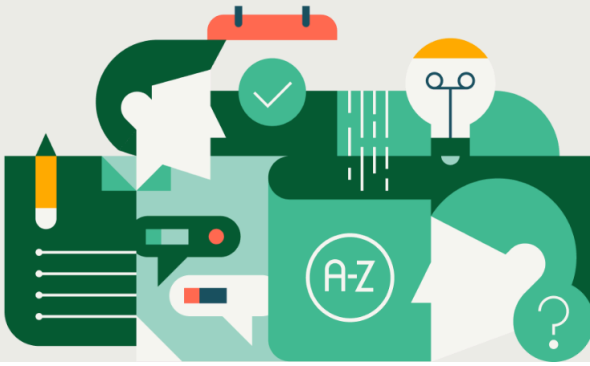
databricks

Databricks 試験ガイド

Databricks Certified Generative AI Engineer Associate



試験ガイドのフィードバックを提供する

情報を更新する

更新	日付
試験公開日	March 18, 2026
試験ガイド更新日	March 18, 2026

試験日の2週間前に、当社のウェブページで試験ガイドをご確認いただき、最新バージョンの試験に向けて準備していることをご確認ください。

この試験ガイドの目的

この試験ガイドの目的は、試験の概要と出題範囲をお伝えし、受験準備が整っているかどうかを判断するお手伝いをすることです。このドキュメントは、試験に変更があった場合に更新され（変更の発効日も記載されます）、自信を持って準備を進めていただけるようになっています。このガイドは、以上に記載された発効日時点で有効な試験バージョンを対象としています。

対象者の説明

Databricks Certified Generative AI Engineer Associate認定試験は、Databricksを使用してLLMを活用したソリューションを設計・実装する個人の能力を評価します。これには、複雑な要件を管理可能なタスクに分解する問題分解、および包括的なソリューションを開発するための現在のGenerative AIランドスケープから適切なモデル、ツール、アプローチを選択することが含まれます。また、セマンティック類似性検索のためのVector Search、モデルとソリューションのデプロイのためのModel Serving、ソリューションのライフサイクル管理のためのMLflow、データガバナンスのためのUnity Catalogなど、Databricks固有のツールも評価されます。この試験に合格した方は、Databricksとそのツールセットをフル活用した高性能なRAGアプリケーションおよびLLMチェーンを構築・デプロイできることが期待されます。

試験について

項目	詳細
採点対象の問題数	採点対象の多肢選択式および複数選択式の問題 45 問
採点対象外の問題	採点対象外の問題が最大 10 問含まれる場合があります（後述の「採点対象外のコンテンツ」を参照）
出題される問題の総数	採点対象の問題 45 問＋採点対象外の問題
制限時間	90 分（採点対象外の問題に対応するための追加時間を含みます）
受験料	200 米ドル（現地の法律に基づき、適用される税金が別途かかります）
受験方法	オンライン監督付き試験のみ
提供言語	英語、日本語、ブラジルポルトガル語（pt-BR）、韓国語
補助資料	使用不可
前提条件	なし（ただし、関連トレーニングの受講を強く推奨します）
推奨される経験	試験の出題範囲に沿った Databricks Platform での実務経験を 6 か月以上有することを強く推奨します
有効期間	2年

再認定：認定ステータスを維持するには、**[2]** 年ごとに再認定が必要です。再認定を受けるには、現行の本番試験を受験する必要があります。

採点対象外のコンテンツ：試験には、将来の利用に向けて統計的な情報を収集するために、採点対象外の問題が含まれる場合があります。これらの問題は試験中に識別されず、スコアにも影響しません。また、これらの問題に対応するための追加時間が制限時間に考慮されています。

推奨される準備

トレーニング

- インストラクター主導**: Generative AI Engineering with Databricks**
- 自己ペース（**Databricks Academy** で利用可能）：
    - Agent Bricks を使用した RAG Agents の構築
    - Databricks でのエージェント型アプリケーションの構築
    - Agent Evaluation on Databricks
    - Databricks におけるエージェントアプリケーションのデプロイとモニタリング

