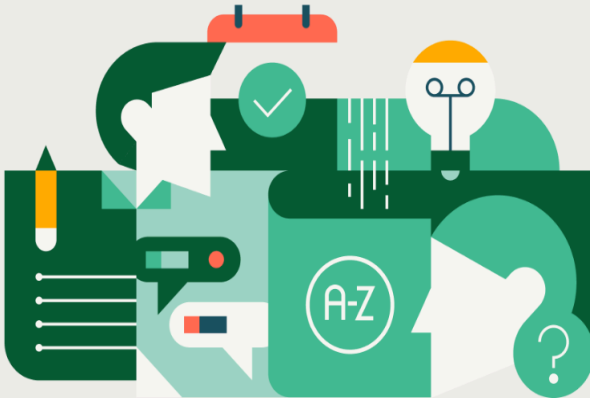
databricks

Guia de Exame Databricks

Exame de Renovação Databricks Certified Data Engineer Associate



Enviar feedback sobre este guia de exame

Objetivo deste guia

O objetivo deste guia é fornecer uma visão geral do exame e do que é abordado nele, ajudando você a avaliar seu preparo para a prova. Este documento será atualizado sempre que houver mudanças no exame (e quando essas mudanças entrarem em vigor), para que você possa se preparar. Este guia abrange a versão do exame vigente em 15 de junho de 2026.

Descrição do público-alvo

Este exame de renovação Databricks Certified Data Engineer Associate avalia a capacidade do indivíduo de utilizar a Databricks Data Intelligence Platform para executar tarefas fundamentais de engenharia de dados. O exame avalia o conhecimento de um Databricks Certified Data Engineer Associate sobre as tarefas relacionadas a ingestão de dados, carregamento de dados, transformação de dados e modelagem — como a capacidade de realizar tarefas de Extract, Transform, Load (ETL) usando PySpark / SQL, trabalhar com Lakeflow Jobs e CI/CD. Por fim, o exame avalia a compreensão dos candidatos sobre técnicas de solução de problemas, monitoramento e otimização, bem como seu conhecimento sobre como implementar governança e segurança no Databricks Platform.

Sobre o exame

- Número de itens avaliados: 25 questões de múltipla escolha avaliadas.
- Tempo limite: 60 minutos.
- Taxa de inscrição: 200 USD
- Formato de aplicação: somente on-line com supervisão (proctored)
- Materiais de apoio: nenhum permitido.
- Pré-requisitos: este exame de renovação está disponível para indivíduos que possuem uma credencial Databricks Certified Data Engineer Associate existente que já tenha expirado ou esteja a até 2 meses de expirar. Recomenda-se fortemente que os candidatos concluam o treinamento sugerido e tenham pelo menos 12 meses de experiência prática com a Databricks Platform.
- Validade: 2 anos.

- Recertificação: os candidatos devem renovar suas credenciais a cada 2 anos concluindo com sucesso o exame vigente ou a versão específica de recertificação disponível naquele ano. Consulte a seção “Getting Ready for the Exam” na página do exame para se preparar para a prova.

Treinamento recomendado

- Com instrutor (Instructor-led): Data Engineering with Databricks
- No seu próprio ritmo (Self-paced, disponível na Databricks Academy):
  - 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

Estrutura do exame

Seção 1: Ingestão e carregamento de dados (32%)

- Usar COPY INTO para realizar a ingestão incremental de arquivos do object storage na nuvem para tabelas Delta governadas pelo Unity Catalog.
- Usar o Auto Loader para ingerir arquivos de forma incremental em tabelas Delta governadas pelo Unity Catalog
- Configurar o Lakeflow Connect para ingerir dados de diversas fontes corporativas de forma confiável em tabelas governadas pelo Unity Catalog.
- Usar clientes JDBC/ODBC ou REST em notebooks para depositar dados no armazenamento em nuvem ou diretamente em tabelas governadas pelo Unity Catalog.
- Priorizar entre Auto Loader, conectores de parceiros do Lakeflow Connect e outros métodos de ingestão com base em volume, frequência, tipo de dados e necessidades de governança.
- Ingerir dados semiestruturados e não estruturados (por exemplo, JSON e dados aninhados) por meio do Lakeflow Connect e de managed connectors em tabelas Delta.

