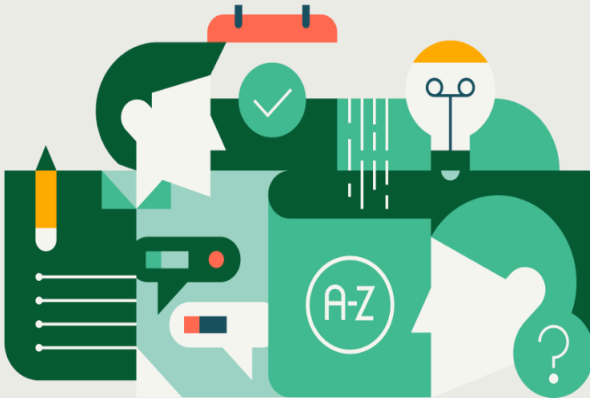
databricks

Databricks 試験ガイド

Databricks 認定 データエンジニアアソシエイト



試験ガイドへのフィードバックを送信

本試験ガイドの目的

本試験ガイドは、試験の概要と出題範囲を説明し、受験の準備状況を判断していただくことを目的としています。本ドキュメントは、試験に変更が加えられるたびに（およびその変更が有効になる時点で）更新されるため、常に最新の状態で準備を進めることができます。本ガイドは、2026 年 5 月 4 日時点の試験バージョンに対応しています。

対象者について

Databricks 認定データエンジニアアソシエイト認定試験では、Databricks Data Intelligence Platform を活用して基礎的なデータエンジニアリングタスクを実行する能力を評価します。本試験では、Data Intelligence Platform とそのワークスペース、アーキテクチャ、および機能に関する知識に加え、データの取り込み、データのロード、データ変換とモデリングに関連するタスク（PySpark を使用した抽出・変換・ロード（ETL）タスクの実行、Lakeflow Jobs の利用、CI/CD など）に関する知識を評価します。最後に、本試験では、トラブルシューティング、モニタリング、最適化の手法に関する理解、および Databricks Platform におけるガバナンスとセキュリティの実現に関する知識を評価します。

試験について

- 採点対象の項目数: 採点対象となる選択式問題 45 問。
- 制限時間: 90 分。
- 受験料: 200 USD、および現地の法律で定められた該当する税金。● 実施方法: オンラインまたはテストセンター
- 試験補助ツール: 使用は一切認められません。
- 前提条件: 必須の前提条件はありません。ただし、コースの受講と Databricks での 6 か月間の実務経験を強く推奨します。
- 有効期間: 2 年間。
- 再認定: 認定を維持するには、2 年ごとに再認定が必要です。再認定を受けるには、その時点で提供されている最新の試験を受験する必要があります。再受験に向けた準備については、試験の Web ページにある「Getting Ready for the Exam」セクションをご確認ください。
- 採点対象外のコンテンツ: 試験には、将来の利用に向けた統計情報を収集するために、採点対象外の項目が含まれる場合があります。これらの項目は試験フォーム上では区別されておらず、スコアには影響しません。これらのコンテンツのための時間は、別途考慮されています。

推奨トレーニング

- 講師によるトレーニング: Data Engineering with Databricks
- セルフペース（Databricks Academy で提供）:
  - Data Ingestion with Lakeflow Connect
  - Deploy Workloads with Lakeflow Jobs
  - DevOps Essentials for Data Engineering
  - Data Interoperability with Unity Catalog
  - Build Data Pipelines with Lakeflow Spark Declarative pipeline
  - Get Started with Data Governance on Databricks

試験のアウトライン

セクション **1: Databricks Intelligence Platform**（**6%**）

- アーキテクチャ、Delta Lake、Unity Catalog など、Databricks Data Intelligence Platform の主要なコンポーネントを理解する。
- Databricks Data Intelligence Platform のコンピュートサービスについて、その特性、制限、コストモデルを含めて理解し、各ワークロードのユースケースに最適な選択肢を選定する。

