databricks

Databricks 시험 가이드

Databricks 인증
Data Engineer Associate

시험 가이드 피드백 제공

이 시험 가이드의 목적

이 시험 가이드는 시험의 개요와 출제 범위를 제공하여 응시 준비 상태를 판단하는 데 도움을 주는 것을 목적으로 합니다. 이 문서는 시험에 변경 사항이 있을 때마다(그리고 해당 변경 사항이 적용되는 시점에) 업데이트되므로 미리 대비할 수 있습니다. 본 문서는 2026년 5월 4일 기준 시험 버전을 다룹니다.

대상 응시자 설명

Databricks Certified Data Engineer Associate 인증 시험은 Databricks Data Intelligence Platform을 활용하여 기초적인 데이터 엔지니어링 작업을 수행하는 능력을 평가합니다. 이 시험은 Data Intelligence Platform과 그 workspace, 아키텍처 및 기능에 대한 지식, 그리고 데이터 수집, 데이터 로딩, 데이터 변환 및 모델링과 관련된 작업(예: PySpark를 사용한 Extract, Transform, Load(ETL) 작업 수행, Lakeflow Jobs 및 CI/CD 활용)에 대한 응시자의 지식을 평가합니다. 마지막으로, 이 시험은 문제 해결, 모니터링, 최적화 기법에 대한 이해와 Databricks Platform 내에서 거버넌스와 보안을 구현하는 방법에 대한 응시자의 지식을 평가합니다.

시험 정보

- 채점 문항 수: 채점되는 객관식 문항 45개.
- 제한 시간: 90분.
- 응시료: 미화 200달러(USD 200), 현지 법률에 따라 부과되는 세금 별도.
- 응시 방식: 온라인 또는 시험 센터
- 시험 보조 도구: 허용되지 않음.
- 사전 요건: 필수 요건 없음. 다만 과정 수강과 Databricks 실무 경험 6개월을 적극 권장합니다.
- 유효 기간: 2년.
- 재인증: 인증을 유지하려면 2년마다 재인증이 필요합니다. 재인증을 위해서는 현재 진행 중인 실제 시험에 응시해야 합니다. 시험에 다시 응시할 준비를 하려면 시험 웹페이지의 “Getting Ready for the Exam” 섹션을 검토하시기 바랍니다.
- 채점되지 않는 콘텐츠: 시험에는 향후 활용을 위한 통계 정보를 수집하기 위해 채점되지 않는 문항이 포함될 수 있습니다. 이러한 문항은 시험지에 별도로 표시되지 않으며 점수에 영향을 주지 않습니다. 이 콘텐츠를 위한 추가 시간이 반영되어 있습니다.

권장 교육

- 강사 주도(Instructor-led): Data Engineering with Databricks
- 자기 주도(Self-paced, Databricks Academy에서 제공):
    - 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

시험 개요

섹션 **1: Databricks Intelligence Platform (6%)**

- Databricks Data Intelligence Platform의 핵심 구성 요소(예: 아키텍처, Delta Lake, Unity Catalog)를 이해합니다.
- Databricks Data Intelligence Platform의 compute 서비스에 대해 그 특성, 제한 사항, 비용 모델을 포함하여 이해하고, 각 워크로드 사용 사례에 가장 적합한 옵션을 선택합니다.

섹션 **2:** 데이터 수집 및 로딩 **(21%)**

- 배치, 스트리밍, 증분 로딩을 포함한 데이터 수집 패턴을 구현하고 그 세부 사항을 파악하며, 로컬 파일, Lakeflow Connect standard connector, Lakeflow Connect managed connector와 같은 소스에서 데이터를 가져옵니다.
- COPY INTO 명령을 사용하여 클라우드 오브젝트 스토리지(ADLS/S3/GCS)의 파일을 Unity Catalog로 관리되는 테이블에 증분 방식으로 로드합니다.
- 배치 모드(예: 디렉터리 목록 또는 파일 알림)에서 schema enforcement 및 schema evolution과 함께 Auto Loader를 사용하여 데이터를 Unity Catalog로 관리되는 테이블에 적재합니다.
- 다양한 엔터프라이즈 소스에서 데이터를 Unity Catalog로 관리되는 테이블에 안정적으로 수집하도록 Lakeflow Connect를 구성합니다.
- 노트북에서 JDBC/ODBC 또는 REST 클라이언트를 사용하여 데이터를 클라우드 스토리지에 적재하거나 Unity Catalog로 관리되는 테이블에 직접 적재하며, 일반적으로 Lakeflow Jobs로 오케스트레이션하고 예약합니다.
- 데이터 볼륨, 수집 빈도, 데이터 유형, Unity Catalog를 통한 거버넌스 요구 사항과 같은 기술적 요건에 따라 Auto Loader, Lakeflow Connect(standard 및 managed connector), 파트너 커넥터, 기타 수집 방식 중에서 우선순위를 정합니다.
- 반정형 및 비정형 데이터(예: JSON 및 중첩 데이터)를 Lakeflow Connect 및 기타 managed connector를 통해 Unity Catalog로 관리되는 Delta 테이블에 수집합니다.

