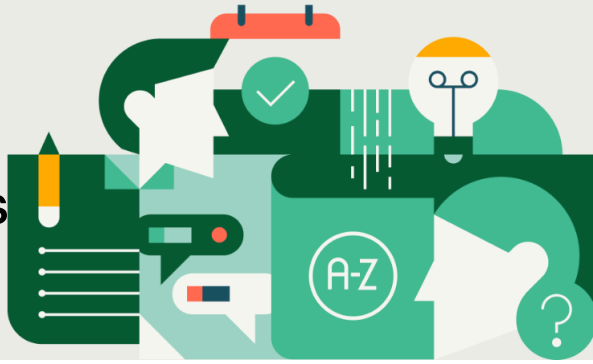
databricks

Databricks Guia Do Exame

Databricks Engenheiro de Dados Certificado Profissional



Fornecer feedback sobre o guia do exame

Finalidade deste Guia do Exame

O objetivo deste guia é fornecer uma visão geral dos tópicos cobertos e ajudá-lo a avaliar seu nível de preparação. Este documento será atualizado sempre que houver alterações em um exame (e quando essas alterações entrarem em vigor) para que você possa estar preparado. **Esta versão cobre o exame atualmente ativo a partir de 30 de setembro de 2025. Volte duas semanas antes de fazer o exame para ter certeza de que você tem a versão mais atual.**

Descrição do público

O exame Databricks Certified Data Engineering Professional valida as habilidades avançadas de um candidato na criação, otimização e manutenção de soluções de engenharia de dados de nível de produção na Plataforma Databricks Lakehouse. Os candidatos bem-sucedidos demonstram experiência nos principais recursos da plataforma, como Delta Lake, Unity Catalog, Auto Loader, Spark Declarative Pipelines, Databricks Compute (incluindo Serverless), Lakeflow Jobs e Medallion Architecture. Esta certificação avalia a capacidade de projetar pipelines ETL seguros, confiáveis e econômicos, processar dados complexos de diversas fontes usando Python e SQL e aplicar práticas recomendadas em gerenciamento de esquema, observabilidade, governança e otimização de desempenho. Os candidatos também são testados na implementação de cargas de trabalho streaming, orquestração de Workflows, uso de DevOps & CI/CD e implantação com ferramentas como Databricks CLI, REST API e Asset Bundles. Os profissionais que obtêm essa certificação comprovadamente têm o conhecimento e a experiência prática necessários para entregar soluções de engenharia de dados prontas para produção no Databricks. Recomenda-se fortemente um ou mais anos de experiência na plataforma Lakehouse.

Sobre o Exame

- Número de itens: 59 questões pontuadas de múltipla escolha
- Tempo limite: 120 minutos
- Taxa de inscrição: USD 200, mais impostos aplicáveis, conforme exigido pela legislação local
- Método de entrega: online com supervisão remota
- Auxiliares de teste: nenhum permitido.
- Pré-requisito: Nenhum exigido; participação no curso e 1 ano de experiência prática no Databricks é altamente recomendado
- Validade: 2 anos

- Recertificação: A recertificação é necessária a cada dois anos para manter seu status de certificado. Para se recertificar, é preciso realizar o exame completo atualmente disponível. Consulte a seção "Preparando-se para o exame" na página do exame para se preparar para fazê-lo 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 são considerados um tempo adicional para esse conteúdo.

Treinamento recomendado

- Instrutor liderou Advanced Data Engineering With Databricks
- Auto-cadenciado (disponível em Databricks Academy):
    - Databricks Streaming e Spark Declarative Pipelines
    - Databricks Data Privacy
    - Databricks Performance Optimization
    - Implantação automatizada com Databricks Asset Bundle

