HISTORYコマンドが実行されました
- D. テーブルにOPTIMIZEコマンドがありませんでした
- E. テーブルに対してHISTORYコマンドが実行されました

正解: ([正解を表示します](#))

The VACUUM command is used to remove files that are no longer referenced by a Delta table and are older than the retention threshold<sup>1</sup>. The default retention period is 7 days<sup>2</sup>, but it can be changed by setting the `delta.logRetentionDuration` and `delta.deletedFileRetentionDuration` configurations<sup>3</sup>. If the VACUUM command was run on the table with a retention period shorter than 3 days, then the data files that were needed to restore the table to a 3-day-old version would have been deleted. The other commands do not delete data files from the table. The TIME TRAVEL command is used to query a historical version of the table<sup>4</sup>. The DELETE HISTORY command is not a valid command in Delta Lake. The OPTIMIZE command is used to improve the performance of the table by compacting small files into larger ones<sup>5</sup>. The HISTORY command is used to retrieve information about the operations performed on the table. References: 1: VACUUM | Databricks on AWS 2: Work with Delta Lake table history | Databricks on AWS 3: [Delta Lake configuration | Databricks on AWS] 4: Work with Delta Lake table history - Azure Databricks 5: [OPTIMIZE | Databricks on AWS] : [HISTORY | Databricks on AWS]

質問: 3

データ エンジニアは、組織の既存の SQLite データベースのデータを使用して、Databricks にテーブルを作成する必要があります。

次のコマンドを実行します。

次のコード行のうち、上記の空白を埋めてタスクを正常に完了するものはどれですか？

- A. `org.apache.spark.sql.jdbc`
- B. オートローダー
- C. デルタ
- D. `sqlite`
- E. `org.apache.spark.sql.sqlite`

正解: ([正解を表示します](#))

In the given command, a data engineer is trying to create a table in Databricks using data from an SQLite database. The correct option to fill in the blank is "sqlite" because it specifies the type of database being connected to in a JDBC connection string. The USING clause should be followed by the format of the data, and since we are connecting to an SQLite database, "sqlite" would be appropriate here. References:

- \* Create a table using JDBC
- \* JDBC connection string
- \* SQLite JDBC driver

質問: 4

データエンジニアとデータアナリストが協力してデータパイプラインを作成しています。データエンジニアはPythonを使用してパイプラインのRaw、Bronze、Silverレイヤーに取り組んでおり、データアナリストはSQLを使用してパイプラインのGoldレイヤーに取り組んでいます。パイプライン

インのRawソースはストリーミング入力です。彼らは現在、パイプラインをDelta Live Tablesを使用するように移行したいと考えています。

Delta Live Tables に移行するときに、パイプラインにどのような変更を加える必要がありますか？

- A. パイプラインには、SQL と Python の異なるノートブック ソースを含めることができます。
- B. パイプラインは完全に SQL で記述する必要があります。
- C. パイプラインは完全に Python で記述する必要があります。
- D. パイプラインでは、ストリーミング ソースの代わりにバッチ ソースを使用する必要があります。

正解: [\(正解を表示します\)](#)

When migrating to Delta Live Tables (DLT) with a data pipeline that involves different programming languages across various data layers, the migration does not require unifying the pipeline into a single language. Delta Live Tables support multi-language pipelines, allowing data engineers and data analysts to work in their preferred languages, such as Python for data engineering tasks (raw, bronze, and silver layers) and SQL for data analytics tasks (gold layer). This capability is particularly beneficial in collaborative settings and leverages the strengths of each language for different stages of data processing.

References: Databricks documentation on Delta Live Tables: Delta Live Tables Guide

#### 質問: 5

データ エンジニアは、特定の培地内の細菌の個体数を計算する関数を Databricks Notebook に記述しました。

アナリストはノートブックでこの関数を使用しますが、間違っただータ型の入力引数を指定することがあり、実行中にエラーが発生する可能性があります。

入力として誤ったデータ型が指定されているかどうかをデータ エンジニアが迅速に識別するのに役立つ Databricks の機能はどれですか。

- A. データ エンジニアは、変数が何であるかを調べるために print ステートメントを追加する必要があります。
- B. Databricks デバッガーは、間違っただータ型が送信された場合にエラーを発生させるブレークポイントを有効にします。
- C. Spark ユーザー インターフェイスには、このセッションで使用される変数を含むデバッグ タブがあります。
- D. Databricks デバッガーでは、変数エクスプローラーを使用して変数の値を一目で確認できます。

正解: [D \(コメントを公表する\)](#)

#### 質問: 6

各地域の総売上額を計算し、その結果をregion\_salesという新しいデータフレームに保存します。予想される結果は次のとおりです。

どのコードが期待される結果を生成するでしょうか？

- A.** `region_sales = sales_df.groupBy("region").agg(sum("sales_amount").alias("total_sales_amount"))`
- B.** 地域別売上 - `sales_df.agg(sum("sales_amount").groupBy("region").alias("total sales amount"))`
- C.** `region_sales = sales_df. sum ("sale_a iTiount") . groupBy("region") . alias("total_sale3_amount")`
- D.** `region_sales= sales_df.groupBy("category").sum(nsales_amount").alias("t_otal_sales_ amount")`
- 正解: **A** ([コメントを发表する](#))

#### 質問: 7

データエンジニアとデータアナリストが協力してデータパイプラインを作成しています。データエンジニアはPythonを使用してパイプラインのRaw、Bronze、Silverレイヤーに取り組んでおり、データアナリストはSQLを使用してパイプラインのGoldレイヤーに取り組んでいます。パイプラインのRawソースはストリーミング入力です。彼らは現在、パイプラインをDelta Live Tablesに移行したいと考えています。

Delta Live Tables に移行するときに、パイプラインに次のどの変更を加える必要がありますか？

- A.** これらの変更は必要ありません
- B.** パイプラインはメダリオンベースのマルチホップアーキテクチャの使用を停止する必要があります
- C.** パイプラインは完全にSQLで記述する必要があります
- D.** パイプラインはストリーミングソースの代わりにバッチソースを使用する必要があります
- E.** パイプラインは完全にPythonで記述する必要があります

正解: ([正解を表示します](#))

Delta Live Tables is a declarative framework for building reliable, maintainable, and testable data processing pipelines. You define the transformations to perform on your data and Delta Live Tables manages task orchestration, cluster management, monitoring, data quality, and error handling. Delta Live Tables supports both SQL and Python as the languages for defining your datasets and expectations. Delta Live Tables also supports both streaming and batch sources, and can handle both append-only and upsert data patterns. Delta Live Tables follows the medallion lakehouse architecture, which consists of three layers of data: bronze, silver, and gold. Therefore, migrating to Delta Live Tables does not require any of the changes listed in the options B, C, D, or E. The data engineer and data analyst can use the same languages, sources, and architecture as before, and simply declare their datasets and expectations using Delta Live Tables syntax. References:

- \* What is Delta Live Tables?
- \* Transform data with Delta Live Tables
- \* What is the medallion lakehouse architecture?

#### 質問: 8

あるデータエンジニアはDatabricksにPythonノートブックを持っていますが、セル内の特定のタスクを実行するためにSQLを使用する必要があります。他のすべてのセルでは、それらのセルに変更を加えることなくPythonを使用できるようにしたいと考えています。

データ エンジニアが Python ノートブックのセル内で SQL を使用する方法を説明しているのは次のどれですか。

- A. PythonノートブックではSQLは使用できません
- B. セルをDatabricksクラスタではなくSQLエンドポイントに接続できます
- C. セル内にSQL構文を記述するだけで済みます
- D. セルの最初の行に%sqlを追加できます
- E. ノートブックのデフォルト言語をSQLに変更できます

正解: ([正解を表示します](#))

In Databricks, you can use different languages within the same notebook by using magic commands. Magic commands are special commands that start with a percentage sign (%) and allow you to change the behavior of the cell. To use SQL within a cell of a Python notebook, you can add %sql to the first line of the cell. This will tell Databricks to interpret the rest of the cell as SQL code and execute it against the default database.

You can also specify a different database by using the USE statement. The result of the SQL query will be displayed as a table or a chart, depending on the output mode. You can also assign the result to a Python variable by using the -o option. For example, %sql -o df SELECT \* FROM my\_table will run the SQL query and store the result as a pandas DataFrame in the Python variable df. Option A is incorrect, as it is possible to use SQL in a Python notebook using magic commands. Option B is incorrect, as attaching the cell to a SQL endpoint is not necessary and will not change the language of the cell. Option C is incorrect, as simply writing SQL syntax in the cell will result in a syntax error, as the cell will still be interpreted as Python code. Option E is incorrect, as changing the default language of the notebook to SQL will affect all the cells, not just one. References: Use SQL in Notebooks - Knowledge Base - Noteable, [SQL magic commands - Databricks],

[Databricks SQL Guide - Databricks]

### 質問: 9

外部テーブルを使用するシナリオを特定します。

データ エンジニアは、Parquet Bronze テーブルを作成し、それが外部の場所の特定のパスに保存されるようにする必要があります。

このシナリオではどのテーブルを作成できますか？

- A. 場所が外部場所の特定のパスを指している管理対象テーブル。
- B. カタログが外部場所の特定のパスを指す管理場所を持つ管理テーブル。
- C. スキーマが外部場所の特定のパスを指す場所を管理している外部テーブル。
- D. 場所が外部場所の特定のパスを指している外部テーブル。

正解: ([正解を表示します](#))

**質問: 10**

データ エンジニアは、組み込みの Databricks Notebooks のバージョン管理を使用するか、Databricks Repos を使用してプロジェクトのバージョン管理を行うかを決定する必要があります。

Databricks Notebooks のバージョン管理よりも Databricks Repos を使用する利点は次のどれですか？

- A. Databricks Repos は開発の進捗を自動的に保存します
- B. Databricks Repos は複数のブランチの使用をサポートしています
- C. Databricks Repos を使用すると、ユーザーはノートブックの以前のバージョンに戻すことができます。
- D. Databricks Repos は特定の変更に対してコメントする機能を提供します
- E. Databricks Repos は Databricks Lakehouse プラットフォーム内に完全に格納されています。

正解: [\(正解を表示します\)](#)

Databricks Repos is a visual Git client and API in Databricks that supports common Git operations such as cloning, committing, pushing, pulling, and branch management. Databricks Notebooks versioning is a legacy feature that allows users to link notebooks to GitHub repositories and perform basic Git operations. However, Databricks Notebooks versioning does not support the use of multiple branches for development work, which is an advantage of using Databricks Repos. With Databricks Repos, users can create and manage branches for different features, experiments, or bug fixes, and merge, rebase, or resolve conflicts between them. Databricks recommends using a separate branch for each notebook and following data science and engineering code development best practices using Git for version control, collaboration, and CI/CD. References: Git integration with Databricks Repos - Azure Databricks | Microsoft Learn, Git version control for notebooks (legacy) | Databricks on AWS, Databricks Repos Is Now Generally Available - New 'Files' Feature in ..., Databricks Repos - What it is and how we can use it | Adatis.