섹션 **3:** 데이터 변환 및 모델링 **(22%)**

- PySpark/SQL로 bronze 테이블을 읽어 null을 정리하고 데이터 유형을 표준화한 뒤 새로운 silver 테이블에 기록하는 방식으로 데이터 정제를 구현합니다.
- inner join, left join, broadcast join, 다중
    - 키, cross join, union, union all과 같은 연산으로 DataFrame을 결합합니다.
- 열 이름 추가, 삭제, 분할, 이름 변경, 필터 적용, 배열 분해(explode) 등을 통해 열, 행, 테이블 구조를 조작합니다.
- count, approximate count distinct, mean, summary와 같이 DataFrame에 대한 데이터 중복 제거 연산 및 집계 연산을 수행합니다.
- 기본 튜닝 파라미터(spark.sql.shuffle.partitions:,
    - spark.default.parallelism, spark.executor/driver.memory,
    - spark.sql.autoBroadcastJoinThreshold)를 이해하고 성능을 다시 측정합니다.
- Unity Catalog에서 BI 및 분석 팀을 위한 materialized view, view, streaming table, 테이블과 같은 Gold 계층 객체의 차이점과 구축 방법을 이해합니다.
- 신뢰할 수 있는 Silver 및 Gold 데이터셋을 보장하기 위해 데이터 품질 검사와 검증 규칙을 적용합니다.

섹션 **4: Lakeflow Jobs** 활용 **(16%)**

- 파이프라인 오케스트레이션을 위해 Lakeflow Jobs를 사용하여 제어 흐름(재시도 및 분기, 반복과 같은 조건부 작업)을 구현합니다.
- Lakeflow Jobs와 그 DAG 기반 태스크 그래프를 사용하여 공통 태스크(노트북, SQL 쿼리, 대시보드, 파이프라인 태스크)와 그 종속성을 구성합니다.
- 트리거 유형(예약, 파일 도착, 테이블 업데이트)에 대한 이해를 바탕으로 Lakeflow Jobs를 사용하여 작업 일정을 구현합니다.
- 데이터 가용성과 파이프라인 종속성에 따라 시간 기반 트리거와 데이터 기반 트리거 중에서 선택합니다.

섹션 **5: CI/CD** 구현 **(10%)**

- Databricks Git Folders(이전 명칭 Databricks Repos)에서 브랜치를 생성하고 전환하기, 변경 사항을 커밋하고 푸시하기, Databricks Git 통합을 사용하여 pull request 생성하기를 포함하여 Databricks workspace UI 내에서 코드 개발 워크플로를 관리합니다.
- dev, test, prod 대상에 동일한 코드베이스를 승격하는 과정에서 Automation Bundle(이전 명칭 Databricks Asset Bundles) 변수와 재정의를 사용한 환경별 구성을 이해합니다.
- Declarative Automation Bundles(이전 명칭 Databricks Asset Bundles)를 배포하여 Lakeflow Jobs, Lakeflow Spark Declarative Pipelines 및 기타 workspace 자산을 dev, test, prod 환경에 걸쳐 패키징, 구성, 승격합니다.
- 자동화된 CI/CD 워크플로에서 Declarative Automation Bundles(이전 명칭 Databricks Asset Bundles) 및 기타 workspace 자산을 검증, 배포, 관리하기 위해 Databricks CLI를 이해합니다.