Resumo do exame

Seção 1: Desenvolvendo código para processamento de dados usando Python e SQL

- Usando Python e ferramentas para desenvolvimento
    - Projetar e implementar uma estrutura de projeto Python escalável otimizada para Databricks Asset Bundles (DABs), permitindo desenvolvimento modular, automação de implantação e integração CI/CD.
    - Gerencie e solucione problemas de instalações e dependências de bibliotecas externas de terceiros no Databricks, incluindo pacotes PyPI, rodas locais e arquivos de origem.
    - Desenvolva funções definidas pelo usuário (UDFs) usando Pandas/Python UDF.
- Criando e testando um ETL pipeline com Spark Declarative Pipelines, SQL e Apache Spark na plataforma Databricks.
    - Crie e gerencie pipelines de dados confiáveis e prontos para produção para dados batch e streaming usando Spark Declarative Pipelines e Autoloader.
    - Crie e automatize cargas de trabalho ETL usando Jobs via UI/APIs/CLI.
    - Explicar as vantagens e desvantagens das tabelas streaming em comparação com as views materializadas.
    - Use as APIs APPLY CHANGES para simplificar CDC em Spark Declarative Pipelines.
    - Compare Spark Structured Streaming e Spark Declarative Pipelines para determinar a abordagem ideal para a criação de pipelines ETL escaláveis.
    - Crie um componente pipeline que use operadores de fluxo de controle (por exemplo, if/else, foreach, etc.)
    - Escolha as configurações apropriadas para ambientes e dependências, memória alta para tarefas Notebook e otimização automática para não permitir tentativas.
    - Desenvolva testes de unidade e integração usando *assertDataFrameEqual*, *assertSchemaEqual*, *DataFrame.transform* e estruturas de teste para garantir a correção do código, incluindo um depurador interno.

Seção 2: Ingestão e aquisição de dados:

- Projete e implemente pipelines de ingestão de dados para ingerir com eficiência uma variedade de formatos de dados, incluindo Delta Lake, Parquet, ORC, AVRO, JSON, CSV, XML, TEXT e BINARY de diversas fontes, como barramentos de mensagens e armazenamento em nuvem.
- Crie um pipeline de dados somente *append-only* capaz de lidar com dados de lotes e streaming usando Delta.

Seção 3: Transformação, limpeza e qualidade de dados

- Escreva código Spark SQL e PySpark eficientes para aplicar transformações de dados avançadas, incluindo funções de janela, junções e agregações, para manipular e analisar grandes datasets.
- Desenvolva um processo de quarentena para dados incorretos com Spark Declarative Pipelines ou Autoloader em jobs clássicos.

Seção 4: Compartilhamento de Dados e Federação

- Demonstre Delta Sharing com segurança entre implantações Databricks usando Databricks Sharing (D2D) ou para plataformas externas usando protocolo de compartilhamento aberto (D2O).
- Configure Lakehouse Federation com governança adequada em todos os sistemas de origem suportados.
- Use Delta Share para compartilhar dados em tempo real do Lakehouse para qualquer plataforma de computação.

Seção 5: Monitoramento e alertas

- Monitoramento
    - Use System Tables para observabilidade sobre utilização de recursos, custo, auditoria e carga de trabalho de monitoramento.
    - Use Query Profiler UI e Spark UI para monitorar cargas de trabalho.
    - Use Databricks REST APIs/Databricks CLI para monitoramento de jobs e pipelines.
    - Use Spark Declarative Pipelines Event Logs para monitorar pipelines.
- Alertas
    - Use Alertas SQL para monitorar a qualidade dos dados.
    - Use a interface de usuário Lakeflows e Jobs API para configurar notificações de problemas de desempenho e status do Job.

Seção 6:Otimização de Custo e Desempenho

- Entenda como e por que o uso de tabelas gerenciadas pelo Unity Catalog reduz a sobrecarga operacional e a carga de manutenção.
- Entenda as técnicas de otimização de Delta, como vetores de exclusão e Liquid Clustering.

- Entenda as técnicas de otimização usadas pelo Databricks para garantir o desempenho de consultas em grandes conjuntos de dados (data skipping, remoção de arquivos, etc.).
- Aplique Change Data Feed (CDF) para resolver limitações específicas de tabelas streaming e melhorar a latência.
- Use o query profile para analisar consultas e identificar gargalos, como omissão de dados incorretos, tipos ineficientes de joins e shuffle de dados.

Seção 7: Garantindo a segurança dos dados e conformidade

- Aplicação de mecanismos de segurança de dados.
    - Utilize ACLs para proteger objetos do Workspace seguindo o princípio do privilégio mínimo e aplicando políticas de controle de acesso.
    - Use filtros de linha e máscaras de coluna para filtrar e mascarar dados confidenciais de tabela.
    - Aplique métodos de anonimização e pseudonimização, como Hashing, Tokenização, Supressão e Generalização, a dados **confidenciais.**
- Garantindo conformidade
    - Implemente um pipeline de lotes e streaming compatível que detecta e aplica o mascaramento de PII para garantir a privacidade dos dados.
    - Desenvolva uma solução de limpeza de dados garantindo conformidade com políticas de retenção de dados.

Seção 8: Governança de dados

- Crie e adicione descrições/metadados sobre dados corporativos para torná-los mais detectáveis.
- Demonstrar compreensão do modelo de herança de permissão do Unity Catalog.