セクション **2:** データの取り込みとロード（**21%**）

- バッチ、ストリーミング、増分ロードなどのデータ取り込みパターンを有効化して詳細に把握し、ローカルファイル、Lakeflow Connect 標準コネクター、Lakeflow Connect マネージドコネクターなどのソースからデータをインポートする。
- COPY INTO コマンドを使用して、クラウドオブジェクトストレージ（ADLS/S3/GCS）から Unity Catalog の管理下にあるテーブルにファイルを増分的にロードする。
- バッチモード（例: ディレクトリリストまたはファイル通知）で、スキーマ強制とスキーマ進化を有効にした Auto Loader を使用して、Unity Catalog の管理下にあるテーブルにデータを取り込む。
- 多様なエンタープライズソースから Unity Catalog の管理下にあるテーブルにデータを確実に取り込むように Lakeflow Connect を構成する。
- ノートブックで JDBC/ODBC または REST クライアントを使用して、クラウドストレージまたは Unity Catalog の管理下にあるテーブルに直接データを取り込む。これは通常、Lakeflow Jobs でオーケストレーションおよびスケジュールされる。
- データ量、取り込み頻度、データ型、Unity Catalog でのガバナンス要件などの技術的要件に基づいて、Auto Loader、Lakeflow Connect（標準コネクターおよびマネージドコネクター）、パートナーコネクター、その他の取り込み方法の間で優先順位を付ける。
- Lakeflow Connect やその他のマネージドコネクターを介して、半構造化データおよび非構造化データ（例: JSON やネストされたデータ）を、Unity Catalog の管理下にある Delta テーブルに取り込む。

セクション **3:** データ変換とモデリング（**22%**）

- PySpark/SQL で bronze テーブルを読み取り、null 値をクリーニングし、データ型を標準化して新しい silver テーブルに書き込むことで、データクリーニングを実装する。
- inner join、left join、broadcast join、複数キーによる結合、cross join、union、union all などの操作で DataFrame を結合する。
- 列名の追加、削除、分割、名前変更、フィルタの適用、配列の展開によって、列、行、テーブル構造を操作する。
- count、approximate count distinct、mean、summary など、DataFrame に対する重複排除操作と集計操作を実行する。
- 基本的なチューニングパラメータ（spark.sql.shuffle.partitions:、
- spark.default.parallelism, spark.executor/driver.memory,
- spark.sql.autoBroadcastJoinThreshold）を理解し、パフォーマンスを再測定する。
- Unity Catalog で、BI チームや分析チーム向けのマテリアライズドビュー、ビュー、ストリーミングテーブル、テーブルなどの Gold 層オブジェクトの違いと構築方法を理解する。
- 信頼性の高い Silver および Gold データセットを確保するために、データ品質チェックと検証ルールを適用する。

セクション **4: Lakeflow Jobs** の利用（**16%**）

- パイプラインのオーケストレーションのために、Lakeflow Jobs を使用して制御フロー（再試行、および分岐やループなどの条件付きタスク）を実装する
- Lakeflow Jobs とその DAG ベースのタスクグラフを使用して、一般的なタスク（ノートブック、SQL クエリ、ダッシュボード、パイプラインの各タスク）とそれらの依存関係を構成する
- トリガーの種類（スケジュール、ファイル到着、テーブル更新）を理解した上で、Lakeflow Jobs を使用してジョブのスケジュールを実装する
- データの可用性とパイプラインの依存関係に基づいて、時間ベースのトリガーとデータ駆動型のトリガーのいずれかを選択する。

セクション **5: CI/CD** の実装（**10%**）

- Databricks Git Folders（旧 Databricks Repos）でのブランチの作成と切り替え、変更のコミットとプッシュ、Databricks Git 連携を使用したプルリクエストの作成など、Databricks ワークスペース UI 内でコード開発ワークフローを管理する。
- 同一のコードベースを dev、test、prod の各ターゲットにプロモートしながら、Automation Bundle（旧 Databricks Asset Bundles）の変数とオーバーライドを使用した環境固有の構成を理解する。
- Declarative Automation Bundles（旧 Databricks Asset Bundles）をデプロイして、Lakeflow Jobs、Lakeflow Spark Declarative Pipelines、その他のワークスペースアセットを dev、test、prod の各環境にわたってパッケージ化、構成、プロモートする。
- 自動化された CI/CD ワークフローで Declarative Automation Bundles（旧 Databricks Asset Bundles）やその他のワークスペースアセットを検証、デプロイ、管理するために、Databricks CLI を理解する。