**質問: 11**

count\_if 関数と x が null の場合のカウントの使用方法を特定します。以下のデータを含むテーブル random\_values を検討します。

以下のクエリの実出力は何でしょうか？

count\_if(col > 1) を count\_1 として選択する

a. count\_1(\*) を count\_2.count(col1) として count\_3 を random\_values col1 から取得する

0

1

2

NULL -

2

3

**A. 4 6 5**

B. 366

C. 365

D. 466

正解: ([正解を表示します](#))

質問: 12

データエンジニアはDatabricks上のPythonノートブックでデータ処理を行っていましたが、出力が期待どおりではないことに気づきました。データエンジニアは、コードをステップ実行し、実行中に特定の変数の値を確認することで、この問題を調査したいと考えています。

データエンジニアは、コード実行と変数をリアルタイムで検査するためにどのツールを使用する必要がありますか？

A. ジョブ実行ダッシュボード

B. クラスターログ

C. SQL アナリティクス

D. Python ノートブック対話型デバッガー

正解: D ([コメントを發表する](#))

質問: 13

データエンジニアは、様々なデータソースにリンクされた複数の外部テーブルを管理していません。データエンジニアは、これらの外部テーブルを効率的に管理し、特定の外部テーブルへのアクセスに必要な権限のみがユーザーに付与されるようにしたいと考えています。

データエンジニアはこれらの外部テーブルへのアクセスをどのように管理する必要がありますか？

A. すべての外部テーブルへのフルアクセス権を持つ単一のユーザー ロールを作成し、それをすべてのユーザーに割り当てます。

B. Databricks ワークスペース レベルで権限を付与します。これはすべての外部テーブルに自動的に適用されます。

C. Unity Catalog を使用して、各外部テーブルのアクセス制御と権限を個別に管理します。

D. コンテナ レベルで Azure Blob Storage のアクセス許可を設定し、すべての外部テーブルへのアクセスを許可します。

正解: C ([コメントを發表する](#))

質問: 14

次のシナリオのうち、データエンジニアが新しい Databricks ジョブ タスクの [Depends On] フィールドでタスクを選択する必要があるのはどれですか。

A. 別のタスクを新しいタスクに置き換える必要がある場合

B. 新しいタスクが開始される前に別のタスクが失敗する必要がある場合

C. 別のタスクが新しいタスクと同じ依存ライブラリを持っている場合

D. 別のタスクが可能な限り少ない計算リソースを使用する必要があるとき

E. 新しいタスクを開始する前に別のタスクが正常に完了する必要がある場合

正解: ([正解を表示します](#))

A data engineer can create a multi-task job in Databricks that consists of multiple tasks that run in a specific order. Each task can have one or more dependencies, which are other tasks that must run before the current task. The Depends On field of a new Databricks Job Task allows the data engineer to specify the dependencies of the task. The data engineer should select a task in the Depends On field when they want the new task to run only after the selected task has successfully completed. This can help the data engineer to create a logical sequence of tasks that depend on each other's outputs or results. For example, a data engineer can create a multi-task job that consists of the following tasks:

- \* Task A: Ingest data from a source using Auto Loader
- \* Task B: Transform the data using Spark SQL
- \* Task C: Write the data to a Delta Lake table
- \* Task D: Analyze the data using Spark ML
- \* Task E: Visualize the data using Databricks SQL

In this case, the data engineer can set the dependencies of each task as follows:

- \* Task A: No dependencies
- \* Task B: Depends on Task A
- \* Task C: Depends on Task B
- \* Task D: Depends on Task C
- \* Task E: Depends on Task D

This way, the data engineer can ensure that each task runs only after the previous task has successfully completed, and the data flows smoothly from ingestion to visualization.

The other options are incorrect because they do not describe valid scenarios for selecting a task in the Depends On field. The Depends On field does not affect the following aspects of a task:

- \* Whether the task needs to be replaced by another task
  - \* Whether the task needs to fail before another task begins
  - \* Whether the task has the same dependency libraries as another task
  - \* Whether the task needs to use as little compute resources as possible
- References: Create a multi-task job, Run tasks conditionally in a Databricks job, Databricks Jobs.

質問: 15

高精度なデータを必要とする企業のデータエンジニアは、ストリーミングとバッチの両方のソースからデータを取り込む必要があります。ストリーミング入力を提供するセンサーが取得したデータの一部が、想定されるパラメータから外れている場合があります。このような場合、データは破棄する必要がありますが、ストリーム自体は失敗してはなりません。

Delta Live Tables のどの機能がこの要件を満たしていますか？

- A. 期待値
- B. 変更データキャプチャ
- C. エラー処理
- D. 監視

正解: C ([コメントを發表する](#))

質問: 16

データ エンジニアは、テーブルから読み取り、データを操作し、新しいテーブルにストリーミング書き込みを実行するように構造化ストリーミング ジョブを構成しました。

データ エンジニアが使用するコード ブロックは次のとおりです。

データ エンジニアが、クエリでマイクロバッチを実行して 5 秒ごとにデータを処理するだけの場合、空白を埋めるためにどのコード行を使用する必要がありますか。

- A. trigger(once="5 seconds")
- B. trigger("5 seconds")
- C. trigger(continuous="5 seconds")
- D. trigger(processingTime="5 seconds")

正解: ([正解を表示します](#))

有効的な**Databricks-Certified-Data-Engineer-Associate-JPN**問題集はJPNTTest.com提供され、**Databricks-Certified-Data-Engineer-Associate-JPN**試験に合格することに役に立ちます！JPNTTest.comは今最新**Databricks-Certified-Data-Engineer-Associate-JPN**試験問題集を提供します。JPNTTest.com Databricks-Certified-Data-Engineer-Associate-JPN試験問題集はもう更新されました。ここで**Databricks-Certified-Data-Engineer-Associate-JPN**問題集のテストエンジンを手に入れます。最新版のアクセス、<https://www.jpntest.com/shiken/Databricks-Certified-Data-Engineer-Associate-JPN-mondaishu> **156問、30%ディスカウント**、特別な割引コード: **JPNshiken**」

質問: 17

データ分析チームは、常時接続のSQLエンドポイントに接続した際に、DatabricksのSQLクエリの実行速度が遅すぎることに気づきました。チームメンバーの多くが同じ小さなクエリを同時に実行している際にこの問題が発生すると主張し、データエンジニアリングチームに支援を依頼しました。データエンジニアリングチームは、各チームのクエリが同じSQLエンドポイントを使用していることに気づきました。

データ エンジニアリング チームがチームのクエリのレイテンシを改善するために使用できるアプローチは次のどれですか。

- A. SQL エンドポイントのクラスター サイズを増やすことができます。
- B. SQL エンドポイントのスケーリング範囲の最大境界を増やすことができます。
- C. SQL エンドポイントの自動停止機能をオンにできます。
- D. SQL エンドポイントの Serverless 機能をオンにできます。
- E. SQL エンドポイントのサーバーレス機能をオンにし、スポット インスタンス ポリシーを「信頼性最適化」に変更できます。

正解: ([正解を表示します](#))

<https://community.databricks.com/t5/data-engineering/sequential-vs-concurrency-optimization-questions-from-query/td-p/36696>

**質問: 18**

データ エンジニアは、データ パイプラインの一部として Delta テーブルを使用する必要がありますが、適切な権限があるかどうかはわかりません。

データ エンジニアがテーブルに対する権限を確認できるのは次のどの場所ですか。

- A. Databricks ファイルシステム
- B. ジョブ
- C. ダッシュボード
- D. 休憩
- E. データエクスプローラー

正解: [E \(コメントを发表する\)](#)

Data Explorer is a graphical interface that allows you to browse, create, and manage data objects such as databases, tables, and views in your workspace. You can also review and modify the permissions on these data objects using Data Explorer. To access Data Explorer, you can click on the Data icon in the sidebar, or use the %sql magic command in a notebook. You can then select a database and a table, and click on the Permissions tab to view and edit the access control lists (ACLs) for the table. You can also use SQL commands such as SHOW GRANT and GRANT to query and modify the permissions on a Delta table. References:

- \* Data Explorer
- \* Access control for Delta tables
- \* SHOW GRANT
- \* [GRANT]

**質問: 19**

あるプロジェクトに新しいデータエンジニアリングチームが配属されました。チームは、既存のテーブルを確認するためにデータベースの顧客にアクセスする必要があります。チームには独自のグループチームがあります。

新しいチームにデータベース全体に対する必要な権限を付与するには、次のコマンドのどれを使用できますか？

- A. カタログ顧客のビューをチームに付与します。
- B. データベース 顧客に対する CREATE 権限を team に付与します。
- C. カタログ チームの使用権限を顧客に付与します。
- D. データベース team に対する CREATE 権限を customer に付与します。
- E. データベース customers の使用権限を team に付与します。

