databricks

Guia do exame da Databricks

Databricks Certified Data Engineer Professional

Fornecer feedback sobre o guia do exame

Finalidade deste guia do exame

O objetivo deste guia é fornecer uma visão geral do exame e de sua abordagem para ajudar você a determinar a preparação necessária para realizá-lo. Este documento será atualizado sempre que houver alterações em um exame (e quando essas alterações entrarem em vigor) para que você possa se preparar. **Esta versão abrange a versão atualmente ativa em 1 de março de 2025. Verifique novamente duas semanas antes de fazer o exame para ter certeza de que você tem a versão mais recente.**

Descrição do Público

O exame de certificação Databricks Certified Data Engineer Professional avalia a capacidade de um indivíduo de usar o Databricks para executar tarefas avançadas de engenharia de dados. Ele inclui uma compreensão da plataforma Databricks e de ferramentas de desenvolvimento como Apache Spark, Delta Lake, MLflow e a Databricks CLI e REST API. Também avalia a capacidade de construir pipelines de ETL otimizados e limpos. Além disso, também será avaliada a modelagem de dados em um Lakehouse usando o conhecimento de conceitos gerais de modelagem de dados. Por fim, este exame também incluirá a avaliação da capacidade de garantir que os pipelines de dados sejam seguros, confiáveis, monitorados e testados antes da implantação. Espera-se que os indivíduos aprovados neste exame de certificação concluam tarefas avançadas de engenharia de dados usando Databricks e suas ferramentas associadas.

Sobre a prova

- Número de itens: 60 perguntas de múltipla escolha pontuadas
- Limite de tempo: 120 minutos
- Taxa de inscrição: US$ 200, mais impostos aplicáveis, conforme exigido pela legislação local
- Método de entrega: supervisionado on-line
- Materiais de consulta: nenhum é permitido
- Pré-requisito: nenhum exigido; participação no curso e 1 ano de experiência prática em Databricks são recomendados
- Validade: 2 anos
- Recertificação: A recertificação é necessária a cada dois anos para manter seu estado de certificado. Para recertificar, você deve fazer o exame completo que está atualmente

disponível. Revise a seção “Preparando-se para o exame” na página do exame para se preparar para fazer o exame novamente.
- Conteúdo sem pontuação: os exames podem incluir itens não pontuados para coletar informações estatísticas para uso futuro. Esses itens não são identificados no formulário e não afetam sua pontuação, e é considerado um tempo adicional para esse conteúdo.

Treinamento recomendado

- Individual (disponível na Databricks Academy):
  - Databricks Streaming e Delta Live Tables - Delta Dawn
  - Databricks Data Privacy
  - Databricks Performance Optimization Delta Dawn
  - Teste e Implantação Automatizados com Databricks Asset Bundle

Visão geral do exame

Seção 1: Databricks Tooling

- Explicar como Delta Lake usa o log de transações e o armazenamento de objetos em nuvem para garantir atomicidade e durabilidade
- Descrever como o controle de simultaneidade otimista do Delta Lake fornece isolamento e quais transações podem entrar em conflito
- Descrever a funcionalidade básica do Delta clone.
- Aplicar otimizações comuns de indexação do Delta Lake, incluindo particionamento, zorder, bloom filters e tamanhos de arquivo
- Implementar tabelas Delta otimizadas para o serviço do Databricks SQL
- Comparar diferentes estratégias para particionar dados (por exemplo, identificar colunas de particionamento adequadas para uso)

Seção 2: Processamento de dados (processamento em lote, processamento incremental e otimização)

- Descrever e distinguir dicas de partição: combinar, reparticionar, reparticionar por intervalo e rebalancear
- Comparar diferentes estratégias para particionar dados (por exemplo, identificar colunas de particionamento adequadas para uso)
- Articular como gravar Pyspark dataframes em disco enquanto controla manualmente o tamanho de arquivos parciais individuais
- Articular múltiplas estratégias para atualizar registros 1+ em uma tabela spark (Tipo 1)
- Implementar padrões de design comuns desbloqueados por Structured Streaming e Delta Lake
- Explorar e ajustar informações de estado usando stream-static joins e Delta Lake
- Implementar stream-static joins
- Implementar a lógica necessária para desduplicação usando Spark Structured Streaming
- Habilitar o CDF nas tabelas Delta Lake e redesenhar as etapas de processamento de dados para processar a saída do CDC, em vez do feed incremental da leitura em Structured Streaming
- Aproveitar o CDF para propagar exclusões facilmente