セクション **6:** トラブルシューティング、モニタリング、最適化（**10%**）

- Lakeflow Jobs の実行履歴ビューを使用して現在の実行時間を過去のベースラインと比較し、ジョブのパフォーマンスの傾向を特定する。
- Lakeflow Jobs UI を使用して、ジョブのステータスを解釈し、DAG ベースのタスクグラフを参照して上流のブロッカーを特定し、パイプラインの実行時間と失敗率を追跡することで、パイプラインの健全性をモニタリングする。
- Spark UI でステージレベルのメトリクスを解釈して、データスキュー、シャッフル、ディスクスピルなどの一般的なパフォーマンスのボトルネックを特定する。
- Liquid Clustering と predictive optimization の機能を理解する。
- クラスターの起動失敗、ライブラリの競合、メモリ不足の問題を診断する。

セクション **7:** ガバナンスとセキュリティ（**15%**）

- Unity Catalog におけるマネージドテーブルと外部テーブルを区別し、それらに対して基本的な操作（作成、変更、削除、およびマネージドテーブルと外部テーブル間の変換）を実行する。
- セキュリティ階層の適切なレベルで、プリンシパル（ユーザー、グループ、サービスプリンシパル）に GRANT、REVOKE、DENY の権限を適用して、UI と SQL でアクセス制御を構成する。
- ユーザーグループに基づいてデータの可視性を制限するための、列レベルのマスキングと行レベルセキュリティを理解する。
- 機密データの行レベルフィルタリングと列マスキングを一元的に制御するための Unity Catalog ABAC ポリシーを理解する。

サンプル問題

これらの問題は、以前のバージョンの試験から退役したものです。目的は、試験ガイドに記載されている学習目標を示し、各学習目標に対応したサンプル問題を提供することです。試験ガイドには、試験で出題される可能性のある学習目標が一覧として示されています。認定試験の準備をする最も良い方法は、試験ガイドの試験アウトラインを確認することです。

問題 **1**
学習目標*: Spark UI* でステージレベルのメトリクスを解釈して、データスキュー、シャッフル、ディスクスピルなどの一般的なパフォーマンスのボトルネックを特定する。

あるデータエンジニアが、新しいデータソースをオンボードした後に、バッチジョブの所要時間が 2 倍になったことに気づきました。Spark UI では、最も長いステージにおいて、ほとんどのタスクは 30 秒未満で完了していますが、1 つのタスクだけが 10 分以上かかっています。このステージのタスクサマリでは、shuffle read の最小値/中央値が 400 MB 前後であるのに対し、shuffle read の最大値は 5 GB を超えています。

ジョブの実行時間を短縮する解決策はどれですか？
A. クラスターサイズを大きくしてエグゼキューターを増やし、遅いタスクがより早く完了するようにする

B. skew join 処理を伴う adaptive query execution が有効になっていることを確認し、実行時に過大なパーティションを自動的に分割する
C. spark.sql.shuffle.partitions を減らして、より少ないタスクに処理を集約する
D. 結合の前に salt キーを使用してデータセットを手動で再パーティションし、偏ったキーを均等に分散させる

問題 **2**
学習目標*: Databricks Data Intelligence Platform* のコンピュートサービスについて、その特性、制限、コストモデルを含めて理解し、各ワークロードのユースケースに最適な選択肢を選定する。

あるデータエンジニアは、パイプラインの迅速な反復を必要としつつ、不正な取り込み後の確実なロールバックを維持し、規制コンプライアンスのための監査トレールを確保し、AI と BI の両方のワークロードに対して単一の信頼できる情報源への一貫したアクセスを提供する必要があります。