Seção 2: Transformação e modelagem de dados (24%)

- Aplicar limpeza de dados, lendo tabelas bronze com PySpark/SQL (tratar valores nulos, padronizar tipos de dados) e gravar os resultados em tabelas silver.
- Identificar parâmetros comuns de ajuste do Spark e compreender seu efeito geral no desempenho.
- Diferenciar os objetos da camada Gold no Unity Catalog (materialized views, views, streaming tables, tabelas comuns) e seu uso para BI/analytics.
- Aplicar verificações de qualidade de dados e regras de validação para garantir conjuntos

de dados Silver e Gold confiáveis

Seção 3: Trabalhando com Lakeflow Jobs (12%)

- Implementar fluxos de controle (repetições de tarefas, ramificações, laços) usando Lakeflow Jobs para orquestrar fluxos de trabalho de dados com múltiplas etapas
- Configurar tipos comuns de tarefas e dependências no Lakeflow Jobs usando um fluxo de trabalho baseado em DAG
- Configurar agendamentos e triggers do Lakeflow Jobs, incluindo os triggers scheduled, file arrival, table update e continuous.

Seção 4: Implementação de CI/CD (12%)

- Gerenciar código na interface do Databricks workspace usando Git Folders: criar/alternar branches, fazer commit, push e abrir pull requests.
- Usar os targets de ambiente do Declarative Automation Bundle (dev, test, prod) para promover uma base de código entre ambientes.
- Usar a Databricks CLI para validar e implantar Declarative Automation Bundles (DABs)

Seção 5: Solução de problemas, monitoramento e otimização (12%)

- Identificar tendências no desempenho dos jobs usando a visão de histórico de execuções do Lakeflow Jobs.
- Usar a interface do Lakeflow Jobs para monitorar a integridade do pipeline, os status dos jobs, os grafos de tarefas em DAG, os tempos de execução e as taxas de falha.
- Identificar o propósito e os benefícios do Liquid Clustering e do predictive optimization como recursos gerenciados do Delta Lake.

Seção 6: Governança e segurança (8%)

- Compreender o propósito do mascaramento em nível de coluna e da segurança em nível de linha e identificar quando devem ser aplicados.
- Compreender os recursos das políticas ABAC do Unity Catalog e como elas dão suporte ao controle centralizado de acesso a dados.

Questões de exemplo

Estas questões foram retiradas de uma versão anterior do exame. A ideia é mostrar os objetivos conforme declarados no guia do exame e fornecer uma questão de exemplo alinhada a cada objetivo. O guia do exame lista os objetivos que podem ser abordados em uma prova. A melhor maneira de se preparar para um exame de certificação é revisar a estrutura do exame no guia.

Questão 1
Objetivo: identificar gargalos comuns de desempenho, como distorção de dados (data skew), shuffling e disk spilling, interpretando métricas em nível de estágio na Spark UI.

Um engenheiro de dados percebe que a duração de um job batch dobrou após a integração de uma nova fonte de dados. Na Spark UI, o estágio mais longo mostra que a maioria das tasks termina em menos de 30 segundos, mas uma task leva mais de 10 minutos. O resumo de tasks do estágio mostra uma leitura de shuffle mínima/mediana em torno de 400 MB, enquanto a leitura de shuffle máxima ultrapassa 5 GB.

Qual solução reduz o tempo de execução do job?

A. Aumentar o tamanho do cluster para adicionar mais executores, de modo que a task lenta termine mais rápido
B. Confirmar que o adaptive query execution com tratamento de skew join está ativo para dividir automaticamente a partição superdimensionada em tempo de execução
C. Reduzir spark.sql.shuffle.partitions para agrupar mais trabalho em menos tasks
D. Reparticionar manualmente o conjunto de dados usando uma salt key antes do join para distribuir uniformemente as chaves distorcidas

Questão 2
Objetivo: ingerir dados semiestruturados e não estruturados (por exemplo, JSON e dados aninhados) por meio do Lakeflow Connect e de managed connectors em tabelas Delta.

Um engenheiro de dados precisa configurar um managed connector do Lakeflow Connect para ingerir registros JSON de uma fonte SaaS em uma tabela Delta no Unity Catalog. O payload de origem contém um campo de objeto aninhado chamado 'order_details' com subchaves variáveis que mudam entre as execuções de ingestão. O engenheiro quer que o conector expanda automaticamente as novas subchaves como colunas adicionais na tabela Delta de destino, sem exigir atualizações manuais de esquema.