AI Prep ガイド - すべての Databricks 試験

Databricks AI Prep Guide ガイドでは、すでにお使いのAIチャットボットをDatabricksの任意の認定資格向けパーソナライズされたチュートリアルに変える方法をご紹介します。公式試験ガイドを活用してAIが提供しがちな古い回答を見極め、自分の弱点を把握し、すべてのDatabricks試験で求められる実践的なトレーニングと組み合わせることができます。

ハンズオン演習

何を練習すればよいかわからない場合は、AI Prep ガイド（以上の「AI Prep ガイド」セクションを参照）をご確認ください。このガイドには「ステップ 5 — ハンズオンの最低要件」というプロンプトが含まれており、AI チャットボットがこの試験の目標に対応した 6〜10 個のハンズオンタスクの個別チェックリストを生成します。各タスクは Databricks Free Edition で完了できる小さなものです。

Databricksドキュメント

準備として、この試験でカバーされているプラットフォーム領域に関する公式Databricksドキュメントを学習してください。それは最も最新の権威あるソースです。各トピックをどの程度深く学習するかをジャッジするため、また非推奨の機能ではなく最新の機能を学んでいることを確認するために活用してください。公式ドキュメント：Databricksドキュメント

試験の概要

セクション 1: アプリケーションの設計 (14%)

- 特定のフォーマットで応答を引き出すプロンプトを設計する
- 特定のビジネス要件を達成するためのモデルタスクを選択する
- 目的のモデルの入力と出力に合わせてチェーンコンポーネントを選択する
- ビジネスユースケースの目標を、AI パイプラインに求められる入力と出力の説明に変換する。
- マルチステージの推論のために知識を収集したりアクションを実行したりするツールを定義し、順序付けする
- 問題を解決するために Agent Bricks（Knowledge Assistant、Multiagent Supervisor、Information Extraction）をどのように、またいつ使用するかを判断する

セクション 2: データ準備 (14%)

- 特定のドキュメント構造とモデルの制約に対してチャンキング戦略を適用する
- RAG アプリケーションの品質を低下させるソースドキュメント内の不要なコンテンツをフィルタリングする。
- 提供されたソースデータからドキュメントのコンテンツを抽出してフォーマットするための適切な Python パッケージを選んでください。

- チャンク化されたテキストをUnity CatalogのDelta Lakeテーブルに書き込むための操作とシーケンスを定義する
- 特定のRAGアプリケーションに必要な知識と品質を提供するために必要なソースドキュメントを特定する。
- ツールとメトリクスを使用して検索のパフォーマンスを評価します
- 高度なチャンキング戦略を使用した検索システムの設計
- 情報検索プロセスにおける再ランキングの役割を説明してください

セクション 3: アプリケーション開発 (30%)

- Generative AI アプリケーションで使用する LangChain や類似ツールを選択する。
- 品質や安全性などの一般的な問題を特定するために、レスポンスを定性的に評価する。
- モデルと検索の評価に基づいてチャンキング戦略を選択する
- キーフィールド、用語、およびインテントに基づいて、ユーザーの入力からの追加コンテキストでプロンプトを拡張します。
- ベースラインから望ましい出力へ LLM の応答を調整するプロンプトを作成する
- 否定的な結果を防ぐために LLM ガードレールを実装する
- 開発するアプリケーションの属性に基づいて、最適な LLM を選択する
- ソースドキュメント、想定されるクエリ、および最適化戦略に基づいて、埋め込みモデルのコンテキスト長を選択する。
- モデルのメタデータ/モデルカードに基づいて、タスクに適したモデルをモデルハブまたは marketplace から選択する。
- 実験で生成された一般的なメトリクスに基づいて、特定のタスクに最適なモデルを選択する
- agentic systems の開発に MLflow と Agent Framework を活用する
- Gen AIアプリケーションライフサイクルの評価フェーズとモニタリングフェーズを比較する
- マルチエージェントシステムが Genie Spaces または会話型 API を活用してデータを取得できるようにします。

セクション4: アプリケーションの組み立てとデプロイ (22%)