正解: [\(正解を表示します\)](#)

The correct command to grant the necessary permission on the entire database to the new team is to use the GRANT USAGE command. The GRANT USAGE command grants the principal the ability to access the securable object, such as a database, schema, or table. In this case, the

securable object is the database customers, and the principal is the group team. By granting usage on the database, the team will be able to see what tables already exist in the database. Option E is the only option that uses the correct syntax and the correct privilege type for this scenario. Option A uses the wrong privilege type (VIEW) and the wrong securable object (CATALOG). Option B uses the wrong privilege type (CREATE), which would allow the team to create new tables in the database, but not necessarily see the existing ones. Option C uses the wrong securable object (CATALOG) and the wrong principal (customers). Option D uses the wrong securable object (team) and the wrong principal (customers). References: GRANT, Privilege types, Securable objects, Principals

#### 質問: 20

データエンジニアは、Delta Live Tables (DLT) パイプラインに3つのテーブルを配置しています。各テーブルで無効なレコードを削除するようにパイプラインを設定しています。DLTパイプラインのどこかの時点で、品質上の懸念から一部のデータが削除されていることに気付きました。パイプラインのどのテーブルでデータが削除されているのかを特定したいと考えています。データエンジニアがレコードを削除しているテーブルを識別するために実行できるアプローチは次のどれですか。

- A. DLT パイプラインを開発するときに、テーブルごとに個別の期待値を設定できます。
- B. どのテーブルがレコードを削除しているかを判断できません。
- C. レコードが削除されたときに電子メールで通知するように DLT を設定できます。
- D. DLT パイプライン ページに移動し、各テーブルをクリックして、データ品質の統計を表示できます。
- E. DLT パイプライン ページに移動し、「エラー」ボタンをクリックして、現在のエラーを確認できます。

正解: **D** ([コメントを發表する](#))

One of the features of DLT is that it provides data quality metrics for each dataset in the pipeline, such as the number of records that pass or fail expectations, the number of records that are dropped, and the number of records that are written to the target. These metrics can be accessed from the DLT pipeline page, where the data engineer can click on each table and view the data quality statistics for the latest update or any previous update. This way, they can identify which table is dropping the records and why. References:

- \* Monitor Delta Live Tables pipelines
- \* Manage data quality with Delta Live Tables

#### 質問: 21

データエンジニアは、Deltaテーブルに関連付けられたデータファイルが非常に小さいことに気づきました。パフォーマンスを向上させるため、小さなファイルを圧縮して大きなファイルにしたいと考えています。

小さなファイルを圧縮するために使用できるキーワードは次のどれですか？

- A. 減らす

- B. 最適化
- C. 圧縮
- D. 配布
- E. 真空

正解: ([正解を表示します](#))

The keyword that can be used to compact the small files associated with a Delta table is OPTIMIZE. The OPTIMIZE command performs file compaction on a Delta table by rewriting a set of small files into a set of larger files<sup>1</sup>. This can improve the performance of queries that scan the table by reducing the number of files that need to be read and the amount of metadata that needs to be processed<sup>1</sup>. The OPTIMIZE command can also optionally sort the data within each file by a given set of columns, which can further improve the query performance by enabling data skipping and predicate pushdown<sup>1</sup>. The OPTIMIZE command can be applied to the whole table or to a specific partition of the table<sup>1</sup>.

The other keywords are not suitable for compacting the small files associated with a Delta table. REDUCE is a keyword used in the SQL syntax for aggregating data using a user-defined function<sup>2</sup>. COMPACTION is not a valid keyword in SQL or Python. REPARTITION is a keyword used in the Python syntax for changing the number of partitions of a DataFrame or an RDD<sup>3</sup>. VACUUM is a keyword used to remove files that are no longer referenced by a Delta table and are older than a retention threshold<sup>4</sup>.

:

- 1: OPTIMIZE | Databricks on AWS
- 2: REDUCE | Databricks on AWS
- 3: repartition | Databricks on AWS
- 4: VACUUM | Databricks on AWS

質問: 22

SQL を使用して Delta Live Tables (DLT) テーブルを作成するときに、CREATE LIVE TABLE 構文ではなく CREATE STREAMING LIVE TABLE (以前の CREATE INCREMENTAL LIVE TABLE) 構文を使用する必要があるのはどれですか。

- A. DLT パイプラインの後続のステップが静的である場合は、CREATE STREAMING LIVE TABLE を使用する必要があります。
- B. データを増分的に処理する必要がある場合は、CREATE STREAMING LIVE TABLE を使用する必要があります。
- C. CREATE STREAMING LIVE TABLE は DLT では冗長なので、使用する必要はありません。
- D. 複雑な集計を通じてデータを処理する必要がある場合は、CREATE STREAMING LIVE TABLE を使用する必要があります。
- E. DLT パイプラインの前のステップが静的である場合は、CREATE STREAMING LIVE TABLE を使用する必要があります。

正解: B ([コメントを發表する](#))

A streaming live table or view processes data that has been added only since the last pipeline update.

Streaming tables and views are stateful; if the defining query changes, new data will be processed based on the new query and existing data is not recomputed. This is useful when data needs to be processed incrementally, such as when ingesting streaming data sources or performing incremental loads from batch data sources. A live table or view, on the other hand, may be entirely computed when possible to optimize computation resources and time. This is suitable when data needs to be processed in full, such as when performing complex transformations or aggregations that require scanning all the data. References: Difference between LIVE TABLE and STREAMING LIVE TABLE, CREATE STREAMING TABLE, Load data using streaming tables in Databricks SQL.

### 質問: 23

特定のユースケースに特化したサイロ化されたデータ アーキテクチャを簡素化および統合するために使用できるのは次のどれですか。

- A. どれもでない
- B. データレイク
- C. データウェアハウス
- D. これらすべて
- E. データレイクハウス

正解: **E** ([コメントを发表する](#))

A data lakehouse is a new paradigm that can be used to simplify and unify siloed data architectures that are specialized for specific use cases. A data lakehouse combines the best of both data lakes and data warehouses, providing a single platform that supports diverse data types, open standards, low-cost storage, high- performance queries, ACID transactions, schema enforcement, and governance. A data lakehouse enables data engineers to build reliable and scalable data pipelines that can serve various downstream applications and users, such as data science, machine learning, analytics, and reporting. A data lakehouse leverages the power of Delta Lake, a storage layer that brings reliability and performance to data lakes. References: What is a data lakehouse?, Delta Lake, Lakehouse: A New Generation of Open Platforms that Unify Data Warehousing and Advanced Analytics

### 質問: 24

データエンジニアは、毎朝仕事を始める前に実行される単一タスクのジョブを持っています。上流のデータの問題を特定した後、元のタスクの前に新しいノートブックを実行するための別のタスクを設定する必要があります。

データ エンジニアが新しいタスクを設定するために使用できるアプローチは次のどれですか。

- A. 既存のジョブ内の既存のタスクを複製し、更新して新しいノートブックを実行できます。
- B. 既存のジョブに新しいタスクを作成し、それを元のタスクの依存関係として追加できます。

C. 既存のジョブに新しいタスクを作成し、元のタスクを新しいタスクの依存関係として追加できます。

D. 新しいジョブを最初から作成し、両方のタスクを追加して同時に実行することができます。

E. 既存のタスクを新しいジョブに複製し、それを編集して新しいノートブックを実行できます。

正解: [B \(コメントを发表する\)](#)

To set up the new task to run a new notebook prior to the original task in a single-task Job, the data engineer can use the following approach: In the existing Job, create a new task that corresponds to the new notebook that needs to be run. Set up the new task with the appropriate configuration, specifying the notebook to be executed and any necessary parameters or dependencies. Once the new task is created, designate it as a dependency of the original task in the Job configuration. This ensures that the new task is executed before the original task.

質問: 25

データエンジニアがSpark SQLベースのETLプロセスを開発しています。実行に失敗しました。

データエンジニアがSpark UIを確認すると、次のようなエラーが表示されました。

この問題を解決するために、データ エンジニアが実行する必要がある 2 つの修正アクションはどれですか。

2つの回答を選択してください - (Q) クエリで収集するデータを減らすためにフィルターを絞り込みます

A. 割り当てを確実にするためにシャッフルパーティションを50に固定します。

B. ワーカーノードのサイズをアップサイズし、パーティションの自動シャッフルを有効にする

C. ドライバーノードのサイズを大きくし、パーティションの自動シャッフルを無効にします

D. クエリパフォーマンスを向上させるためにデータセットをキャッシュする

正解: [\(正解を表示します\)](#)

質問: 26

「Students」テーブルのデルタトランザクションログは、`DESCRIBE HISTORY students` コマンドを使用して表示されます。データエンジニアは、ログに記録されているUPDATE操作前のテーブルの状態をクエリする必要があります。

これを実現するためにデータ エンジニアが使用すべきコマンドはどれですか? (2 つ選択してください。)

A. `students@v4` から \* を選択

B. 学生から '2024-04-22T 14:32:47.000+00:00' のタイムスタンプを選択して \* ください

C. `SELECT * FROM students FROM HISTORY VERSION AS OF 3`

D. 学生バージョン 5 以降から \* を選択

E. `SELECT * FROM students TIMESTAMP AS OF '2024-04-22T 14:32:58.000+00:00'`

正解: [\(正解を表示します\)](#)

質問: 27

Delta Live Table パイプラインには、STREAMING LIVE TABLE を使用して定義された 2 つのデータセットが含まれます。LIVE TABLE を使用して Delta Lake テーブル ソースに対して 3 つのデータセットが定義されています。