Qual configuração atende ao requisito?

A. Definir o modo de evolução do esquema como 'rescue' nas configurações de ingestão do connector, para que os registros que contêm subchaves não reconhecidas sejam gravados em uma coluna de dados resgatados separada, em vez de expandir o esquema de destino.
B. Habilitar a opção 'mergeSchema' nas TBLPROPERTIES da tabela Delta de destino, para que o connector mescle as mudanças de esquema recebidas no momento da gravação, sem nenhuma configuração no nível do connector.
C. Definir o modo de evolução do esquema como 'failOnNewColumns' nas configurações de ingestão do connector e executar manualmente ALTER TABLE ADD COLUMN após cada execução de ingestão para incorporar as novas subchaves.
D. Definir o modo de evolução do esquema como 'addNewColumns' nas configurações de ingestão do connector, para que as subchaves recém-descobertas no campo JSON aninhado sejam automaticamente promovidas a colunas de nível superior na tabela Delta de destino.

Questão 3
Objetivo: priorizar entre Auto Loader, conectores de parceiros do Lakeflow Connect e outros métodos de ingestão com base em volume, frequência, tipo de dados e necessidades de governança.

Um engenheiro de dados está construindo pipelines downstream para consumir logs de auditoria do Databricks a partir de um bucket S3 de propriedade do cliente. Antes de implementar a inferência de esquema e o checkpointing, ele quer entender o formato de entrega, a latência típica de ingestão e se os arquivos podem ser sobrescritos.

Qual é o comportamento de armazenamento dos logs de auditoria do Databricks?

A. Os arquivos são entregues em JSON, com o registro de eventos ocorrendo tipicamente em menos de 15 minutos após o início da entrega, e novas entregas podem sobrescrever arquivos existentes
B. Os arquivos são entregues em CSV, com garantias de latência inferior a um minuto, e as sobrescritas nunca ocorrem depois que um arquivo é gravado, para preservar a imutabilidade
C. Os arquivos são entregues em Parquet, com consistência eventual além de 24 horas, e as sobrescritas são desabilitadas para simplificar a ingestão em streaming
D. Os arquivos são entregues em JSON, com entrega em uma cadência batch semanal, e as sobrescritas substituem completamente o conteúdo anterior, sem fazer append.

Questão 4
Objetivo: identificar tendências no desempenho dos jobs usando a visão de histórico de execuções do Lakeflow Jobs.

Um engenheiro de dados habilita o gráfico de tendência de duração na visão de histórico de execuções do Lakeflow Jobs para um job que é executado todas as noites.

Como a visão de histórico de execuções se comporta como resultado da habilitação desse gráfico?

A. A visão define um limite de alerta de referência usando a duração média de execução nas execuções históricas.
B. A visão aplica um indicador visual às linhas em que a duração da execução ultrapassa a média.
C. A visão exibe um gráfico de linhas que traça a duração de cada execução recente ao longo do tempo.
D. A visão gera uma projeção de custo com base na duração média de execução e na taxa de DBU do cluster.

Questão 5
Objetivo: gerenciar código na interface do Databricks workspace usando Git Folders: criar/alternar branches, fazer commit, push e abrir pull requests.

Uma equipe deseja uma forma modular de implantar, versionar e orquestrar pipelines de ETL no Databricks, viabilizando CI/CD e repetibilidade.

Qual recurso atende a esse requisito?

A. Usar modelos no Unity Catalog para representar os jobs de ETL, em que cada modelo armazena o artefato de código do pipeline e o CI/CD promove versões atualizando os aliases de modelo vinculados às tarefas do Job.
B. Empacotar a lógica de transformação como bibliotecas wheel armazenadas em Unity Catalog Volumes e vinculá-las às tasks de Jobs para garantir uma implantação determinística entre os ambientes.
C. Empacotar a lógica da API dentro de um notebook montado em Volume e usar a Jobs API v2 para acionar o notebook, dependendo do histórico de revisões do notebook para funcionar como um sistema de versionamento.
D. Usar DABs para definir recursos e assets de código, versioná-los no Git e promover implantações entre ambientes por meio de ações automatizadas de CI/CD.

Respostas:

1. B
2. D
3. A
4. C
5. D