섹션 **6:** 문제 해결**,** 모니터링**,** 최적화 **(10%)**

- Lakeflow Jobs 실행 기록 보기를 사용하여 현재 실행 시간을 과거 기준선과 비교함으로써 작업 성능의 추세를 파악합니다.
- Lakeflow Jobs UI를 사용하여 작업 상태를 해석하고, DAG 기반 태스크 그래프를 확인하여 업스트림 병목을 찾아내며, 파이프라인 실행 시간과 실패율을 추적함으로써 파이프라인 상태를 모니터링합니다.
- Spark UI에서 스테이지 수준 메트릭을 해석하여 데이터 스큐, 셔플링, 디스크 스필과 같은 일반적인 성능 병목을 파악합니다.
- Liquid Clustering 및 predictive optimization의 기능을 이해합니다.
- 클러스터 시작 실패, 라이브러리 충돌, 메모리 부족(out-of-memory) 문제를 진단합니다.

섹션 **7:** 거버넌스 및 보안 **(15%)**

- Unity Catalog에서 managed 테이블과 external 테이블을 구분하고, 이에 대한 기본 작업(생성, 수정, 삭제, managed 테이블과 external 테이블 간 변환)을 수행합니다.
- 보안 계층의 적절한 수준에서 principal(사용자, 그룹, 서비스 프린시펄)에게 GRANT, REVOKE, DENY 권한을 적용하여 UI와 SQL을 사용해 액세스 제어를 구성합니다.
- 사용자 그룹에 따라 데이터 가시성을 제한하기 위해 열 수준 마스킹과 행 수준 보안을 이해합니다.
- 민감한 데이터에 대한 행 수준 필터링과 열 마스킹을 중앙에서 제어하기 위해 Unity Catalog ABAC 정책을 이해합니다.

샘플 문항

이 문항들은 이전 버전 시험에서 출제되었다가 폐기된 문항입니다. 그 목적은 시험 가이드에 명시된 학습 목표를 보여 주고, 각 목표에 부합하는 샘플 문항을 제공하는 데 있습니다. 시험 가이드에는 시험에서 다뤄질 수 있는 목표가 나열되어 있습니다. 인증 시험을 준비하는 가장 좋은 방법은 시험 가이드의 시험 개요를 검토하는 것입니다.

문항 **1**
목표*: Spark UI*에서 스테이지 수준 메트릭을 해석하여 데이터 스큐, 셔플링, 디스크 스필과 같은 일반적인 성능 병목을 파악합니다.

한 데이터 엔지니어가 새로운 데이터 소스를 도입한 후 배치 작업의 실행 시간이 두 배로 늘어난 것을 발견합니다. Spark UI에서 가장 오래 걸리는 스테이지를 보면 대부분의 태스크는 30초 이내에 완료되지만, 한 태스크는 10분 넘게 걸립니다. 해당 스테이지의 태스크 요약을 보면 셔플 읽기의 최소/중앙값은 약 400MB인 반면 최대값은 5GB를 초과합니다.

작업 실행 시간을 줄이는 해결책은 무엇입니까?

A. 느린 태스크가 더 빨리 완료되도록 클러스터 크기를 늘려 실행자(executor)를 추가한다 B. 초과 크기의 파티션을 런타임에 자동으로 분할하도록 스큐 조인 처리 기능이 포함된 adaptive query execution이 활성화되어 있는지 확인한다
C. 더 적은 수의 태스크로 작업을 통합하도록 spark.sql.shuffle.partitions를 줄인다 D. 조인 전에 salt 키를 사용하여 데이터셋을 수동으로 리파티션하여 치우친 키를 고르게 분산한다

문항 **2**
목표*: Databricks Data Intelligence Platform*의 *compute* 서비스에 대해 그 특성, 제한 사항, 비용 모델을 포함하여 이해하고, 각 워크로드 사용 사례에 가장 적합한 옵션을 선택합니다.

한 데이터 엔지니어는 파이프라인을 빠르게 반복 개발하면서도, 잘못된 수집 이후 안정적인 롤백을 유지하고, 규제 준수를 위한 감사 추적을 보장하며, AI와 BI 워크로드 모두에 대해 단일 진실 공급원(single source of truth)에 일관되게 액세스할 수 있어야 합니다.

이러한 요구 사항을 충족하기 위해 데이터 엔지니어는 어떤 전략을 사용해야 합니까?

A. 롤백을 위해 수동 파일 버전 관리와 야간 복사본을 사용하는 DBFS CSV 스토리지. B. 일관된 액세스와 리니지를 위해 Unity Catalog로 관리되는 Delta Lake ACID 트랜잭션과 시간 여행(time travel).
C. 복구와 거버넌스를 위해 임시 SQL 쿼리만 사용하는 클라우드 오브젝트 스토리지. D. 감사 추적과 BI 배포를 위한 휘발성 인메모리 DataFrame.