テーブルは、継続パイプライン モードを使用してプロダクション モードで実行されるように構成されています。

以前に処理されていないデータが存在し、すべての定義が有効であると仮定すると、[開始] をクリックしてパイプラインを更新した後の予想される結果は何ですか？

- A. パイプラインがシャットダウンされるまで、すべてのデータセットは設定された間隔で更新されます。コンピューティングリソースは、追加のテストを可能にするために保持されます。
- B. すべてのデータセットは一度更新され、パイプラインは処理されずに保持されます。コンピューティングリソースは保持されますが、使用されません。
- C. パイプラインがシャットダウンされるまで、すべてのデータセットは設定された間隔で更新されます。コンピューティングリソースは更新のためにデプロイされ、パイプラインが停止すると終了します。
- D. すべてのデータセットが一度更新され、パイプラインがシャットダウンされます。コンピューティングリソースは終了します。
- E. すべてのデータセットが一度更新され、パイプラインはシャットダウンされます。コンピューティングリソースは、追加のテストのために保持されます。

正解: [C \(コメントを發表する\)](#)

In Production mode, the pipeline runs continuously and updates the output tables whenever new data is available in the input sources. The compute resources are allocated on demand and released when the pipeline is stopped. This mode is suitable for production workloads that require high availability and reliability. References: [Configure pipeline settings for Delta Live Tables](#), [Tutorial: Run your first Delta Live Tables pipeline](#), [Building Reliable Data Pipelines Using DataBricks' Delta Live Tables](#)

#### 質問: 28

新しいデータエンジニアリングチームがELTプロジェクトに割り当てられました。新しいデータエンジニアリングチームには、プロジェクトを完全に管理するために、salesテーブルに対する完全な権限が必要です。

新しいデータ エンジニアリング チームにデータベースの完全な権限を付与するには、どのコマンドを使用できますか？

- A. テーブル販売に関するすべての特権をチームに付与します。
- B. テーブル sales に対する SELECT 権限を team に付与します。
- C. テーブル sales に対する SELECT CREATE MODIFY 権限を team に付与します。
- D. テーブル team のすべての権限を sales に付与します。

正解: [\(正解を表示します\)](#)

To grant full privileges on a table such as 'sales' to a group like 'team', the correct SQL command in Databricks is:

```
GRANT ALL PRIVILEGES ON TABLE sales TO team;
```

This command assigns all available privileges, including SELECT, INSERT, UPDATE, DELETE, and any other data manipulation or definition actions, to the specified team. This is typically necessary when a team needs full control over a table to manage and manipulate it as part of a project or ongoing maintenance.

References: Databricks documentation on SQL permissions: [SQL Permissions in Databricks](#)

**質問: 29**

データエンジニアは、特定のユースケースのために、テーブルストア内の文字列列 city にカスタムロジックを適用する必要があります。このカスタムロジックを大規模に適用するために、データエンジニアはSQLユーザー定義関数 (UDF) を作成したいと考えています。

次のコードブロックのどれがこの SQL UDF を作成しますか？

- A.
- B.
- C.
- D.
- E.

正解: [\(正解を表示します\)](#)

<https://www.databricks.com/blog/2021/10/20/introducing-sql-user-defined-functions.html>

**質問: 30**

データ エンジニアは、/path/to/csv にある CSV ファイルのデータを使用して、Databricks にテーブルを作成する必要があります。

次のコマンドを実行します。

次のコード行のうち、上記の空白を埋めてタスクを正常に完了するのはどれですか？

- A. デルタの使用
- B. CSVから
- C. "path/to/csv" から
- D. CSVの使用
- E. これらのコード行はタスクを正常に完了するのに必要ありません

正解: [D \(コメントを發表する\)](#)

**質問: 31**

データ エンジニアは Unity Catalog でアクセス制御を設定しており、データ アナリストのグループがテーブルをクエリすることはできるが、データを変更できないようにする必要があります。

データ エンジニアはデータ アナリストにどのような権限を与える必要がありますか？

- A. すべての権限
- B. 選択
- C. 変更
- D. 挿入

正解: [\(正解を表示します\)](#)

有効的なDatabricks-Certified-Data-Engineer-Associate-JPN問題集はJPNTest.com提供され、Databricks-Certified-Data-Engineer-Associate-JPN試験に合格することに役に立ちます！JPNTest.comは今最新Databricks-Certified-Data-Engineer-Associate-JPN試験問題集を提供します。JPNTest.com Databricks-Certified-Data-Engineer-Associate-JPN試験問題集はもう更新されました。ここでDatabricks-Certified-Data-Engineer-Associate-JPN問題集のテストエンジンを手に入れます。最新版のアクセス、<https://www.jpntest.com/shiken/Databricks-Certified-Data-Engineer-Associate-JPN-mondaishu> 156問、30%ディスカウント、特別な割引コード: **JPNshiken**」

質問: 32

オープンソーステクノロジーを採用した Databricks Lakehouse プラットフォームの利点は次のどれですか？

- A. クラウド固有の統合
- B. 簡素化されたガバナンス
- C. ストレージの拡張性
- D. ワークロードをスケーリングする能力
- E. ベンダーロックインの回避

正解: (正解を表示します)

One of the benefits of the Databricks Lakehouse Platform embracing open source technologies is that it avoids vendor lock-in. This means that customers can use the same open source tools and frameworks across different cloud providers, and migrate their data and workloads without being tied to a specific vendor. The Databricks Lakehouse Platform is built on open source projects such as Apache Spark, Delta Lake, MLflow, and Redash, which are widely used and trusted by millions of developers. By supporting these open source technologies, the Databricks Lakehouse Platform enables customers to leverage the innovation and community of the open source ecosystem, and avoid the risk of being locked into proprietary or closed solutions. The other options are either not related to open source technologies (A, B, C, D), or not benefits of the Databricks Lakehouse Platform (A, B). References: Databricks Documentation - Built on open source, Databricks Documentation - What is the Lakehouse Platform?, Databricks Blog - Introducing the Databricks Lakehouse Platform.

質問: 33

データエンジニアは、組織の既存のSQLiteデータベースのデータを使用して、Databricksにテーブルを作成する必要があります。そのためには、次のコマンドを実行します。

テーブル jdbc\_customer360 を作成する

USING

OPTIONS (

```
url "jdbc:sqlite:/customers.db", dbtable "customer360"  
)
```

タスクを正常に完了するには、上記の空白を埋めるコードの行はどれですか？

- A. オートローダー
- B. org.apache.spark.sql.jdbc
- C. sqlite
- D. org.apache.spark.sql.sqlite

正解: ([正解を表示します](#))

To create a table in Databricks using data from an SQLite database, the correct syntax involves specifying the format of the data source. The format in the case of using JDBC (Java Database Connectivity) with SQLite is specified by the org.apache.spark.sql.jdbc format. This format allows Spark to interface with various relational databases through JDBC. Here is how the command should be structured:

```
CREATE TABLE jdbc_customer360  
USING org.apache.spark.sql.jdbc  
OPTIONS (  
url "jdbc:sqlite:/customers.db",  
dbtable "customer360"  
)
```

The USING org.apache.spark.sql.jdbc line specifies that the JDBC data source is being used, enabling Spark to interact with the SQLite database via JDBC.

References: Databricks documentation on JDBC: Connecting to SQL Databases using JDBC

#### 質問: 34

CREATE TABLE AS SELECT ステートメントを使用するときに、CSV ではなく Parquet から外部テーブルを作成する利点を説明しているのは次のうちどれですか。

- A. CREATE TABLE AS SELECT 文はファイルでは使用できません
- B. Parquetファイルには明確に定義されたスキーマがあります
- C. Parquetファイルは最適化できる
- D. Parquetファイルはパーティション分割できます
- E. ParquetファイルはDeltaテーブルになります

正解: **B** ([コメントを發表する](#))

Option C is the correct answer because Parquet files have a well-defined schema that is embedded within the data itself. This means that the data types and column names of the Parquet files are automatically detected and preserved when creating an external table from them. This also enables the use of SQL and other structured query languages to access and analyze the data. CSV files, on the other hand, do not have a schema embedded in them, and require specifying the schema explicitly or inferring it from the data when creating an external table from them. This can lead to errors or inconsistencies in the data types and column names, and also increase the processing time and complexity.

CREATE TABLE AS SELECT, Parquet Files, CSV Files, Parquet vs. CSV

**質問: 35**

データエンジニアは、SQLクエリで使用したいPython変数table\_nameを持っています。table\_nameを使用してクエリを実行するPythonコードブロックを作成したいと考えています。

次のような不完全なコードブロックがあります。

```
____(f"SELECT customer_id, spending FROM {table_name}")
```

空欄を埋めてタスクを正常に完了するには、次のうちどれを使用できますか？

- A. spark.delta.sql
- B. spark.delta.table
- C. spark.table
- D. dbutils.sql
- E. spark.sql

正解: ([正解を表示します](#))

The spark.sql method can be used to execute SQL queries programmatically and return the result as a DataFrame. The spark.sql method accepts a string argument that contains a valid SQL statement. The data engineer can use a formatted string literal (f-string) to insert the Python variable table\_name into the SQL query. The other methods are either invalid or not suitable for running SQL queries. References: Running SQL Queries Programmatically, Formatted string literals, spark.sql

**質問: 36**

データエンジニアが同じコマンドセルにPythonとSQLを記述しようとしてエラーが発生しています。エンジニアは、SelectステートメントでPython変数を使用できると考えていました。コマンドが失敗する理由は何ですか？

- A. Databricksはセルごとに1つの言語をサポートします。
- B. Databricksは複数の言語をサポートしますが、ノートブックごとに1つだけです。
- C. Databricksは同じセル内での言語の相互運用性をサポートしていますが、ScalaとSQL間のみです。
- D. Databricksは、特殊文字が使用されている場合にのみ言語の相互運用性をサポートします。

正解: ([正解を表示します](#))