- 前処理と後処理を含む pyfunc モデルを使用してチェーンをコーディングする
- Model Serving エンドポイントからリソースへのアクセスを制御する
- 要件に従ってシンプルなチェーンをコーディングする
- RAG アプリケーションを作成するために必要な基本要素を選択してください。モデルフレーバー、埋め込みモデル、リトリーバー、依存関係、入力例、モデルシグネチャ
- MLflow を使用してモデルを Unity Catalog に登録する
- Vector Search インデックスを作成してクエリーを実行する
- Foundation Model APIsを活用したLLMアプリケーションを提供する方法を特定する。
- Mosaic AI Vector Search の主要なキーの概念とコンポーネントを説明してください。
- バッチ推論ワークロードを特定し、ai_query() を適切に適用する
- 埋め込みの数、更新頻度、レイテンシ、およびコスト要件に基づいて、特定のソリューション向けに Vector Search を設定する
- 中間メモリや構造化された情報を保存および取得するために、永続的なデータストアを設定します。

- Vector Search インデックスの更新、環境間でのプロンプトのプロモーション、エージェントの個々のコンポーネントのテストなど、CI/CD のベストプラクティスを適用する。
- 特定のアプリケーション要件に基づいて、マネージド、外部、およびカスタムの MCP サーバーを統合する。
- プロンプトのバージョン管理を適用するし、プロンプトのライフサイクルを管理する
- エージェントの使用シナリオに適したインタラクティブなユーザー向けインターフェース（Apps、Slack、Teams など）を開発する。

セクション5: ガバナンス (8%)

- パフォーマンス目標を達成するためのガードレールとして、マスキング技術を使用します。
- Gen AIアプリケーションへの悪意のあるユーザー入力から保護するためのガードレール技術を選択してください。
- 法的リスクを回避するために、データソースの法的・ライセンス要件を使用する。
- GenAIアプリケーションにデータを供給するデータソースにおける問題のあるテキストの軽減策として、代替手段を推奨する

セクション 6: 評価とモニタリング (12%)

- 定量的な評価メトリクスのセットに基づいてLLMの選択（サイズとアーキテクチャ）を行う
- 特定のLLMデプロイメントシナリオに対して監視するキーメトリクスを選択する
- MLflow のスコアリングとトレーシングを使用してエージェントのパフォーマンスを評価する
- 推論ログを使用して、展開されたRAGアプリケーションのパフォーマンスを評価する
- Databricks の機能を使用して LLM のコストを管理する
- 推論テーブルとAgent Monitoringを使用して、ライブLLM endpointをトラッキングする
- グラウンドトゥルースを必要とする評価ジャッジを特定する
- AI Gateway（Inference Tables、Usage Tables、レート制限）を使用して、Agent Framework を介して展開された LLM またはエージェントを追跡する。
- エージェントとLLMsの評価にDatabricksのカスタムScorerを使用する
- SME のフィードバックを取り入れてエージェントのパフォーマンスを向上させます。

サンプル問題

サンプル問題の目的は、試験ガイドに記載されている目標を示し、各目標に対応する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目標が記載されています。最善の準備方法は、以上の試験日程に対応した試験の全アウトラインを確認することです。

問題 1

目標：特定のドキュメント構造とモデルの制約に対してチャンキング戦略を適用する

Generative AI エンジニアが、最大 1 億件までしか格納できないベクターデータベースに 1 億 5,000 万件のエンベディングをロードしようとしています。レコード数を削減するために取ることができる 2 つのアクションはどれですか？

A. ドキュメントのチャンクサイズを増やす
B. チャンク間のオーバーラップを減らす
C. ドキュメントのチャンクサイズを小さくする
D. チャンク間のオーバーラップを増やす
E. より小さな埋め込みモデルを使用する

質問 2

目標：特定の ***RAG*** アプリケーションに必要な知識と品質を提供するために必要なソースドキュメントを特定する。

Generative AI エンジニアが、自動車部品の販売を支援するために開発中のカスタマー向け GenAI アプリケーションからのレスポンスを評価しています。このアプリケーションでは、質問に回答するためにお客様が account_id と transaction_id を明示的に入力する必要があります。リリース後、注文や請求の詳細については適切に回答できたが、配送や予定到着日に関する質問には正確に回答できなかったというフィードバックがありました。このアプリケーションのこれらの質問への回答能力を向上させるには、どのアプローチが有効ですか？