Seção 9: Depuração e implantação

- Depuração e solução de problemas
    - Identifique informações de diagnóstico pertinentes usando Spark UI, logs de cluster, tabelas do sistema e perfis de consultas para solucionar erros.
    - Analise os erros e corrija as execuções de jobs com falha com reparos de Jobs e substituições de parâmetros.
    - Use logs de eventos do Spark Declarative Pipelines e do Spark UI para depurar Spark Declarative Pipelines e Spark Pipelines.
- Implantando CI/CD
    - Crie e implante recursos Databricks usando Databricks Asset Bundles.
    - Configure e integre com fluxos de Job CI/CD baseados em Git usando Pastas Git Databricks para notebooks e implantação de código.

Seção 10: Modelagem de dados

- Projete e implemente modelos de dados escalonáveis usando Delta Lake para gerenciar grandes conjuntos de dados.
- Simplifique as decisões de layout de dados e otimize o desempenho da query usando Liquid Clustering.
- Identifique os benefícios de usar Liquid Clustering sobre particionamento e Z-Order.
- Projetar modelos dimensionais para cargas de trabalho analíticas, garantindo consultas e agregação eficientes.

Exemplos de perguntas

Essas perguntas foram retiradas de uma versão anterior do exame. O objetivo é mostrar como os tópicos estão declarados no guia do exame e fornecer um exemplo de pergunta que se alinhe com cada um. O guia lista os objetivos que podem ser abordados. A melhor maneira de se preparar para a certificação é revisar o resumo apresentado no guia.

Pergunta 1

Objetivo: Entender as operações de metastore do catálogo do Delta Lake e o comportamento do ACID compliance.

Uma tabela Delta Lake foi criada com a consulta:

```
CREATE TABLE prod.sales_by_stor
USING DELTA
LOCATION "/mnt/prod/sales_by_store"
```

Percebendo que a consulta original tinha um erro tipográfico, o código abaixo foi executado:

```
ALTER TABLE prod.sales_by_stor RENAME TO prod.sales_by_store
```

Qual resultado ocorrerá após a execução do segundo comando?

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

Pergunta 2

Objetivo: Entender o comportamento do Spark Structured Streaming e determinar a abordagem ideal para pipelines prontos para SLA de produção.

Um Job Structured Streaming implantado na produção vem sofrendo atrasos durante os horários de pico do dia. Atualmente, durante a execução normal, cada microlote de dados é processado em menos de 3 segundos. Durante as horas de pico do dia, o tempo de execução de cada microlote torna-se muito inconsistente, às vezes excedendo 30 segundos. A escrita streaming está atualmente configurada com um intervalo de disparo de 10 segundos.

Mantendo todas as outras variáveis constantes e supondo que os registros precisam ser processados em menos de 10 segundos, qual ajuste atenderá ao requisito?

A. Use a opção *trigger once* e configure um Job Databricks para executar a query a cada 8 segundos; isso garante que todos os registros em atraso sejam processados com cada lote.
B. Diminua o intervalo de disparo para 5 segundos; acionar lotes com mais frequência

pode impedir que registros acumulem e que lotes grandes causem vazamento.

C. Diminua o intervalo de disparo para 5 segundos; o acionamento de lotes com mais frequência permite que os executores parados comecem a processar o próximo lote enquanto tarefas em execução mais longas de lotes anteriores são concluídas.

D. O intervalo de disparo não pode ser modificado sem modificar o diretório do ponto de verificação; para manter o estado de transmissão atual, aumente o número de partições de shuffle para maximizar o paralelismo.

Pergunta 3

Objetivo: Aplicar métodos de anonimização e pseudonimização, como Hashing, Tokenização, Supressão e generalização para dados confidenciais

A equipe de engenharia de dados está migrando um sistema corporativo com milhares de tabelas e views para o Lakehouse. Eles planejam implementar a arquitetura de destino usando uma série de tabelas Bronze, Silver e Gold. As tabelas Bronze serão usadas quase exclusivamente para cargas de trabalho de engenharia de dados de produção, enquanto as tabelas Silver serão usadas para dar suporte a cargas de trabalho de engenharia de dados e aprendizado de máquina. As tabelas Gold servirão em grande parte para fins de Business Intelligence e relatórios. Embora as informações de identificação pessoal (PIIs) existam em todos os níveis de dados, as regras de pseudonimização e anonimização estão em vigor para todos os dados nos níveis Silver e Gold.

A organização está interessada em reduzir as preocupações com a segurança e, ao mesmo tempo, maximizar a capacidade de colaboração entre equipes diversas.