**質問: 37**

Spark SQLの配列関数によって提供される利点は次のどれですか？

- A. 一度にさまざまなタイプのデータを処理する機能
- B. 特定のパーティションとウィンドウ内のデータを操作する機能
- C. 指定された間隔で時間関連のデータを処理する機能
- D. JSONファイルから取り込んだ複雑なネストされたデータを扱う機能
- E. 手続きの自動化のためにテーブルの配列を操作する機能

正解: ([正解を表示します](#))

The array functions from Spark SQL are a subset of the collection functions that operate on array columns<sup>1</sup>. They provide an ability to work with complex, nested data ingested from JSON files or other sources<sup>2</sup>. For example, the explode function can be used to transform an array column into multiple rows, one for each element in the array<sup>3</sup>. The array\_contains function can be used to check if a value is present in an array column<sup>4</sup>. The array\_join function can be used to concatenate all elements of an array column with a delimiter. These functions can be useful for processing JSON data that may have nested arrays or objects. References: 1: Spark SQL, Built-in Functions - Apache Spark 2: Spark SQL Array Functions Complete List - Spark By Examples 3: Spark SQL Array Functions - Syntax and Examples - DWgeek.

com 4: Spark SQL, Built-in Functions - Apache Spark : Spark SQL, Built-in Functions - Apache Spark :

[Working with Nested Data Using Higher Order Functions in SQL on Databricks - The Databricks Blog]

### 質問: 38

ある組織は、AWS上のDatabricksワークスペースに保存されている大規模なデータセットを、AzureでホストされているDatabricksワークスペースを持つパートナー組織と共有する計画を立てています。データエンジニアは、安全かつ効率的なデータ共有を確保しながら、データ転送コストを最小限に抑えたいと考えています。

クラウド間のデータ共有に関連するデータ送信コストを削減する戦略はどれですか？

- A. AWS と Azure 間の VPN 接続を構成して、データ共有を高速化します。
- B. 追加の設定なしでデルタ共有を使用する
- C. 共有前にデータセットをCloudflare R2オブジェクトストレージに移行する
- D. 送信コストを監視せずに、署名済み URL 経由でデータを共有する

正解: ([正解を表示します](#))

### 質問: 39

あるデータエンジニアは、複数のソースからのストリーミングデータとバッチデータの両方を処理するETLパイプラインを設計しています。このパイプラインは、データ品質の確保、スキーマの進化への対応、そして容易なメンテナンスを実現する必要があります。チームはこれらの目標を達成するために、DatabricksのDelta Live Tables (DLT)の活用を検討しています。彼らは、このユースケースに適したDLTの主な機能とメリットを理解したいと考えています。

Delta Live Tables (DLT) が適切な選択である理由は何ですか？

- A. 手動スキーマの強制、高い運用オーバーヘッド、限られたスケーラビリティ
- B. バッチ処理のみをサポートし、データのバージョン管理はサポートされず、インフラストラクチャのコストが高くなります。
- C. データ品質チェック用のカスタムコードが必要、ストリーミングデータはサポートされていない、パイプラインのメンテナンスが複雑
- D. 自動データ品質チェック、スキーマ進化の組み込みサポート、宣言型パイプライン開発

正解: ([正解を表示します](#))

質問: 40

Databricks メダリオン アーキテクチャにおける Silver レイヤーの主な機能は何ですか？

- A. さらなる処理のためにデータを検証、クリーンアップ、重複排除する
- B. 監査目的のみで履歴データを保存する
- C. ビジネス分析のためのデータの集約と拡充
- D. 生のデータを元の状態で取り込む

正解: ([正解を表示します](#))

質問: 41

データエンジニアは、SQLクエリで使用したいPython変数table\_nameを持っています。table\_nameを使用してクエリを実行するPythonコードブロックを構築したいと考えています。

次のような不完全なコード ブロックがあります。

```
____(f"SELECT customer_id, spend FROM {table_name}")
```

タスクを正常に完了するには、次のどれを使用して空欄を埋めることができますか？

- A. spark.delta.table
- B. dbutils.sql
- C. spark.table
- D. spark.delta.sql
- E. spark.sql

正解: **E** ([コメントを公表する](#))

The spark.sql method can be used to execute SQL queries programmatically and return the result as a DataFrame. The spark.sql method accepts a string argument that contains a valid SQL statement. The data engineer can use a formatted string literal (f-string) to insert the Python variable table\_name into the SQL query. The other methods are either invalid or not suitable for running SQL queries. References: Running SQL Queries Programmatically, Formatted string literals, spark.sql

質問: 42

Databricks の AutoLoader の機能は何ですか？

- A. Auto Loader は、クラウドストレージから新しいファイルを自動的に取り込んで処理し、スキーマの進化をサポートせずにバッチ データとストリーミング データを処理します。
- B. Auto Loader は、クラウドストレージから新しいファイルを自動的に取り込んで処理し、スキーマの進化をサポートせずにストリーミング データのみを処理します。
- C. Auto Loader は、クラウドストレージから新しいファイルを自動的に取り込んで処理し、スキーマの進化をサポートしながらバッチ データを処理します。
- D. Auto Loader は、クラウドストレージから新しいファイルを自動的に取り込んで処理し、スキーマの進化をサポートしながらバッチ データとストリーミング データの両方を処理します。

正解: ([正解を表示します](#))

**質問: 43**

エンジニアリングマネージャーは、Databricks SQLクエリを使用して、最近のプロジェクトのパフォーマンスを監視したいと考えています。プロジェクトのリリース後1週間は、クエリ結果を1分ごとに更新したいと考えています。しかし、クエリに使用されるコンピューティングリソースが稼働し続け、プロジェクトリリース後1週間を過ぎても組織に多大なコストがかかることを懸念しています。

プロジェクトのリリース後 1 週間を過ぎてもクエリによって組織にコストが発生しないようにするために、エンジニアリング チームが使用できるアプローチは次のどれですか。

- A. SQL エンドポイントによって消費される DBU の数に制限を設定できます。
- B. クエリの更新スケジュールを、一定回数の更新後に終了するように設定できます。
- C. プロジェクトのリリース後 1 週間を過ぎても、クエリによって組織にコストが発生しないことを保証できません。
- D. クエリの更新スケジュールを管理できるユーザーの数に制限を設定できます。
- E. クエリ スケジューラで、クエリの更新スケジュールを特定の日付に終了するように設定できます。

正解: [E \(コメントを發表する\)](#)

In Databricks SQL, you can use scheduled query executions to update your dashboards or enable routine alerts. By default, your queries do not have a schedule. To set the schedule, you can use the dropdown pickers to specify the frequency, period, starting time, and time zone. You can also choose to end the schedule on a certain date by selecting the End date checkbox and picking a date from the calendar. This way, you can ensure that the query does not run beyond the first week of the project's release and does not incur any additional cost. Option A is incorrect, as setting a limit to the number of DBUs does not stop the query from running. Option B is incorrect, as there is no option to end the schedule after a certain number of refreshes. Option C is incorrect, as there is a way to ensure the query does not cost the organization money beyond the first week of the project's release. Option D is incorrect, as setting a limit to the number of individuals who can manage the query's refresh schedule does not affect the query's execution or cost. References: Schedule a query, Schedule a query - Azure Databricks - Databricks SQL

**質問: 44**

ブロンズ テーブルと生データの間を説明するのは次のどれですか。

- A. ブロンズ テーブルには、生データ ファイルよりも少ないデータが含まれています。
- B. ブロンズ テーブルには、生データよりも真実に近いデータが含まれています。
- C. ブロンズ テーブルには集計が含まれますが、生のデータは集計されません。
- D. ブロンズ テーブルには、生データよりも精度の低いデータ ビューが含まれています。
- E. ブロンズ テーブルには、スキーマが適用された生データが含まれています。

正解: [\(正解を表示します\)](#)

Bronze tables are the first layer of a medallion architecture, which is a data design pattern used to organize data in a lakehouse. Bronze tables contain raw data ingested from various sources, such as RDBMS data, JSON files, IoT data, etc. The table structures in this layer correspond to the source system table structures

"as-is", along with any additional metadata columns that capture the load date/time, process ID, etc. The only transformation applied to the raw data in this layer is to apply a schema, which defines the column names and data types of the table. The schema can be inferred from the data source or specified explicitly. Applying a schema to the raw data enables the use of SQL and other structured query languages to access and analyze the data. Therefore, option E is the correct answer. References: What is a Medallion Architecture?, Raw Data Ingestion into Delta Lake Bronze tables using Azure Synapse Mapping Data Flow, Apache Spark + Delta Lake concepts, Delta Lake Architecture & Azure Databricks Workspace.

**質問: 45**

データエンジニアは、オンプレミスのPostgreSQLデータベースの売上データとAzure Synapseの顧客データを統合し、包括的なレポートを作成する必要があります。目標は、データの重複を回避し、最新の情報を確保することです。データエンジニアは、Databricksを使用してどのようにこれを実現すればよいのでしょうか？

- A. Lakehouse Federation を使用して両方のデータソースを直接クエリします
- B. Databricks にデータを取り込むためのカスタム ETL パイプラインを開発する
- C. 両方のソースのデータを手動で単一のデータベースに同期します
- D. 両方のソースからデータをCSVファイルにエクスポートし、Databricksにアップロードします。

正解: [A \(コメントを發表する\)](#)

**質問: 46**

データエンジニアは、いくつかのテーブルからデータエンティティを作成したいと考えています。このデータエンティティは、他のセッションの他のデータエンジニアによって使用される必要があります、物理的な場所に保存する必要があります。

データ エンジニアが作成する必要があるデータ エンティティは次のどれですか。

- A. データベース
- B. 関数
- C. 表示
- D. 一時ビュー
- E. テーブル

正解: [\(正解を表示します\)](#)

A table is a data entity that is stored in a physical location and can be accessed by other data engineers in other sessions. A table can be created from one or more tables using the CREATE TABLE or CREATE TABLE AS SELECT commands. A table can also be registered from an existing DataFrame using the spark.

catalog.createTable method. A table can be queried using SQL or DataFrame APIs. A table can also be updated, deleted, or appended using the MERGE INTO command or the DeltaTable API.