これらの要件を満たすために、データエンジニアはどの戦略を使用すべきですか？

A. 手動のファイルバージョニングとロールバック用の夜間コピーを伴う DBFS 上の CSV ストレージ。
B. Delta Lake の ACID トランザクションとタイムトラベルを使用し、一貫したアクセスとリネージのために Unity Catalog で管理する。
C. クラウドオブジェクトストレージのみを使用し、リカバリとガバナンスにはアドホックな SQL クエリを使用する。
D. 監査トレールと BI 配信のために、一時的なインメモリの DataFrame を使用する。

問題 **3**
学習目標*:* バッチ、ストリーミング、増分ロードなどのデータ取り込みパターンを有効化して詳細に把握し、ローカルファイル、*Lakeflow Connect* 標準コネクター、*Lakeflow Connect* マネージドコネクターなどのソースからデータをインポートする。

あるデータエンジニアは、顧客所有の S3 バケットから Databricks の監査ログを取り込む下流のパイプラインを構築しています。スキーマ推論とチェックポイント処理を実装する前に、配信形式、一般的な取り込みのレイテンシ、およびファイルが上書きされる可能性があるかどうかを把握したいと考えています。

Databricks の監査ログのストレージの動作はどのようなものですか？

A. ファイルは JSON として配信され、一般的には配信開始後 15 分以内にイベントがログに記録され、新しい配信によって既存のファイルが上書きされることがある
B. ファイルは CSV として配信され、1 分未満のレイテンシが保証され、不変性を保つためにファイルが一度書き込まれると上書きは一切発生しない
C. ファイルは Parquet として配信され、24 時間を超える結果整合性を持ち、ストリーミング取り込みを簡素化するために上書きは無効化されている
D. ファイルは JSON として週次のバッチ間隔で配信され、上書きは追記せずに以前のコンテンツを完全に置き換える

問題 **4**
学習目標*:* クラスターの起動失敗、ライブラリの競合、メモリ不足の問題を診断する。

あるデータエンジニアリングチームは、キュレーションされた Delta テーブルに対して、一日中アドホックな SQL クエリを実行する複数のビジネスアナリストをサポートしています。チームは、非常に大きなクラスターへの不必要なスケーリングを避けてコストを管理しながら、効率的なクエリパフォーマンス、高速なクラスター起動、および複数の同時ユーザーへのサポートを確保する必要があります。

これらの要件を満たすクラスター構成はどれですか？

A. スケジュールされた ETL ワークフロー向けに設計された、オートスケーリング付きの job cluster
B. 固定数のワーカーノードで構成された all-purpose cluster
C. オートスケーリングを有効にした high-concurrency cluster
D. 軽量な開発タスク向けに構成された single-node cluster

問題 **5**
学習目標*: Databricks Git Folders*（旧 *Databricks Repos*）でのブランチの作成と切り替え、変更のコミットとプッシュ、*Databricks Git* 連携を使用したプルリクエストの作成など、*Databricks* ワークスペース *UI* 内でコード開発ワークフローを管理する。

あるチームは、Databricks で ETL パイプラインをデプロイ、バージョン管理、およびオーケストレーションするモジュラーな方法を希望しており、CI/CD と再現性を実現したいと考えています。
この要件をサポートする機能はどれですか？

A. Unity Catalog のモデルを使用して ETL ジョブを表現し、各モデルにパイプラインのコードアーティファクトを保存し、CI/CD が Job タスクに紐づいたモデルエイリアスを更新することでバージョンをプロモートする。
B. 変換ロジックを Unity Catalog Volumes に保存された wheel ライブラリとしてパッケージ化し、Jobs タスクにバインドして、環境間での決定的なデプロイを確保する。
C. API ロジックを Volume にマウントされたノートブック内にパッケージ化し、Jobs API v2 を使用してノートブックをトリガーし、バージョニングシステムとしてノートブックのリビジョン履歴に依存する。
D. DABs を使用してリソースとコードアセットを定義し、Git でバージョン管理し、自動化された CI/CD アクションを通じて環境間でデプロイをプロモートする。

解答:

1. B

2. B

3. A

4. C

5. D