Qual declaração exemplifica as melhores práticas para a implementação desse sistema?

A. O isolamento de tabelas em bancos de dados separados com base em níveis de qualidade de dados permite o gerenciamento fácil de permissões por meio de ACLs de banco de dados e permite a separação física de locais de armazenamento default para tabelas gerenciadas.

B. O armazenamento de todas as tabelas de produção em um único banco de dados fornece uma view unificada de todos os ativos de dados disponíveis em todo o Lakehouse, simplificando a capacidade de descoberta, concedendo a todos os usuários privilégios de view nesse banco de dados.

C. Como os bancos de dados no Databricks são meramente uma construção lógica, as escolhas em torno da organização do banco de dados não afetam a segurança ou a capacidade de descoberta no Lakehouse.

D. Trabalhar no banco de dados default do Databricks fornece a maior segurança ao trabalhar com tabelas gerenciadas, pois elas serão criadas no DBFS root.

Pergunta 4

Objetivo: Projetar e implementar modelos de dados escalonáveis usando Delta Lake para gerenciar grandes conjuntos de dados.

Uma tabela Delta Lake que representa metadados sobre postagens de conteúdo de usuários tem o seguinte `esquema`:

```
user_id LONG, post_text STRING, post_id STRING, longitude FLOAT,
latitude FLOAT, post_time TIMESTAMP, date DATE
```

Com base no esquema acima, qual coluna é uma boa candidata para particionar a tabela Delta?

A. post_id
B. post_time
C. date
D. user_id

Pergunta 5
Objetivo: Demonstrar compreensão do modelo de herança de permissão Unity Catalog

Uma tabela chamada *user_ltv* está sendo usada para criar uma view que será usada por analistas de dados em várias equipes. Os usuários no Workspace são configurados em grupos, que são usados para configurar o acesso a dados usando ACLs.

A **tabela user_ltv** tem o seguinte esquema:

```
email STRING, age INT, ltv INT
```

A seguinte definição de view é executada:

```
CREATE VIEW email_ltv AS
SELECT
CASE WHEN
  is_member('marketing') THEN email
  ELSE 'REDACTED'
END AS email,
ltv
FROM user_ltv
```

Um analista que não é membro do grupo de marketing executa a seguinte consulta:

```
SELECT * FROM email_ltv
```

Qual será o resultado dessa consulta?

A. Apenas as **colunas email** e **ltv** serão retornadas; a coluna email conterá a string "**REDACTED**" em cada linha.
B. Três colunas serão retornadas, mas uma coluna será nomeada "**REDACTED**" e conterá apenas valores nulos.
C. Somente as **colunas email** e **ltv** serão retornadas; a coluna email conterá todos os valores nulos.
D. As **colunas email** e **ltv** serão retornadas com os valores em **user_ltv**.

Pergunta 6:
Objetivo: *Escolha as configurações apropriadas para ambientes e dependências, alta memória para tarefas de Notebooks e otimização automática para impedir retentativas.*

A equipe de relatórios de negócios exige que os dados de seus painéis sejam atualizados a cada hora. O tempo total de processamento do pipeline que extrai, transforma e carrega os dados do pipeline leva 10 minutos.

Supondo condições normais de operação, qual configuração atenderá aos requisitos do contrato de nível de serviço com o menor custo?

A. Agende um Job para executar o pipeline uma vez por hora em um cluster interativo dedicado.
B. Agende um Job para executar o pipeline uma vez por hora em um novo job cluster.
C. Agende um Job Structured Streaming com um intervalo de trigger de 60 minutos.
D. Configure um Job que execute toda vez que novos dados cheguarem em um determinado diretório.

Pergunta 7:
Objetivo – *Entender o ambiente de desenvolvimento de Notebooks, o gerenciamento de variáveis e a criação de código seguro e configurável.*

A equipe de segurança está explorando se o módulo de segredos Databricks pode ou não ser usado para conexão com um banco de dados externo.

Depois de testar o código com todas as variáveis Python sendo definidas com strings, eles carregam a senha para o módulo secrets e configuram as permissões corretas para o usuário ativo no momento. Em seguida, eles modificam seu código para o seguinte (deixando todas as outras variáveis inalteradas).

```
password = dbutils.secrets.get(scope="db_creds", key="jdbc_password")

print(password)

df = (spark
  .read
  .format("jdbc")
  .option("url", connection)
  .option("dbtable", tablename)
  .option("user", username)
  .option("password", password)
  )
```

Qual instrução descreve o que acontecerá quando o código acima for executado?