- Demonstrar como o particionamento adequado de dados permite o simples arquivamento ou exclusão de dados
- Articular como “pequenos” (tiny files, scanning overhead, over partitioning, etc) induzem problemas de desempenho nas queries do Spark

Seção 3: Modelagem de dados
- Descrever o objetivo das transformações de dados durante a promoção de bronze para prata
- Discutir como Change Data Feed (CDF) aborda dificuldades passadas na propagação de atualizações e exclusões na arquitetura Lakehouse
- Aplicar o Delta Lake clone para saber como o clone superficial (shallow) e profundo interagem com as tabelas de origem/destino
- Desenvolver uma tabela multiplex bronze para evitar armadilhas comuns ao tentar produzir cargas de trabalho em transmissão
- Implementar práticas recomendadas para dados em transmissão de tabelas multiplex bronze
- Aplicar processamento incremental, aplicação de qualidade e desduplicação para processar dados de bronze a prata
- Tomar decisões informadas sobre como impor a qualidade dos dados com base nos pontos fortes e nas limitações de várias abordagens no Delta Lake
- Implementar tabelas evitando problemas causados pela falta de restrições de key estrangeira
- Adicionar restrições às tabelas do Delta Lake para evitar a gravação de dados incorretos
- Implementar tabelas de lookup e descrever as desvantagens dos modelos de dados normalizados
- Diagramar arquiteturas e operações necessárias para implementar várias tabelas de dimensões que mudam lentamente usando Delta Lake com cargas de trabalho para transmissão e lote
- Implementar tabelas SCD Tipos 0, 1 e 2

Seção 4: Segurança e Governança
- Criar views dinâmicas para realizar o mascaramento de dados
- Usar views dinâmicas para controlar o acesso a linhas e colunas

Seção 5: Monitoramento e Registro em log
- Descrever os elementos da Spark UI para auxiliar na análise de desempenho, depuração de aplicativos e ajuste de aplicações Spark
- Inspecionar cronogramas e métricas de eventos para estágios e jobs realizados em um cluster
- Tirar conclusões das informações apresentadas na Spark UI, na Ganglia UI e na Cluster UI para avaliar problemas de desempenho e depurar aplicações com falha
- Desenvolver sistemas que controlem SLAs de custo e latência para production streaming jobs
- Implantar e monitorar jobs de transmissão e lote

Seção 6: Testes e Implantação
- Adaptar um padrão de dependência de notebook para usar dependências de arquivo Python

- Adaptar o código Python mantido como Wheels para direcionar importações usando caminhos relativos
- Reparar e executar novamente jobs com falha
- Criar jobs com base em casos de uso e padrões comuns
- Criar jobs multi-task com dependências múltiplas
- Desenvolver sistemas que controlem SLAs de custo e latência para streaming jobs de produção
- Configurar o Databricks CLI e executar os comandos básicos para interagir com o espaço de trabalho e os clusters
- Executar comandos do CLI para implantar e monitorar Databricks jobs
- Usar a REST API para clonar um job, trigger uma execução e exportar a saída da execução

Exemplos de perguntas

Estas perguntas foram retiradas de uma versão anterior da prova. O propósito é mostrar os objetivos, como estão indicados no guia do exame, e oferecer um exemplo de pergunta que se alinhe ao 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 o esboço dele no guia do exame.

Pergunta 1
Objetivo: Identificar os resultados da execução de um comando em uma tabela do Delta Lake criada com uma query

Uma tabela do Delta Lake foi criada com a query:

```
CREATE TABLE dev.my_table
USING DELTA
LOCATION "/mnt/dev/my_table"
```

Percebendo que a tabela precisa ser utilizada por outros e que seu nome está incorreto, o código abaixo foi executado:

```
ALTER TABLE dev.my_table RENAME TO dev.our_table
```

Qual resultado ocorrerá após executar o segundo comando?