A. すべての自動車部品に関する会社の配送ポリシーと支払い条件を含むベクターストアを作成する
B. transaction_id をプライマリキーとして、請求書データと予定配達日を格納した feature store テーブルを作成する
C. 期待される到着日のサンプルデータをチューニングデータセットとして提供し、モデルが最新の配送情報を持つように定期的にファインチューニングを行います。
D. 注文が行われた日時を入力としてチャットプロンプトに追加し、どの配送方法も14日を超えることはないため、その日付に14日を追加するようモデルに指示してください。

質問 3

目標：提供されたソースデータからドキュメントのコンテンツを抽出し、フォーマットするために適切な ***Python*** パッケージを選択する。

Generative AI エンジニアが、.jpeg や .png などの形式の画像ファイルとしてスキャン・保存されたソースドキュメントから取得したコンテキストに依存する RAG アプリケーションを構築しています。できるだけ少ないコード量でソリューションを開発したいと考えています。ソースドキュメントからテキストを抽出するには、どの Python パッケージを使用すべきですか？

A. beautifulsoup
B. scrapy
C. pytesseract
D. pyquery

質問 4

目標：ソースドキュメント、想定されるクエリ、および最適化戦略に基づいて、埋め込みモデルのコンテキスト長を選択する。

Generative AI エンジニアが LLM ベースのアプリケーションを作成しています。リトリーバー用のドキュメントは、それぞれ最大512トークンにチャンク分割されています。このエンジニアは、このアプリケーションではコストとレイテンシが品質よりも重要であることを把握しています。選択できるコンテキスト長のレベルがいくつかあります。どれが要件を満たしますか？

A. コンテキスト長 512: 最小モデルは 0.13GB、埋め込み次元は 384
B. コンテキスト長 514: 最小モデルは 0.44GB、埋め込み次元は 768
C. コンテキスト長 2048：最小モデルは 11GB、埋め込み次元は 2560
D. コンテキスト長 32768: 最小モデルは 14GB、埋め込み次元は 4096

質問 5

目的：開発するアプリケーションの属性に基づいて、最適な ***LLM*** を選択する。

Generative AI エンジニアが、段落程度の長さのメモフィールドを、メモの意図を示す1文の要約に更新し、アプリケーションのフロントエンドに収まるようにするアプリケーションを構築したいと考えています。このアプリケーションに適した LLMs を評価する際、どの自然言語処理タスクカテゴリを使用すべきですか？

A. text2text Generation
B. Sentencizer
C. テキスト分類
D. 要約

質問 6

目標**:** エンベディングの数、更新頻度、レイテンシ、およびコスト要件に基づいて、特定のソリューション向けに ***Vector Search*** を設定する。

オンライン小売業者に勤めるGenerative AI エンジニアが、Vector Search とメタデータフィルタリングを組み合わせて検索機能の改善を試みています。検索は最大で毎秒80件に達する可能性があり、レイテンシが最も重要な指標です。レイテンシを損なわずに精度が向上するのであれば、初期開発コストは問いません。在庫は全国で1億点あります。このエンジニアはどのように設定すべきでしょうか？

A. GTE Large埋め込みモデルを活用し、ハイブリッド検索と再ランキングを有効にした標準的なVector Searchを使用します。
B. GTE Large埋め込みモデルを活用し、ハイブリッド検索と再ランキングを有効にしたストレージ最適化 Vector Searchを使用します。
C. カスタム埋め込みモデルをファインチューニングし、標準的な Vector Search を使用し、ハイブリッド検索とリランキングはオフにします。
D. カスタム埋め込みモデルをファインチューニングし、ストレージ最適化された Vector Search を使用し、ハイブリッド検索とリランキングはオフのままにします。

質問 7

目標：***Vector Search*** インデックスの更新、環境間でのプロンプトの昇格、エージェントの個々のコンポーネントのテストなど、***CI/CD*** のベストプラクティスを適用する。