A. A conexão com a tabela externa será bem-sucedida; a cadeia de caracteres **"REDACTED"** será impressa.
B. A conexão com a tabela externa será bem-sucedida; o valor da cadeia de caracteres da **senha** será impresso em texto sem formatação.
C. Uma caixa de entrada interativa aparecerá no Notebook; se a senha correta for fornecida, a conexão será bem-sucedida e a senha será impressa em texto simples.
D. Uma caixa de entrada interativa aparecerá no Notebook; se a senha correta for fornecida, a conexão será bem-sucedida e a senha codificada será salva no DBFS.

Pergunta 8:
Objetivo: Compreender as técnicas de otimização usadas pelo Databricks para garantir o desempenho de consultas em grandes conjuntos de dados (data skipping, remoção de arquivos, etc.).

Uma tarefa de ingestão de dados requer que um dataset JSON de 1 TB seja escrito em Parquet com um tamanho de arquivo parcial de destino de 512 MB. Como Parquet está sendo usado em vez de Delta Lake, os recursos internos de dimensionamento de arquivo, como Auto-Optimize & Auto-Compaction, não podem ser usados.

Qual estratégia produzirá o melhor desempenho sem shuffle de dados?

A. Ingerir os dados, executar as transformações narrow, reparticionar para 2.048 partições (1 TB*1024*1024/512) e, em seguida, escrever em Parquet.
B. Defina **spark.sql.adaptive.advisoryPartitionSizeInBytes** como 512 MB bytes, ingira os dados, execute as transformações estreitas, consolide em 2.048 partições (1TB*1024*1024/512) e, em seguida, escreva em Parquet.
C. Defina **spark.sql.files.maxPartitionBytes** como 512 MB, faça a ingestão de dados, execute as transformações narrow e escreva em Parquet.
D. Defina **spark.sql.shuffle.partitions** para 2.048 partições (1TB*1024*1024/512), faça a ingestão de dados, execute as transformações narrow, otimize os dados ordenando-os (que reparticiona automaticamente os dados) e, em seguida, escreva em Parquet.

Pergunta 9:
Objetivo: Aplicar o clone do Delta Lake para saber como os clones superficiais e profundos interagem com as tabelas de origem/destino.

A equipe de marketing está procurando compartilhar dados em uma tabela agregada com a organização de vendas, mas os nomes de campo usados pelas equipes não correspondem e vários campos específicos de marketing não foram aprovados para a organização de vendas.

Qual solução aborda a situação enfatizando a simplicidade?

A. Criar uma view na tabela marketing selecionando apenas os campos aprovados para a equipe de vendas; dar alias aos nomes de quaisquer campos que devem ser padronizados para as convenções de nomenclatura de vendas.
B. Crie uma nova tabela com o esquema necessário e use a funcionalidade DEEP CLONE do Delta Lake para sincronizar as alterações confirmadas de uma tabela para a tabela correspondente.
C. Use uma instrução CTAS para criar uma tabela derivada da tabela marketing; configure um Job de produção para propagar alterações.
D. Adicione uma escrita de tabela paralela ao pipeline de produção atual, atualizando uma nova tabela de vendas que varia conforme necessário da tabela de marketing.

Pergunta 10:
Objetivo: Criar um Job multitarefa com várias dependências

Um Job Databricks foi configurado com três tarefas, cada uma das quais é um Databricks Notebook. A tarefa A não depende de outras tarefas. As tarefas B e C são executadas em paralelo, com cada uma tendo uma dependência serial da tarefa A.

Qual será o estado resultante se as tarefas A e B forem concluídas com êxito, mas a tarefa C falhar durante uma execução agendada?

A. Toda a lógica expressa no Notebook associada às tarefas A e B terá sido concluída com sucesso; algumas operações na tarefa C podem ter sido concluídas com êxito.
B. A menos que todas as tarefas sejam concluídas com êxito, nenhuma alteração será confirmada no Lakehouse, como a tarefa C falhou, todas as confirmações serão revertidas automaticamente.
C. Toda a lógica expressa no Notebook associada às tarefas A e B terá sido concluída com sucesso; quaisquer alterações feitas na tarefa C serão revertidas devido à falha na tarefa.
D. Como todas as tarefas são gerenciadas como um gráfico de dependência, nenhuma alteração será confirmada no Lakehouse até que todas as tarefas tenham sido concluídas com êxito.

Respostas
Pergunta 1: D
Pergunta 2: B
Pergunta 3: A
Pergunta 4: C
Pergunta 5: A
Pergunta 6: B
Pergunta 7: A
Pergunta 8: C
Pergunta 9: A
Pergunta 10:A