문항 **3**
목표*:* 배치, 스트리밍, 증분 로딩을 포함한 데이터 수집 패턴을 구현하고 그 세부 사항을 파악하며, 로컬 파일*, Lakeflow Connect standard connector, Lakeflow Connect managed connector*와 같은 소스에서 데이터를 가져옵니다.

한 데이터 엔지니어가 고객 소유의 S3 버킷에서 Databricks 감사 로그를 소비하는 다운스트림 파이프라인을 구축하고 있습니다. 스키마 추론과 체크포인팅을 구현하기 전에, 이 엔지니어는 전달 형식, 일반적인 수집 지연 시간, 파일이 덮어쓰기될 수 있는지 여부를 이해하고자 합니다.

Databricks 감사 로그 스토리지의 동작은 무엇입니까?

A. 파일은 JSON으로 전달되며 전달이 시작된 후 일반적으로 15분 이내에 이벤트가 로깅되고, 새로운 전달이 기존 파일을 덮어쓸 수 있다
B. 파일은 CSV로 전달되며 1분 미만의 지연 시간을 보장하고, 불변성을 유지하기 위해 파일이 한 번 기록되면 덮어쓰기가 발생하지 않는다
C. 파일은 Parquet로 전달되며 24시간이 지난 후 최종 일관성을 갖고, 스트리밍 수집을 단순화하기 위해 덮어쓰기가 비활성화되어 있다

D. 파일은 JSON으로 전달되며 주간 배치 주기로 전달되고, 덮어쓰기는 기존 내용을 추가 없이 완전히 대체한다

문항 **4**
목표*:* 클러스터 시작 실패, 라이브러리 충돌, 메모리 부족*(out-of-memory)* 문제를 진단합니다.

한 데이터 엔지니어링 팀이 정제된 Delta 테이블에 대해 하루 종일 임시 SQL 쿼리를 실행하는 여러 비즈니스 분석가를 지원합니다. 이 팀은 매우 큰 클러스터로의 불필요한 확장을 피해 비용을 관리하면서도, 효율적인 쿼리 성능, 빠른 클러스터 시작, 다수의 동시 사용자 지원을 보장해야 합니다.

이러한 요구 사항을 충족하는 클러스터 구성은 무엇입니까?

A. 예약된 ETL 워크플로를 위해 설계된, autoscaling이 적용된 작업 클러스터
B. 고정된 수의 작업자 노드로 구성된 다목적(all-purpose) 클러스터
C. autoscaling이 활성화된 높은 동시성(high-concurrency) 클러스터
D. 경량 개발 작업을 위해 구성된 단일 노드(single-node) 클러스터

문항 **5**
목표*: Databricks Git Folders(*이전 명칭 *Databricks Repos)*에서 브랜치를 생성하고 전환하기, 변경 사항을 커밋하고 푸시하기*, Databricks Git* 통합을 사용하여 *pull request* 생성하기를 포함하여 *Databricks workspace UI* 내에서 코드 개발 워크플로를 관리합니다.

한 팀이 Databricks에서 ETL 파이프라인을 배포, 버전 관리, 오케스트레이션하는 모듈식 방법을 원하며, 이를 통해 CI/CD와 재현성을 가능하게 하고자 합니다.

이 요구 사항을 지원하는 기능은 무엇입니까?

A. Unity Catalog의 모델을 사용하여 ETL 작업을 표현하고, 각 모델이 파이프라인 코드 아티팩트를 저장하며, CI/CD가 Job 태스크에 연결된 모델 별칭을 업데이트하여 버전을 승격한다. B. 변환 로직을 Unity Catalog Volumes에 저장된 wheel 라이브러리로 패키징하고 이를 Jobs 태스크에 바인딩하여 여러 환경에 걸쳐 결정적인(deterministic) 배포를 보장한다.
C. API 로직을 Volume에 마운트된 노트북 안에 패키징하고, Jobs API v2를 사용하여 노트북을 트리거하며, 노트북 개정 이력을 버전 관리 시스템으로 활용한다. D. DABs를 사용하여 리소스와 코드 자산을 정의하고, 이를 Git에서 버전 관리하며, 자동화된 CI/CD 작업을 통해 여러 환경에 걸쳐 배포를 승격한다.

정답:

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