Generative AI エンジニアが、開発、ステージング、本番運用にわたってエージェントのプロンプトテンプレートを管理する必要があります。チームはゲート付きリリースプロセスを必要としています。プロンプトは開発環境で更新され、自動テストによってステージングで検証され、承認後にのみ本番運用に昇格されます。このソリューションはバージョン履歴を保持し、必要に応じて以前のプロンプトバージョンへのロールバックを可能にする必要があります。この昇格ワークフローをサポートするアプローチはどれですか？

A. プロンプトテンプレートをアプリケーションリポジトリに保存し、テストが通過した後にステージングブランチを本番運用ブランチにマージすることでプロモートします。
B. プロンプトをMLflowのバージョンとして追跡し、合格後にエイリアスを使用してプロンプトをプロモートします。
C. CI ランナー上の JSON ファイルにプロンプトを保存し、実行のたびに本番環境のプロンプトを上書きします。
D. プロンプトをDelta テーブルに格納し、一貫性を確保するためにデプロイのたびに本番環境のテーブルを上書きします。

問題 8

目標：エージェントの使用シナリオに適したインタラクティブなユーザー向けインターフェースを開発する（***Apps***、***Slack***、***Teams*** など）。

Generative AI エンジニアが、カスタマーサポートエージェントが質問を行い、社内の PDF に基づいた回答を受け取れる Databricks App を構築しています。要件は次のとおりです。ユーザーは企業のアイデンティティで認証する必要があり、アプリはブラウザに長期有効なトークンを公開せずに Mosaic AI Agent endpoint を呼び出す必要があり、回答へのアクセスは各ユーザーの権限を尊重する必要があります。これらの要件を満たすアプローチはどれですか？

A. Databricks App のバックエンドを使用して、アプリのクレデンシャルで Agent endpoint を呼び出し、アプリの認証済みコンテキストを通じてユーザーのアイデンティティ／権限を適用します。
B. Databricks のパーソナルアクセストークン（PAT）をアプリの JavaScript に保存し、ブラウザから Agent の endpoint を直接呼び出す。
C. Agent endpoint を公開し、アプリのフロントエンドに埋め込まれた API キーで保護します。
D. Agent がID確認なしに読み取れるよう、PDFをパブリックバケットにエクスポートします。

質問 9

目標：特定のアプリケーション要件に基づいて、マネージド、外部、およびカスタムの ***MCP*** サーバーを統合する。

Generative AI エンジニアが、インターネットデータソースから事実に基づく情報にアクセスし、外部 API を使用してウェブ検索を実行する必要があるリサーチアシスタントエージェントを構築しています。Databricks は

このインターネットデータソース向けにマネージド MCP サーバーを提供しており、キーを必要とする外部 API 向けには外部 MCP サーバーが利用可能です。このアプリケーションは、本番運用において両方のデータソースへの安定したアクセスを確保しながら、メンテナンスのオーバーヘッドを最小限に抑える必要があります。エンジニアがこれらのデータソースをエージェントに統合するために取るべき2つのアクションはどれですか？

A. インターネットリソースと外部 API の両方をラップし、エージェントが呼び出せる単一の統合インターフェースとして提供するカスタム MCP サーバーを構築します。
B. マネージドウェブブラウザー MCP サーバーを使用して、情報を取得するためにインターネットリソースにプログラム的にアクセスしてください。
C. Unity Catalog の外部テーブルを設定して、インターネットリソースのコンテンツと検索結果をキャッシュし、エージェントがオフラインでアクセスできるようにします。
D. エージェントのMCP サーバー設定で、サーバータイプを「managed」と指定し、インターネットリソースのサーバー識別子を指定することで、マネージドMCP サーバーを設定します。
E. 外部MCPサーバーの接続情報を提供し、API キーをDatabricks Secretsに保存して、MCPサーバーの設定でそれを参照することで、外部MCPサーバーをデプロイします。