References:

- \* Create a table
- \* Create a table from a query result
- \* Register a table from a DataFrame
- \* [Query a table]
- \* [Update, delete, or merge into a table]

有効的なDatabricks-Certified-Data-Engineer-Associate-JPN問題集はJPNTTest.com提供され、Databricks-Certified-Data-Engineer-Associate-JPN試験に合格することに役に立ちます！JPNTTest.comは今最新Databricks-Certified-Data-Engineer-Associate-JPN試験問題集を提供します。JPNTTest.com Databricks-Certified-Data-Engineer-Associate-JPN試験問題集はもう更新されました。ここでDatabricks-Certified-Data-Engineer-Associate-JPN問題集のテストエンジンを手に入れます。最新版のアクセス、<https://www.jpntest.com/shiken/Databricks-Certified-Data-Engineer-Associate-JPN-mondaishu> 156問、30%ディスカウント、特別な割引コード: **JPNshiken**

質問: 47

データエンジニアは、eコマース取引のデルタテーブルのデータレイアウトとクエリパフォーマンスを最適化する必要があります。テーブルは「purchase\_date」という日付列でパーティション分割されています。これは時間ベースのクエリには役立ちますが、高カーディナリティ列であるユーザー統計「customer\_id」の検索には適していません。

テーブルは通常、「customer\_id」のフィルターを使用してクエリされます。

特定の日付範囲内で「d」というデータが生成されますが、このデータは各パーティション内の複数のファイルに分散されているため、完全なパーティション スキャンが必要となり、実行時間とコストが増加します。

データ エンジニアは、効率的な読み取りのためにデータ レイアウトをどのように最適化すればよいでしょうか。

- A. 「customer\_id」と「purchase\_date」で Liquid クラスタリングを実装するテーブルを変更します。
- B. テーブルを「customer\_id」でパーティション分割するように変更します。
- C. パフォーマンス向上のため、頻繁な読み取りがキャッシュされるように、クラスター上でデルタ キャッシュを有効にします。
- D. 既存のパーティションを維持しながら、「customerid」に Liquid Clustering を実装するテーブルを変更します。

正解: ([正解を表示します](#))

**質問: 48**

データエンジニアリングチームには2つのテーブルがあります。1つ目のテーブル

「march\_transactions」は、3月のすべての小売取引のコレクションです。2つ目のテーブル

「april\_transactions」は、4月のすべての小売取引のコレクションです。これらのテーブル間に重複レコードはありません。

march\_transactions と april\_transactions のすべてのレコードを重複レコードなしで含む新しいテーブル all\_transactions を作成するには、次のコマンドのどれを実行する必要がありますか？

**A.** all\_transactions テーブルを CREATE する

march\_transactions から \* を選択

内部結合 SELECT \* FROM april\_transactions;

**B.** all\_transactions テーブルを作成する

march\_transactions から \* を選択

UNION SELECT \* FROM april\_transactions;

**C.** all\_transactions テーブルを作成する

march\_transactions から \* を選択

外部結合 SELECT \* FROM april\_transactions;

**D.** all\_transactions テーブルを作成する

march\_transactions から \* を選択

INTERSECT SELECT \* from april\_transactions;

**E.** すべてのトランザクションをテーブルとして作成する

march\_transactions から \* を選択

MERGE SELECT \* FROM april\_transactions;

正解: **B** ([コメントを發表する](#))

The correct command to create a new table that contains all records from two tables without duplicate records is to use the UNION operator. The UNION operator combines the results of two queries and removes any duplicate rows. The INNER JOIN, OUTER JOIN, and MERGE operators do not remove duplicate rows, and the INTERSECT operator only returns the rows that are common to both tables. Therefore, option B is the only correct answer. References: Databricks SQL Reference - UNION, Databricks SQL Reference - JOIN, Databricks SQL Reference - MERGE, [Databricks SQL Reference - INTERSECT]

**質問: 49**

ゴールド テーブルとシルバー テーブルの関係を説明するのは次のどれですか。

**A.** Gold テーブルには Silver テーブルよりも集計が含まれる可能性が高くなります。

**B.** ゴールド テーブルには、シルバー テーブルよりも貴重なデータが含まれる可能性が高くなります。

**C.** ゴールド テーブルには、シルバー テーブルよりもデータのビューがあまり洗練されていない可能性が高くなります。

**D.** Gold テーブルには Silver テーブルよりも多くのデータが含まれる可能性が高くなります。

E. ゴールド テーブルには、シルバー テーブルよりも真実のデータが含まれる可能性が高くなります。

正解: [A \(コメントを發表する\)](#)

According to the medallion lakehouse architecture, gold tables are the final layer of data that powers analytics, machine learning, and production applications. They are often highly refined and aggregated, containing data that has been transformed into knowledge, rather than just information. Silver tables, on the other hand, are the intermediate layer of data that represents a validated, enriched version of the raw data from the bronze layer. They provide an enterprise view of all its key business entities, concepts and transactions, but they may not have all the aggregations and calculations that are required for specific use cases. Therefore, gold tables are more likely to contain aggregations than silver tables. References:

\* What is the medallion lakehouse architecture?

\* What is a Medallion Architecture?

質問: 50

CREATE TABLE AS SELECT ステートメントを使用するときに、CSV ではなく Parquet から外部テーブルを作成する利点を説明しているのは次のどれですか。

- A. Parquetファイルはパーティション分割できます
- B. CREATE TABLE AS SELECT 文はファイルでは使用できません
- C. Parquetファイルには明確に定義されたスキーマがあります
- D. Parquetファイルは最適化できる
- E. ParquetファイルはDeltaテーブルになります

正解: [\(正解を表示します\)](#)

Option C is the correct answer because Parquet files have a well-defined schema that is embedded within the data itself. This means that the data types and column names of the Parquet files are automatically detected and preserved when creating an external table from them. This also enables the use of SQL and other structured query languages to access and analyze the data. CSV files, on the other hand, do not have a schema embedded in them, and require specifying the schema explicitly or inferring it from the data when creating an external table from them. This can lead to errors or inconsistencies in the data types and column names, and also increase the processing time and complexity.

CREATE TABLE AS SELECT, Parquet Files, CSV Files, Parquet vs. CSV

質問: 51

データ エンジニアは、Databricks SQL ダッシュボードを 1 日に 1 回更新するようにスケジュール設定したいと考えていますが、関連付けられている SQL エンドポイントは必要なときにのみ実行したいと考えています。

データ エンジニアがダッシュボードの更新スケジュールで使用される SQL エンドポイントの合計実行時間を最小限に抑えるために使用できるアプローチは次のどれですか。

- A. ダッシュボードの SQL エンドポイントが各クエリの SQL エンドポイントと一致していることを確認できます。
- B. ダッシュボードの SQL エンドポイントをサーバーレスに設定できます。
- C. SQL エンドポイントの自動停止機能をオンにできます。
- D. SQL エンドポイントのクラスター サイズを縮小できます。
- E. ダッシュボードの SQL エンドポイントが、含まれているクエリの SQL エンドポイントの 1 つではないことを確認できます。

正解: [B \(コメントを发表する\)](#)

A serverless SQL endpoint is a compute resource that is automatically managed by Databricks and scales up or down based on the workload. A serverless SQL endpoint can be used to run queries and dashboards without requiring manual configuration or management. A serverless SQL endpoint is only active when it is needed and shuts down automatically when idle, minimizing the total running time and cost. A serverless SQL endpoint can be created and assigned to a dashboard using the Databricks SQL UI or the SQL Analytics API. References:

- \* Create a serverless SQL endpoint
- \* Assign a SQL endpoint to a dashboard
- \* SQL Analytics API

質問: 52

データエンジニアは毎日ステートメントを実行し、前日の売上をテーブルtransactionsにコピーします。各日の売上は、`transactions/raw`という場所にある個別のファイルに保存されます。

今日、データ エンジニアは次のコマンドを実行してこのタスクを完了します。

今日コマンドを実行した後、データ エンジニアはテーブル トランザクション内のレコード数が変更されていないことに気がきました。

次のどれが、ステートメントがテーブルに新しいレコードをコピーしなかった理由を説明していますか？

- A. コピーするファイルの形式が `FORMAT_OPTIONS` キーワードに含まれていませんでした。
- B. コピーするファイルの名前が `FILES` キーワードに含まれていませんでした。
- C. 前日のファイルはすでにテーブルにコピーされています。
- D. `PARQUET` ファイル形式は `COPY INTO` をサポートしていません。
- E. `COPY INTO` ステートメントでは、コピーされた行を表示するためにテーブルを更新する必要があります。

正解: [\(正解を表示します\)](#)

The `COPY INTO` statement is an idempotent operation, which means that it will skip any files that have already been loaded into the target table. This ensures that the data is not duplicated or corrupted by multiple attempts to load the same file. Therefore, if the data engineer runs the same command every day without specifying the names of the files to be copied with the `FILES` keyword or a glob pattern with the `PATTERN` keyword, the statement will only copy the first file that matches the source location and ignore the rest. To avoid this problem, the data engineer should either use the `FILES` or `PATTERN` keywords to filter the files to be copied based on the

date or some other criteria, or delete the files from the source location after they are copied into the table2. References: 1: COPY INTO | Databricks on AWS 2: Get started using COPY INTO to load data | Databricks on AWS

**質問: 53**

Databricksワークフローは、ノートブックのエラーにより最終ステージで失敗します。このワークフローは毎日実行されています。データエンジニアはエラーを修正し、パイプラインを再実行したいと考えています。このワークフローの実行には、非常にコストと時間がかかります。

ダウンタイムとコストを最小限に抑えるために、データエンジニアはどのようなアクションを実行する必要がありますか？

- A. 別のクラスターに切り替える
- B. 修復実行
- C. ワークフロー全体を再実行します
- D. クラスターを再起動します

正解: ([正解を表示します](#))

**質問: 54**

次の構造化ストリーミングクエリのうち、シルバーテーブルからゴールドテーブルへのホップを実行しているのはどれですか。

- A.
- B.
- C.
- D.
- E.