A. A alteração do nome da tabela é registrada no log de transações do Delta.
B. A referência da tabela no metastore é atualizada e todos os arquivos de dados são movidos.
C. A referência da tabela no metastore é atualizada e nenhum dado é alterado.
D. Um novo log de transações do Delta é criado para a tabela renomeada.
E. Todos os arquivos e metadados relacionados são descartados e recriados em uma única transação ACID.

Pergunta 2
Objetivo: Desduplicar dados em registros processados anteriormente à medida que são inseridos na tabela Delta.

Um engenheiro de dados está desenvolvendo um fluxo de trabalho de ETL que pode detectar registros duplicados que chegam atrasados de sua única fonte. O engenheiro de dados sabe que pode desduplicar os registros em lote, mas está procurando outra solução.

Qual abordagem permite ao engenheiro de dados desduplicar dados em registros processados anteriormente à medida que são inseridos em uma tabela Delta?

A. **`VACUUM`** a tabela Delta após a conclusão de cada lote.
B. Confiar no Delta Lake schema enforcement para evitar registros duplicados.
C. Definir a configuração **`delta.deduplicate = true`**.
D. Executar um full outer join em um key exclusivo e substituir os dados existentes.
E. Executar um insert-only merge com uma condição correspondente em um key exclusivo.

Pergunta 3
Objetivo: Identificar como configurar todas as tabelas no Lakehouse como tabelas externas e não gerenciadas do Delta Lake

O arquiteto de dados determinou que todas as tabelas no Lakehouse sejam configuradas como tabelas externas e não gerenciadas do Delta Lake.

Qual abordagem garantirá que esse requisito seja atendido?

A. Sempre que uma tabela for criada, certificar-se de que a palavra-chave **`LOCATION`** seja usada.
B. Sempre que uma tabela for criada, certificar-se de que a palavra-chave **`EXTERNAL`** seja usada na instrução **`CREATE TABLE`**.
C. Quando o espaço de trabalho estiver sendo configurado, certificar-se de que o armazenamento externo de objetos em nuvem tenha sido montado.
D. Sempre que um banco de dados for criado, certificar-se de que a palavra-chave **`LOCATION`** seja usada.
E. Sempre que uma tabela for criada, certificar-se de que as palavras-chaves **`LOCATION`** e **`UNMANAGED`** sejam usadas.

Pergunta 4
Objetivo: Descrever os controles de permissão para Databricks jobs

Uma equipe de engenharia de dados está tentando transferir a propriedade de seus Databricks

Workflows de um indivíduo que trocou de equipe. No entanto, eles não têm certeza de como funcionam os controles de permissão especificamente para Databricks Jobs work.

Qual instrução descreve corretamente os controles de permissão para Databricks jobs?

A. O criador de um Databricks job sempre terá privilégios de "Proprietário"; essa configuração não pode ser alterada.
B. As Databricks jobs devem ter exatamente um proprietário; os privilégios de "Proprietário" não podem ser atribuídos a um grupo.
C. Além do grupo default "admins", apenas usuários individuais podem receber privilégios em Jobs.
D. Apenas administradores do espaço de trabalho podem garantir privilégios de "Proprietário" a um grupo.
E. Um usuário só poderá transferir a propriedade do job para um grupo se também for membro desse grupo.

Pergunta 5
Objetivo: Identificar métodos para instalar pacotes python...
Um engenheiro de dados precisa usar um pacote Python para processar dados. Como resultado, ele precisa instalar o pacote Python em todos os nós do cluster atualmente ativo.

O que descreve um método de instalação de um pacote Python com escopo no nível do notebook para todos os nós no cluster atualmente ativo?

A. Usar **`%pip install`** em uma célula de notebook
B. Usar **`%sh pip install`** em uma célula de notebook
C. Executar **`source env/bin/activate`** um script de configuração em um notebook
D. Instalar bibliotecas de PyPI usando a IU do cluster
E. Usar **`b`** em uma célula de notebook

Respostas
Pergunta 1: C
Pergunta 2: E
Pergunta 3: A
Pergunta 4: B
Pergunta 5: A