質問 10

目的***: SME*** のフィードバックを取り入れてエージェントのパフォーマンスを向上させる

Generative AI エンジニアが、操作チームが社内で使用しているカスタマーサポート用 RAG アシスタントの評価を担当しています。4人のドメインエキスパートが、事実の正確さ、完全性、有用性などの観点から、毎週サンプリングされた回答を MLflow でレビューしています。数回のラウンドを経て、エンジニアは同じ回答に対するエキスパートの評価が大きくばらつき、モデルの改善をトラッキングするための評価データが信頼できないものになっていることに気づきました。エンジニアは、評価者の不一致を減らしつつ、反復的な品質改善をサポートできる、信頼性の高い評価プロセスを必要としています。エンジニアは何をすべきでしょうか？

A. LLM-as-a-judge を使用して過去および将来のレスポンスを再採点し、専門家間の意見の相違を調整する代わりに、モデルが生成した評価を主要な真実のソースとして扱います。
B. 明確なルーブリックを定義し、基準についてSMEをキャリブレーションし、一貫したAgent Evaluationのためにmlflow.genai.evaluate()で統一された判断を使用します。
C. 各回答についてすべてのドメインエキスパートのスコアを平均し、その混合スコアをモデルチューニングの決定的なベンチマークとして直接使用します。
D. すべての専門家がすでに合意している回答のみからベンチマークを構築し、一貫性を向上させるために評価セットから意見が分かれているケースを除外する。

解答キーと解説

#	解答	解説
1	A, B	チャンクを大きくし、オーバーラップを減らすと、どちらもチャンクの総数が減り、埋め込み（embedding）数が 1 億件の上限を下回ります。チャンクを小さくしたり、オーバーラップを増やしたりすると、件数は増加します。
2	B	予定納期は構造化されたトランザクション単位の事実であり、transaction_id をキーとする特徴量ストア（feature store）のテーブルで扱うのが最適です。ポリシーのベクトルストアやファインチューニングでは適切ではありません。固定の 14 日を加算する方法は不正確です。
3	C	ソースはスキャンされた画像（.jpeg/.png）であるため OCR が必要です。pytesseract は最小限のコードで画像からテキストを抽出します。beautifulsoup、scrapy、pyquery は HTML／Web スクレイピング用のツールであり、OCR ではありません。
4	A	チャンクは 512 トークンに制限されており、品質よりもコストとレイテンシが優先されるため、コンテキスト 512、0.13 GB、384 次元のモデルが最も効率的な選択です。より大きなコンテキストや次元数のモデルは、このケースでは利点がなく、コストとレイテンシだけを増やします。
5	D	段落を一文の要約に凝縮するのは要約（summarization）タスクであるため、候補となる LLM は Summarization NLP タスクカテゴリで評価すべきです。
6	C	レイテンシが最重要指標であるため、レイテンシを増やすハイブリッド検索とリランキングは無効のままにします。チームは初期コストを許容しているため、ファインチューニング済みのカスタム埋め込みモデルで精度を高め、標準のベクトル検索でスループット要件を満たします。
7	B	プロンプトを MLflow のバージョンとして追跡し、エイリアスで昇格させることで、完全なバージョン履歴とロールバックを備えたゲート付きワークフローが実現します。ブランチのマージ、JSON の上書き、または Delta の上書きでは、適切なバージョン管理が失われます。
8	A	Databricks App のバックエンドはトークンをサーバー側で保持し、企業の ID でユーザーを認証し、ユーザーごとの権限を適用するため、すべての要件を満たします。PAT／API キーをブラウザーに埋め込むことや、データを公開することは制約に違反します。
9	D, E	インターネットソースについては、エージェントの MCP 構成を通じてマネージド MCP サーバーを設定し（D）、外部 MCP サーバーは API キーを Databricks Secrets に格納してデプロイします（E）。これにより、メンテナンスを最小限に抑えつつ、本番環境での信頼性の高いアクセスを確保できます。カスタムラッパーや外部テーブルによるキャッシュはオーバーヘッドを増やします。
10	B	評価者間の不一致は、明確なルーブリックを定義して SME をキャリブレーションし、整合した判定を mlflow.genai.evaluate() で使用することで修正します。ノイズの多いスコアの平均化、意見が分かれたケースの除外、または LLM ジャッジへの盲目的な依存では、根本原因に対処できません。