正解: ([正解を表示します](#))

The best practice is to use "Complete" as output mode instead of "append" when working with aggregated tables. Since gold layer is work final aggregated tables, the only option with output mode as complete is option E.

**質問: 55**

あるデータエンジニアは、小売業のユースケースにおいて、データ分析ダッシュボードへの入力データのクリーン度をDatabricks SQLダッシュボードで監視しています。このジョブには、売上が0である店舗レベルのレコードの数を返すDatabricks SQLクエリが含まれています。データエンジニアは、この値が0より大きい場合、メッセージングWebhookを介してチーム全体に通知したいと考えています。

売上が0ドルの店舗数が0より大きい場合に、データエンジニアがメッセージングWebhookを介してチーム全体に通知するために使用できるアプローチは次のどれですか。

- A. カスタムテンプレートを使用してアラートを設定できます。
- B. 新しい電子メールアラートの送信先を指定してアラートを設定できます。
- C. 1回限りの通知でアラートを設定できます。

D. 新しい Webhook アラートの送信先を持つアラートを設定できます。

E. 通知なしでアラートを設定できます。

正解: [D \(コメントを發表する\)](#)

A webhook alert destination is a notification destination that allows Databricks to send HTTP POST requests to a third-party endpoint when an alert is triggered. This enables the data engineer to integrate Databricks alerts with their preferred messaging or collaboration platform, such as Slack, Microsoft Teams, or PagerDuty. To set up a webhook alert destination, the data engineer needs to create and configure a webhook connector in their messaging platform, and then add the webhook URL to the Databricks notification destination. After that, the data engineer can create an alert for their Databricks SQL query, and select the webhook alert destination as the notification destination. The alert can be configured with a custom condition, such as when the number of stores with \$0 in sales is greater than zero, and a custom message template, such as "Alert: {number\_of\_stores} stores have \$0 in sales". The alert can also be configured with a recurrence interval, such as every hour, to check the query result periodically. When the alert condition is met, the data engineer and their team will receive a notification via the messaging webhook, with the custom message and a link to the Databricks SQL query. The other options are either not suitable for sending notifications via a messaging webhook (A, B, E), or not suitable for sending recurring notifications. References: Databricks Documentation - Manage notification destinations, Databricks Documentation - Create alerts for Databricks SQL queries, Databricks Documentation - Configure alert conditions and messages.

質問: 56

データ エンジニアに新しいデータ レコードが渡されました。

id STRING = 'a1'

rank INTEGER = 6

rating FLOAT = 9.4

既存の Delta テーブル my\_table に新しいレコードを追加するために使用できる SQL コマンドは次のどれですか。

A. INSERT INTO my\_table VALUES ('a1', 6, 9.4)

B. my\_table UNION VALUES ('a1', 6, 9.4)

C. INSERT VALUES ('a1', 6, 9.4) INTO my\_table

D. UPDATE my\_table VALUES ('a1', 6, 9.4)

E. UPDATE VALUES ('a1', 6, 9.4) my\_table

正解: [\(正解を表示します\)](#)

To append a new record to an existing Delta table, you can use the INSERT INTO statement with the VALUES clause. This statement will insert one or more rows into the table with the specified values.

Option A is the only code block that follows this syntax correctly. Option B is incorrect, as it uses the UNION operator, which will return a new table that is the union of two tables, not append to an existing table. Option C is incorrect, as it uses the INSERT VALUES statement, which is not a valid SQL syntax.

Option D is incorrect, as it uses the UPDATE statement, which will modify existing rows in the table, not append new rows. Option E is incorrect, as it uses the UPDATE VALUES statement, which is also not a valid SQL syntax. References: Insert data into a table using SQL | Databricks on AWS, Insert data into a table using SQL - Azure Databricks, Delta Lake Quickstart - Azure Databricks

**質問: 57**

データエンジニアはコストを削減し、クラウド支出を最適化したいと考えています。既存のSLAを維持しながらクラウドコストを削減するために、Databricks Serverlessを使用することを決定しました。

Databricks Serverless への移行の最初のステップは何ですか？

- A. 低頻度の BI ダッシュボードとアドホック SQL 分析
- B. 最新の Databricks ランタイムおよび Unity Catalog と互換性のある、頻繁に実行される効率的な Python ベースのデータ変換パイプライン
- C. 最新の Databricks ランタイムおよび Unity Catalog と互換性のある、頻繁に実行される効率的な Scala ベースのデータ変換パイプライン
- D. ソース API、ファイル、JDBC/ODBC 接続からの取り込みを含む従来の取り込みパイプライン

正解: [\(正解を表示します\)](#)

**質問: 58**

新しいデータエンジニアリングチームがELTプロジェクトに割り当てられました。新しいデータエンジニアリングチームには、プロジェクトを完全に管理するために、salesテーブルに対する完全な権限が必要です。

新しいデータ エンジニアリング チームにデータベースに対する完全な権限を付与するには、次のコマンドのどれを使用できますか？

- A. テーブル sales のすべての権限を team に付与します。
- B. テーブル sales に対する SELECT CREATE MODIFY 権限を team に付与します。
- C. テーブル sales に対する SELECT 権限を team に付与します。
- D. テーブル sales の使用権限を team に付与します。
- E. テーブル team のすべての権限を sales に付与します。

正解: [\(正解を表示します\)](#)

To grant full permissions on a table to a user or a group, you can use the GRANT ALL PRIVILEGES ON TABLE statement. This statement will grant all the possible privileges on the table, such as SELECT, CREATE, MODIFY, DROP, ALTER, etc. Option A is the only code block that follows this syntax correctly. Option B is incorrect, as it does not grant all the possible privileges on the table, but only a subset of them. Option C is incorrect, as it only grants the SELECT privilege on the table, which is not enough to fully manage the project. Option D is incorrect, as it grants the USAGE privilege on the table, which is not a valid privilege for tables. Option E is incorrect, as it grants all the privileges on the table team to the user or group sales, which is the opposite of what the question asks. References: Grant privileges on a table using

質問: 59

アセットバンドルの構造は何ですか？

- A. メタデータを伴わずにワークスペース アセットのみを含む圧縮アーカイブ (ZIP)。
- B. ランタイム環境とアセットのソースコードを含むDockerイメージ
- C. プロジェクトの成果物、リソース、および構成を指定する YAML 構成ファイル。
- D. 新しいワークスペースに移行されるアセットの名前を列挙した単一のプレーンテキスト ファイル。

正解: [\(正解を表示します\)](#)

質問: 60

Databricks を Unity Catalog と連携して使用している企業のデータエンジニアは、Unity Catalog に対応した Databricks ワークスペースを使用している外部パートナーとテーブルコレクションを共有する必要があります。データエンジニアは、これを実現するために Delta Sharing を使用することにしました。

Delta Sharing を設定するために、データ エンジニアが外部パートナーに最初に要求する必要がある情報は何ですか？

- A. Databricks クラスターの名前
- B. DatabricksワークスペースのIPアドレス
- C. Unity Catalogメタストアの共有識別子
- D. Databricksアカウントのパスワード

正解: [C \(コメントを發表する\)](#)

質問: 61

データ エンジニアは、テーブルから読み取り、データを操作し、新しいテーブルにストリーミング書き込みを実行するように構造化ストリーミング ジョブを構成しました。

データ エンジニアが使用するコード ブロックは次のとおりです。

データ エンジニアが、クエリで利用可能なすべてのデータを必要な数のバッチで処理することだけを望む場合、データ エンジニアは空白を埋めるために次のどのコード行を使用する必要がありますか。

- A. 処理時間(1)
- B. トリガー(availableNow=True)
- C. トリガー(parallelBatch=True)
- D. トリガー(処理時間="1回")
- E. トリガー(連続="1回")

正解: [B \(コメントを發表する\)](#)

<https://spark.apache.org/docs/latest/api/python/reference/pyspark.ss/api/pyspark.sql.streaming.DataStreamWriter.trigger.html>

有効的なDatabricks-Certified-Data-Engineer-Associate-JPN問題集はJPNTTest.com提供され、Databricks-Certified-Data-Engineer-Associate-JPN試験に合格することに役に立ちます！JPNTTest.comは今最新Databricks-Certified-Data-Engineer-Associate-JPN試験問題集を提供します。JPNTTest.com Databricks-Certified-Data-Engineer-Associate-JPN試験問題集はもう更新されました。ここでDatabricks-Certified-Data-Engineer-Associate-JPN問題集のテストエンジンを手に入れます。最新版のアクセス、<https://www.jpntest.com/shiken/Databricks-Certified-Data-Engineer-Associate-JPN-mondaishu> 156問、30%ディスカウント、特別な割引コード: **JPNshiken**」

### 質問: 62

データエンジニアリングチームは、Databricks SQLクエリを実行されていないSQLエンドポイントに送信すると、実行速度が遅くなることに気づきました。データエンジニアリングチームはこの問題を解決したいと考えています。

このシナリオで結果を返すまでの時間を短縮するために、チームが使用できるアプローチは次のどれですか。

- A. SQL エンドポイントのサーバーレス機能をオンにし、スポット インスタンス ポリシーを「信頼性最適化」に変更できます。
- B. SQL エンドポイントの自動停止機能をオンにすることができます。
- C. SQL エンドポイントのクラスター サイズを増やすことができます。
- D. SQL エンドポイントの Serverless 機能をオンにできます。
- E. SQL エンドポイントのスケールアップ範囲の上限を増やすことができます

正解: D ([コメントを發表する](#))

Option D is the correct answer because it enables the Serverless feature for the SQL endpoint, which allows the endpoint to automatically scale up and down based on the query load. This way, the endpoint can handle more concurrent queries and reduce the time it takes to return results. The Serverless feature also reduces the cold start time of the endpoint, which is the time it takes to start the cluster when a query is submitted to a non- running endpoint. The Serverless feature is available for both AWS and Azure Databricks platforms.

Databricks SQL Serverless, Serverless SQL endpoints, New Performance Improvements in Databricks SQL

### 質問: 63

重複レコードの書き込みを回避しながら Delta テーブルにデータを書き込むために使用できるコマンドはどれですか。

- A. ドロップ
- B. 無視
- C. ゴー

D. 追加

E. 挿入

正解: ([正解を表示します](#))

The MERGE command can be used to upsert data from a source table, view, or DataFrame into a target Delta table. It allows you to specify conditions for matching and updating existing records, and inserting new records when no match is found. This way, you can avoid writing duplicate records into a Delta table<sup>1</sup>. The other commands (DROP, IGNORE, APPEND, INSERT) do not have this functionality and may result in duplicate records or data loss<sup>234</sup>. References: 1: Upsert into a Delta Lake table using merge | Databricks on AWS 2: SQL DELETE | Databricks on AWS 3: SQL INSERT INTO | Databricks on AWS 4: SQL UPDATE | Databricks on AWS

質問: 64

データ エンジニアは、バッチ取り込みパイプラインの一部として次のコード ブロックを使用して、コンポーザブル テーブルから読み取ります。

トランザクション テーブルがストリーム ソースである場合にこのコード ブロックが機能するには、次のどの変更を加える必要がありますか。

- A. 予測をストリーム対応の予測関数に置き換える
- B. スキーマ(schema)をオプション("maxFilesPerTrigger", 1)に置き換えます
- C. transactions」をDeltaテーブルの場所へのパスに置き換えます
- D. format("delta") を format("stream") に置き換えます。
- E. spark.read を spark.readStream に置き換えます

正解: ([正解を表示します](#))

To read from a stream source, the data engineer needs to use the spark.readStream method instead of the spark.

read method. The spark.readStream method returns a DataStreamReader object that can be used to specify the details of the input source, such as the format, the schema, the path, and the options. The spark.read method is only suitable for batch processing, not streaming processing. The other changes are not necessary or correct for reading from a stream source. References: Structured Streaming Programming Guide, Read a stream, Databricks Data Sources

質問: 65

データエンジニアは個人用ラップトップで作業しており、クラウドストレージ上のDelta Lakeに保存されているデータに対して複雑な変換を実行する必要があります。エンジニアは、Databricks クラスターと連携し、ローカルIDEで作業するためにDatabricks Connectを使用することにしました。

Databricks Connect を使用すると、エンジニアはどのようにして Databricks クラスターと対話しながら、ローカル マシン上でシームレスにコードを開発、テスト、デバッグできるようになるのでしょうか。

- A. Databricks ランタイムを模倣したローカル環境を提供することで、エンジニアが好みの IDE を使用してコードを開発、テスト、デバッグできるようになります。
- B. Databricksランタイムを模倣したローカル環境を提供することで、エンジニアはDatabricks独自のWebインターフェースを通じてのみコードを開発、テスト、デバッグできるようになります。
- C. Databricks ランタイムを模倣したローカル環境を提供することで、エンジニアは Databricks に必要な特定の IDE を使用してコードを開発、テスト、デバッグできるようになります。
- D. ネットワーク接続を必要とせずにローカルマシンからSparkジョブを直接実行できるようになることで

正解: ([正解を表示します](#))

**質問: 66**

データエンジニアがクラウドストレージから新しいデータを取り込むためのスクリプトを作成しています。スキーマが変更された場合、取り込みは失敗するはずですが、変更の下流ソースが見つかり、意図した変更であることが検証されるまで、取り込みは失敗するはずですが。

どのコマンドが要件を満たしますか？

- A. レスキュー
- B. 新しい列を追加する
- C. なし
- D. 新しい列の失敗

正解: ([正解を表示します](#))

**質問: 67**

データエンジニアは、クラウド内の社内カスタム定義ネットワーク内にあるデータベースに保存されているデータに対して探索的分析を実施する必要があります。このタスクにはSQLを使用しています。

どのタイプの SQL ウェアハウスを使用すると、データ エンジニアが大量のクエリを迅速かつコスト効率よく処理できるようになりますか？

- A. クラシック SQL ウェアハウス
- B. ノートブック向けサーバーレスコンピューティング
- C. プロ SQL ウェアハウス
- D. サーバーレス SQL ウェアハウス

正解: ([正解を表示します](#))

**質問: 68**

Auto Loader と互換性のあるワークロードの種類は何ですか？

- A. バッチワークロード
- B. サーバーレスワークロード
- C. 機械学習ワークロード
- D. ストリーミングワークロード

正解: ([正解を表示します](#))

**質問: 69**

データエンジニアは、パートナー組織にDatabricksアカウントの導入を支援しています。両チームはビジネスユースケースを共有しています。データエンジニアは、Unity-Catalogで管理されている差分テーブルと、それらのテーブルを作成するノートブックジョブの一部をパートナー組織と共有する必要があります。

データ エンジニアは、必要な情報をシームレスに共有するにはどうすればよいでしょうか？

- A. データとノートブックは、Unity Catalog を使用するだけで簡単に共有できます。
- B. すべてのコードを圧縮して電子メールで共有し、データレイクからのデータ取り込みを可能にします
- C. Delta Sharing を介して必要なデータセットとノートブックを共有します。Unity Catalog を介して権限を管理します。
- D. Github 経由でコードベースへのアクセスを共有し、データレイクからデータセットを取り込むことを許可します。

正解: ([正解を表示します](#))

**質問: 70**

データ エンジニアが単一ノード クラスターを使用するシナリオを説明しているのは次のどれですか。

- A. 少量のデータを使って対話的に作業しているとき
- B. できるだけ早く更新される自動レポートを実行している場合
- C. Databricks SQL内でSQLを操作している場合
- D. 大規模なデータに自動的にスケールする能力について懸念がある場合
- E. 大量のデータを含むレポートを手動で実行する場合

正解: ([正解を表示します](#))

The scenario in which a data engineer will want to use a single-node cluster is when they are working interactively with a small amount of data. A single-node cluster is a cluster consisting of an Apache Spark driver and no Spark workers<sup>1</sup>. A single-node cluster supports Spark jobs and all Spark data sources, including Delta Lake<sup>1</sup>. A single-node cluster is helpful for single-node machine learning workloads that use Spark to load and save data, and for lightweight exploratory data analysis<sup>1</sup>. A single-node cluster can run Spark locally, spawn one executor thread per logical core in the cluster, and save all log output in the driver log<sup>1</sup>. A single-node cluster can be created by selecting the Single Node button when configuring a cluster<sup>1</sup>.

The other options are not suitable for using a single-node cluster. When running automated reports to be refreshed as quickly as possible, a data engineer will want to use a multi-node cluster that can scale up and down automatically based on the workload demand<sup>2</sup>. When working with SQL within Databricks SQL, a data engineer will want to use a SQL Endpoint that can execute SQL queries on a serverless pool or an existing cluster<sup>3</sup>. When concerned about the ability to automatically scale with larger data, a data engineer will want to use a multi-node cluster that can leverage the Databricks Lakehouse Platform and the Delta Engine to handle large-scale

data processing efficiently and reliably<sup>4</sup>. When manually running reports with a large amount of data, a data engineer will want to use a multi-node cluster that can distribute the computation across multiple workers and leverage the Spark UI to monitor the performance and troubleshoot the issues.

:

- 1: Single Node clusters | Databricks on AWS
- 2: Autoscaling | Databricks on AWS
- 3: SQL Endpoints | Databricks on AWS
- 4: Databricks Lakehouse Platform | Databricks on AWS
- 5: [Spark UI | Databricks on AWS]

#### 質問: 71

Databricks Lakehouse プラットフォームを使用する利点のうち、Delta Lake によって提供されるものはどれですか？

- A. さまざまな言語を使用して同じデータを操作する能力
- B. 1つのノートブックでリアルタイムに共同作業できる機能
- C. クエリ失敗時のアラートを設定する機能
- D. バッチおよびストリーミングワークロードをサポートする機能
- E. 複雑なデータ操作を分散する機能

正解: **D** ([コメントを发表する](#))

Delta Lake is the optimized storage layer that provides the foundation for storing data and tables in the Databricks lakehouse. Delta Lake is fully compatible with Apache Spark APIs, and was developed for tight integration with Structured Streaming, allowing you to easily use a single copy of data for both batch and streaming operations and providing incremental processing at scale<sup>1</sup>. Delta Lake supports upserts using the merge operation, which enables you to efficiently update existing data or insert new data into your Delta tables<sup>2</sup>. Delta Lake also provides time travel capabilities, which allow you to query previous versions of your data or roll back to a specific point in time<sup>3</sup>. References: 1: What is Delta Lake? | Databricks on AWS 2: Upsert into a table using merge | Databricks on AWS 3: [Query an older snapshot of a table (time travel) | Databricks on AWS] Learn more

有効的な**Databricks-Certified-Data-Engineer-Associate-JPN**問題集はJPNTTest.com提供され、**Databricks-Certified-Data-Engineer-Associate-JPN**試験に合格することに役に立ちます！JPNTTest.comは今最新**Databricks-Certified-Data-Engineer-Associate-JPN**試験問題集を提供します。JPNTTest.com **Databricks-Certified-Data-Engineer-Associate-JPN**試験問題集はもう更新されました。ここで**Databricks-Certified-Data-Engineer-Associate-JPN**問題集のテストエンジンを手に入れます。最新版のアクセス、<https://www.jpntest.com/shiken/Databricks->

[Certified-Data-Engineer-Associate-JPN-mondaishu](#) 156問、30%ディスカウント、特別な割引  
コード: **JPNshiken**」